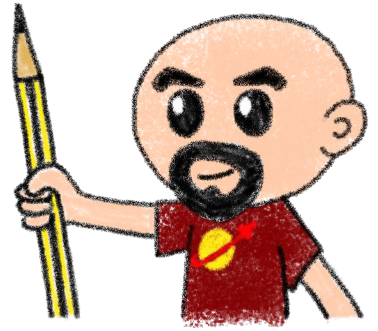
OVHcloud

insight//.

Kubernetes Operators:

Operating Cloud Native services at scale

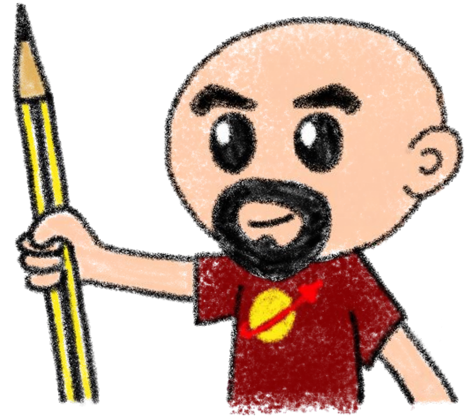
Horacio Gonzalez 2021-02-05



@LostInBrittany

Who are we?

Introducing myself and introducing ~~OVH~~ OVHcloud



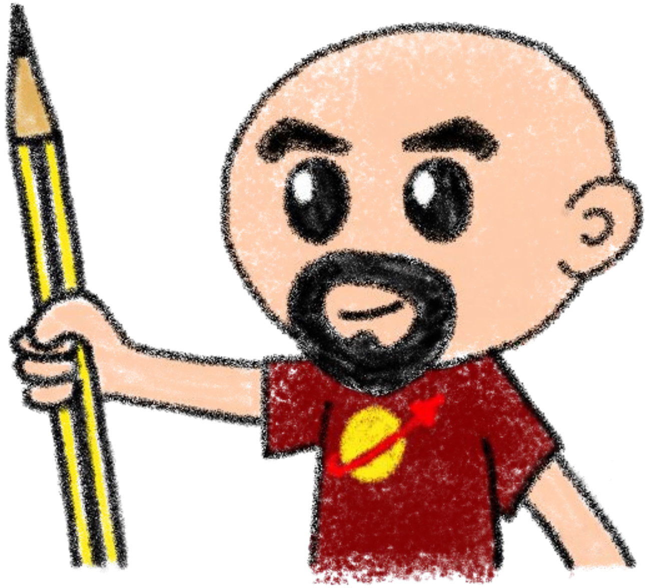
Horacio Gonzalez

@LostInBrittany

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

OVHcloud

DevRel Leader



DevFest du Bout du Monde

Finist Devs

GDG Finistère

Google Developers Experts 2019

Web Technologies GDE

Flutter

OVHcloud: A global leader

- 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

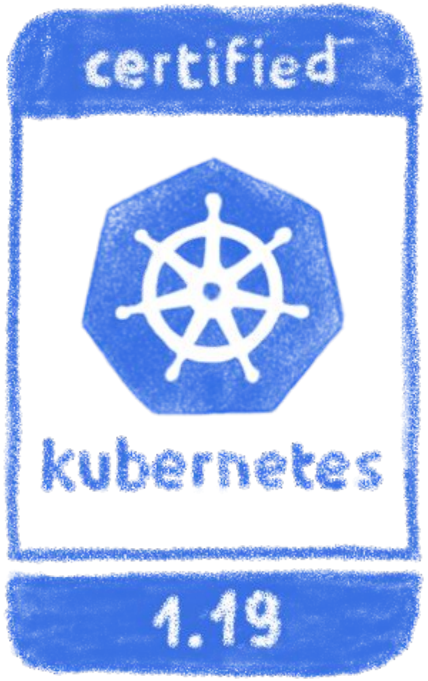
1.5 Billion Euros Invested
since 2016

P.U.E. 1.09
Energy efficiency indicator

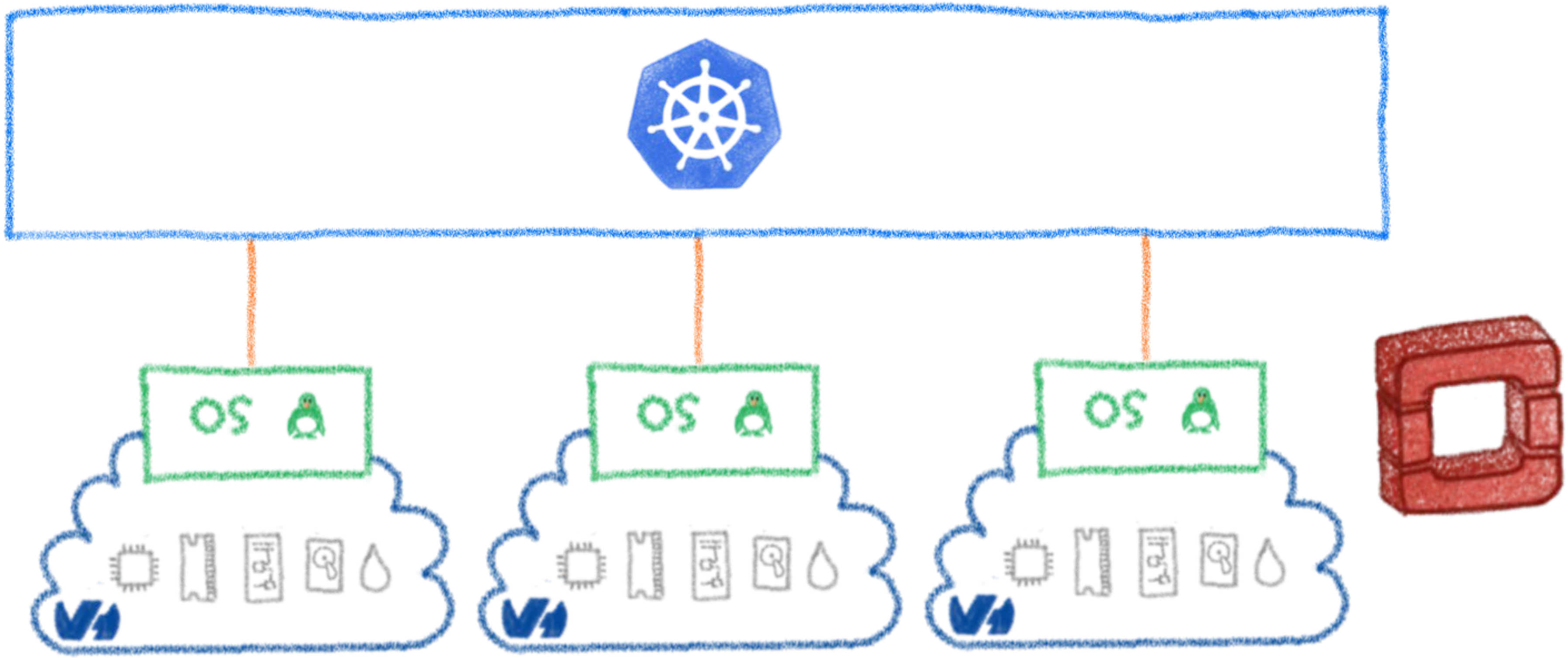
20 Years in Business
Disrupting since 1999

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

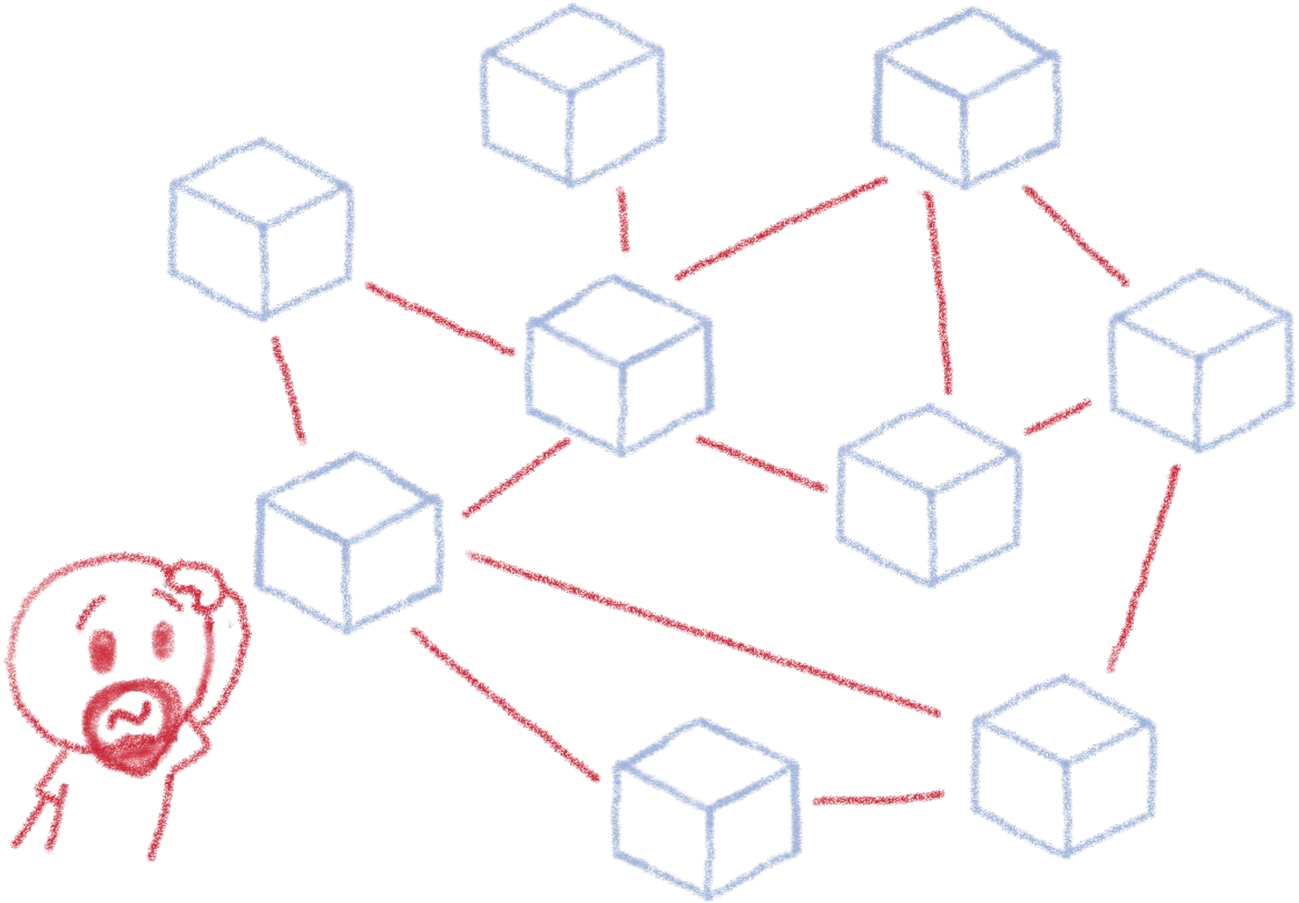
Large instance range ... and more to come

Predictible pricing

Operating Kubernetes

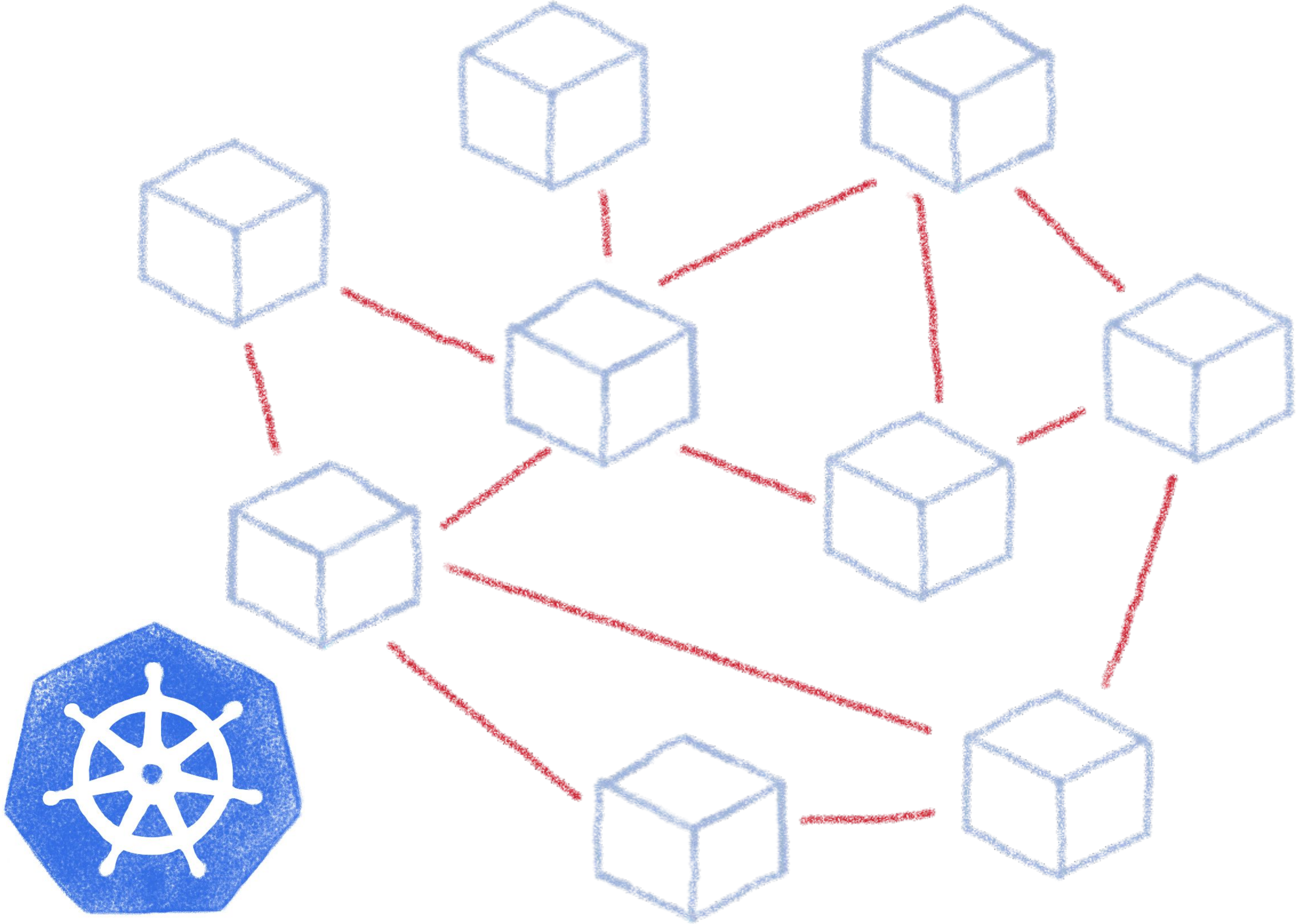
Easier said than done

Operating microservices?

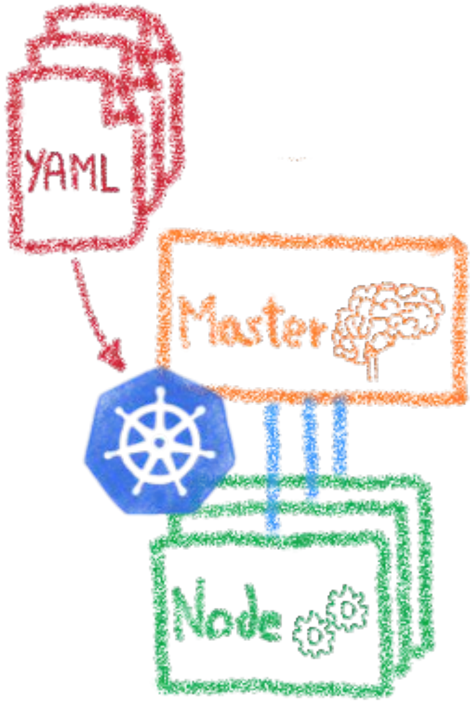


Are you sure you want to operate them by hand?

