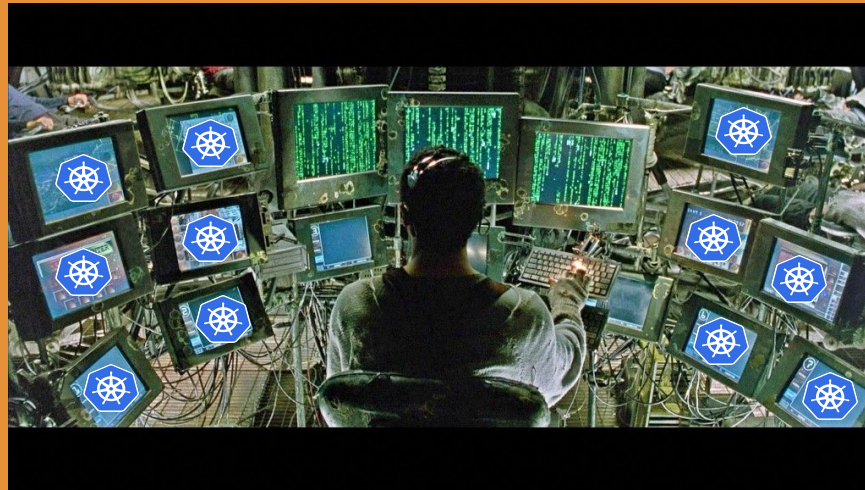


Les opérateurs Kubernetes à la portée de n'importe quel développeur.



**THANK YOU**

# Faisons connaissance

👉 Qui utilise Kubernetes ?

👉 Qui utilise Kubernetes en production (pour de vrai) ?

👉 Qui a déjà utilisé un opérateur ?

👉 Qui a déjà développé un opérateur ?

👉 Qui est dev ? Ops ? Les deux ?



# Quel type de dev ?



👊 Go ?

👋 Helm  
?

👉 Java ?

👿 VB  
?

# Disclaimer

→ Explications simplifiées pour que l'on parle le même langage

→ Je ne suis pas Ops / Expert Kubernetes

→ Je suis plutôt Dev













Pourquoi faire des choses manuellement  
quand il est possible de les faire faire par  
une application ?

**YAML**



**MANIFESTS EVERYWHERE**

# Principes de base des opérateurs



## Pattern clairement définis dans Kubernetes

*“The Operator pattern aims to capture the key aim of a human operator who is managing a service or set of services. Human operators who look after specific applications and services have deep knowledge of how the system ought to behave, how to deploy it, and how to react if there are problems.”*

*People who run workloads on Kubernetes often like to use automation to take care of repeatable tasks. The Operator pattern captures how you can write code to automate a task beyond what Kubernetes itself provides.”*



## API Kubernetes

- Possible d'ajouter des API dans Kubernetes à celles existantes (Deployment, Service, ...)
- Permet de modifier (étendre) le comportement de Kubernetes sans devoir modifier le code de Kubernetes
- Basée sur une custom resources (CRDs)

# Custom Resource Definition et Custom Resource

## → Custom Resource Definition = CRD

→ “Sorte” de schéma / classe pour les objets Kubernetes

## → Custom Resource : CR

→ Sorte “d’objet instanciant la classe (CRD)”

```
1 apiVersion: apiextensions.k8s.io/v1
2 kind: CustomResourceDefinition
3 metadata:
4   name: helloworldcustomresources.fr.wilda
5 spec:
6   group: fr.wilda
7   names:
8     kind: HelloWorldCustomResource
9     plural: helloworldcustomresources
10    shortNames:
11      - hw
12    singular: helloworldcustomresource
13  scope: Namespaced
14  versions:
15    - name: v1
16      schema:
17        openAPIV3Schema:
18          properties:
19            spec:
20              properties:
21                → name:
22                  type: string
23              type: object
24            status:
25              type: object
26          type: object
27  served: true
28  storage: true
```

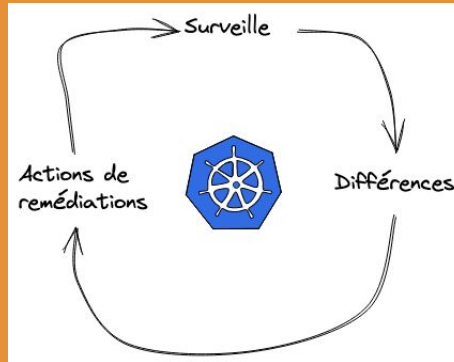
```
1 apiVersion: "fr.wilda/v1"
2 kind: HelloWorldCustomResource
3 metadata:
4   name: hello-world
5   namespace: test-hw-crd
6 spec:
7   → name: stef
```

