OVHcloud

DEVOPS.BARCELONA

Let's dive into Kubernetes operator creation

Horacio Gonzalez

2023-11-09

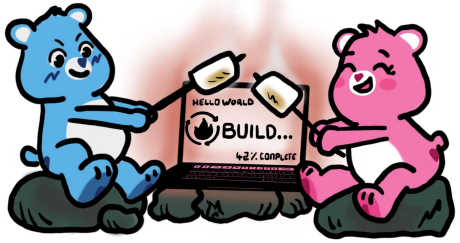
Who are we?

Introducing myself and introducing OVHcloud

Horacio Gonzalez

@LostInBrittany

Spaniard Lost in Brittany



Finist Devs

GDG Finistère

OVHcloud

DevRel Leader

Experts 2023

Experts 2023

OVHcloud

PRIVATE CLOUD

PUBLIC CLOUD

BARE METAL

WEB CLOUD

PAAS

33 Data Centers
in 13 locations

44 Points of Presence
on a 36 TBPS Bandwidth
Network

2800+ Employees
worldwide

117K+ Private Cloud
VM running

340K Public Cloud
Instances running

400K Physical Servers
in our data centers

1 Million+ Servers
produced since 1999

1.6 Million Customers
across 140 countries

308 Million euros adjusted
EBITDA (2022)

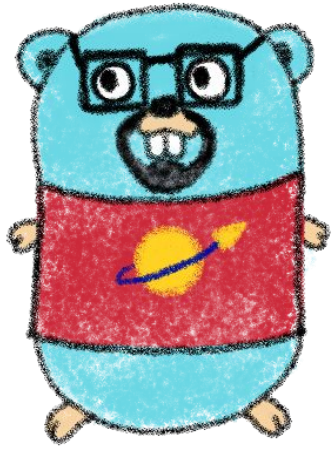
P.U.E. 1,10 – 1,30
Energy efficiency indicator

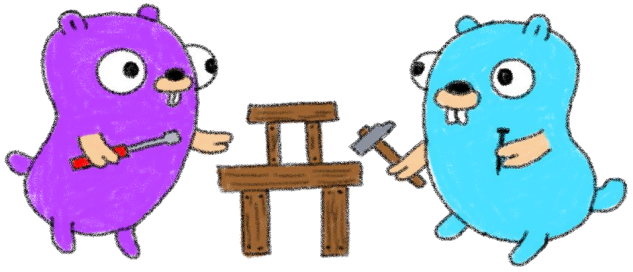
Over 20 Years in Business
Disrupting since 1999

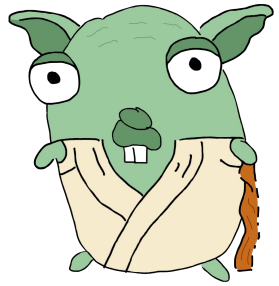
@LostInBrittany

Warning

Gophers, gophers everywhere!



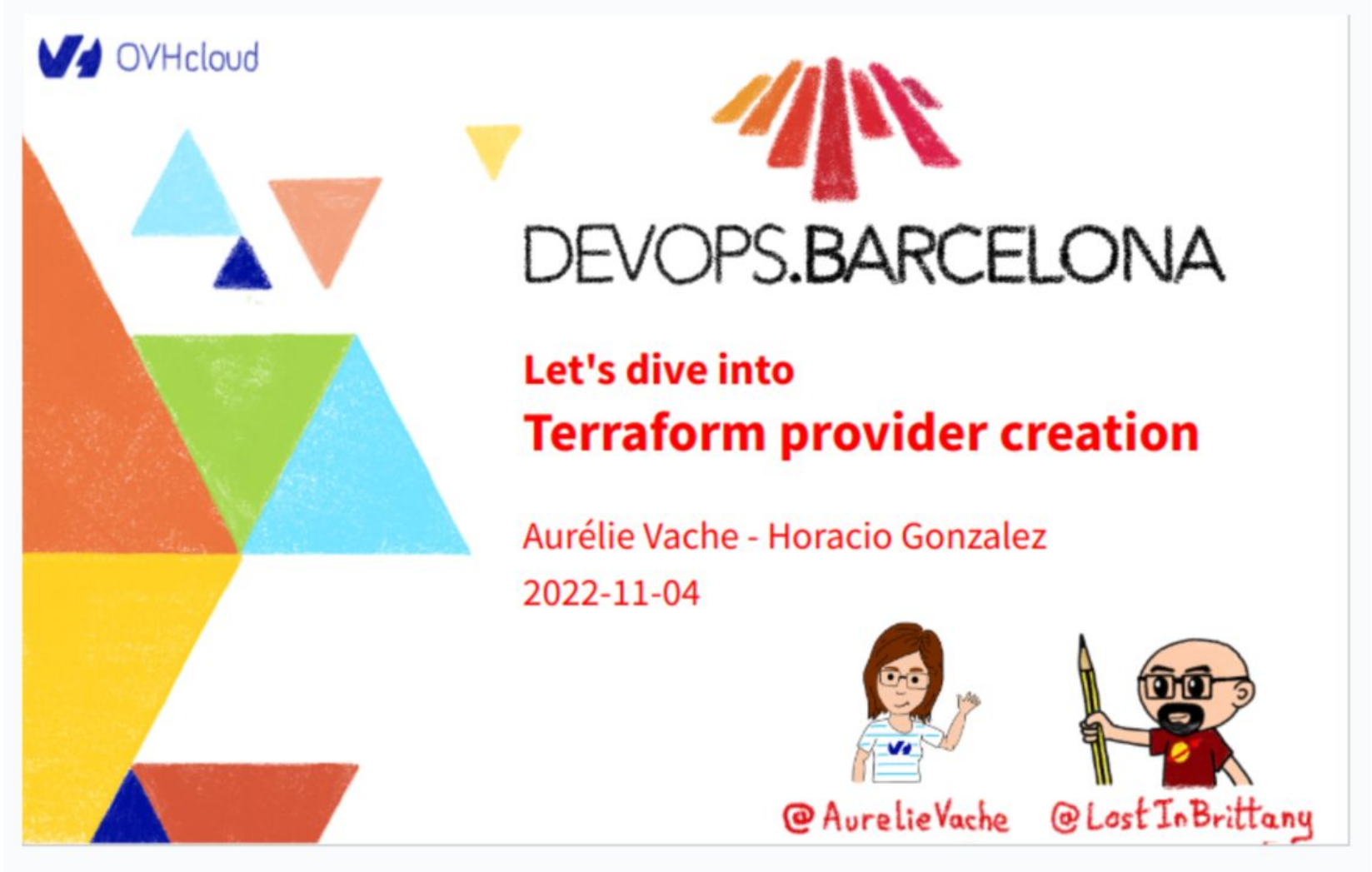




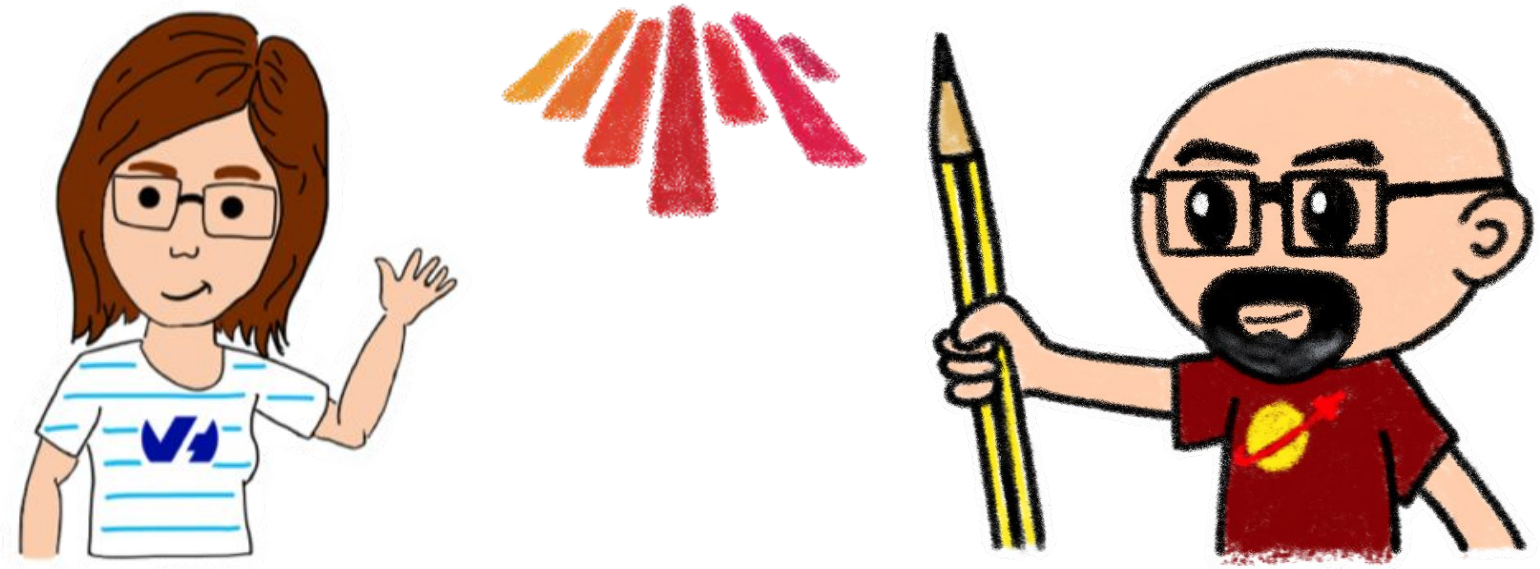
OVHcloud

@LostInBrittany

Last year in DevOps Barcelona



We proposed a sequel for this year



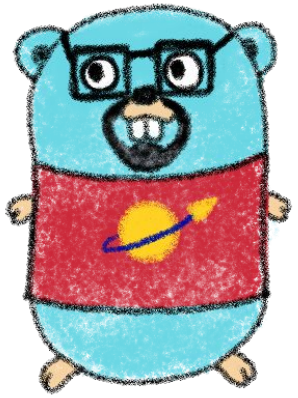
Let's dive into Kubernetes operator creation

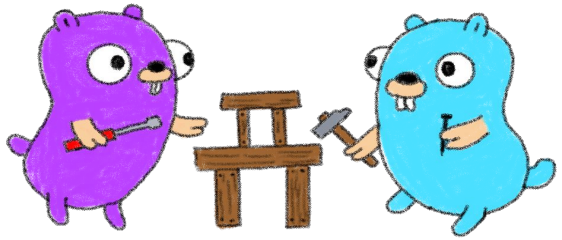
DEVOPS.BARCELONA

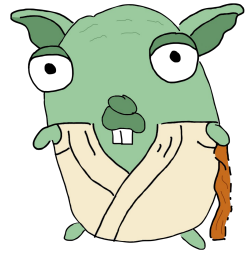
But at the end Aurélie couldn't come

So I must do it alone… Wish me luck!

And why Gophers?



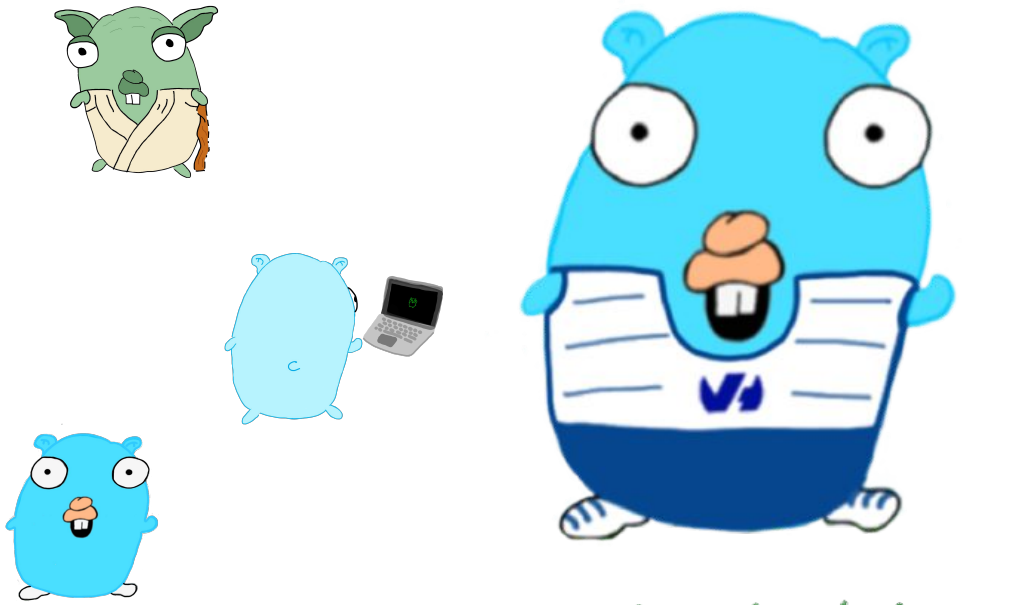


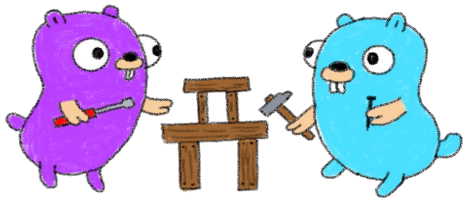


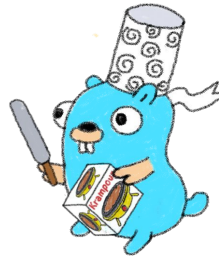
Because we love Gophers, of course!