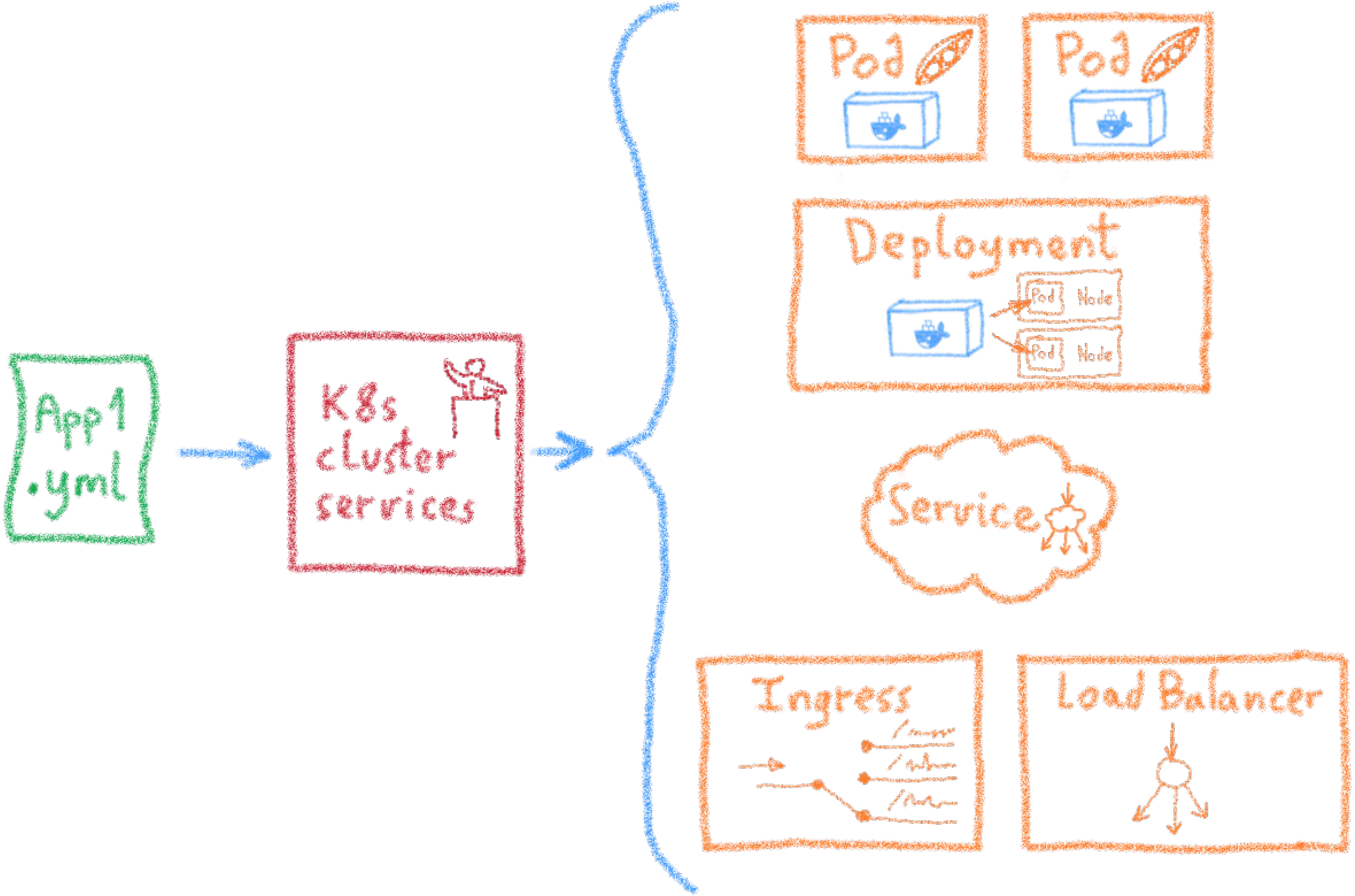
Taming microservices with Kubernetes



Declarative infrastructure



Desired State Management



Ingress

Services

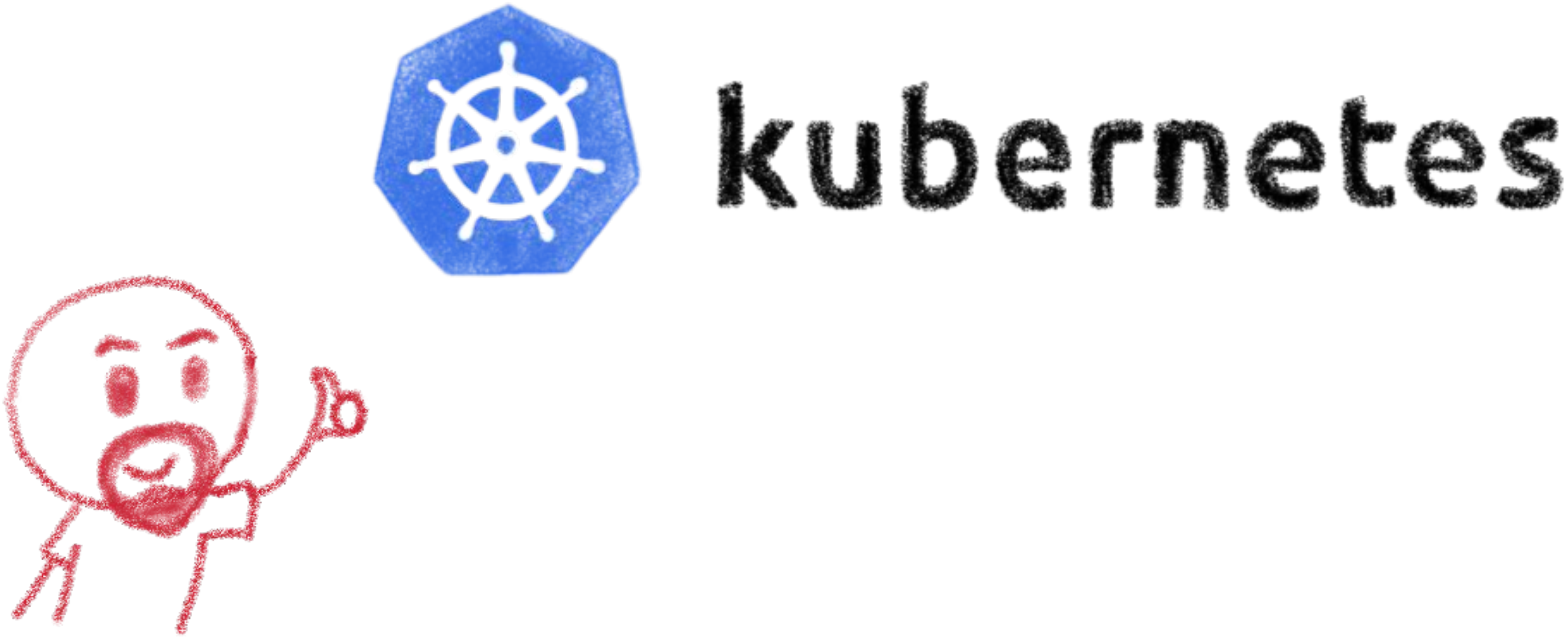
Deployments

Pods

Sidecars

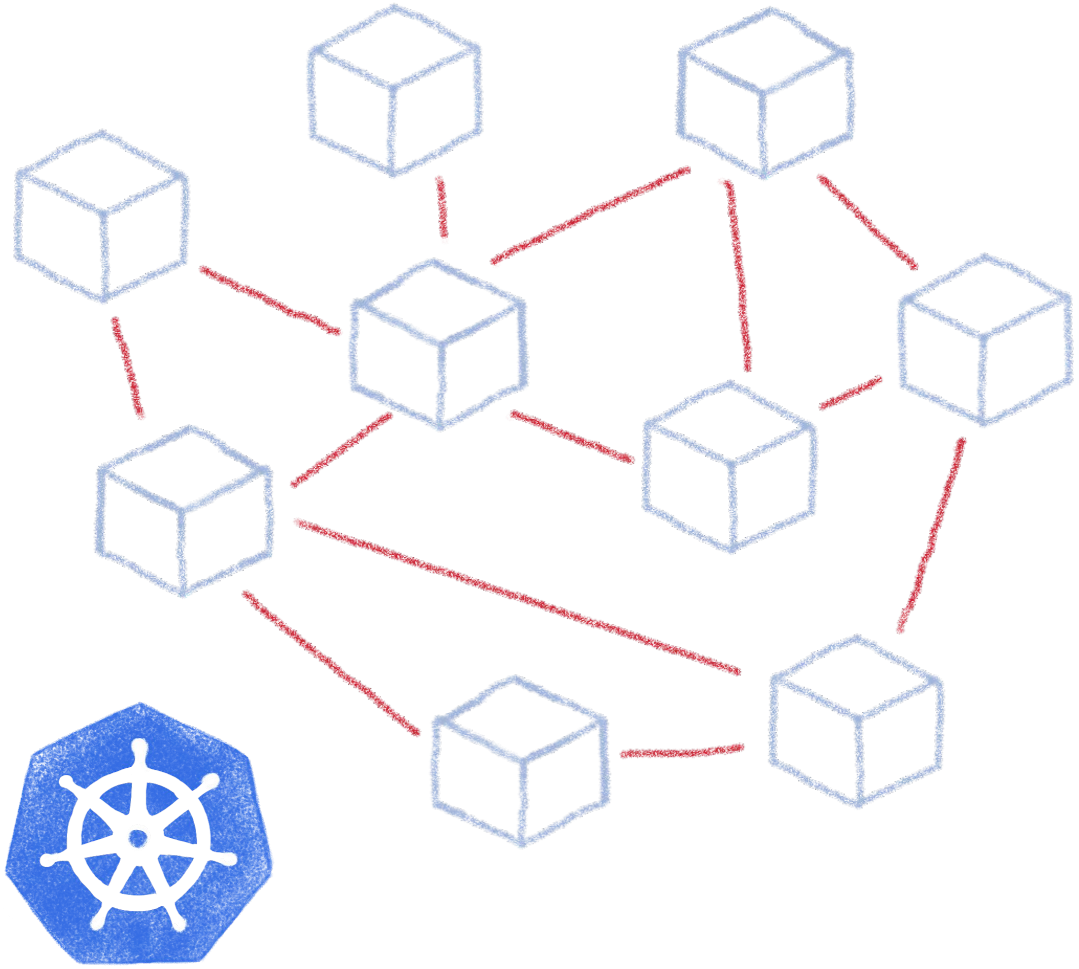
Replica Sets

Beyond a simple deployment



Everything is good now, isn't it?

Complex deployments



Ingress

Services

Deployments

Pods

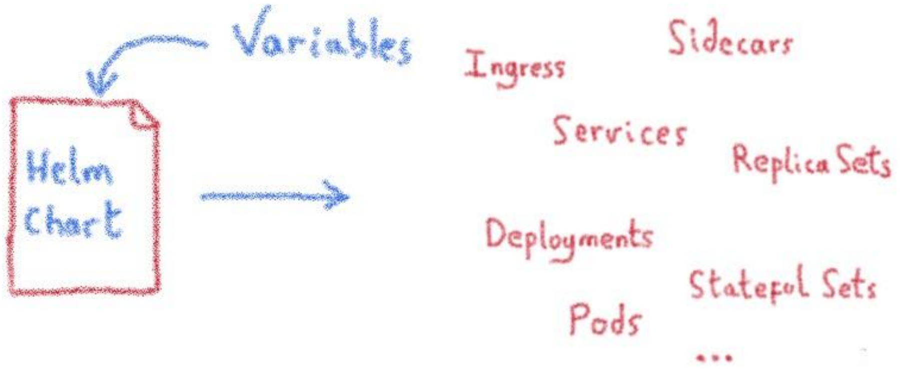
Sidecars

Replica Sets

Stateful Sets

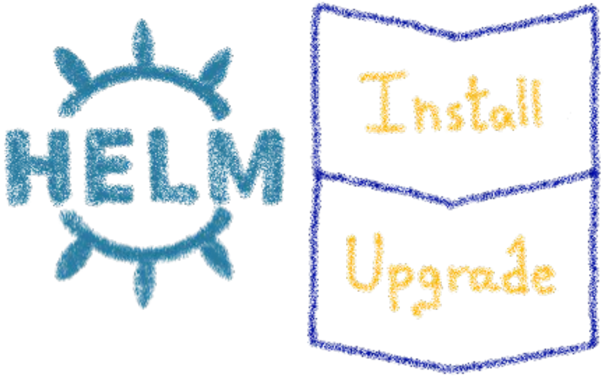
Complex deployments

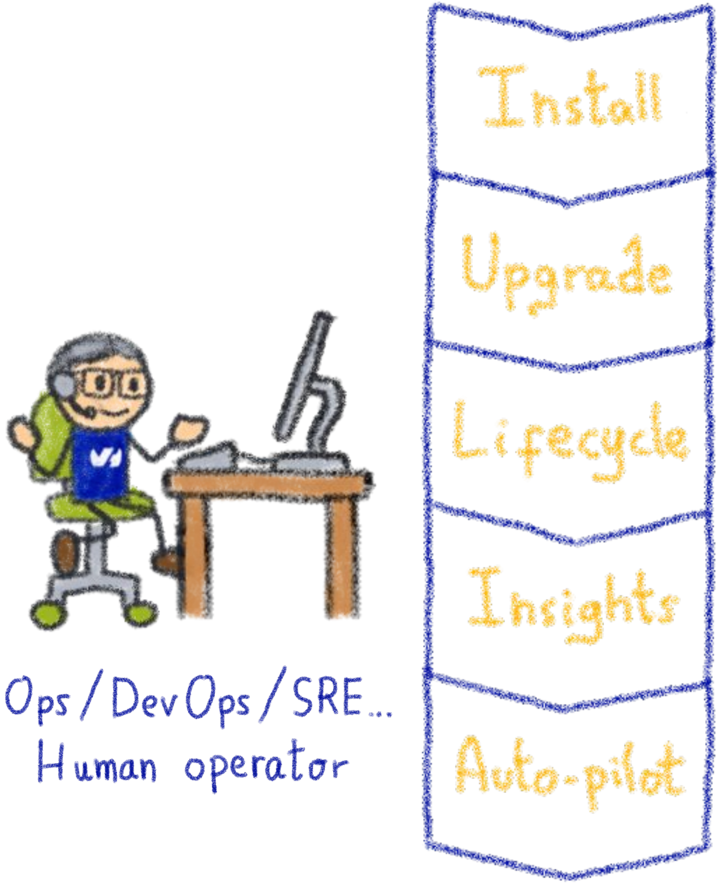
A package manager for Kubernetes



- Manage complexity
- Easy upgrades V.41 → V.42
- Simple sharing
- Easy rollbacks V.43 → V.42