## Boucle de réconciliation

→ Élément central de la définition du pattern contrôleur

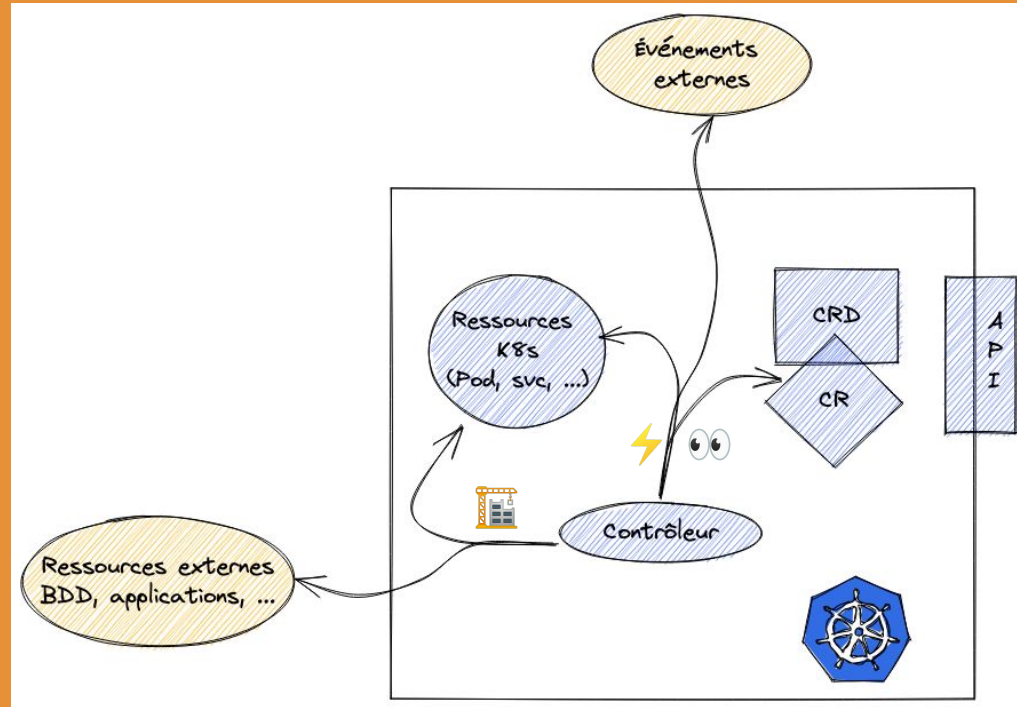


*“In Kubernetes, controllers are control loops that watch the state of your cluster, then make or request changes where needed. Each controller tries to move the current cluster state closer to the desired state.”*