And because Golang and Kubernetes are so linked…

Credit where it is due



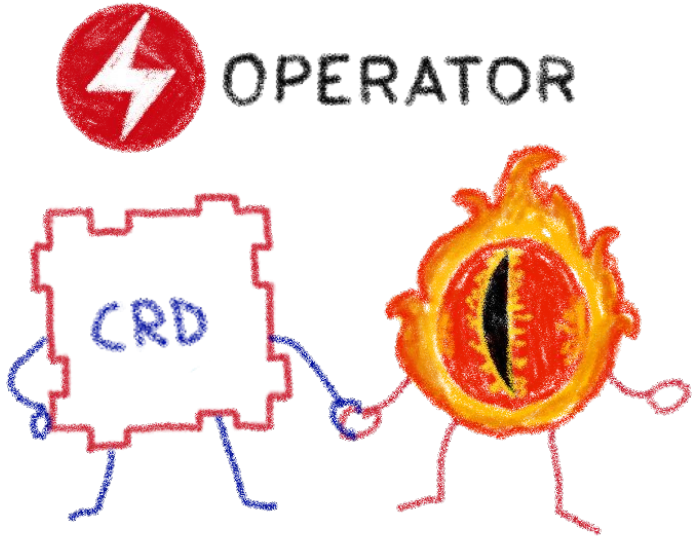




All the gophers you will see are drawn by Aurélie and Horacio, and are based on the Go mascot designed by Renee French which is licensed under CC BY 3.0.

Kubernetes operators

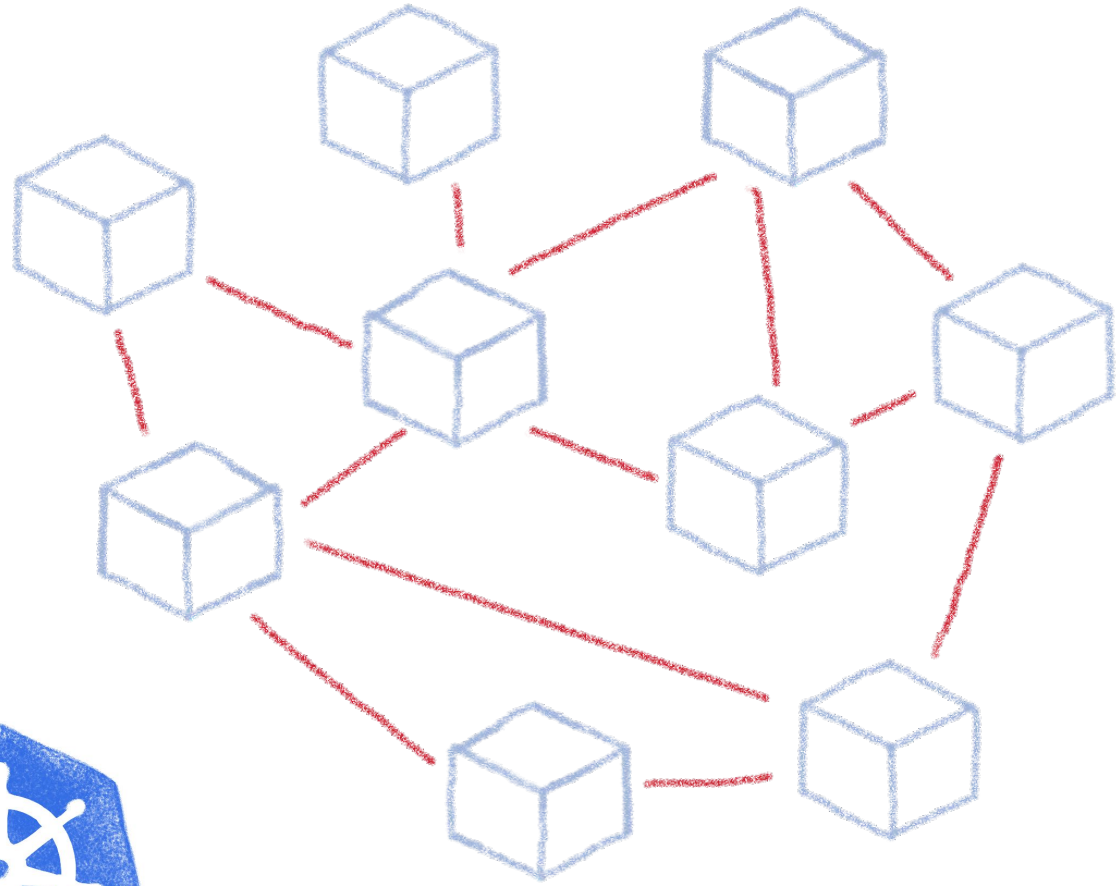
Helping to tame the complexity of K8s Ops



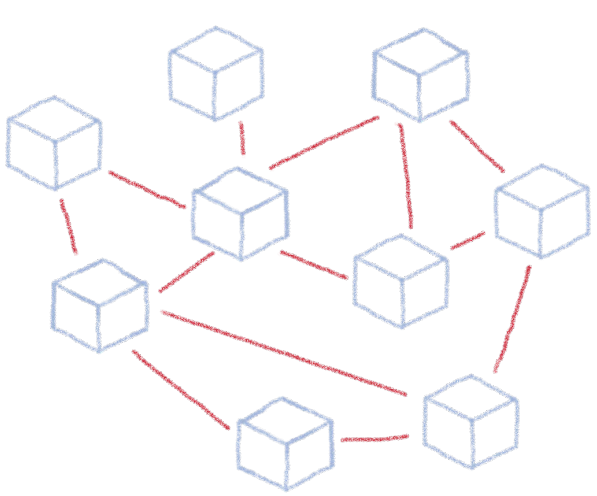
OVHcloud

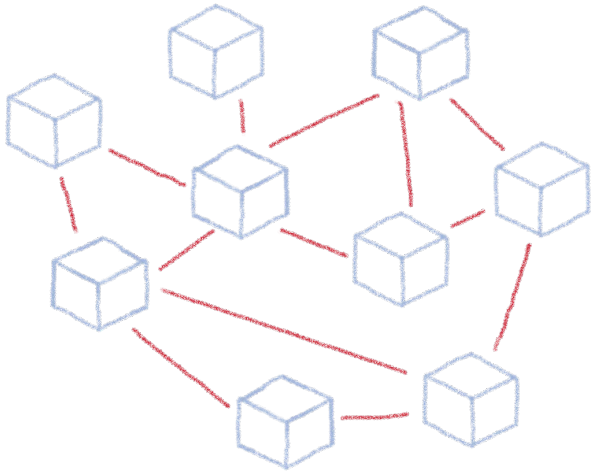
@LostInBrittany

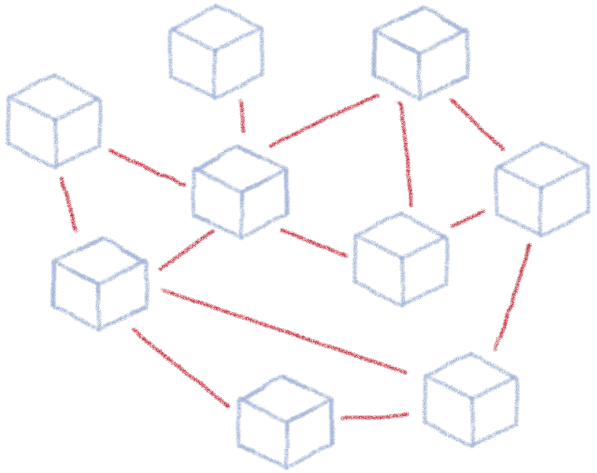
Taming microservices with Kubernetes



What about complex deployments







Ingress

Services

Deployments

Pods

Sidecars

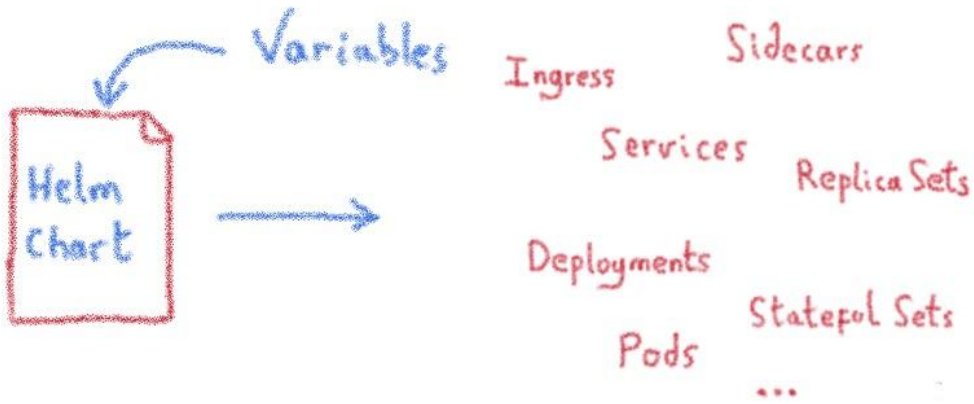
Replica Sets

Statepul Sets

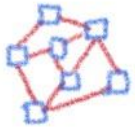
@LostInBrittany

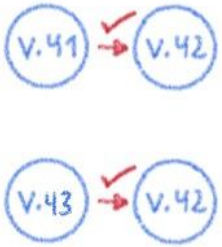
Tools like Helm helps with complexity

A package manager for Kubernetes

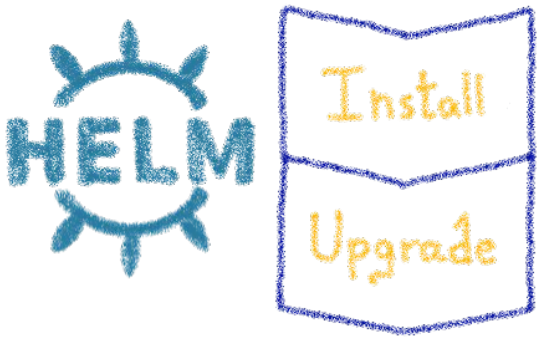


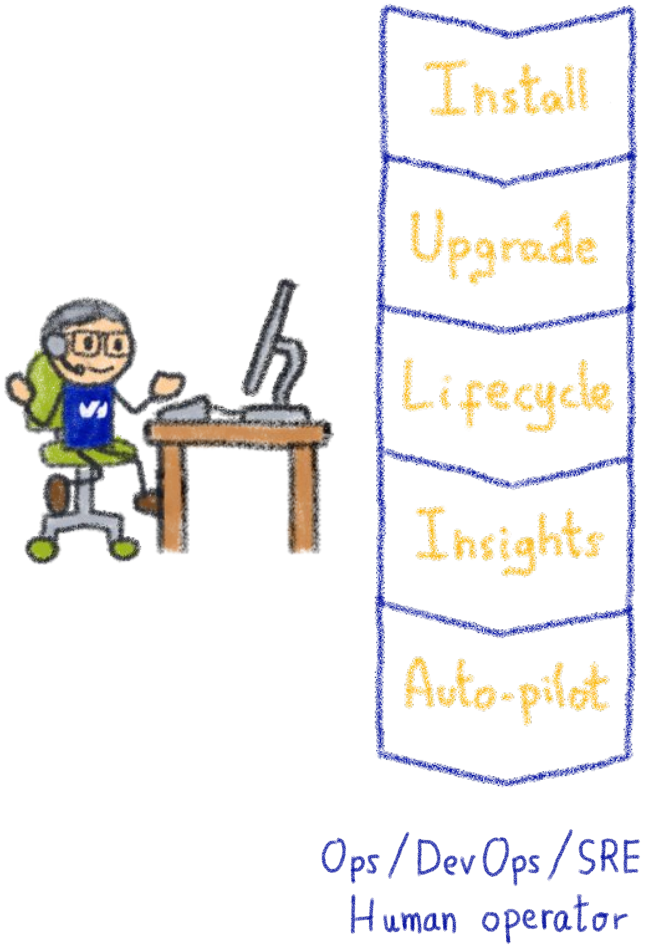
- Manage complexity
- Simple sharing
- Easy upgrades
- Easy rollbacks