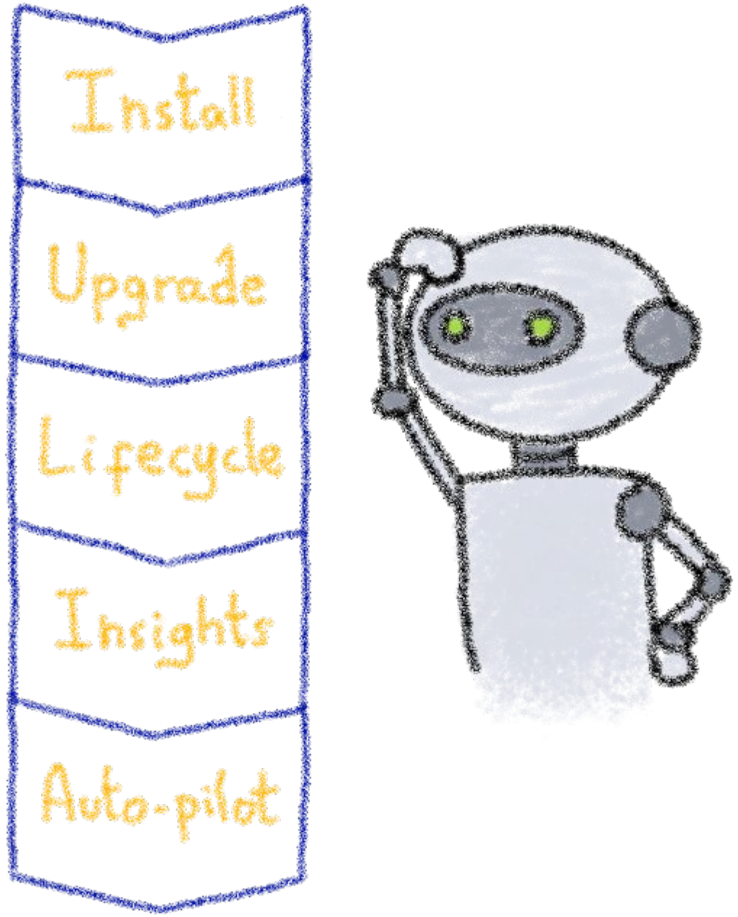
Helm Charts are configuration





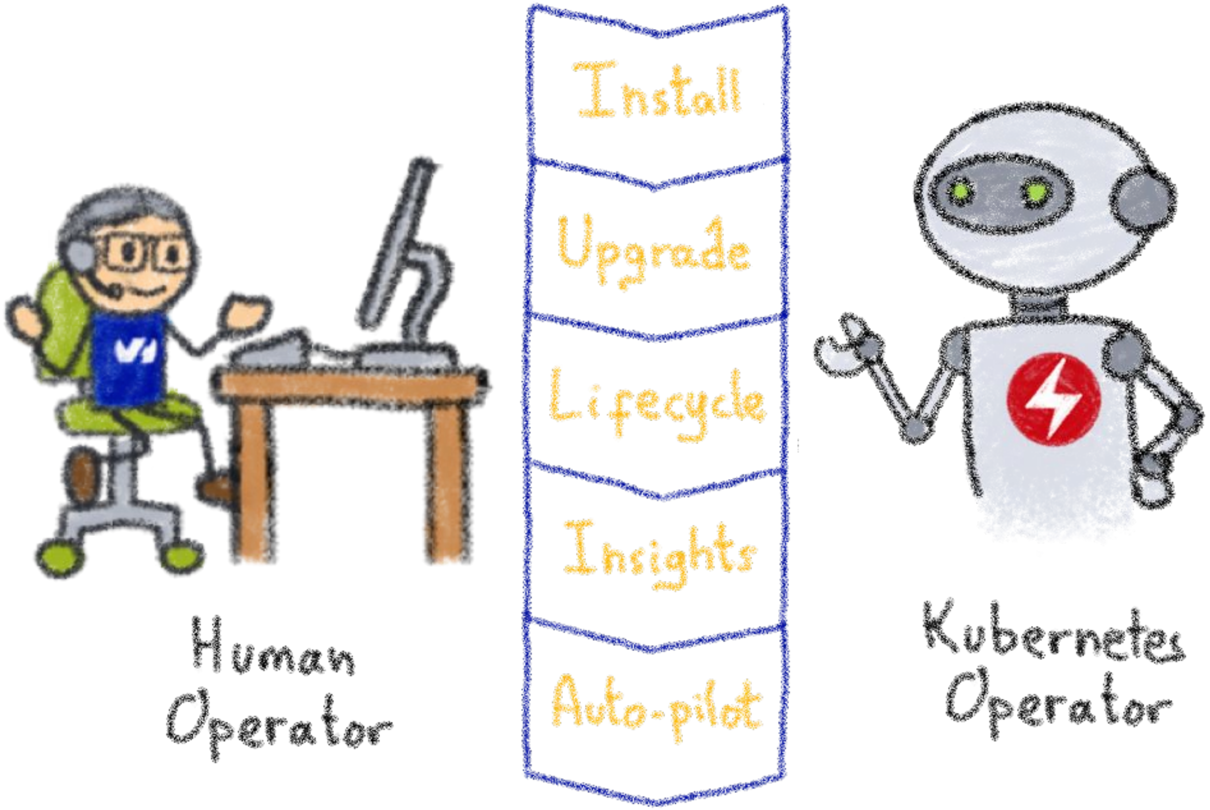
Operating is more than installs & upgrades

Kubernetes is about automation



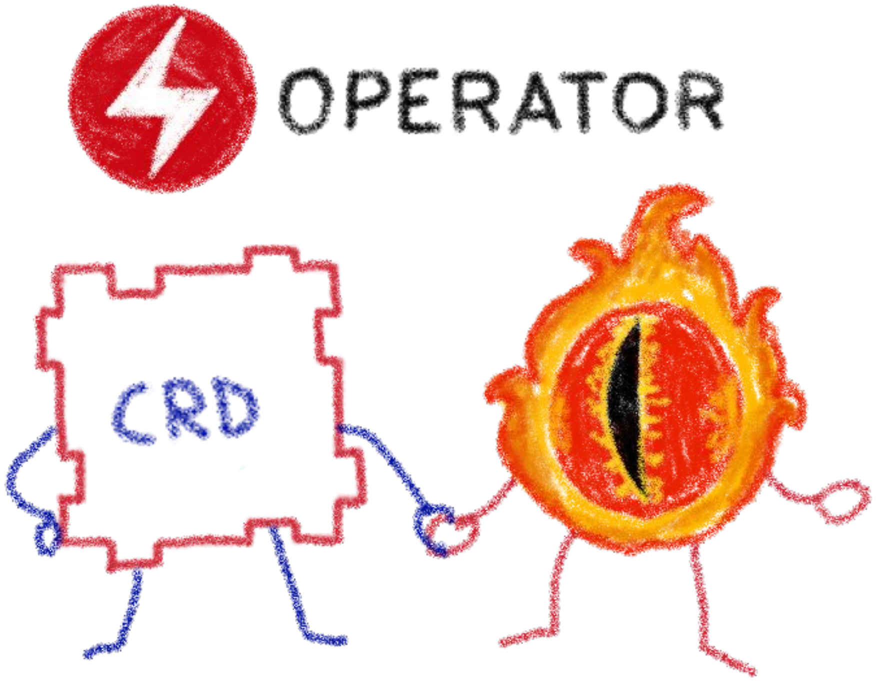
How about automating human operators?

Kubernetes Operators



A Kubernetes version of the human operator

Building operators

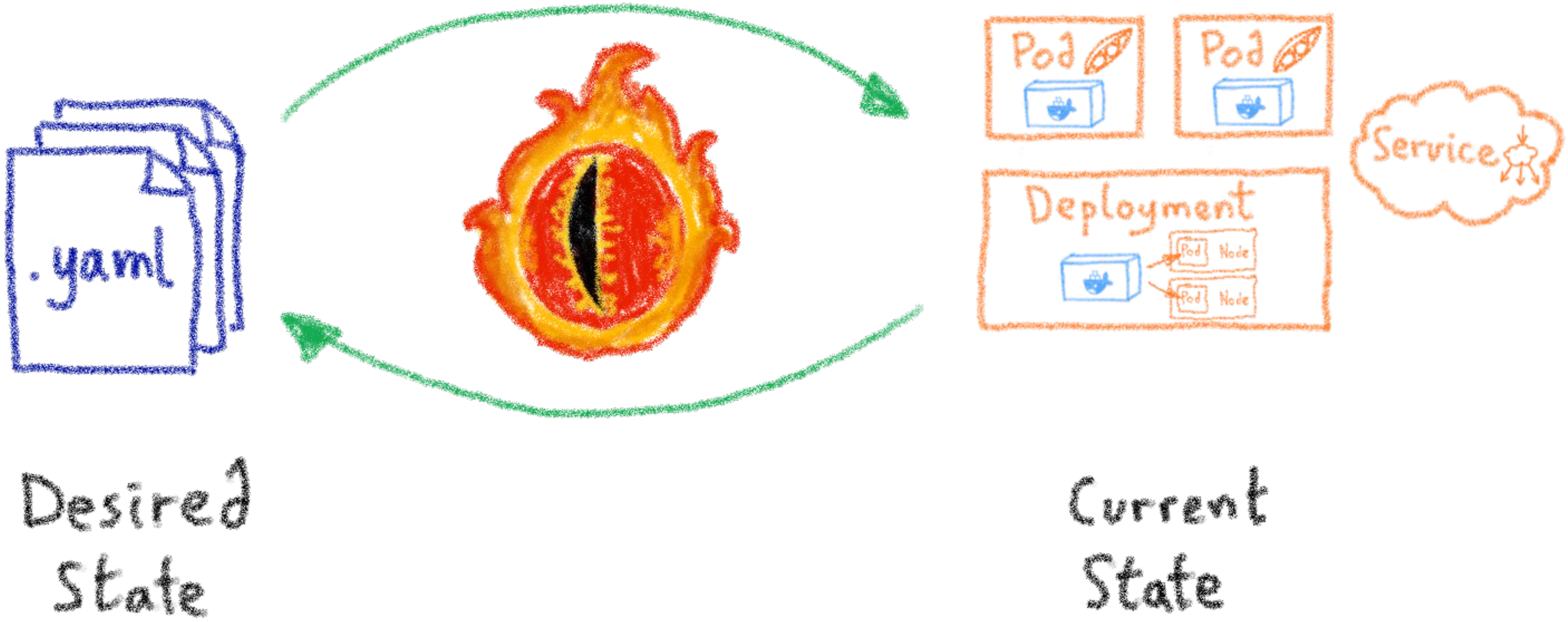


Basic K8s elements: Controllers and Custom Resources

Kubernetes Controllers

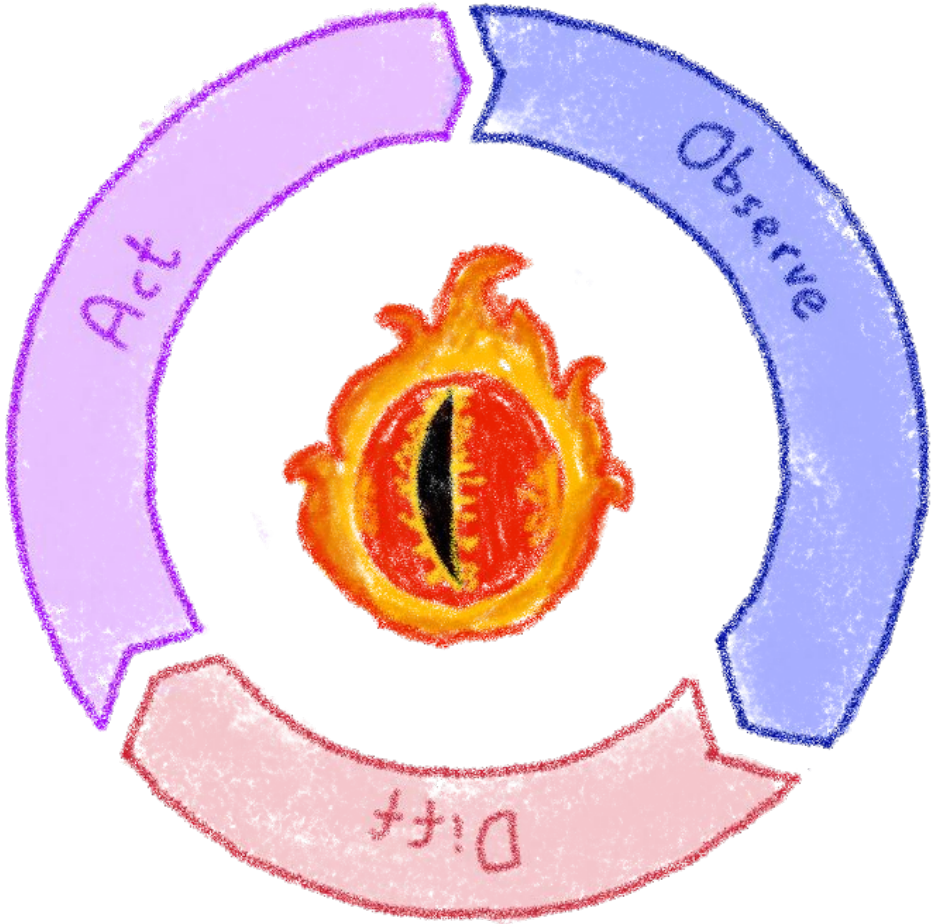
Keeping an eye on the resources

A control loop



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

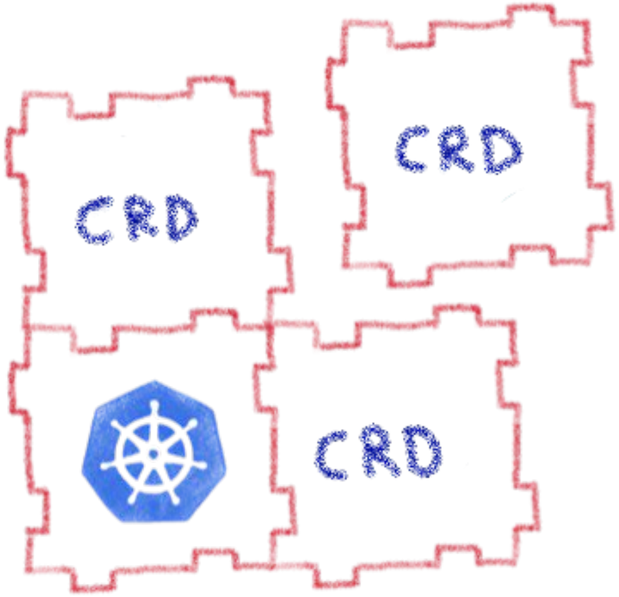
A reconcile loop



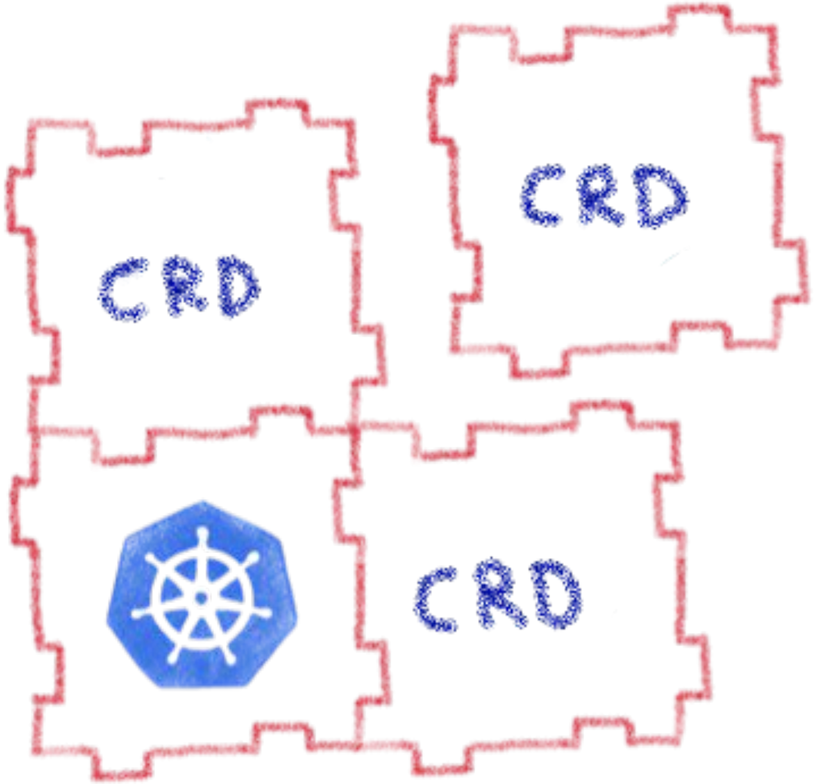
Strives to reconcile current state and desired state