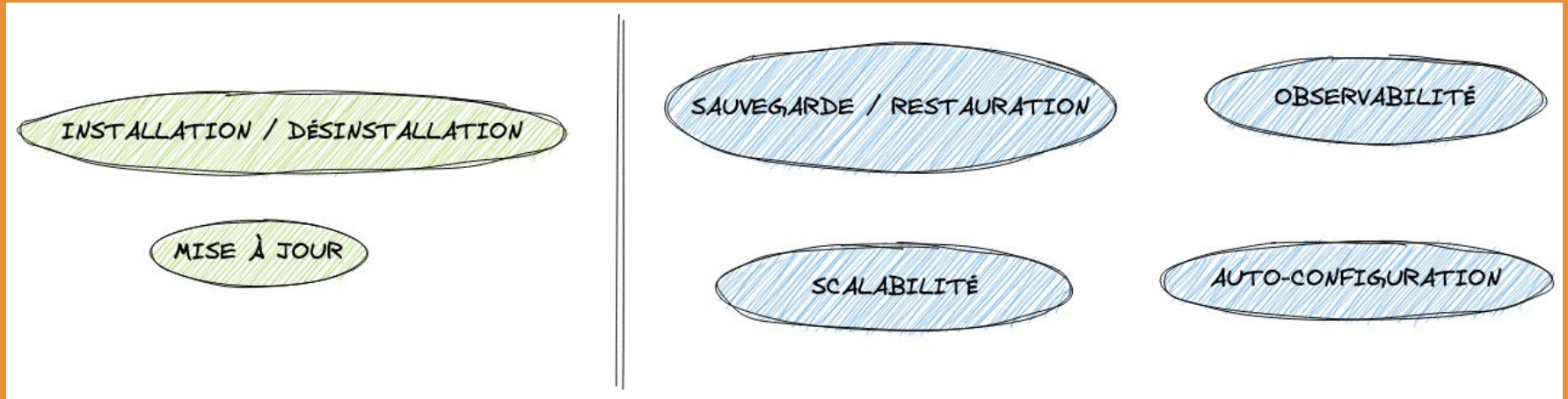
# Un opérateur pour les gouverner tous

- API + CRD (et CR) + Contrôleur
- Gère une et une seule application (déployée ou non dans Kubernetes)
- Essaie de maintenir l'état de l'application avec ce qui est déclaré dans la CR





# Différentes fonctionnalités d'un opérateur



## 👉 One more thing !

- ➔ Un opérateur n'est qu'un Pod qui exécute une application
- ➔ Il peut donc être développé avec n'importe quel langage
- ➔ Il suffit d'appeler les API Kubernetes ou d'utiliser un client compatible

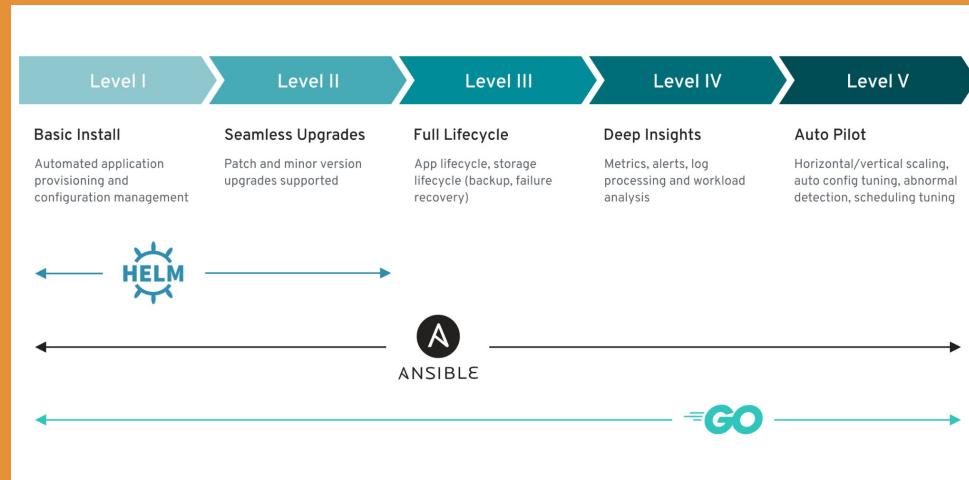


A collection of various metal wrenches scattered on a dark, textured background. The wrenches are of different sizes and orientations, some showing brand names like 'CHROM' and '59'.

# Frameworks

# Simplifier le développement et le packaging

- Plus simples de prise en main que les API ou clients
- Permet de se concentrer sur le code “métier”
- Le plus connu : Operator SDK





# ✨ Les principales fonctionnalités du Framework

- Basé sur Kubebuilder qui permet la création d'API basées sur les CRDs
- Création (scaffolding) du squelette applicatif
- Facilitation de la génération des CRD et du contrôleur
- Gestion de la boucle de réconciliation
- Exécution locale avec debugger possible
- Packaging de l'opérateur simplifié
- Intégration d'Operator Lifecycle Manager (OLM)

# Et en java ?



The screenshot shows the homepage of the Java Operator SDK. The page has an orange header with a navigation menu containing links for HOME, DOCS, CODE OF CONDUCT, and RELEASES, along with icons for Discord and GitHub. The main content area features the Java logo and the text "[ JAVA OPERATOR SDK ]". Below this are three buttons: "GET STARTED" (black with white text), "CONTRIBUTE" (white with orange text and a GitHub icon), and "DISCORD CHANNEL" (white with orange text and a Discord icon). The bottom section is white and contains the text "Sponsored by:" followed by the logos for Container Solutions and Red Hat.