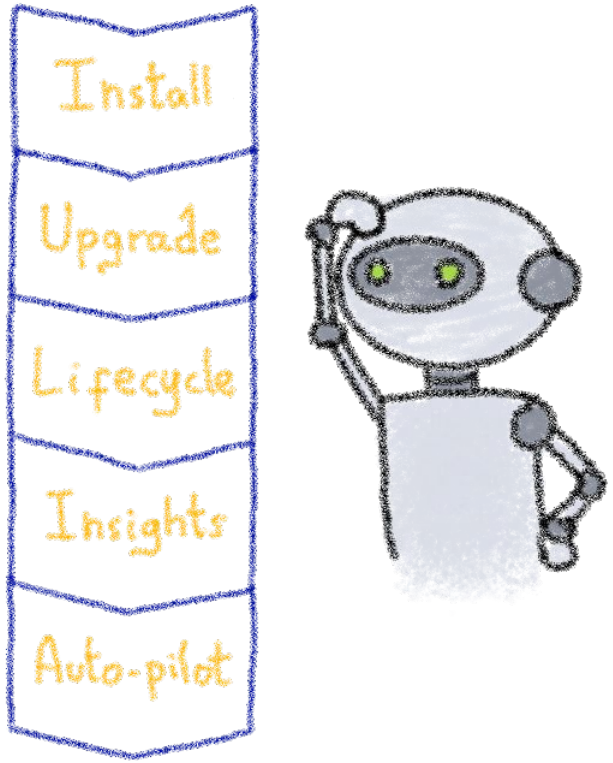
Helm Charts are configuration





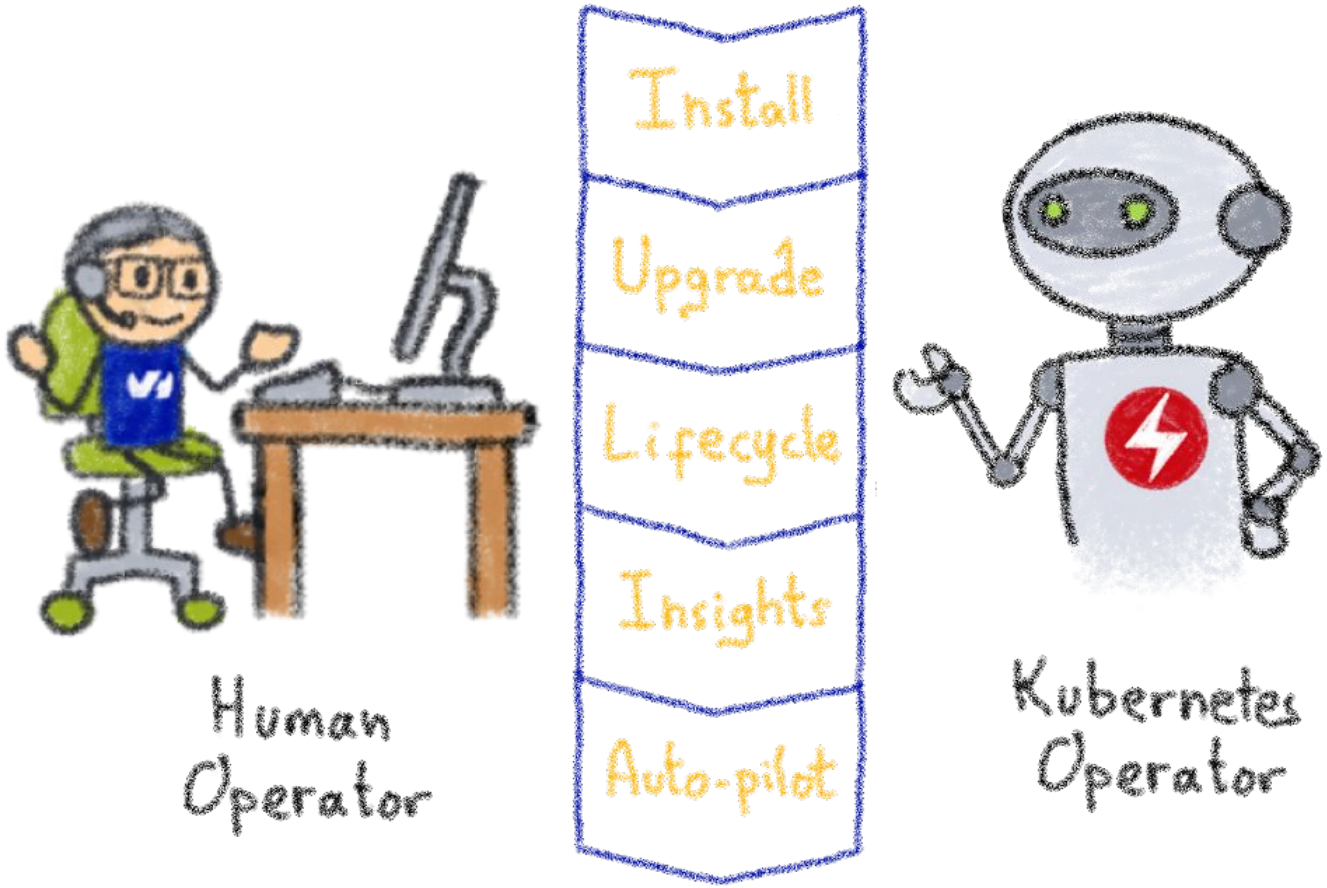
Operating is more than installs & upgrades

Kubernetes is about automation



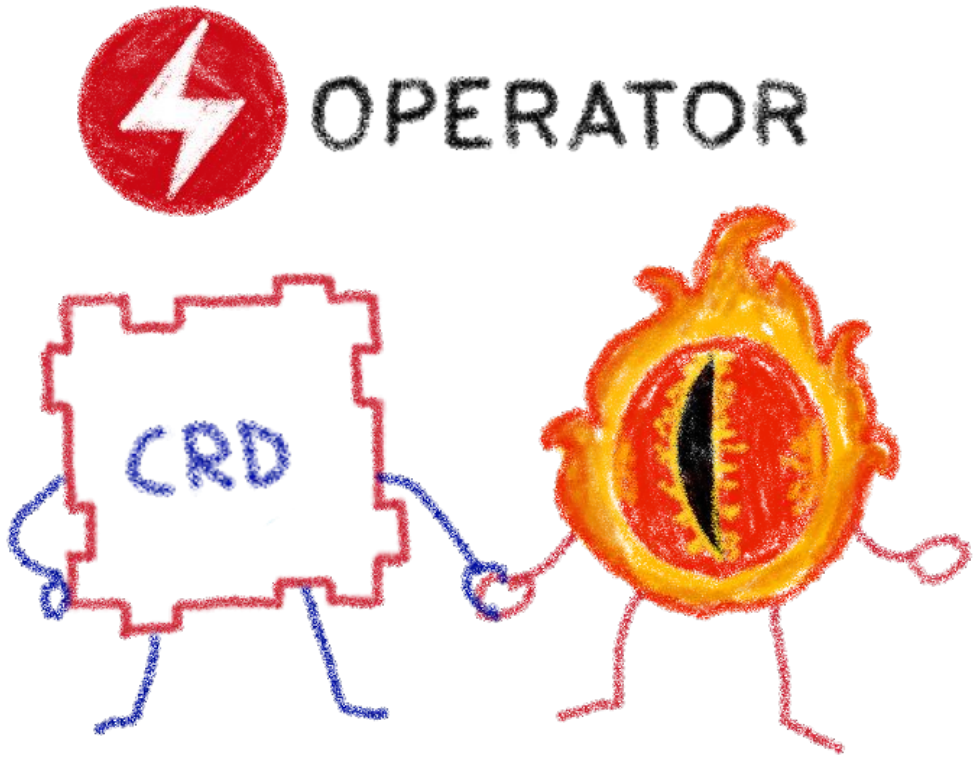
How about automating human operators?

Kubernetes Operators



A Kubernetes version of the human operator

Building operators



Basic K8s elements: Controllers and Custom Resources

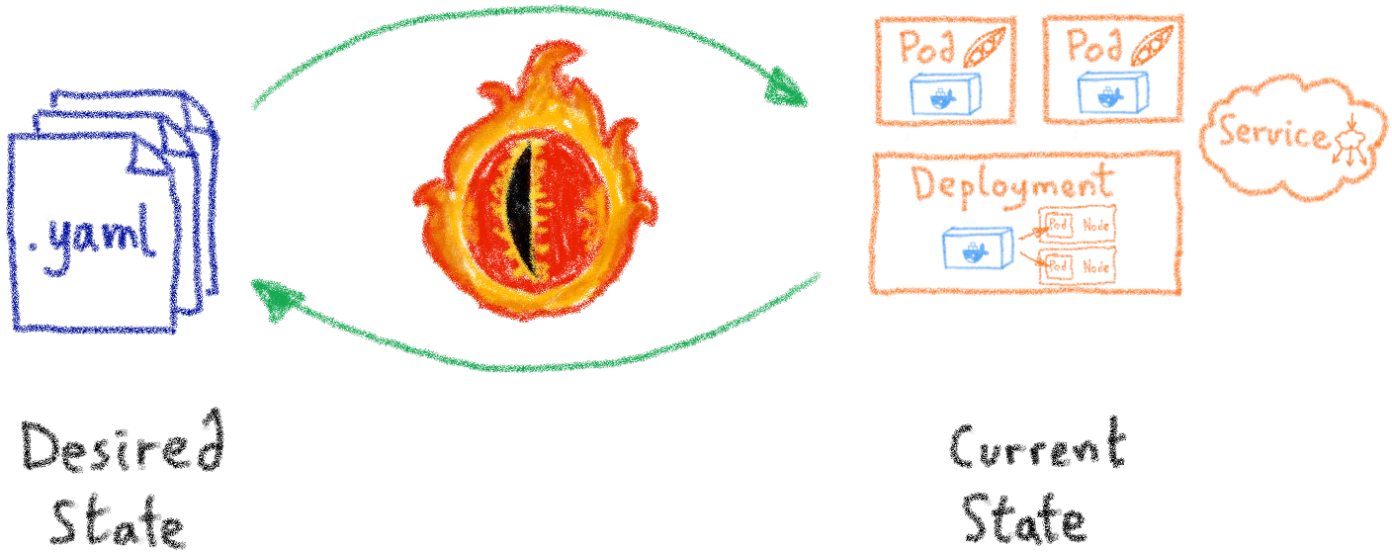
Kubernetes Controllers

Keeping an eye on the resources

OVHcloud

@LostInBrittany

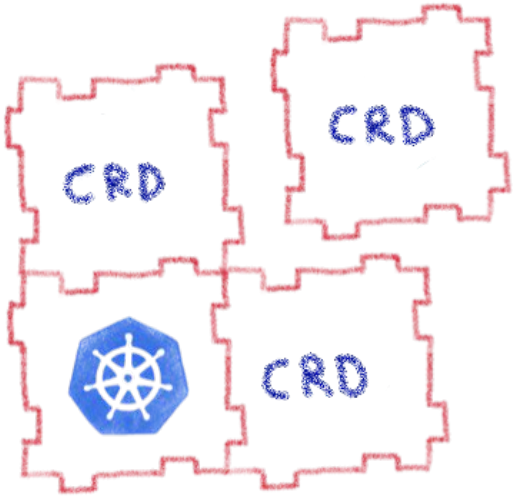
A reconcile loop



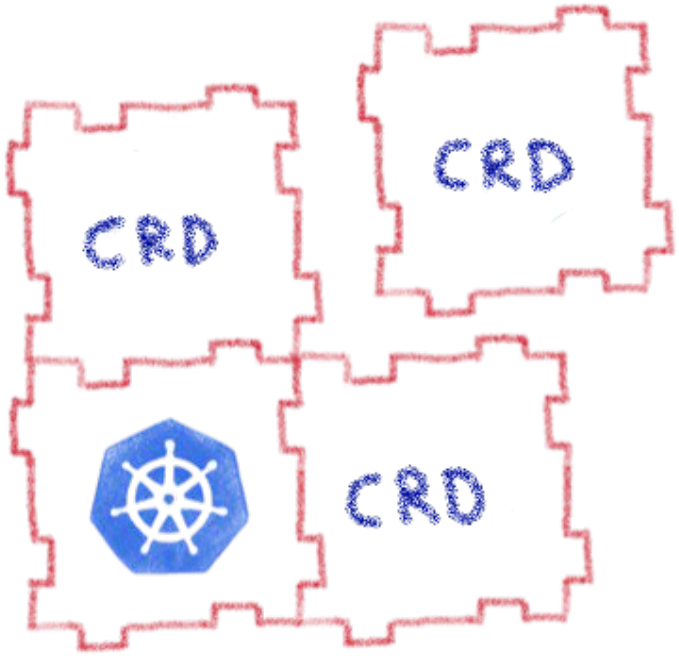
Controllers watch the state of the cluster,
and make or request changes where needed

Custom Resource Definitions

Extending Kubernetes API

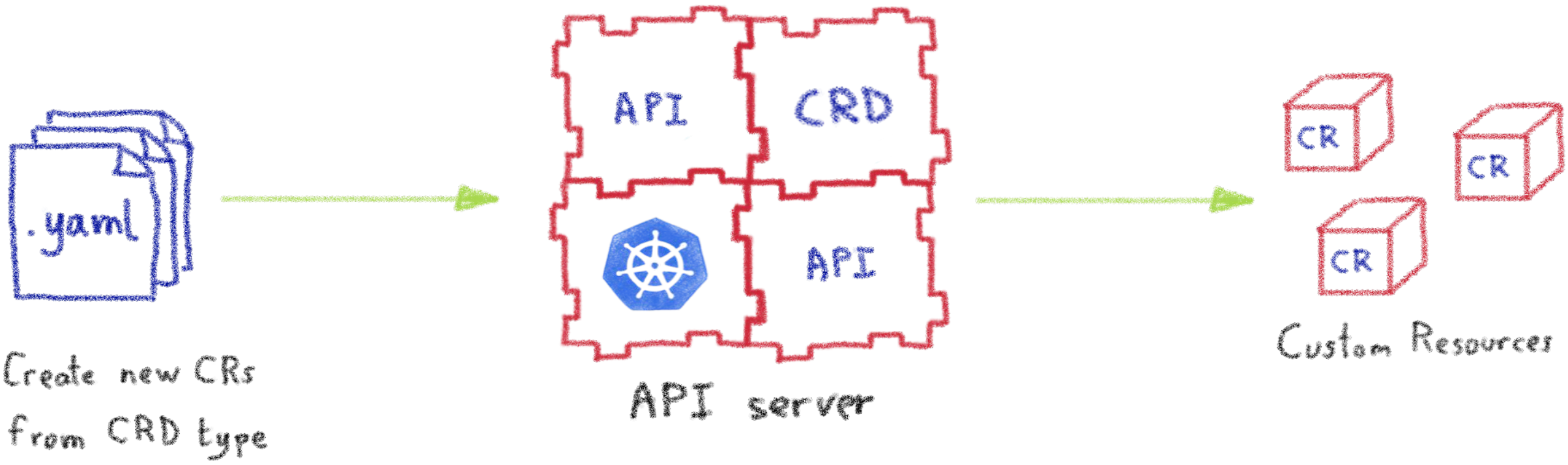


Extending Kubernetes API



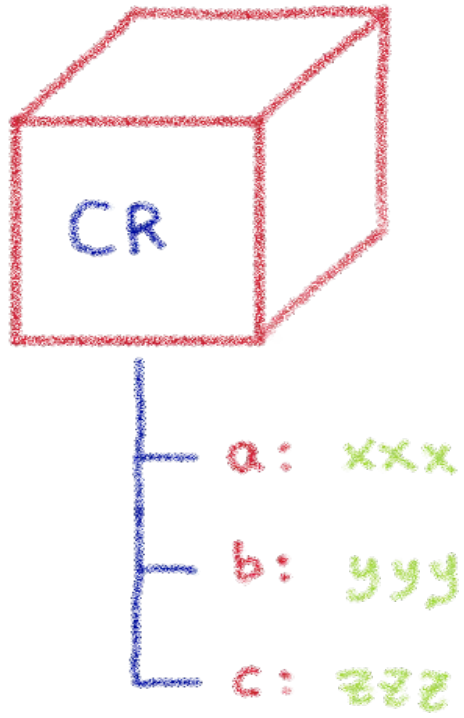
By defining new types of resources,
internal or external to the cluster