Custom Resource Definitions

Extending Kubernetes API



Extending Kubernetes API

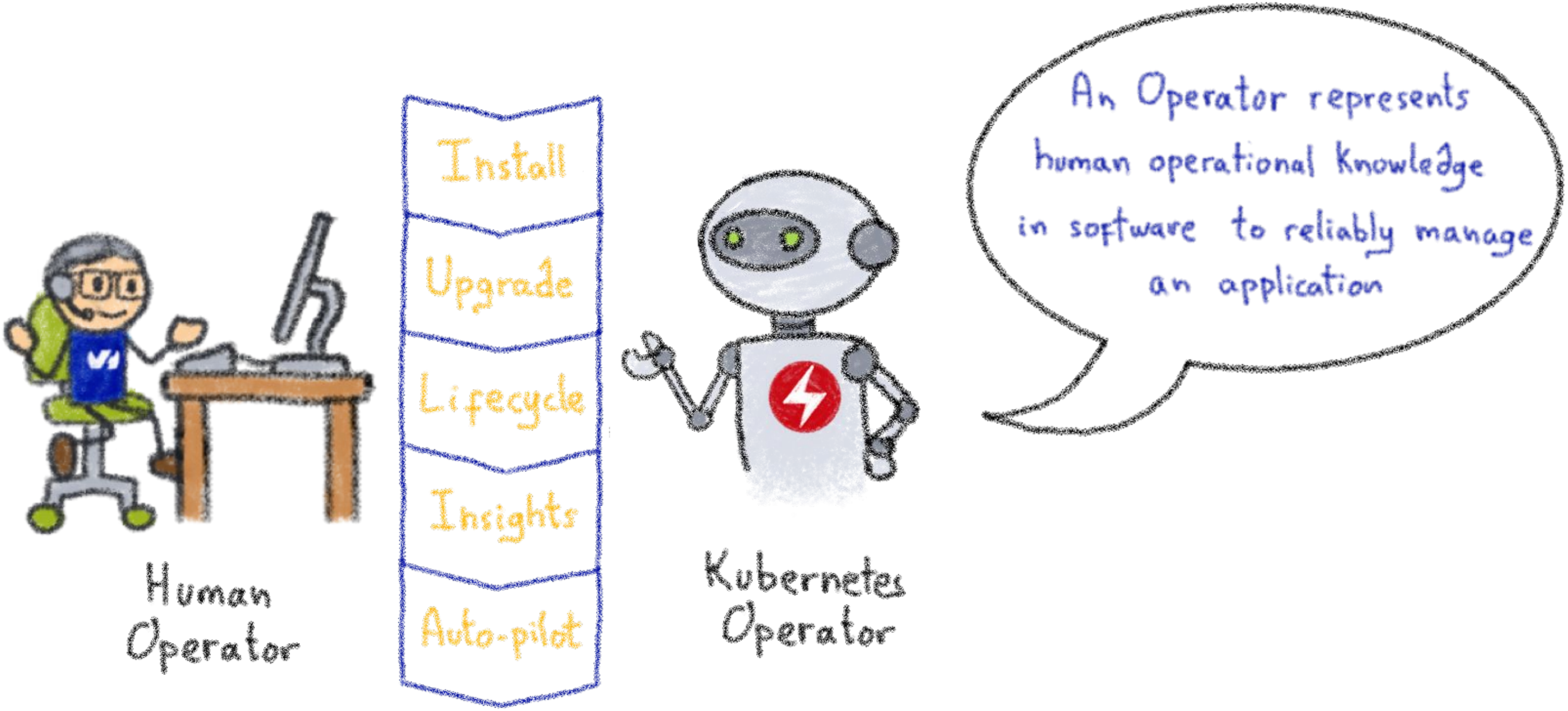


By defining new types of resources

Kubernetes Operator

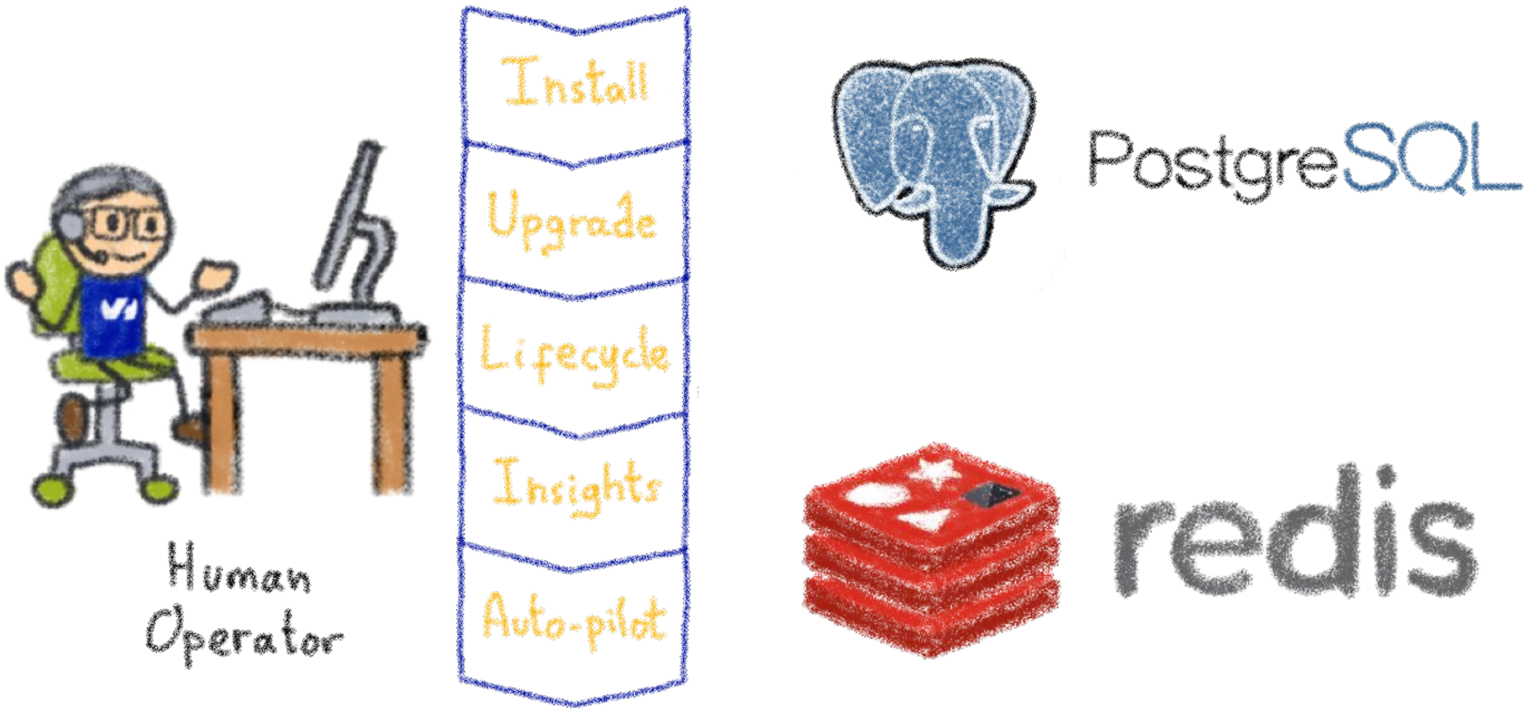
Automating operations

What's a Kubernetes Operator?



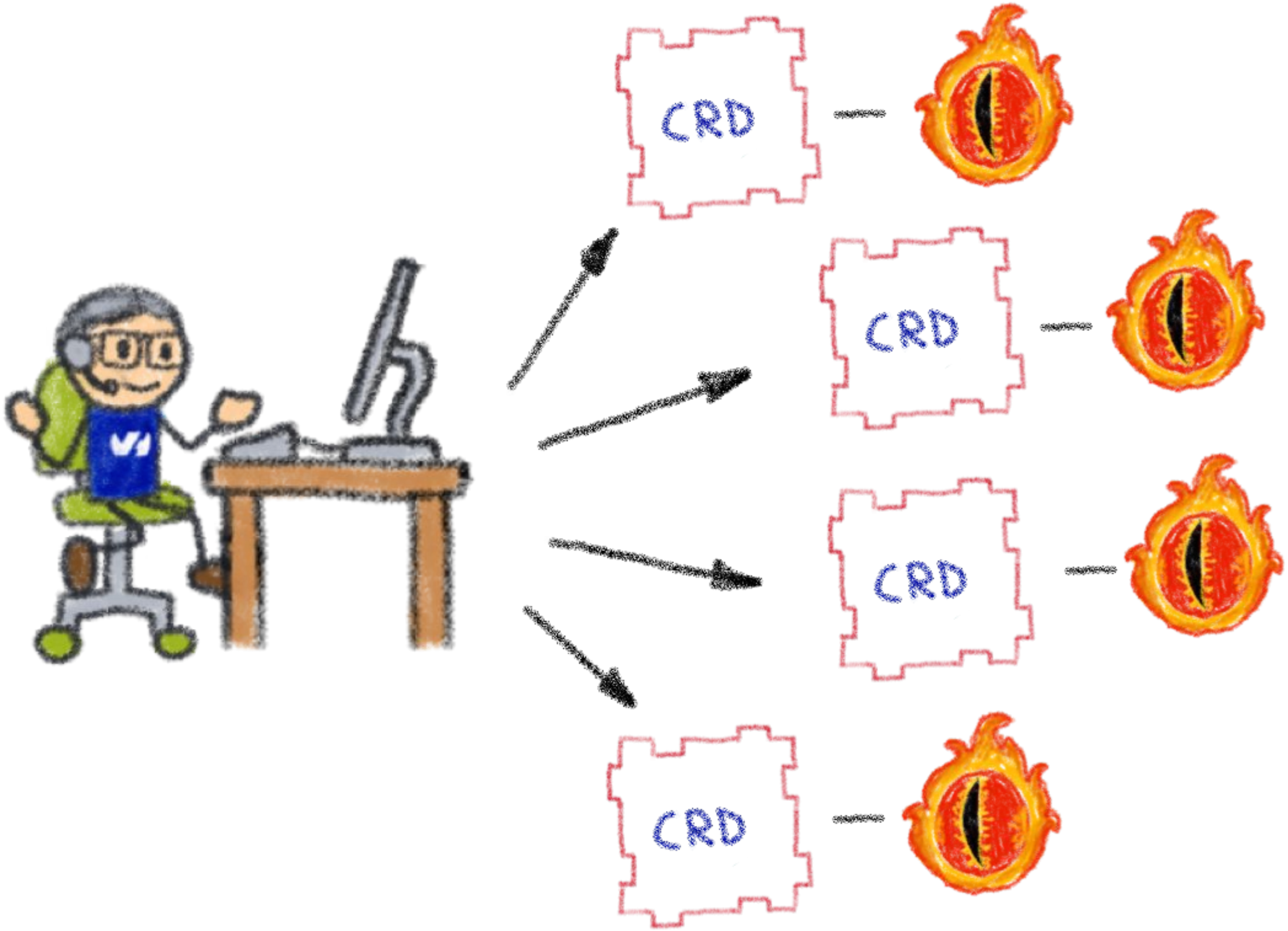
An Operator represents human operational knowledge in software to reliably manage an application

Example: databases



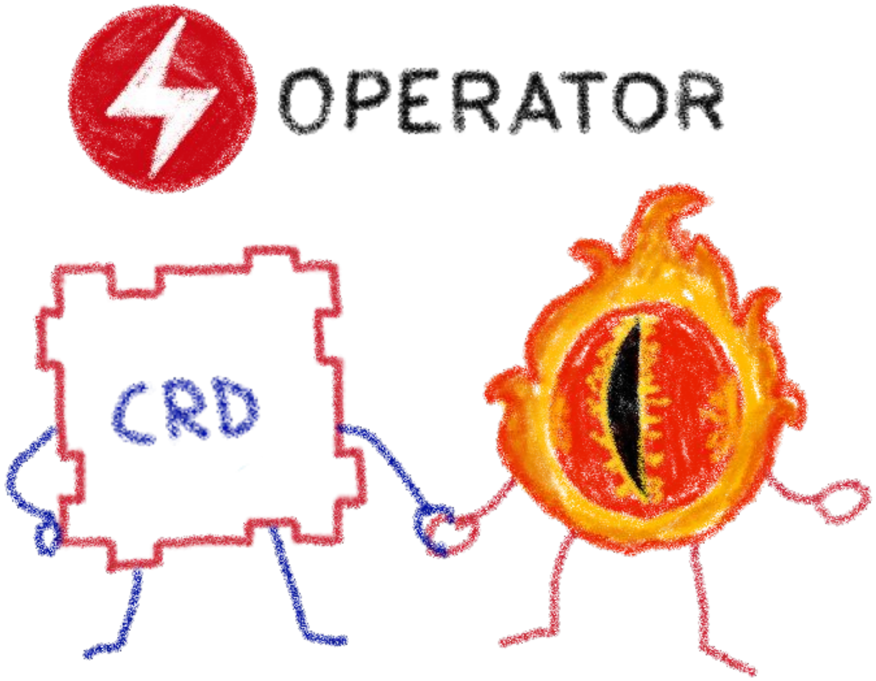
Things like adding an instance to a pool, doing a backup, sharding...

Knowledge encoded in CRDs and Controllers



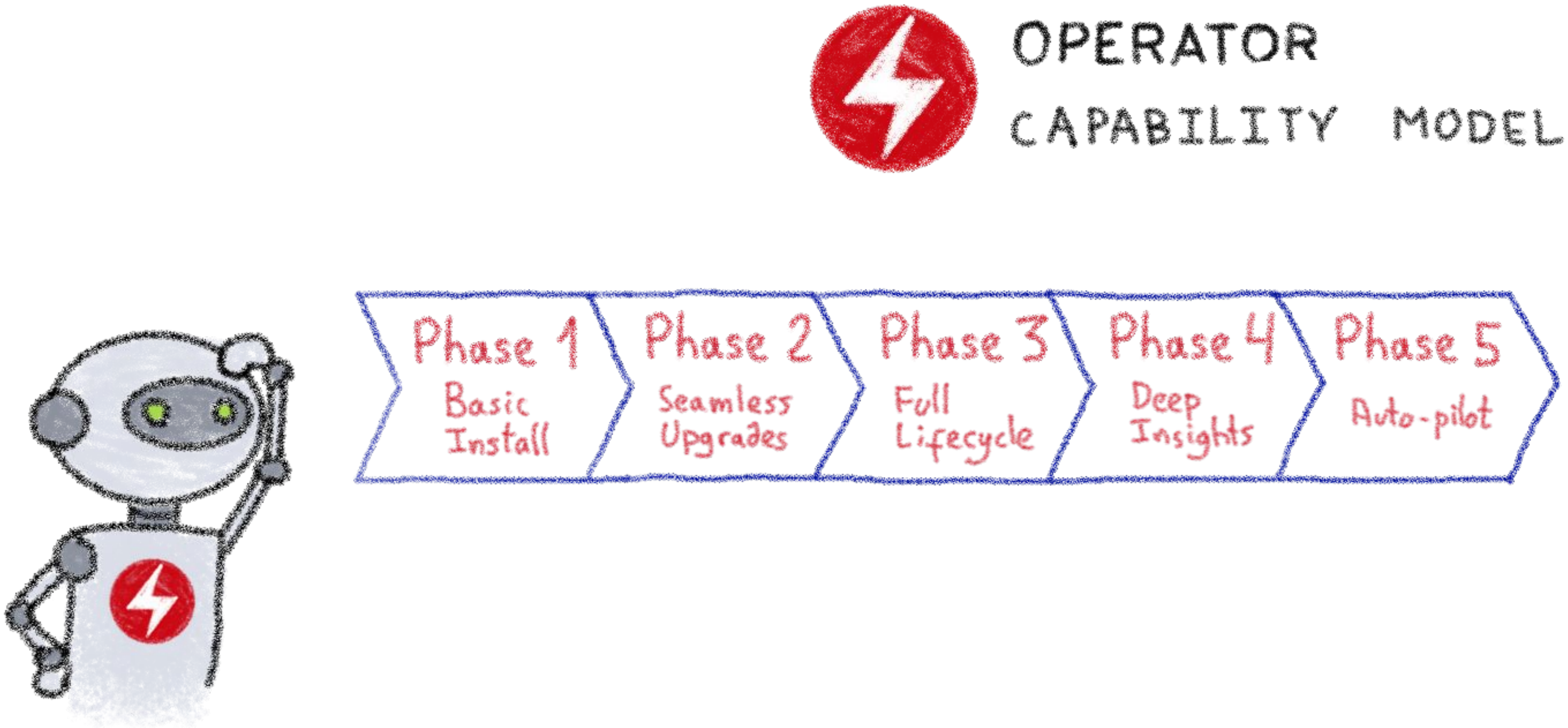
Encapsulating business logic in CRDs & Controllers