[ JAVA OPERATOR SDK ]

HOME DOCS CODE OF CONDUCT RELEASES

[ JAVA OPERATOR SDK ]

GET STARTED CONTRIBUTE DISCORD CHANNEL

Sponsored by:

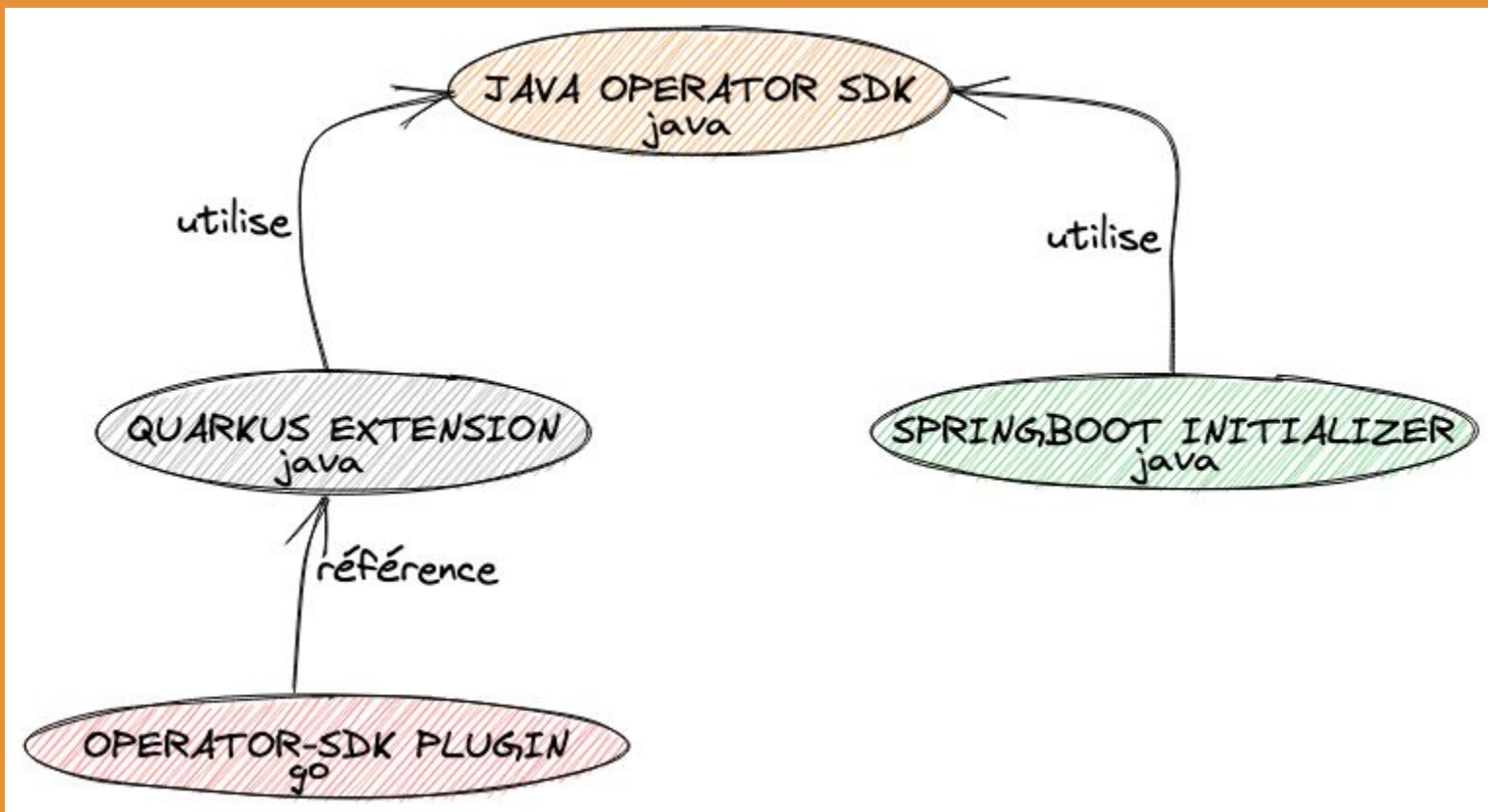
Container Solutions & Red Hat

 Java Operator SDK

# ✨ Les principales fonctionnalités du Framework

- ➔ Client Kubernetes Fabric 8 pour manipuler les ressources Kubernetes
- ➔ Création (scaffolding) du squelette applicatif pour le mode Quarkus
- ➔ Facilitation de la génération des CRD et du contrôleur (annotations)
- ➔ Gestion de la boucle de réconciliation
- ➔ Exécution locale avec debugger possible
- ➔ Intégration avec Spring Boot et Quarkus
- ➔ Packaging de l'opérateur simplifié