With a CRD you can create CR in the cluster



They are the blueprints of the Custom Resources

Custom Resources are simply data



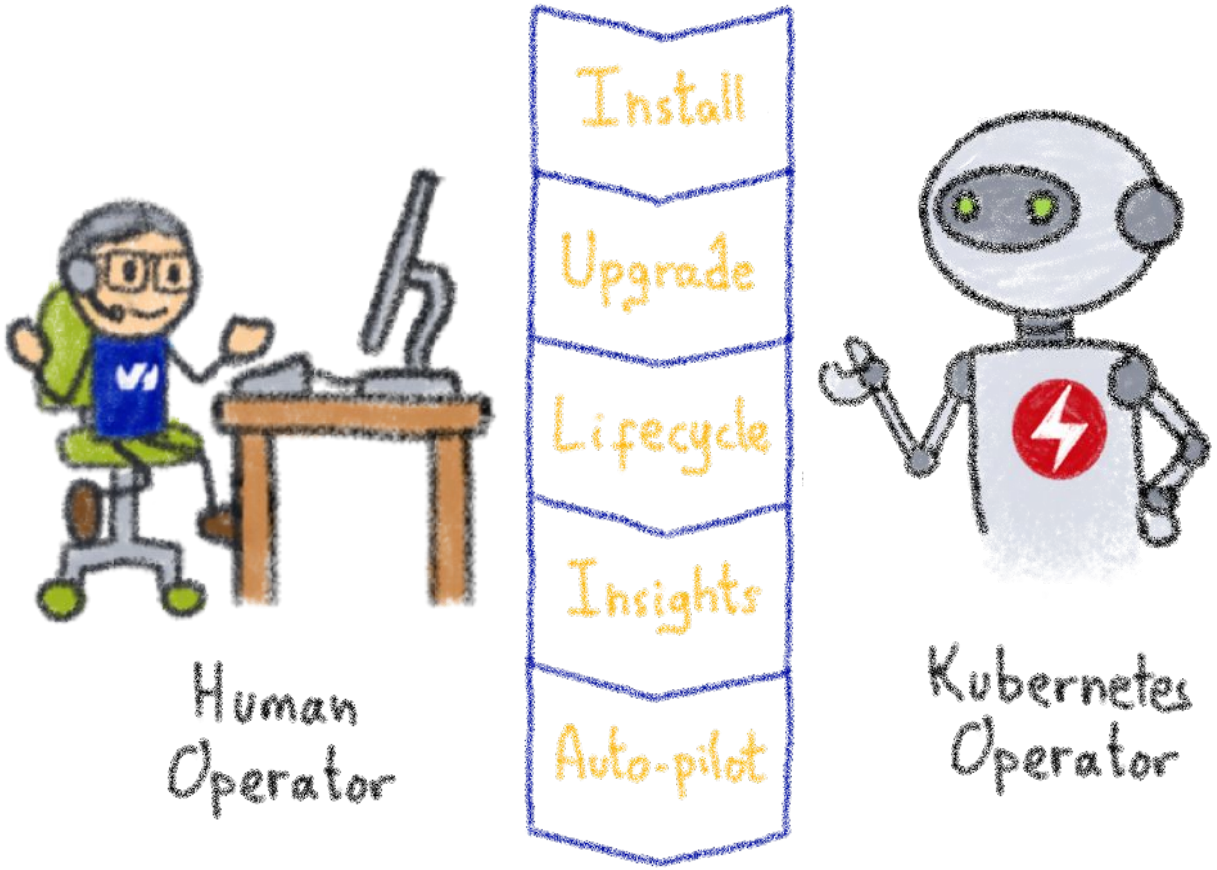
Only data, properties, no logic

All the logic must be in the Controller

Kubernetes Operator

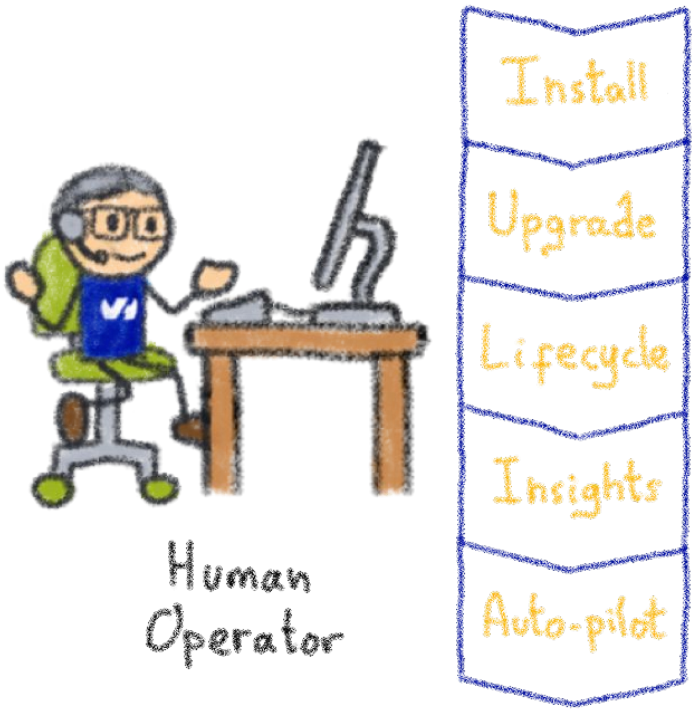
Automating operations

What's a Kubernetes Operator?



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

Example: databases

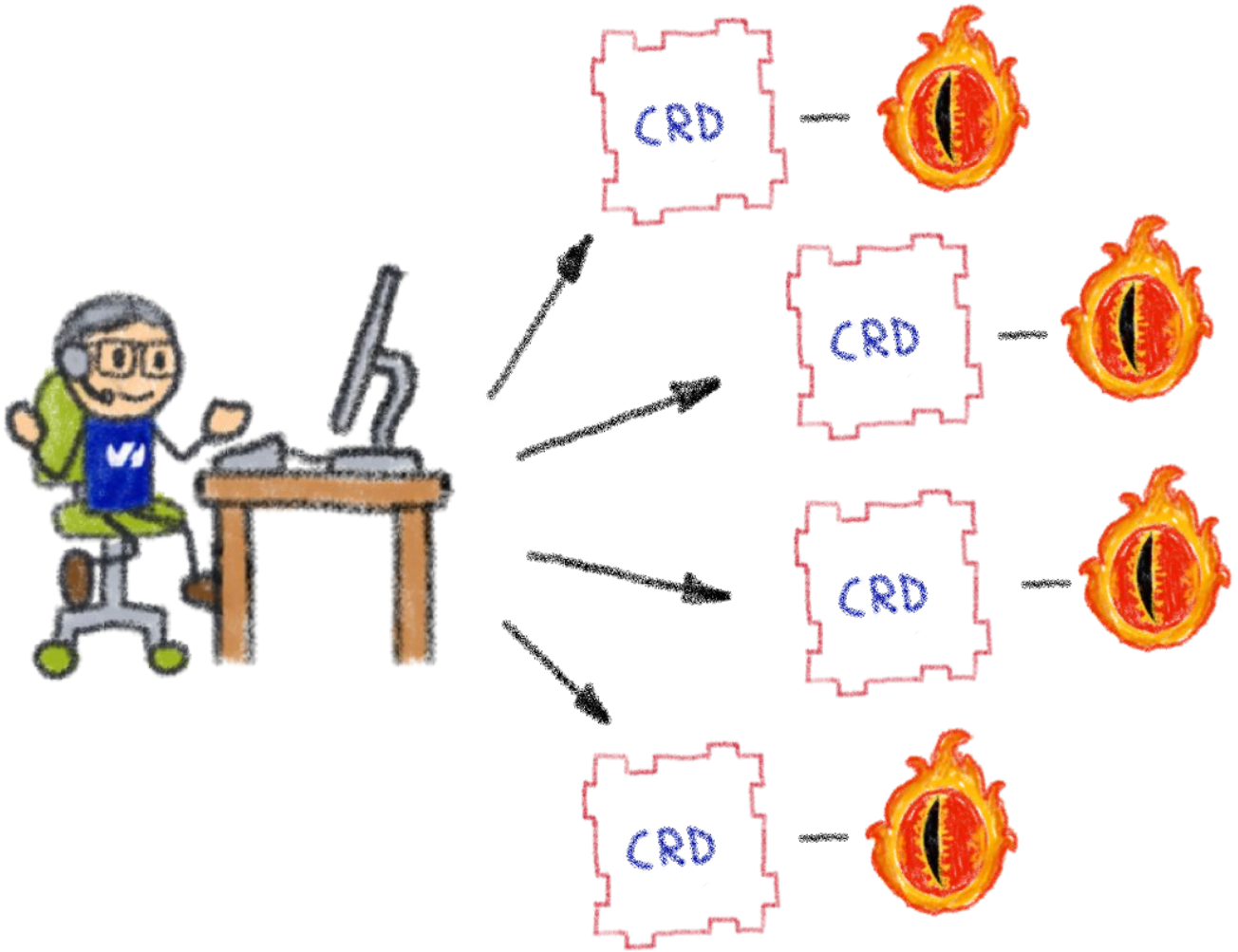


PostgreSQL

redis

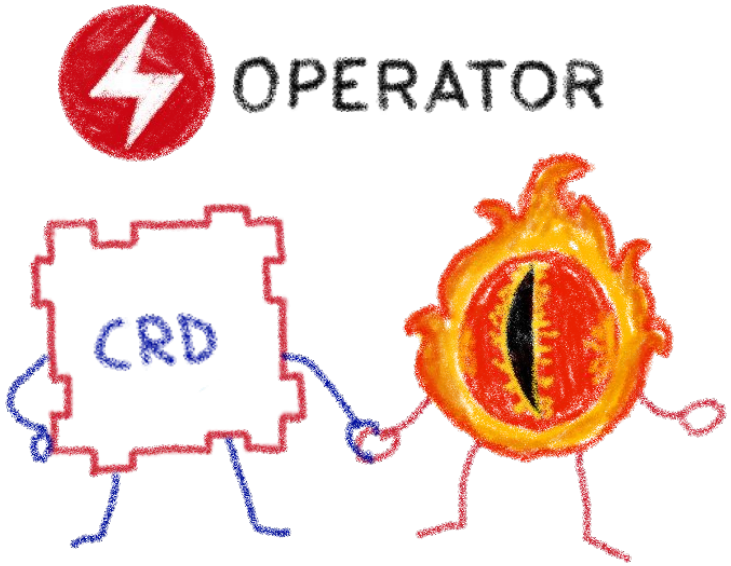
Things like adding an instance to a pool,
doing a backup, sharding...

Knowledge encoded in CRDs and Controllers



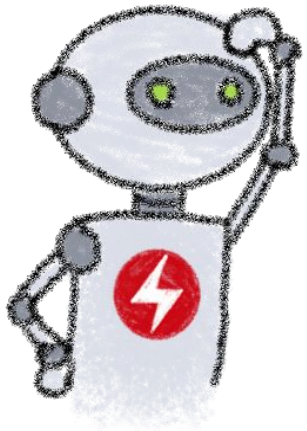
Encapsulating business logic in CRDs & Controllers

Custom Controllers for Custom Resources



Operators implement and manage Custom Resources using custom reconciliation logic

Operator Capability Model

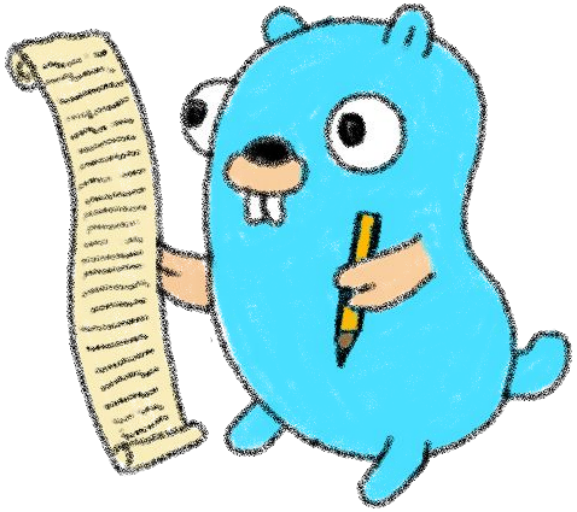


OPERATOR CAPABILITY MODEL

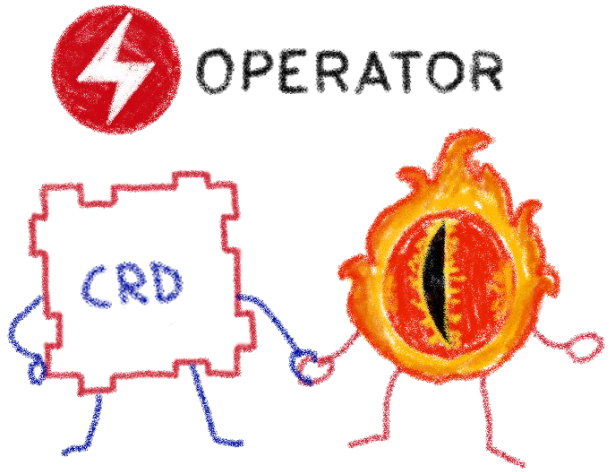
Gauging the operator maturity

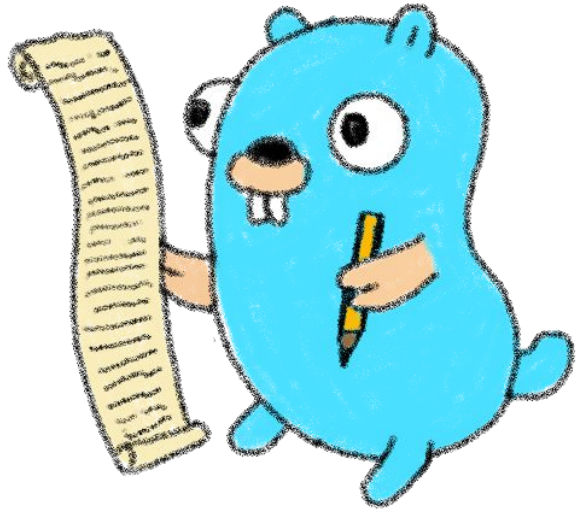
How can we write Operators?