Custom Controllers for Custom Resources



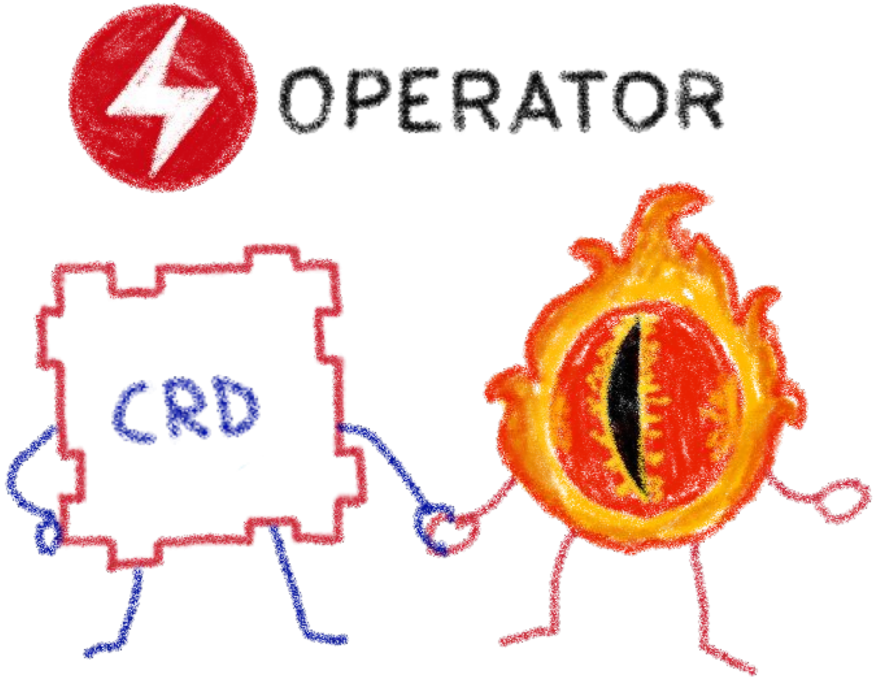
Operators implement and manage Custom Resources
using custom reconciliation logic

Operator Capability Model



Gauging the operator maturity

How to write an Operator



1- Create a new project

2- Write the CRDs to define new resource APIs

3- Specify resources to watch

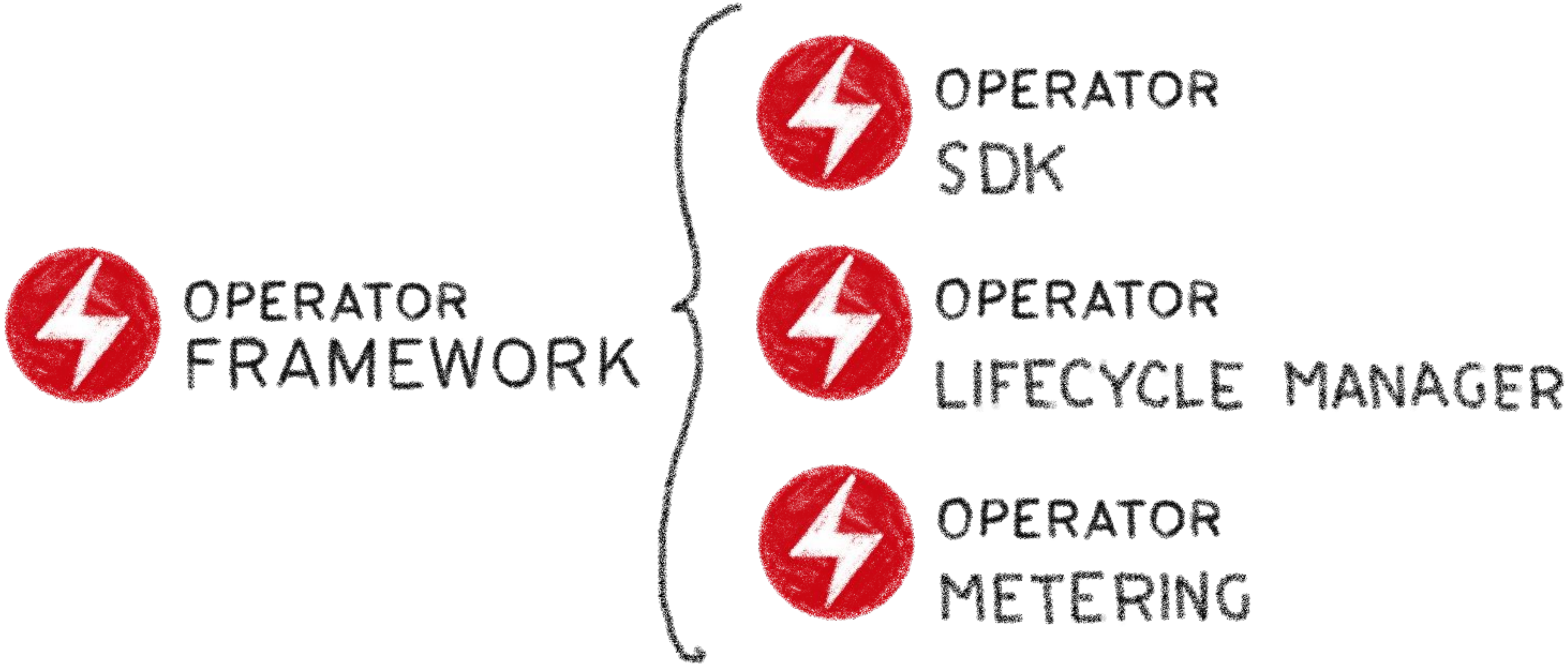
4- Define the reconciliation logic in the Controllers

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

BUILD

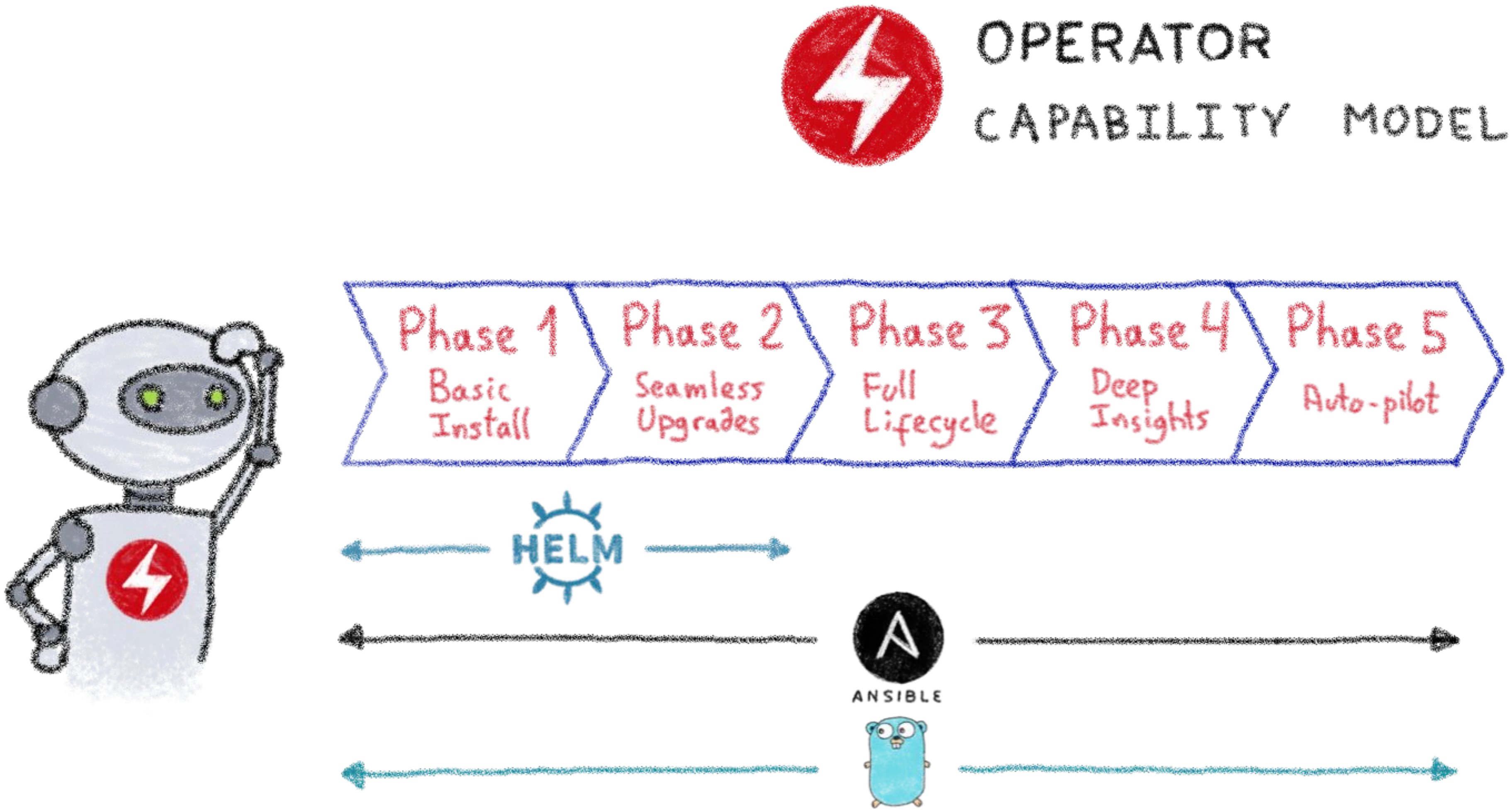
TEST

ITERATE

HELM
ANSIBLE

Three different ways to build an Operator

Operator SDK and Capability Model

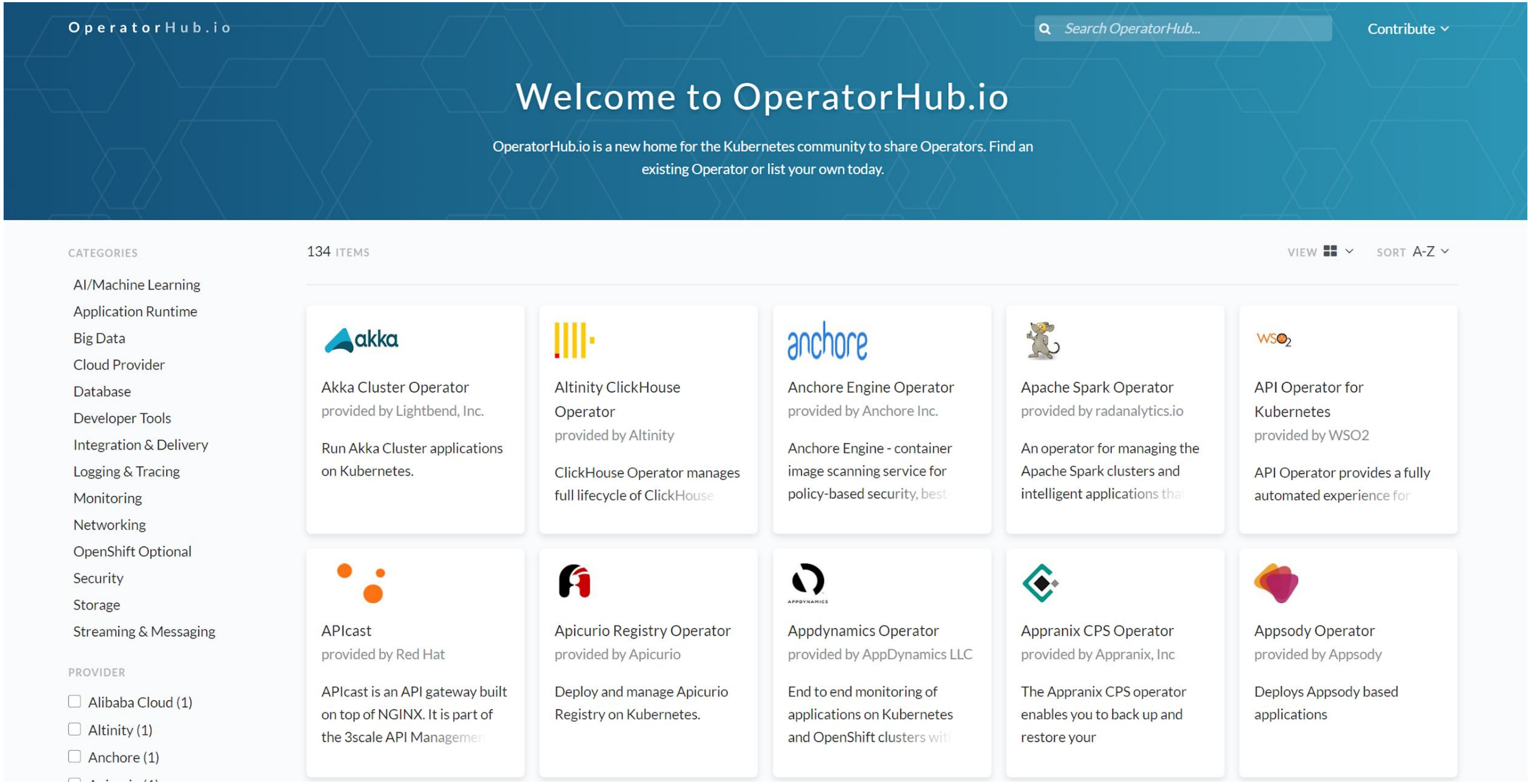


Operator Lifecycle Manager

OPERATOR
LIFECYCLE MANAGER

INSTALL
MANAGE
UPDATE

OperatorHub.io



Welcome to OperatorHub.io

OperatorHub.io is a new home for the Kubernetes community to share Operators. Find an existing Operator or list your own today.

CATEGORIES

- AI/Machine Learning
- Application Runtime
- Big Data
- Cloud Provider
- Database
- Developer Tools
- Integration & Delivery
- Logging & Tracing
- Monitoring
- Networking
- OpenShift Optional
- Security
- Storage
- Streaming & Messaging

PROVIDER

- □ Alibaba Cloud (1)
- □ Altinity (1)
- □ Anchore (1)

134 ITEMS

Akka Cluster Operator
provided by Lightbend, Inc.
Run Akka Cluster applications on Kubernetes.

Altinity ClickHouse Operator
provided by Altinity
ClickHouse Operator manages full lifecycle of ClickHouse

Anchore Engine Operator
provided by Anchore Inc.
Anchore Engine - container image scanning service for policy-based security, best

Apache Spark Operator
provided by radanalytics.io
An operator for managing the Apache Spark clusters and intelligent applications tha

API Operator for Kubernetes
provided by WSO2
API Operator provides a fully automated experience for

APIcast
provided by Red Hat
APIcast is an API gateway built on top of NGINX. It is part of the 3scale API Managemen

Apicurio Registry Operator
provided by Apicurio
Deploy and manage Apicurio Registry on Kubernetes.