# Les différents composants



# Démos





# 👋 Hello world !!

- ➔ Scaffolding du projet Quarkus
- ➔ Création des éléments de base
- ➔ Création de la CRD / API
- ➔ Exécution en mode local / debugging
- ➔ Exécution en mode “développeur”



CommitStrip.com



# Bienvenue dans le monde réel

- Création de la CRD
- Gestion du déploiement d'un Nginx
- Gestion d'une suppression d'une ressource
- Limite sur le nombre de Pods
- Packaging et déploiement dans Kubernetes






**I KNOW KUBERNETES OPERATOR**



# 💡 Ce qu'il faut retenir en Java

- Projet jeune mais actif : v3.x (plugin operator-sdk / extension Quarkus pas toujours synchronisées)
- Possibilité de faire quasiment tout
- Mode local avec le debugging et pas à pas
- Génération / mise à jour de la CRD
- Très (trop ?) orienté Quarkus
- Java 🥰
- La suite : OLM, EventSource plus simples, ...





# Ce qu'il faut retenir en Go

- Un des (le plus ?) Frameworks le plus utilisé
- Possibilité de tout faire (OLM y compris)
- Mode local avec le debugging et pas à pas
- Génération / mise à jour de la CRD
- Go ...





# Ce qu'il faut retenir avec Helm

- Permet de réutiliser des charts existants
- Se limite à des opérateurs simples
- Mode local sans le debugging et pas à pas
- Helm ...
- La suite : contrôleur custom en Go





# The Cloud with European values



**OVHcloud**



Bare Metal Cloud



Hosted Private Cloud



Public Cloud



Web Cloud



Télécom





# Stéphane Philippart

👉 Baby DevRel@OVHCloud 🥑 🦄 🎉

👉 Co-créateur de TADx (meetups Agile, Dev, DevOps)

💰 Voucher pour essayer : TrialFR042022

🔗 Démonstrations :

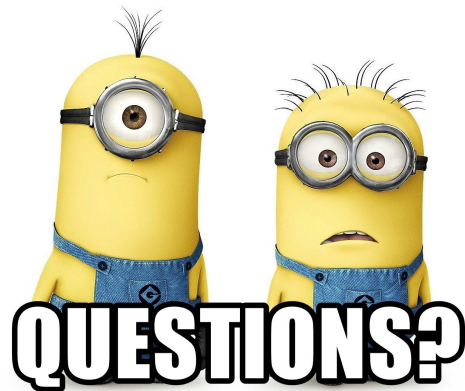
- <https://github.com/philippart-s/breizh-camp-helm-operator>
- <https://github.com/philippart-s/breizh-camp-go-operator>
- <https://github.com/philippart-s/breizh-camp-java-operator>

🐦 [@wildagxs](https://twitter.com/wildagxs)

📄 <https://philippart-s.github.io/blog>

🐙 <https://github.com/philippart-s/>

🌐 <https://www.linkedin.com/in/philippartstephane/>



<https://tinyurl.com/ye25r3dm>

# Liens

<https://www.hiclipart.com/>

<https://kubernetes.io/docs/concepts/architecture/controller/>

<https://kubernetes.io/docs/concepts/extend-kubernetes/api-extension/custom-resources/>

<https://javaoperatorsdk.io/>

<https://quarkiverse.github.io/quarkiverse-docs/quarkus-operator-sdk/dev/index.html>

[https://quarkus.io/guides/all-config#quarkus-core\\_quarkus.native.container-runtime](https://quarkus.io/guides/all-config#quarkus-core_quarkus.native.container-runtime)

<https://github.com/fabric8io/kubernetes-client/blob/master/doc/CHEATSHEET.md>

<https://stackoverflow.com/a/61437982>



© ArtVstudio