Which language? Any framework?



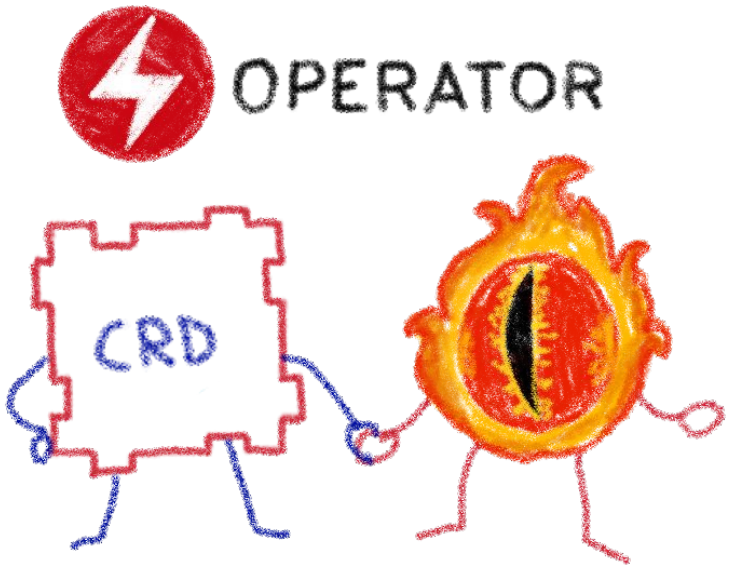
They are simply pods and manifests





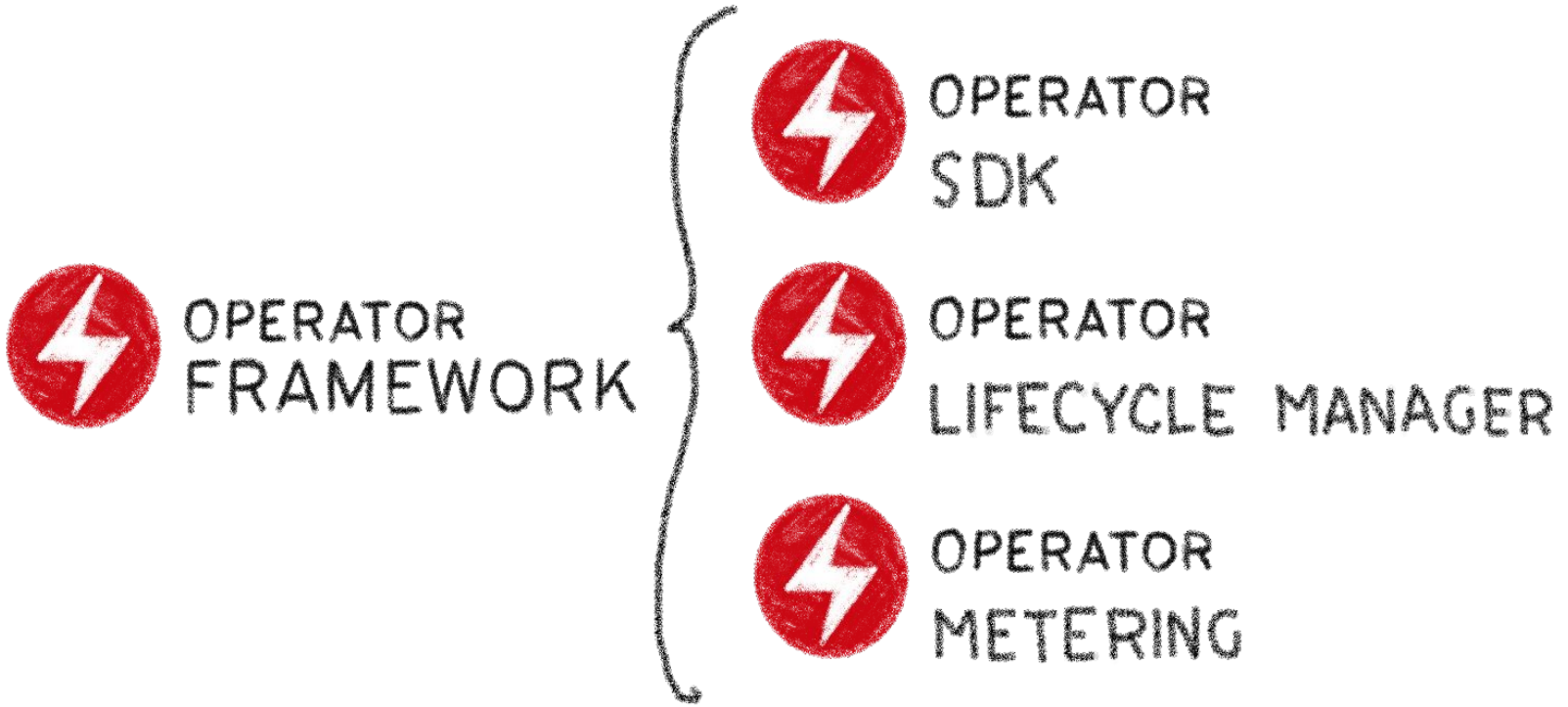
You can simply call Kubernetes APIs
or use a compatible client

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

The Operator Framework

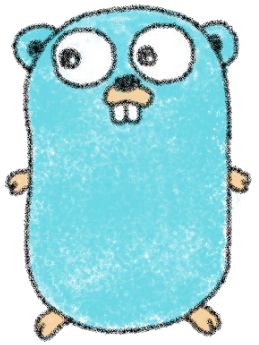


Open source framework to accelerate the development of an Operator

Operator SDK

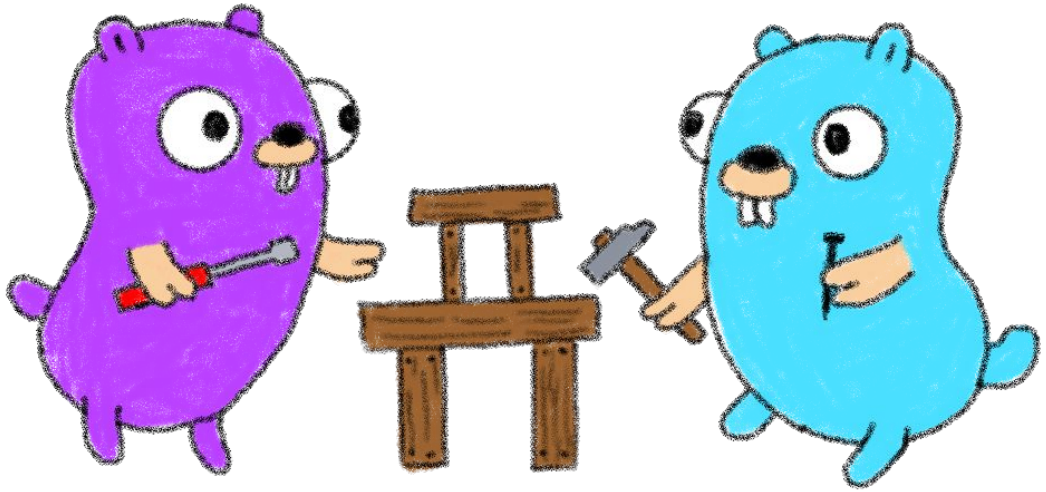
OPERATOR
SDK

BUILD
TEST
ITERATE



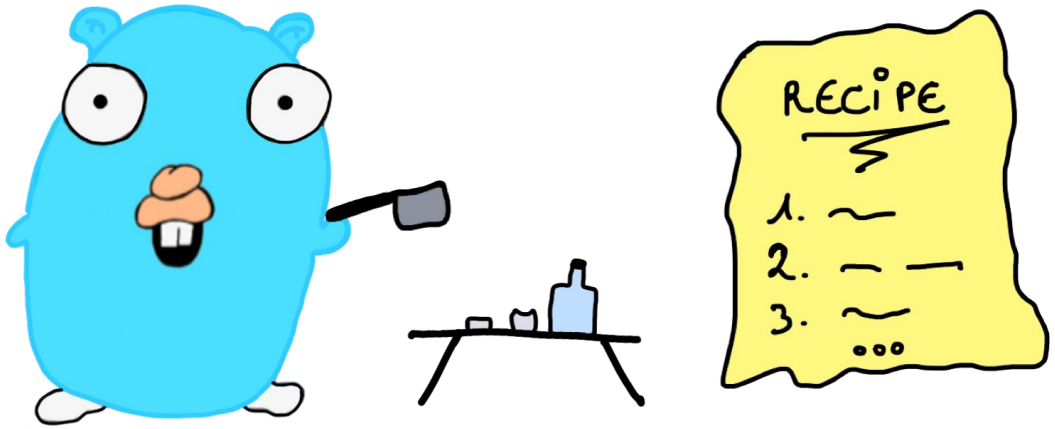
Our objective

Why? Because we can!



What do we want?

- In a simple and easy Kubernetes operator
- Handle cute Gophers



- In **Javascript,** because it's very expressive and easy to understand… and I like it 😁

All the code is available

README.md

Let's dive into Kubernetes operator creation

This repository stores all the code for my talk *Let's dive into Kubernetes operator creation*, that I have given at:

- 2023-11-09 - DevOps Barcelona - Slides

Packages

No packages published
Publish your first package

Languages

TypeScript 80.9% JavaScript 15.8%
HTML 2.1% Other 1.2%

Suggested Workflows

Based on your tech stack

Datadog Synthetics Configure
Run Datadog Synthetic tests within your GitHub Actions workflow

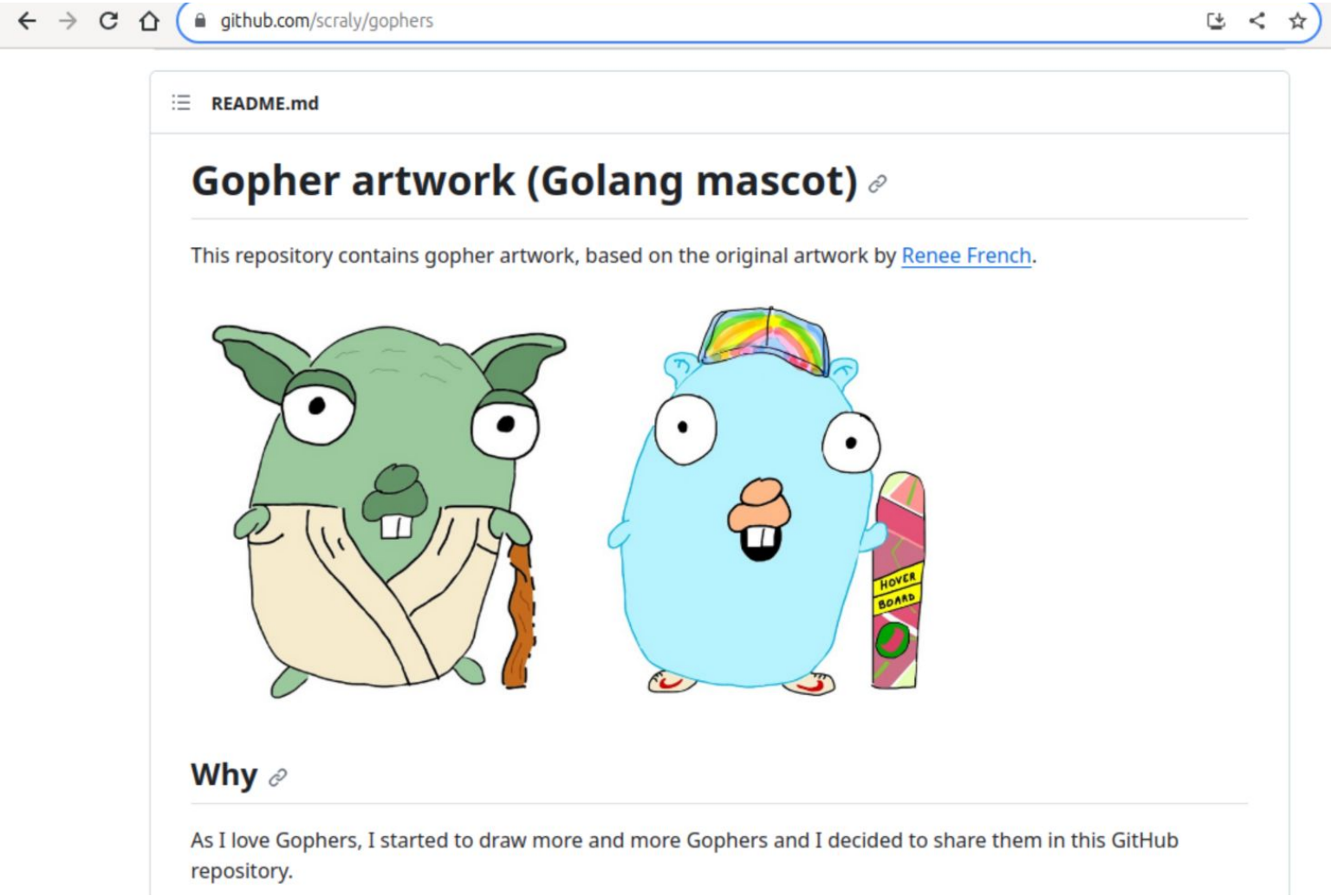
Gulp Configure
Build a NodeJS project with npm and gulp.