Appdynamics Operator
provided by AppDynamics LLC
End to end monitoring of applications on Kubernetes and OpenShift clusters wit

Apprenix CPS Operator
provided by Appranix, Inc
The Appranix CPS operator enables you to back up and restore your

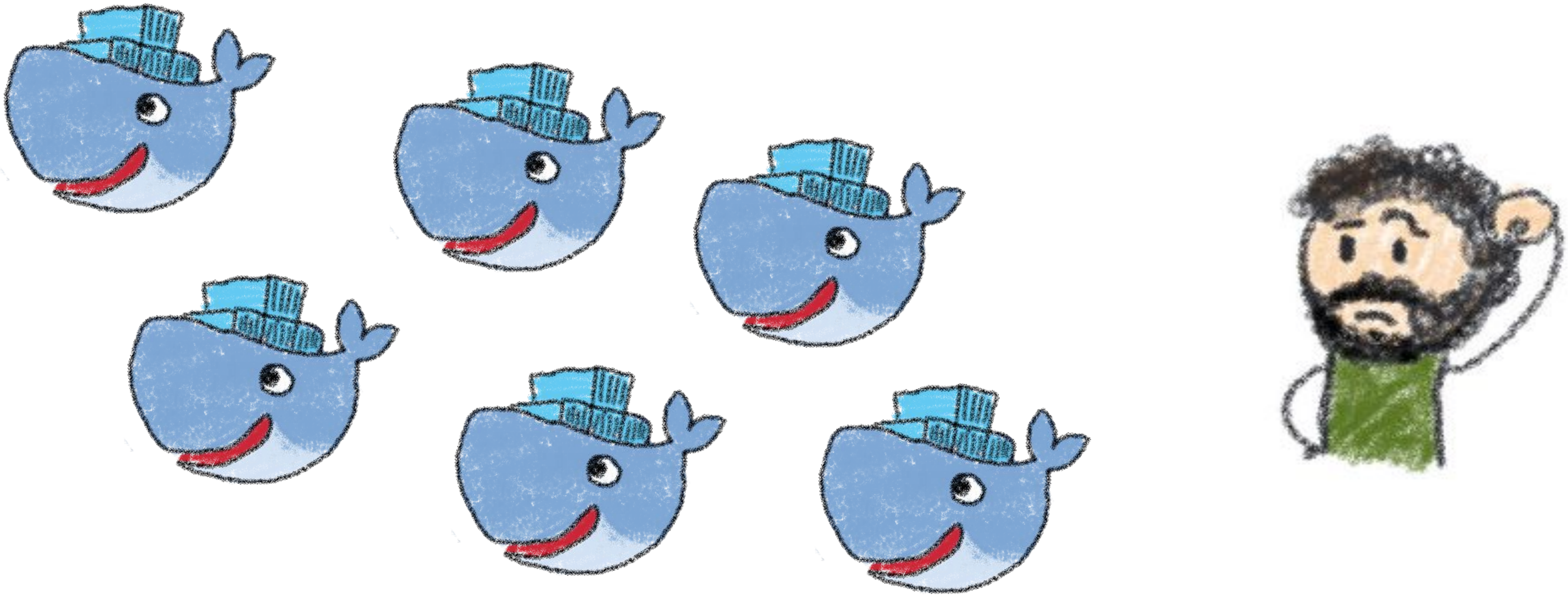
Appsody Operator
provided by Appsody
Deploys Appsody based applications

Harbor Operator

Managing private registries at scale

HARBOR

We wanted to build a new product



OVHcloud Managed Private Registry

Looking at the Open Source world

HARBOR
Portus
docker REGISTRY

Two main alternatives around Docker Registry

Harbor has more community traction

HARBOR

★ Star 11.5k

Fork 3.1k

CLOUD NATIVE COMPUTING FOUNDATION

HELM

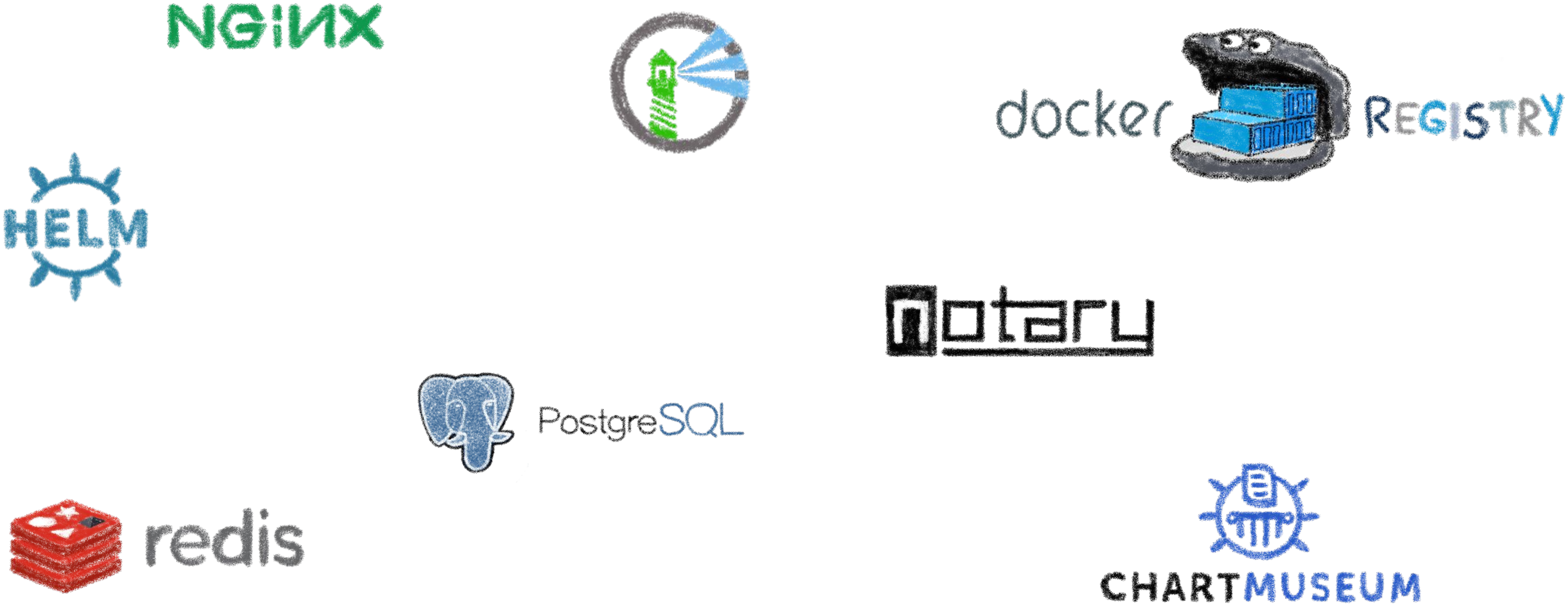
Portus

★ Star 2.6k

Fork 454

Two main alternatives

Harbor has lots of components



But it has a Helm Chart

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

```
$ helm install harbor
```

What about configuration?

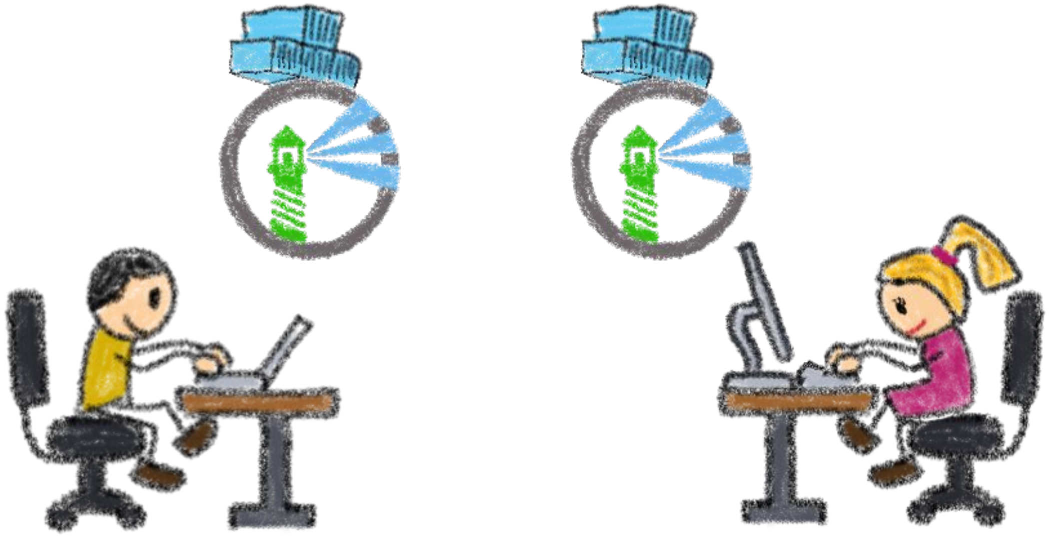
Installing a 200 GB K8s volume?

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

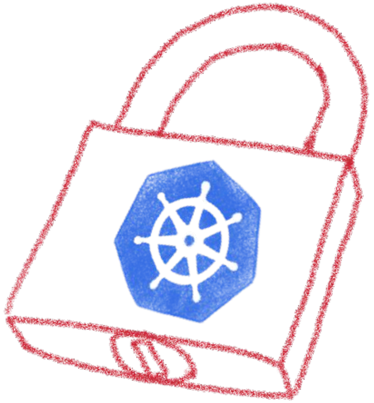
PostgreSQL

Object Storage

} as a Service

Reusing existing services

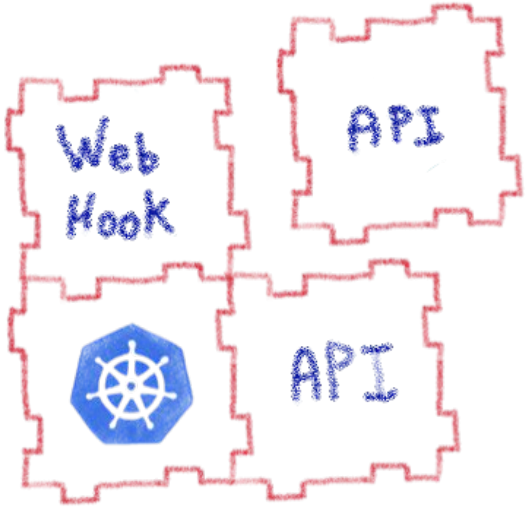
Using the platform



RBAC

Security policies

API inputs validation



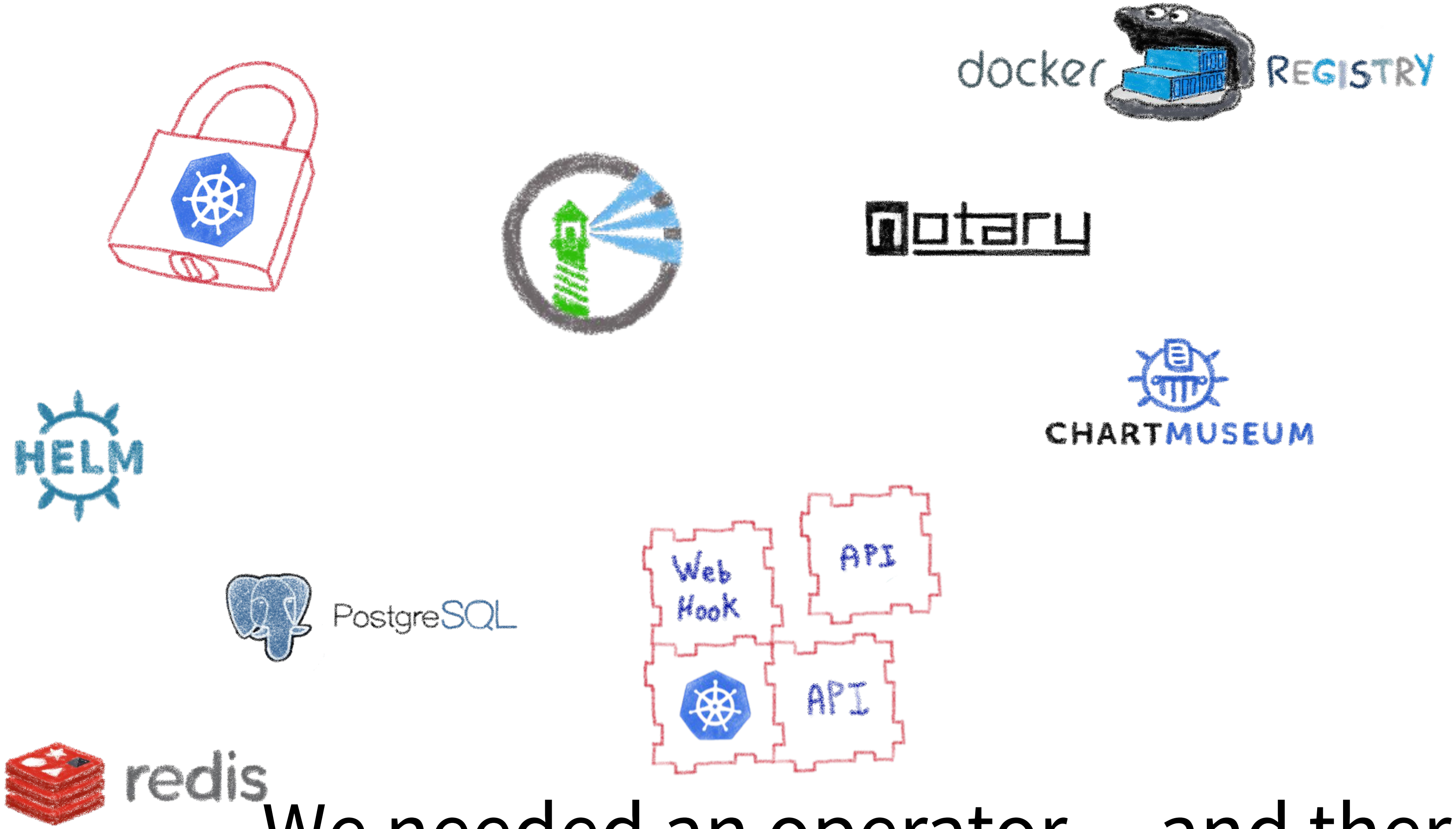
Modularity & Extensibility

APIception

Webhooks

Kubernetes tooling to the rescue

Let's automate it



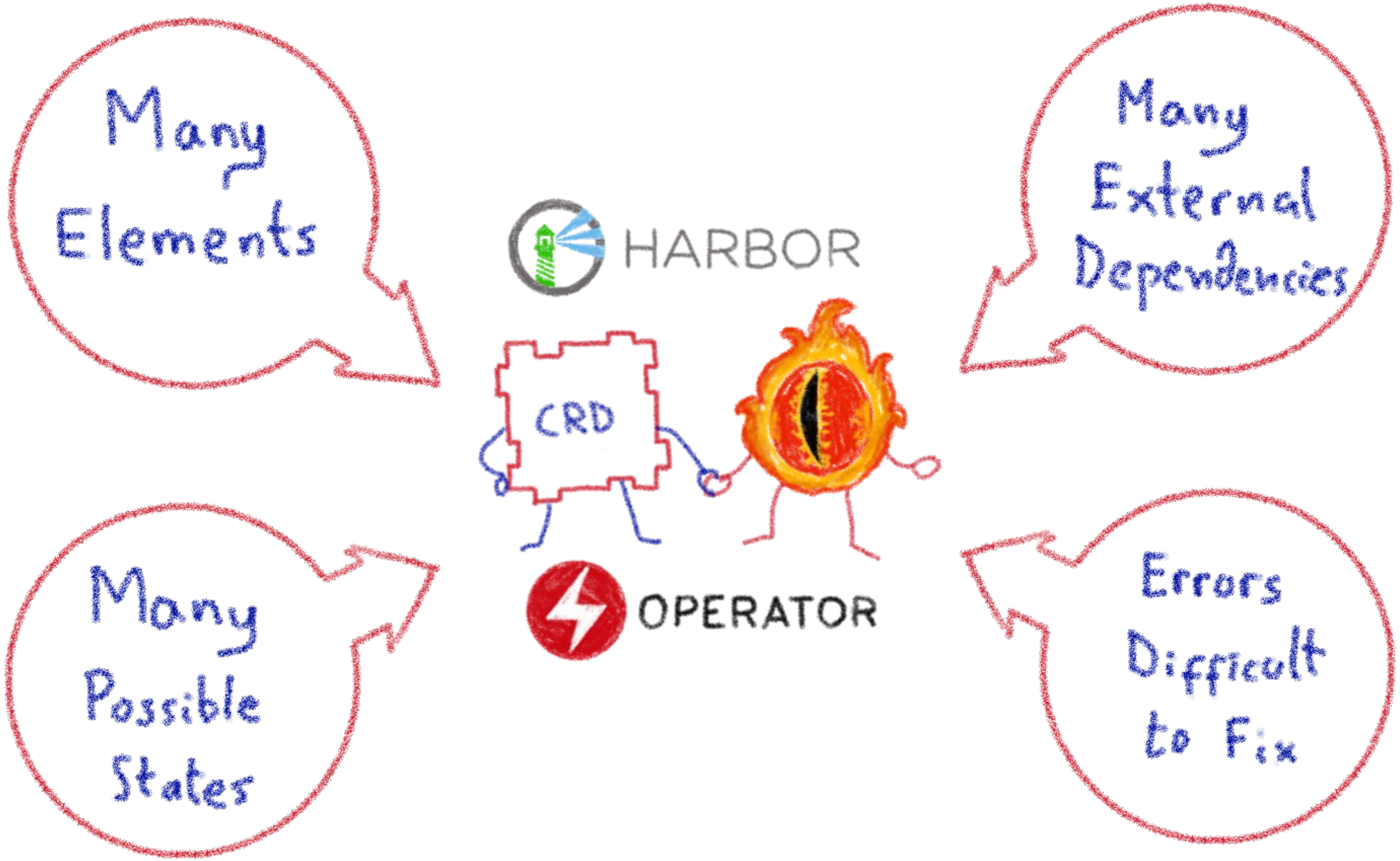
We needed an operator… and there wasn't any

Working with the community

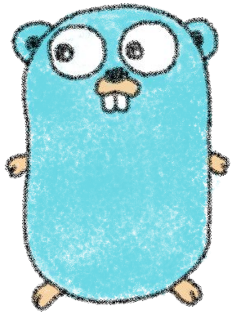
We need an Operator for HARBOR, we are coding it. Interested?
Oh yeah! We would love it!