Deno Configure
Test your Deno project

More workflows Dismiss suggestions

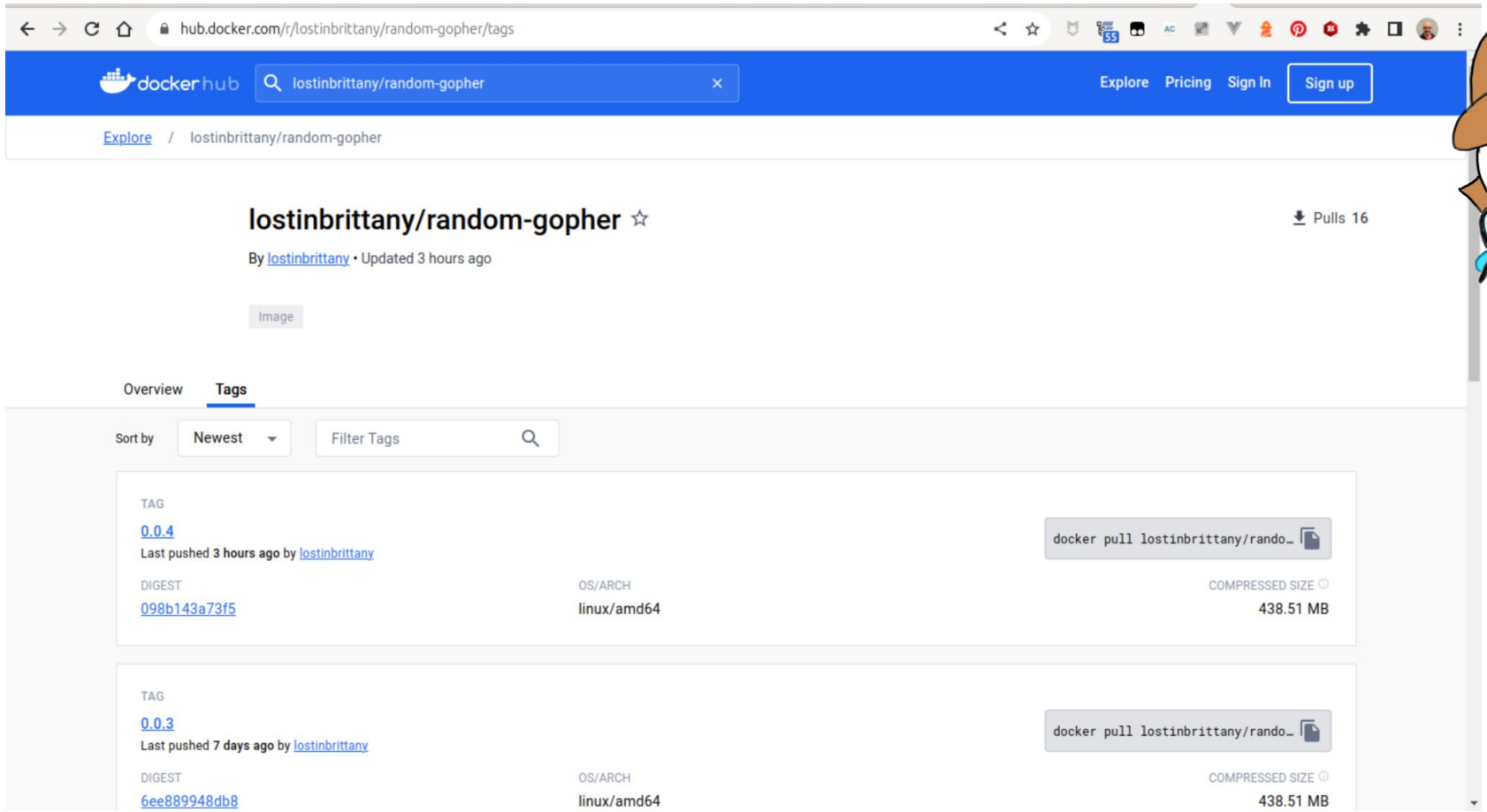
https://github.com/LostInBrittany/lets-dive-into-kubernetes-operator-creation

Aurélie's Gopher repository



https://github.com/scraly/gophers

random-gopher container



https://hub.docker.com/r/lostinbrittany/random-gopher

OVHcloud

@LostInBrittany

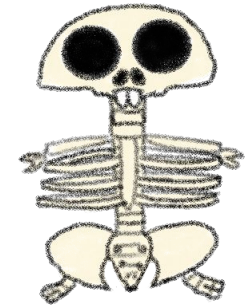
random-gopher container

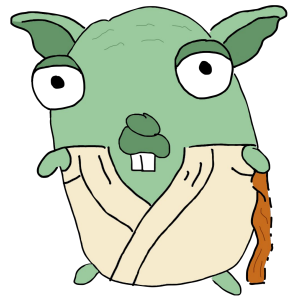
```
import express from 'express';

import { readdir } from 'node:fs/promises';
import path  from 'node:path';

let app = express();
let chosenGopher;

async function initFiles() {
   try {
       const files = await readdir('gophers');
       const gophers = files.filter(
           (item) => item.endsWith('png') || item.endsWith('jpg')
       );
       const randomIndex = Math.floor((Math.random()*gophers.length));
       chosenGopher = gophers[randomIndex];
       console.log(chosenGopher);
     } catch (err) {
       console.error(err);
     }
}
```



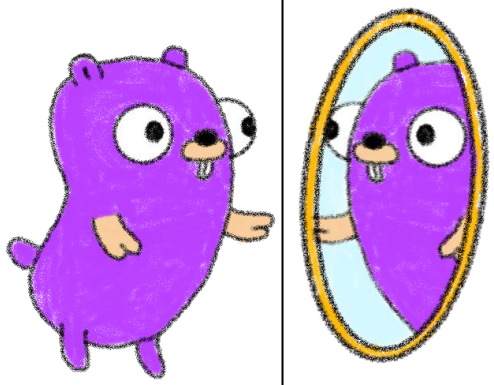


At startup it chooses and exposes a random gopher

OVHcloud

@LostInBrittany

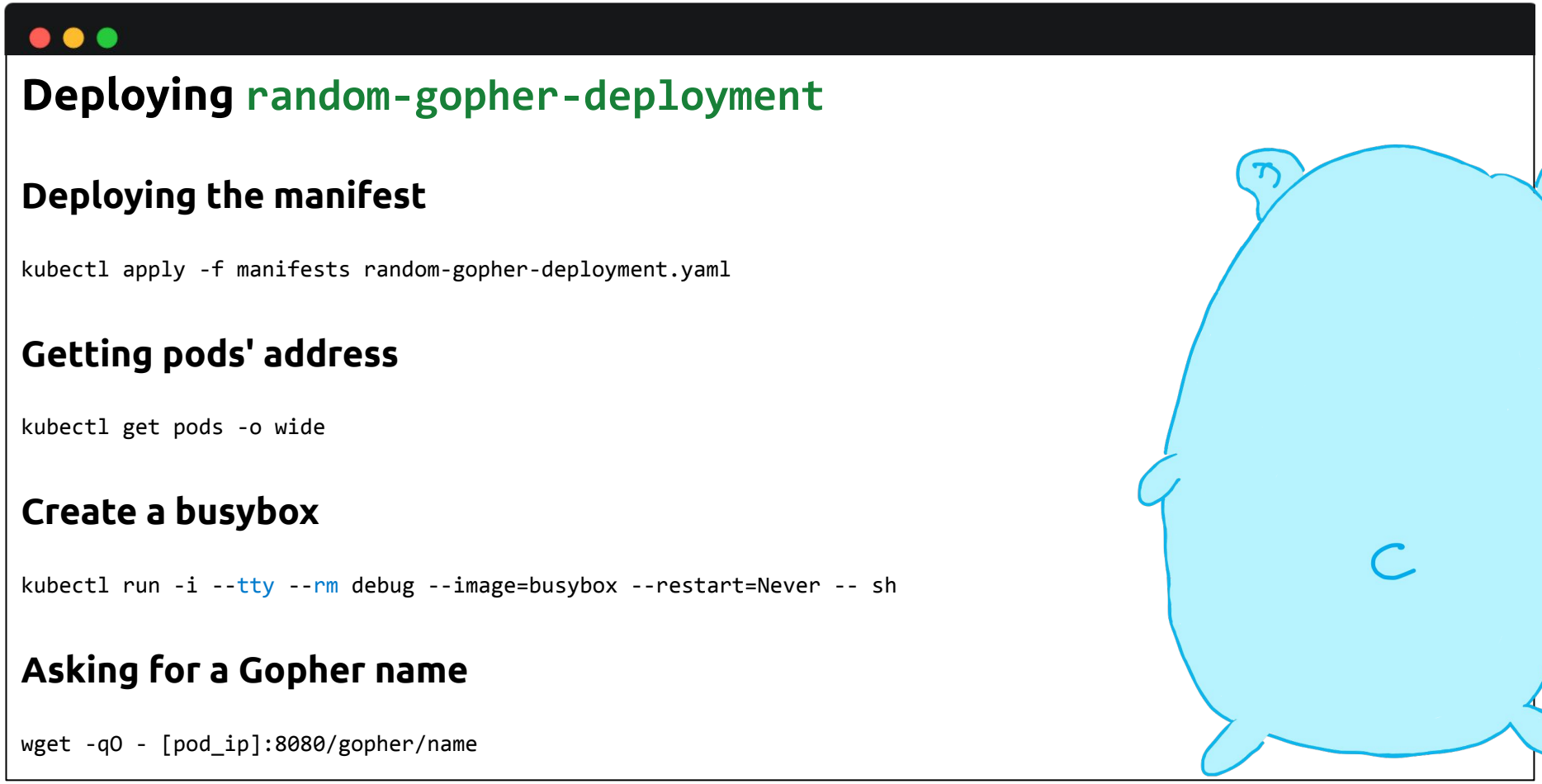
random-gopher-deployment



```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
 name: random-gopher
spec:
 selector:
   matchLabels:
     run: random-gopher
 replicas: 10
 template:
   metadata:
     labels:
       run: random-gopher
   spec:
     containers:
     - name: random-gopher
       image: lostinbrittany/random-gopher:0.0.4
       ports:
       - containerPort: 8080
```

Deploying lots of random-gophers in the cluster

Applying it to the cluster



Deploying `random-gopher-deployment`

Deploying the manifest

```
kubectl apply -f manifests random-gopher-deployment.yaml
```

Getting pods' address

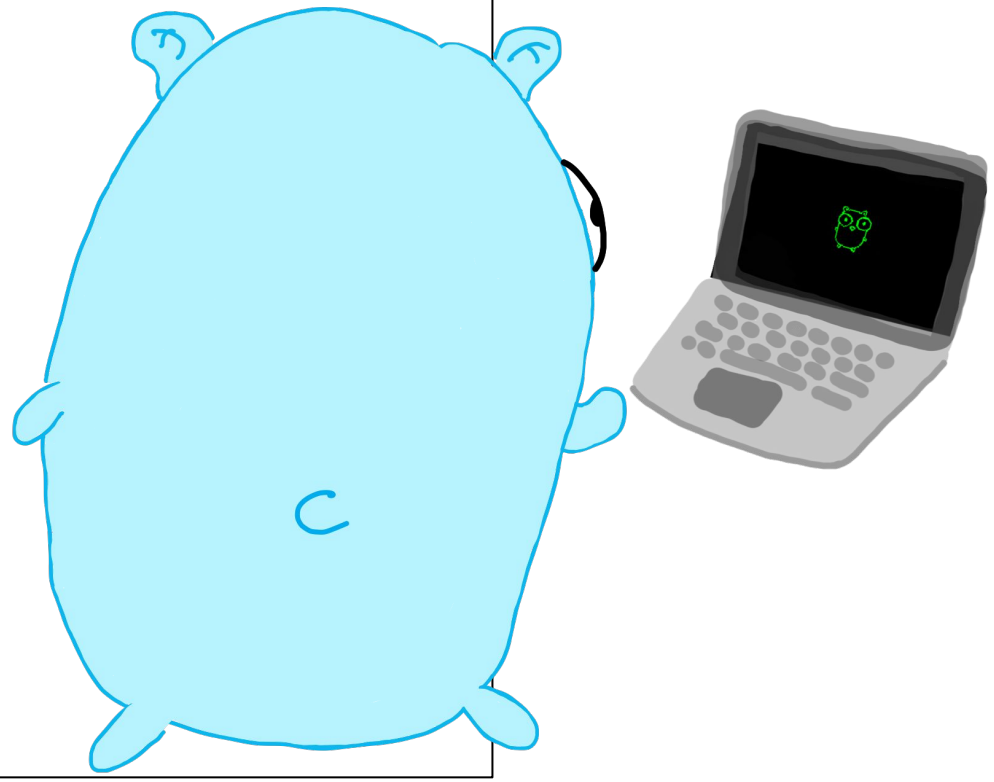
```
kubectl get pods -o wide
```

Create a busybox

```
kubectl run -i --tty --rm debug --image=busybox --restart=Never -- sh
```

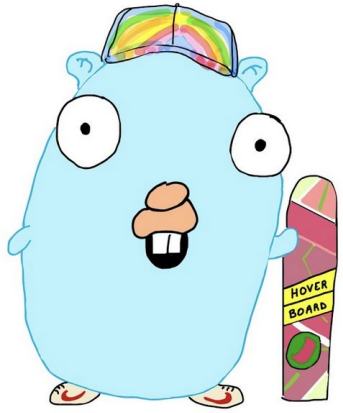
Asking for a Gopher name

```
wget -qO - [pod_ip]:8080/gopher/name
```



Let's switch to the terminal…

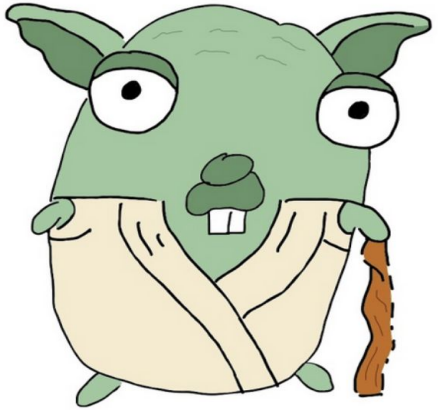
We also have an API for Gophers

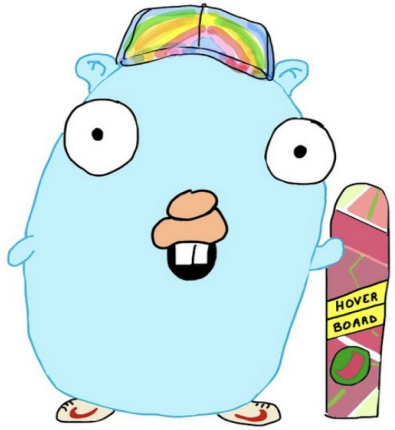


gophers-api

This simple API handle a list of Gophers. It alllows to:

- list the existing Gophers
- display the information about a Gopher
- create a new Gopher
- delete a Gopher
- update the path and the URL of a Gopher





https://github.com/LostInBrittany/lets-dive-into-kubernetes-operator-creation/tree/main/gopher-api-and-ui

OVHcloud

@LostInBrittany

We also have an API for Gophers

Swagger
Supported by SMARTBEAR

Explore

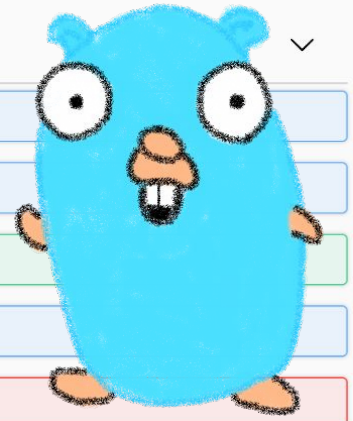
gophers-api 0.1.0

HTTP server that handle cute Gophers.

Schemes
HTTP

default

- GET /healthz
- GET /gophers
- POST /gopher Add a new Gopher
- GET /gopher
- DELETE /gopher
- PUT /gopher



CRUD

Create

Read

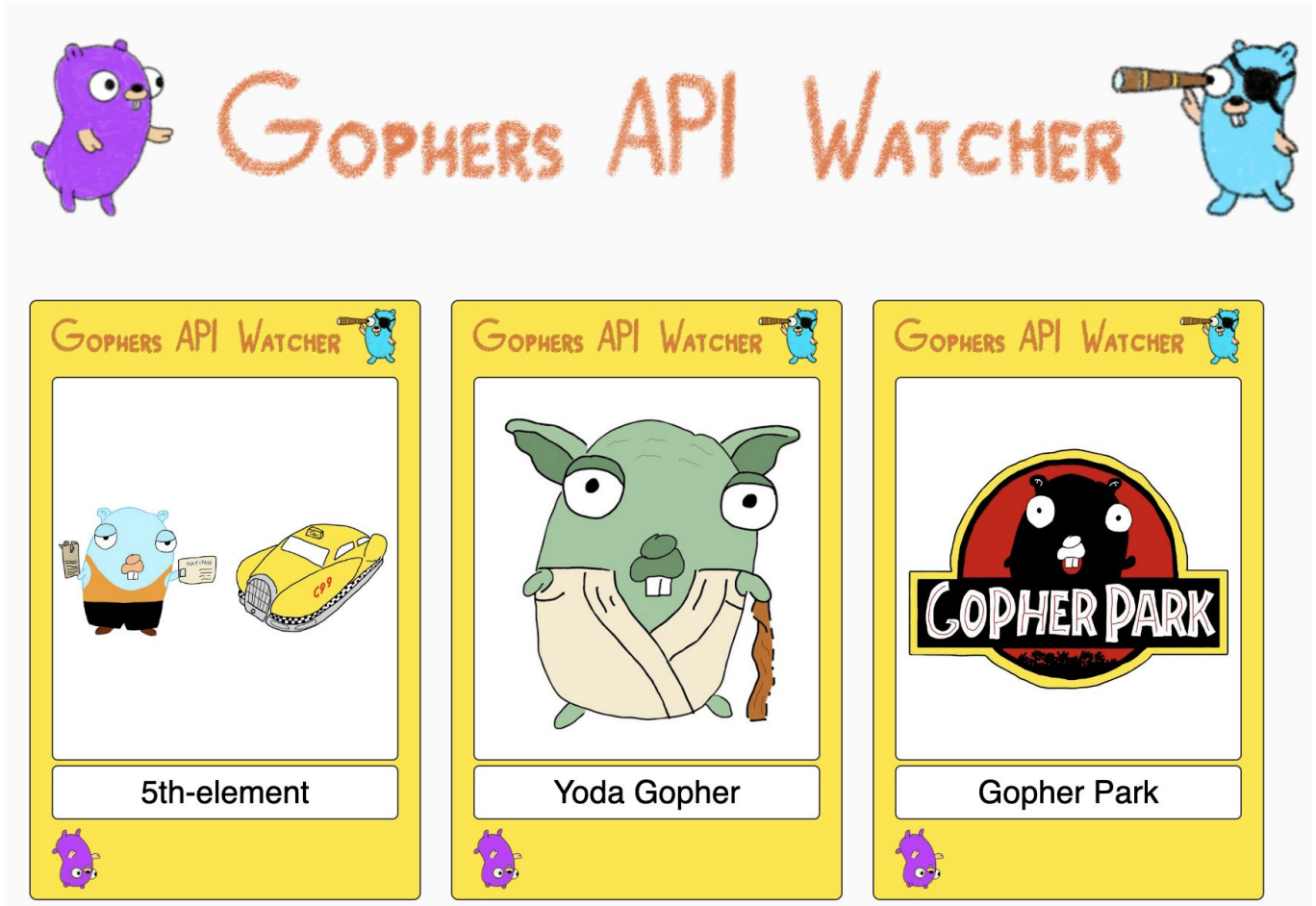
Update

Delete

OVHcloud

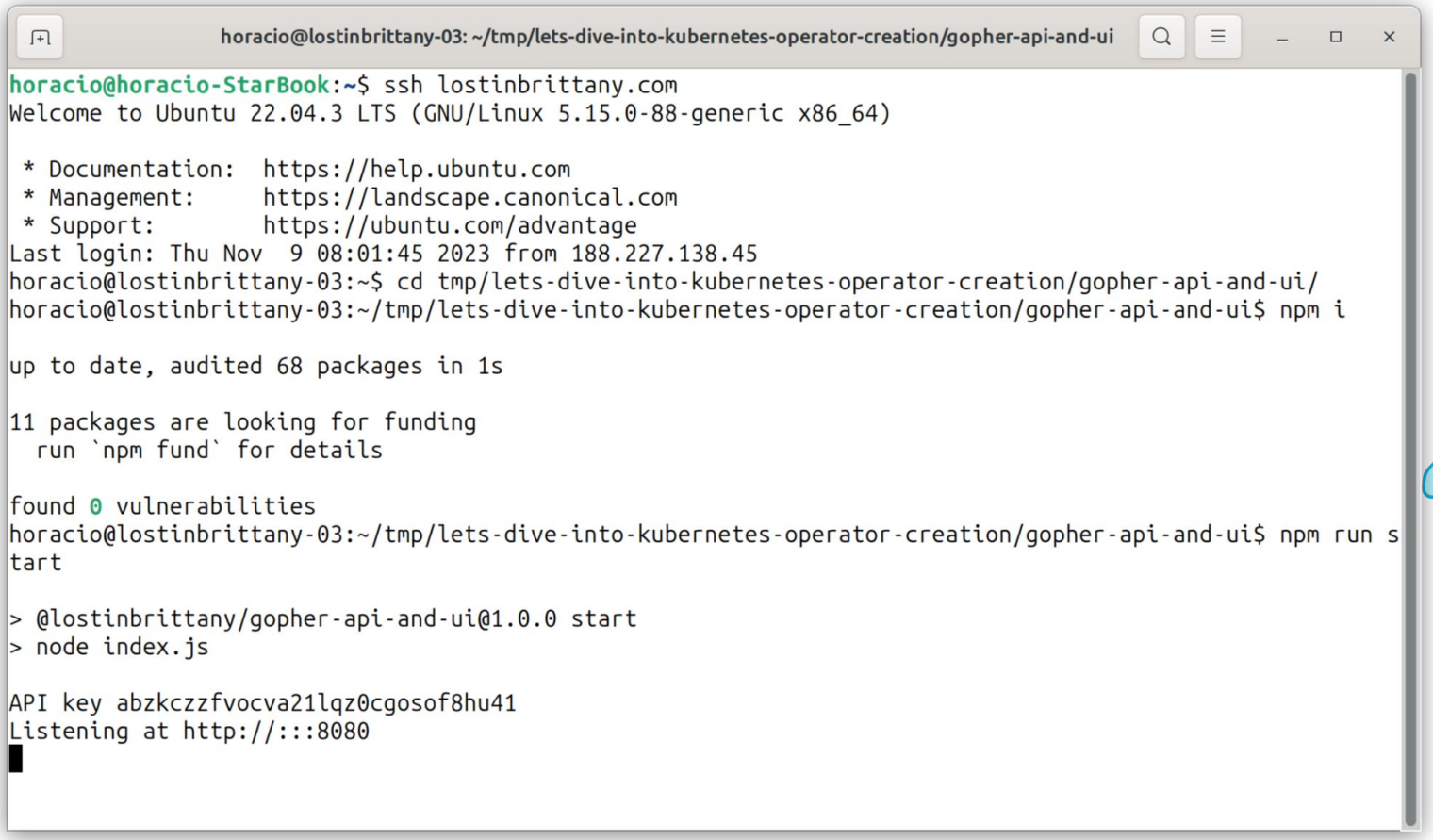
@LostInBrittany

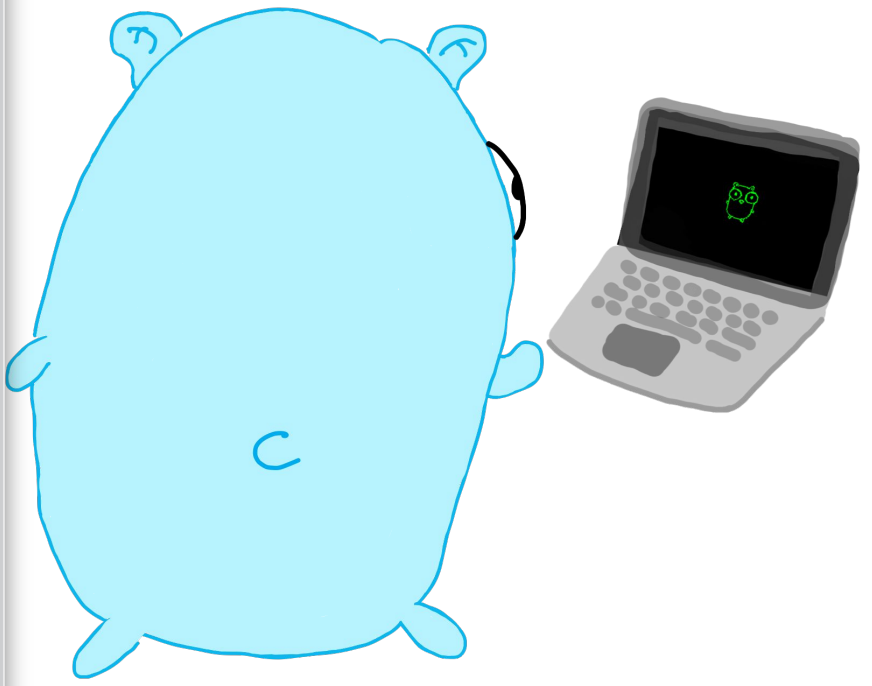
And an UI to see the Gophers in the API



https://github.com/LostInBrittany/gophers-api-watcher

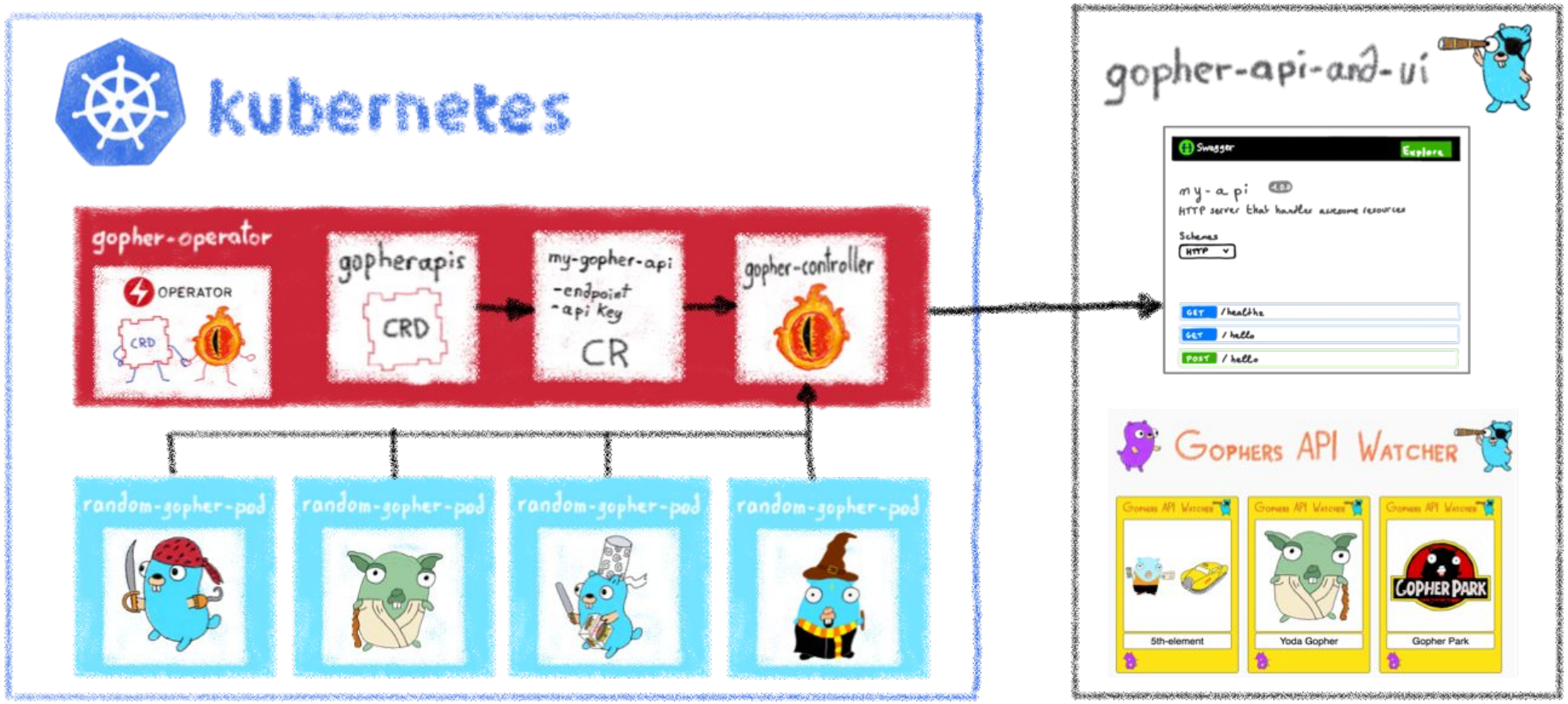
Deploying the API in an instance





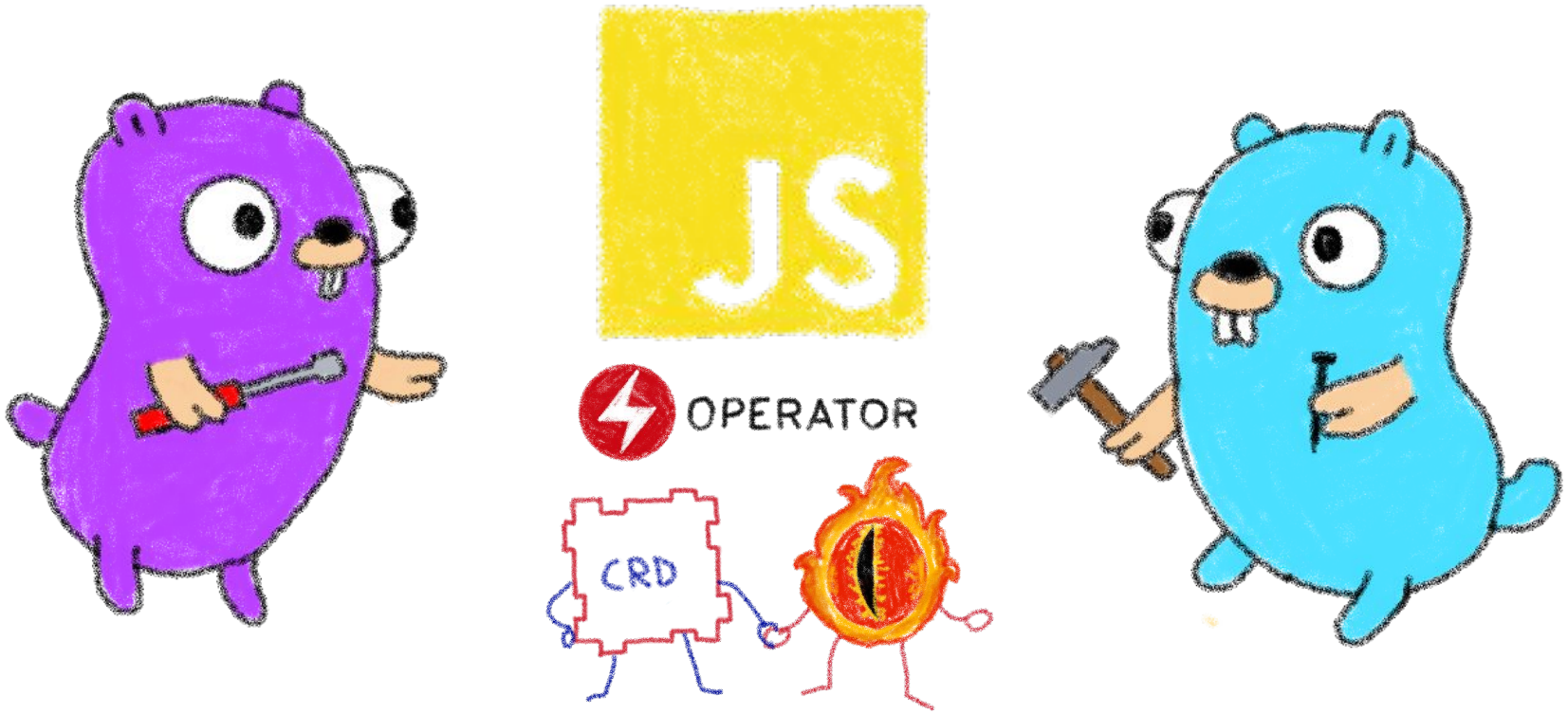
Let's switch to the terminal…

What we want



An operator to feed the API with the deployed pods info

And we are doing it in the simplest way



In JavaScript, yeah!

Taking as base `k8s-operator-node`

github.com/dot-i/k8s-operator-node

README.md

NodeJS Kubernetes operator framework

build passing | version v1.3.8 | node >=10

The **NodeJS** operator framework for **Kubernetes** is implemented in TypeScript, but can be called from either Javascript or TypeScript.