Harbor community also needed the operator

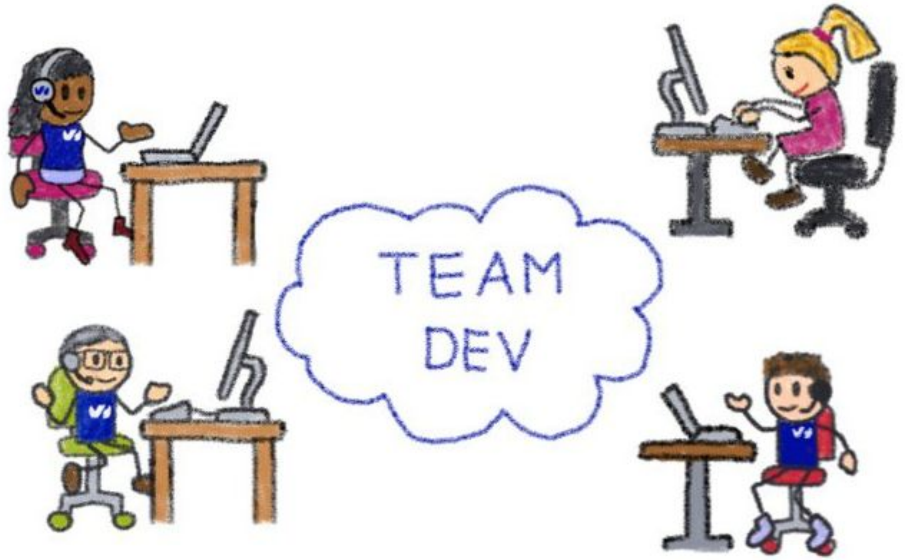
The challenge: reconciliation loop



The Harbor Operator



Written in Go



7 Components
- Config Map
- Secrets
- Ingress
- Certificates
- Deployments
- Services

1 CRD & 1 Controller

kubebuilder

OPENTRACING

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

It's Open Source

Donated by OVHcloud

to the

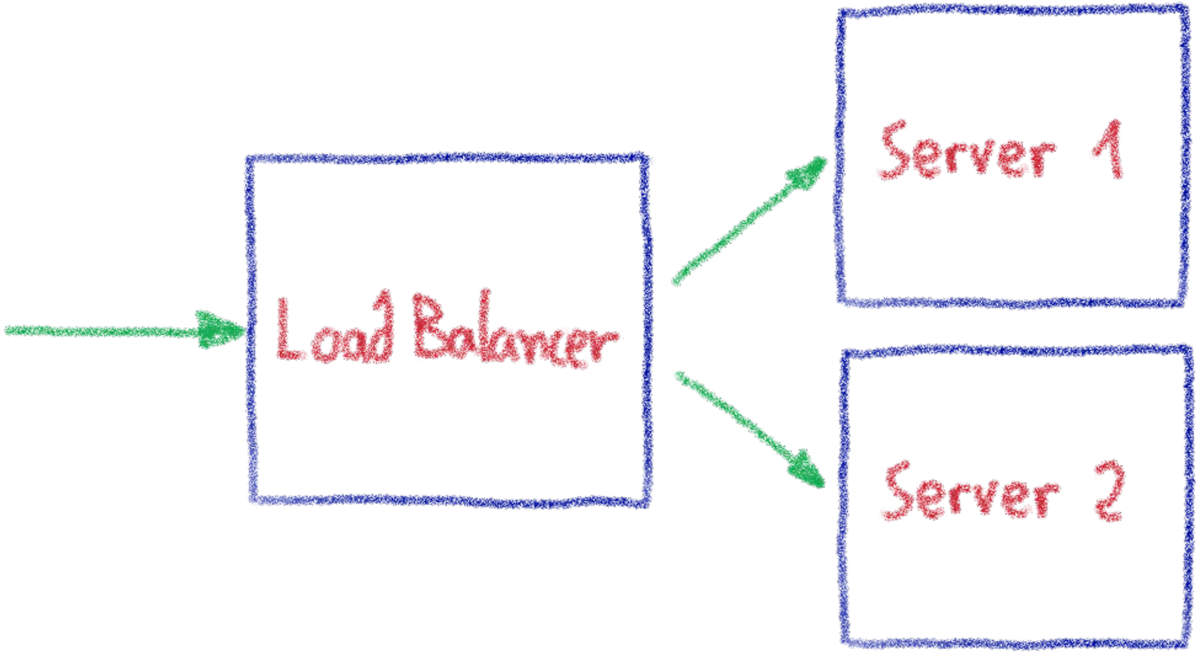
CLOUD NATIVE
COMPUTING FOUNDATION

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

LoadBalancer Operator

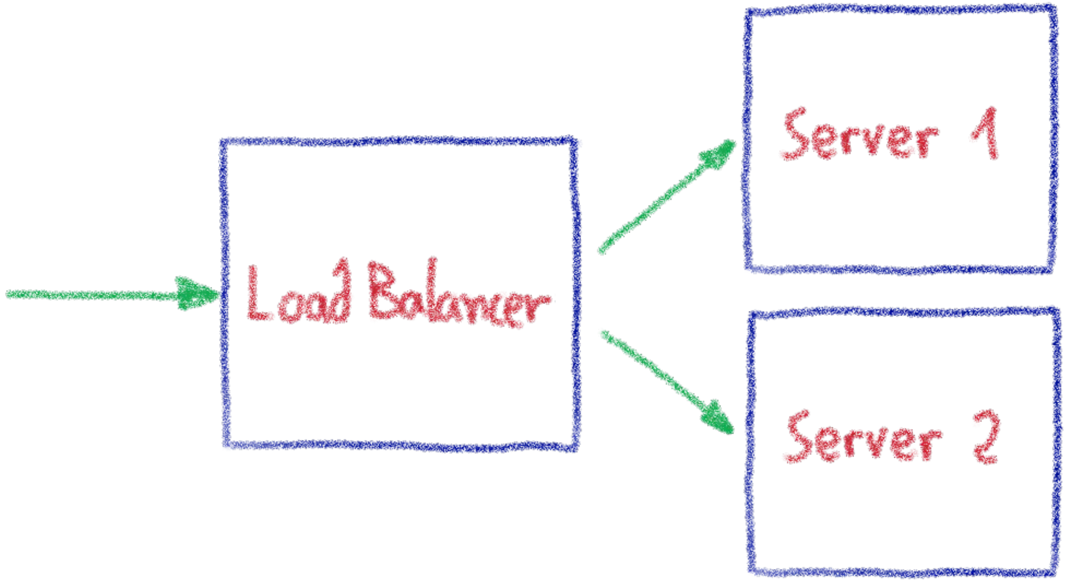
A managed LoadBalancer at scale

Load Balancer: a critical cog



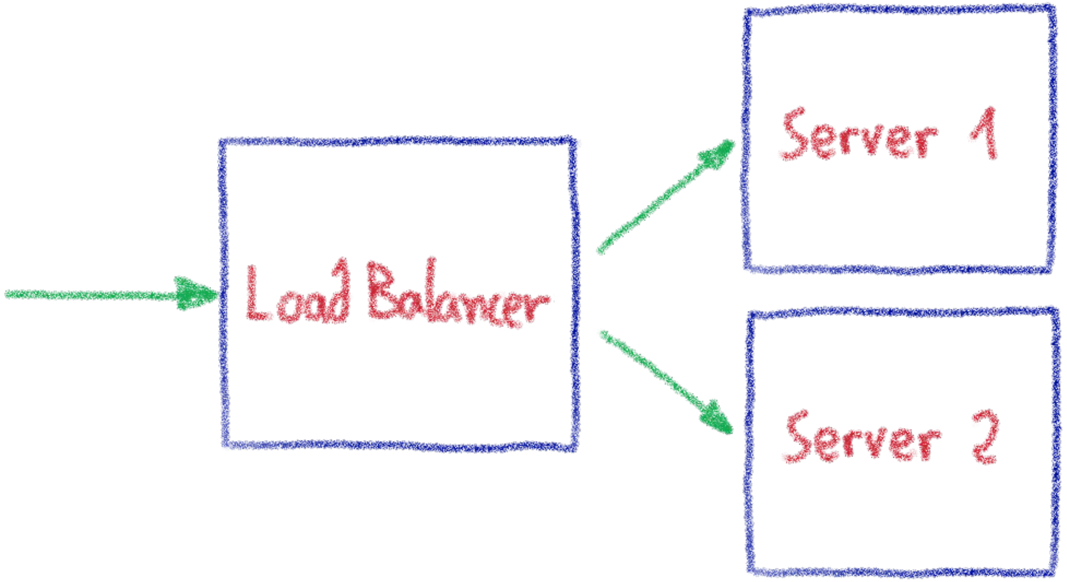
Cornerstone of any Cloud Provider's infrastructure

Our legacy Load Balancer stack



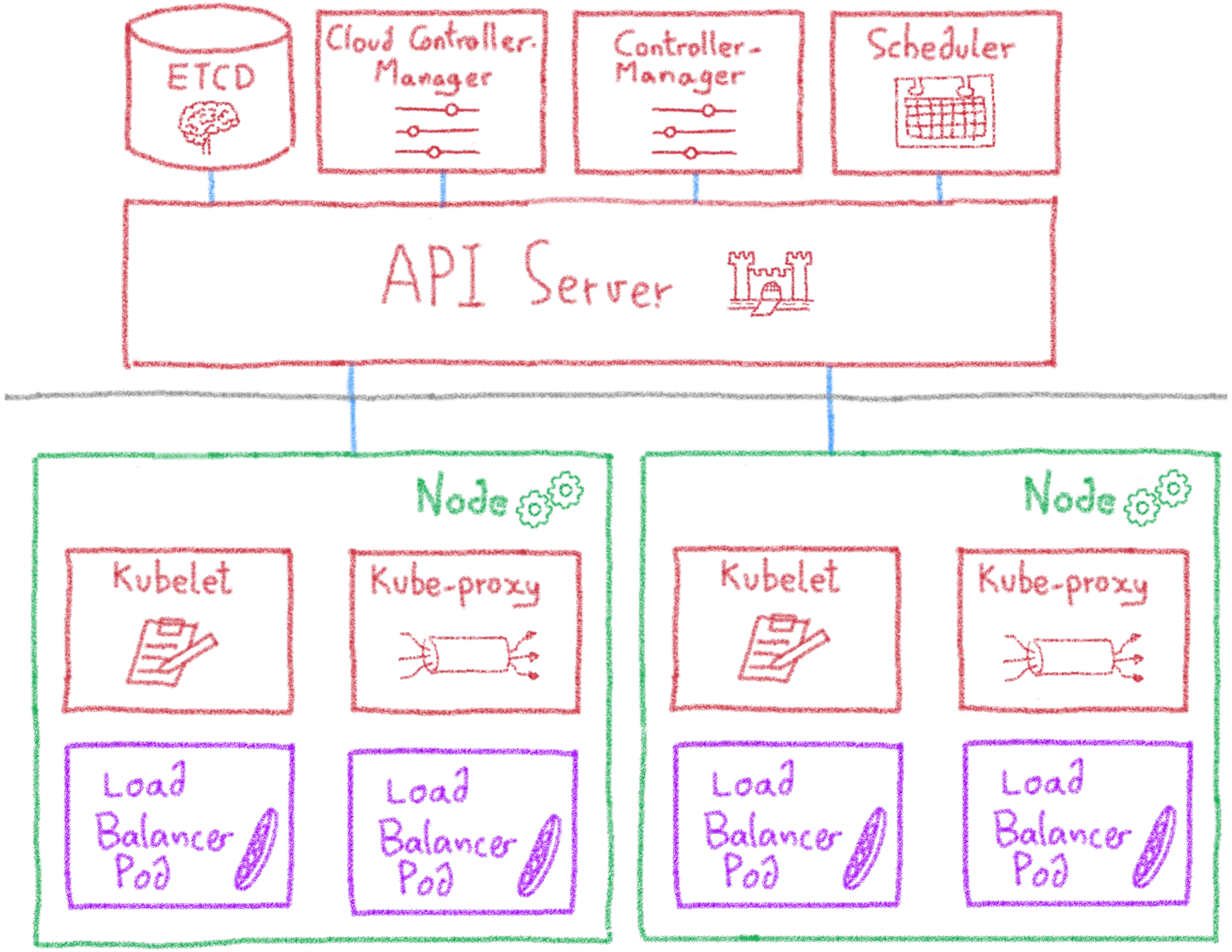
- Excellent performances
  - Built on bare metal servers + BGP
  - Custom made servers tuned for network traffic
- Carry the TLS termination
  - SSL / LetsEncrypt
- Not cloud ready
  - Piloted by configuration files
  - Long configuration loading time
- Custom made hardware
  - Slower to build
  - Needs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