The operator framework is implemented for server-side use with node using the `@kubernetes/client-node` library.

Installation

```
npm install @dot-i/k8s-operator
```

Basic usage

Operator class

To implement your operator and watch one or more resources, create a sub-class from `Operator`.

```
import Operator from '@dot-i/k8s-operator';

export default class MyOperator extends Operator {
    protected async init() {
        // ...
    }
}
```

dot-i Nico Francois

dependabot[bot]

Languages

TypeScript 93.6% JavaScript 6.1%

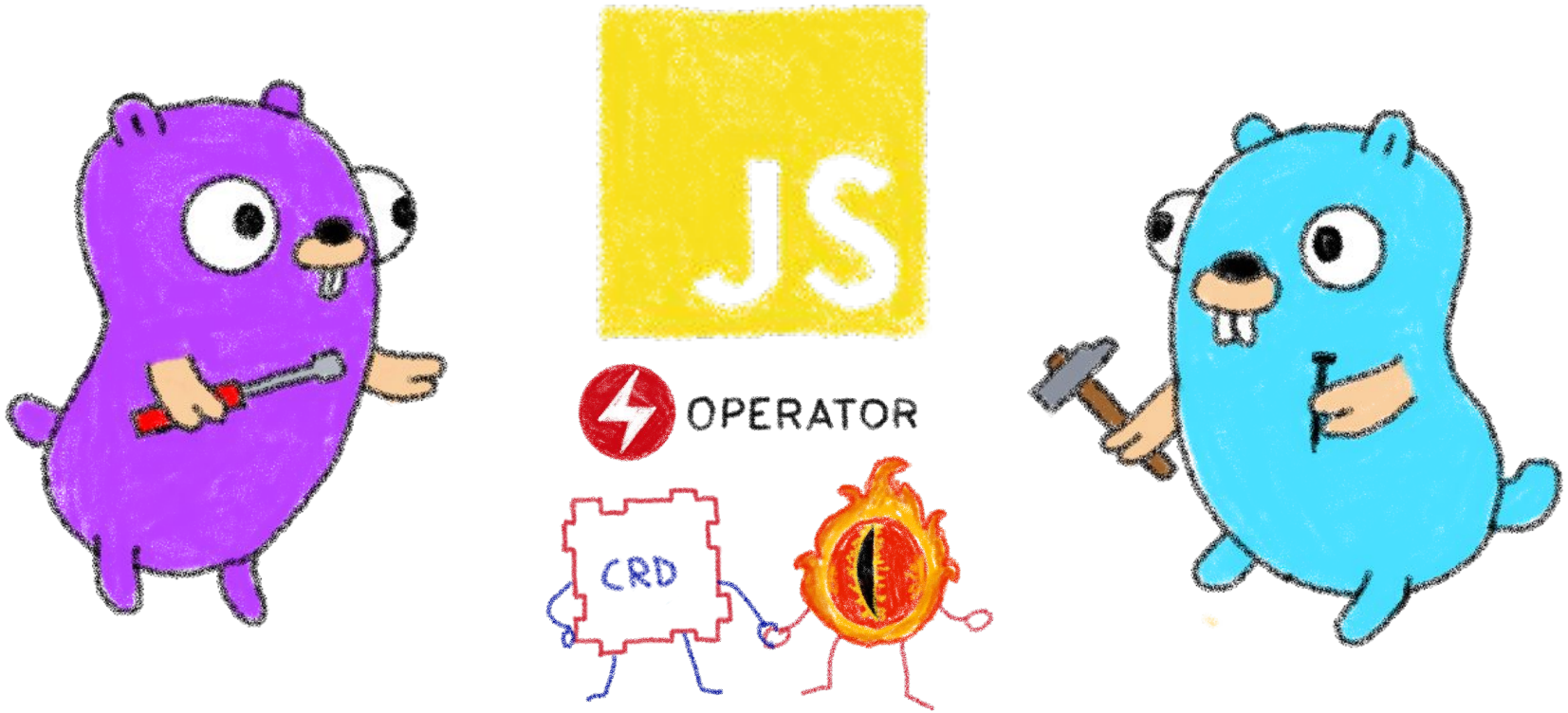
Shell 0.3%

https://github.com/dot-i/k8s-operator-node

OVHcloud

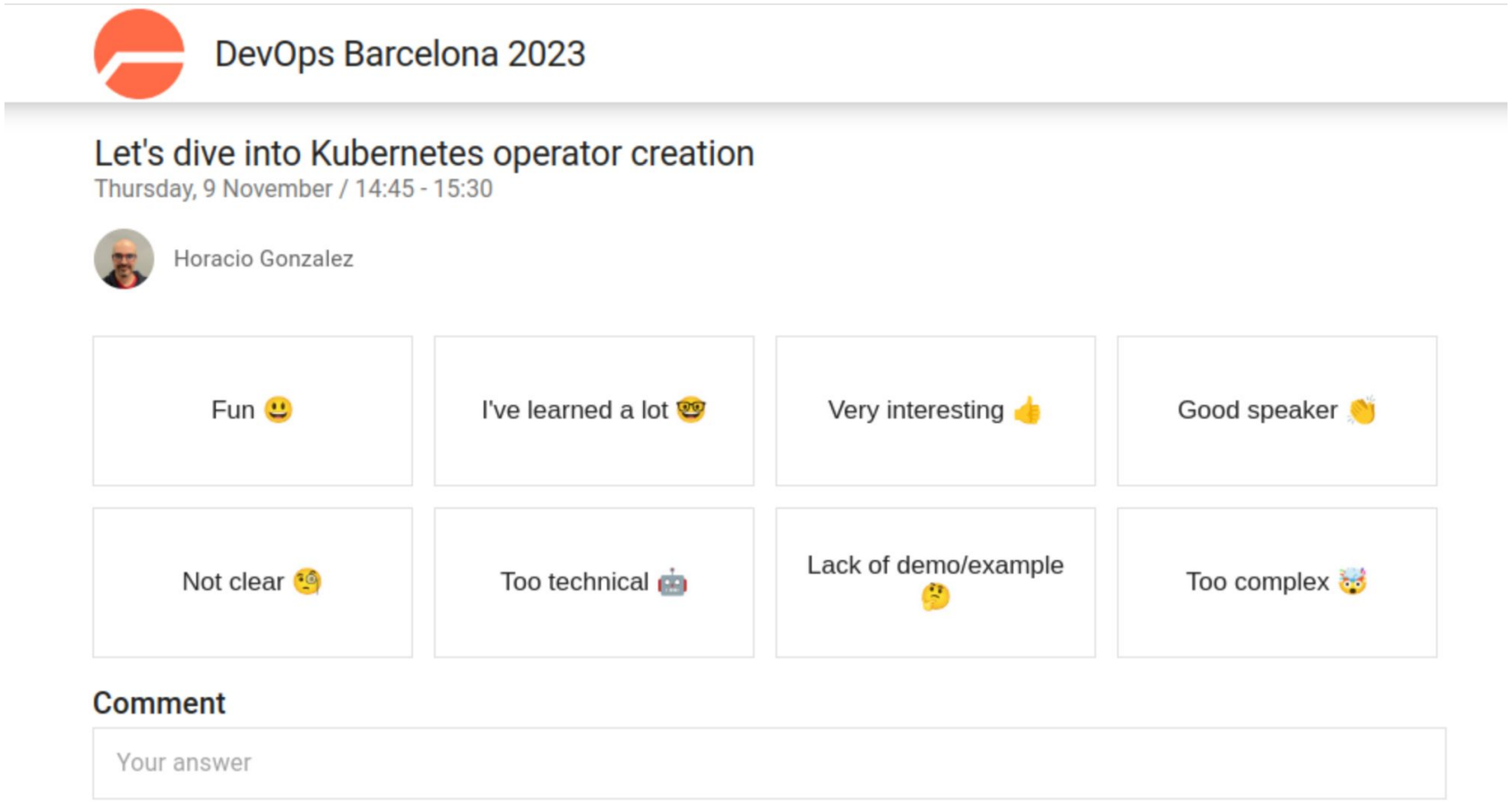
@LostInBrittany

Building the gopher operator



Let's switch to VS Code…

We ❤️ feedbacks



https://bit.ly/devopsbcn23-horacio

OVHcloud

@LostInBrittany

That's all, folks!

Thank you all!