Our needs for a new Load Balancer

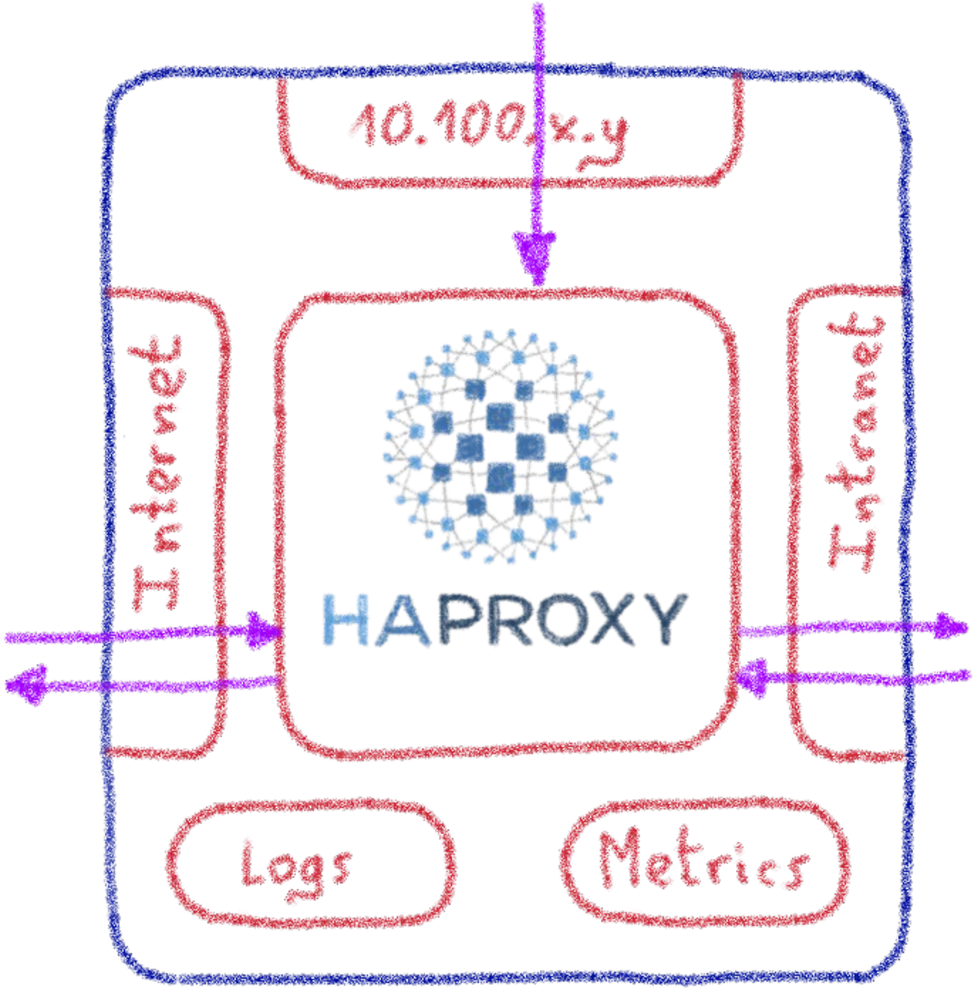


- Supporting mass update
- Quickly reconfigurable
- Available anywhere quickly
- Easily operable
- Integrated into our Public Cloud

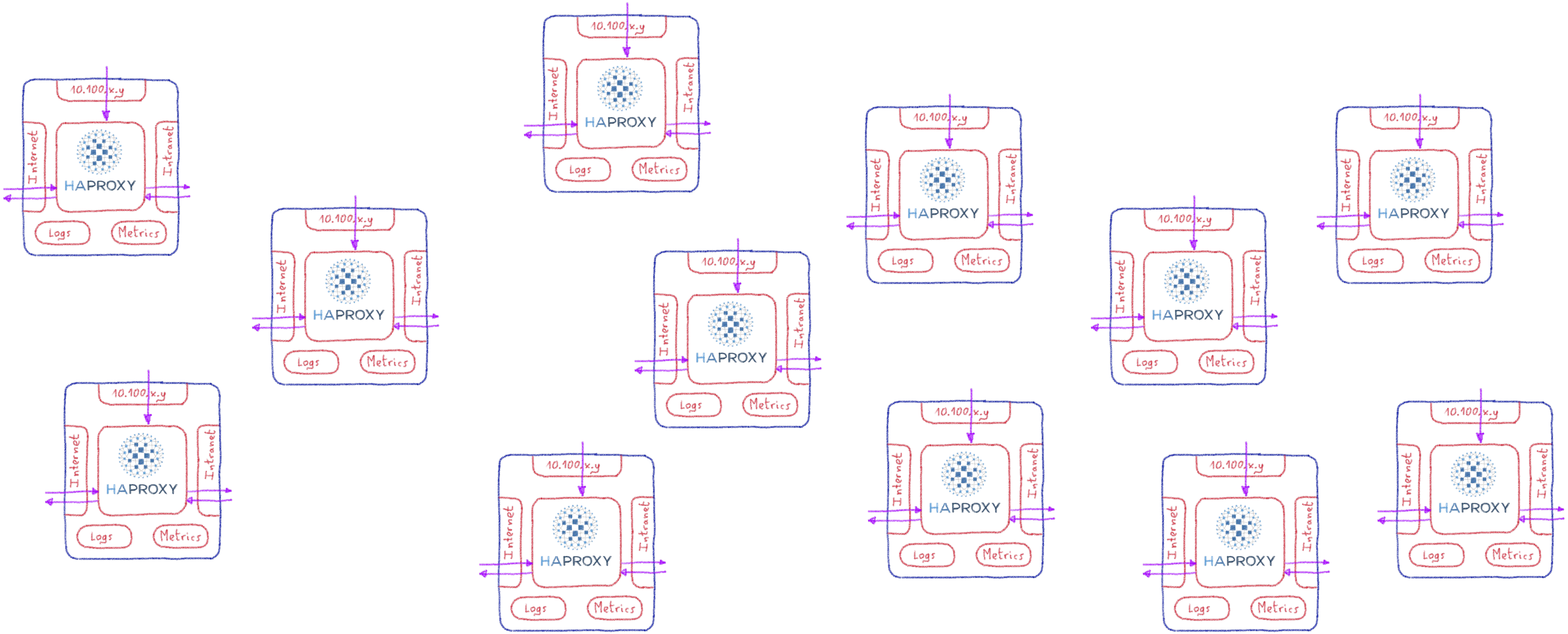
Building it on Kubernetes



A Load Balancer in a pod



Orchestrating one million LBs...



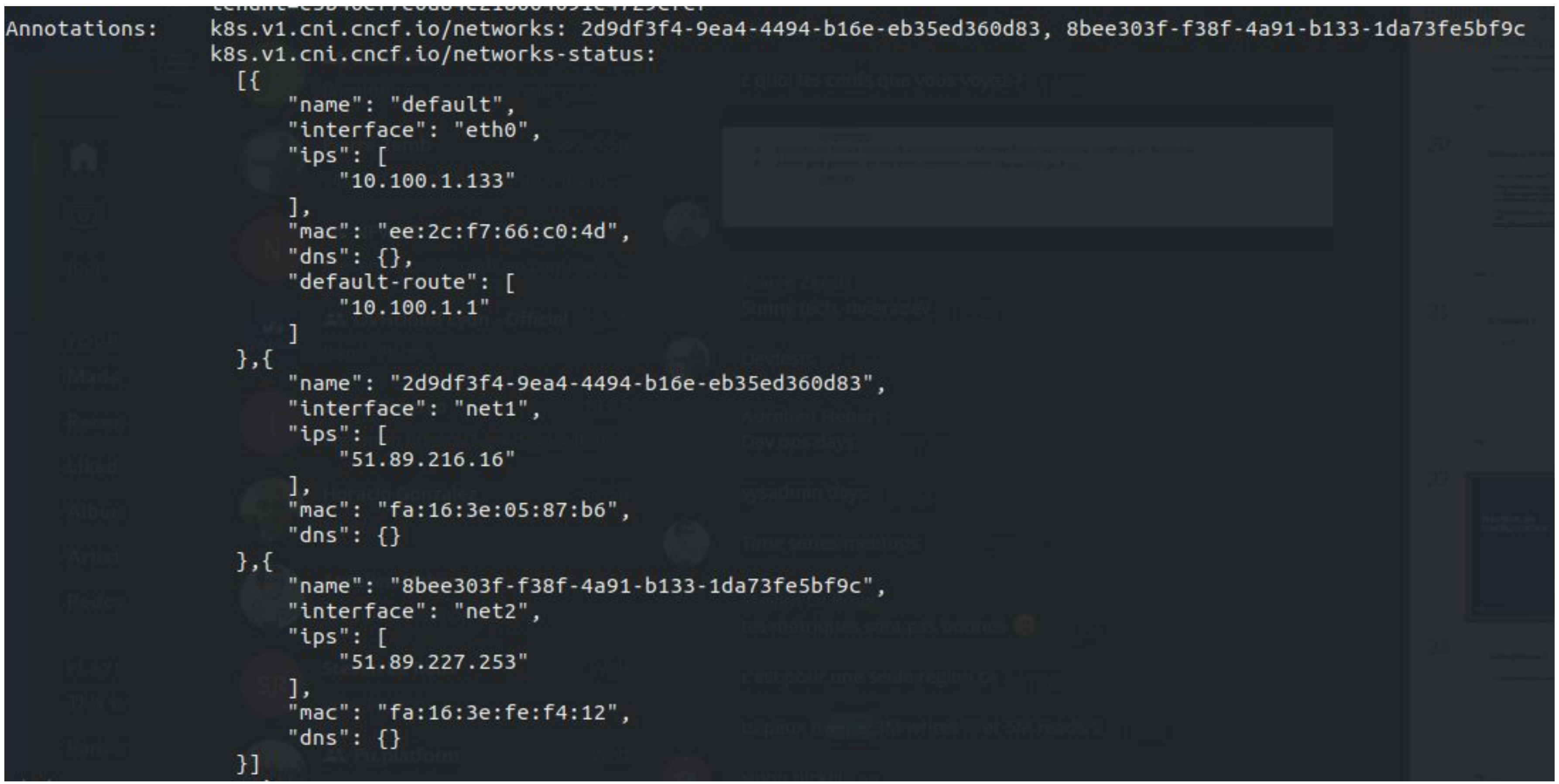
`kubectl apply -f lb` is not an option!

We needed an Operator

Network: multus-cni

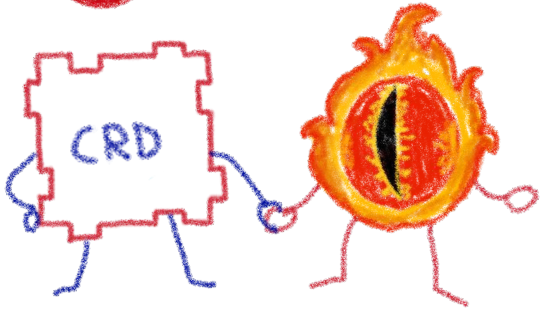
Attaching multiple network interfaces to pods:
Bridge + Host-local

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": {}
                  }]
```

OPERATOR

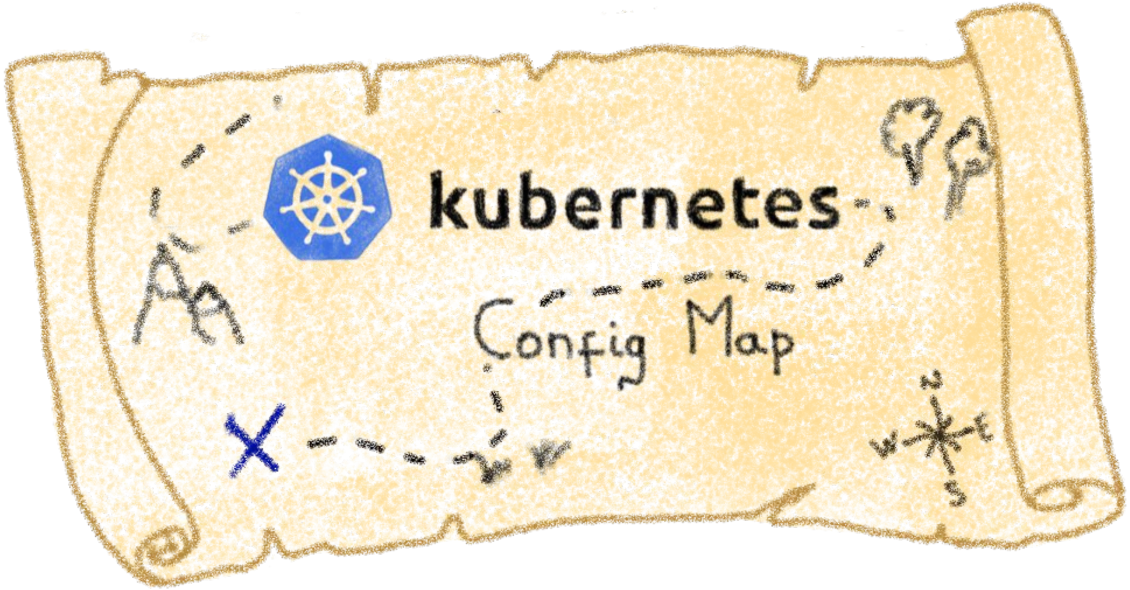


HAPROXY

MULTUS

Using annotations to add interfaces to pod

Config management



Using Config Map

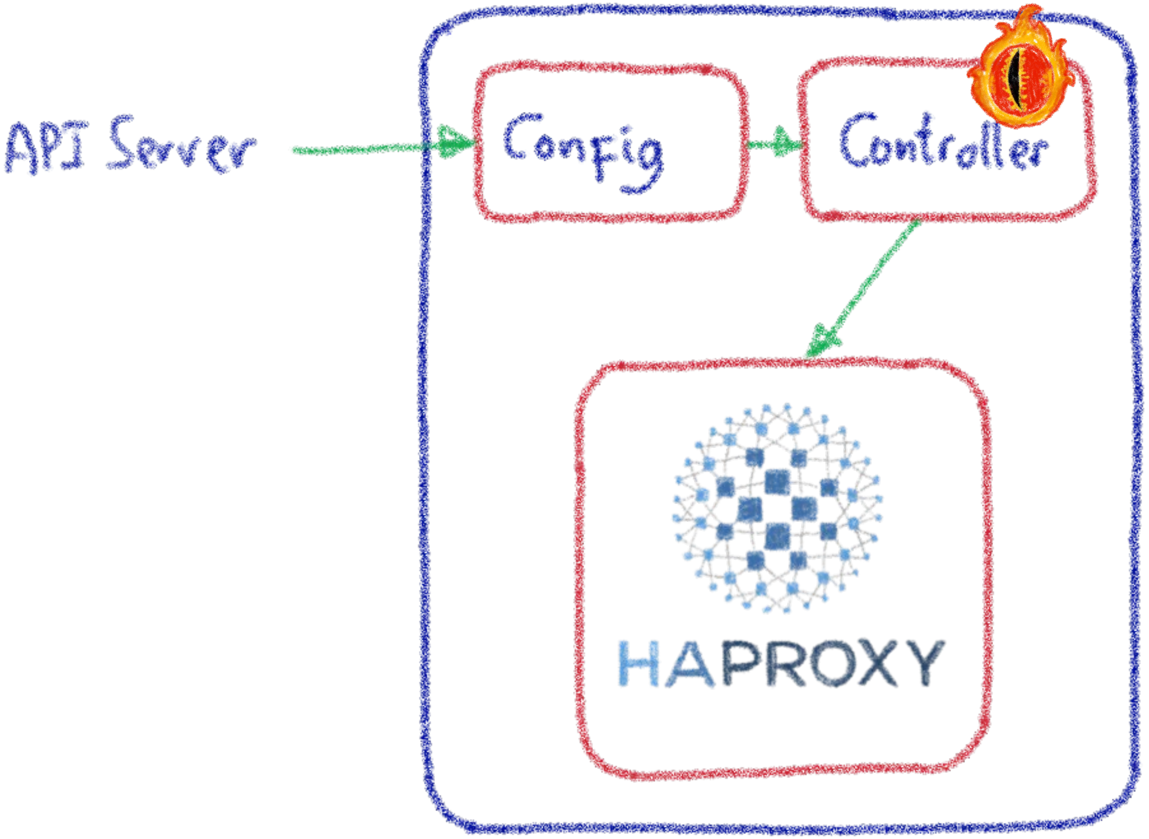
How to detect a change on Config Map files?

Watch + Trigger?

More information on Config Map working

`martensson.io/go-fsnotify-and-kubernetes-configmaps`

A Controller to watch and trigger



Observability

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

That's all, folks!

Thank you all!