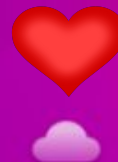




CLOUD SUD

24 MARS 2022

Développer un opérateur
Kubernetes en Java,
c'est possible !



Stéphane PHILIPPART
DevRel@OVHcloud

TOULOUSE - MONTPELLIER - MARSEILLE

alter way



THANK YOU

⚠ Disclaimer ⚠

- Explications simplifiées pour que l'on parle le même langage
- Je ne suis pas Ops / Expert Kubernetes
- Je suis plutôt Dev

Mais c'est quoi un opérateur ?



A glowing blue circular interface with the text "J.A.R.V.I.S" in the center. The interface consists of several concentric rings, some of which are segmented, suggesting a futuristic or technological theme. The text is in a bold, sans-serif font, and the entire graphic is set against a black background.

J.A.R.V.I.S

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
- Basé sur une ou plusieurs [ressources](#) (CRDs)

Custom Resource Definition et Custom Resource

➔ Custom Resource Definition = CRD

➔ “Sorte” de schema / 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
```

carbon
carbon.now.sh

Boucle de réconciliation

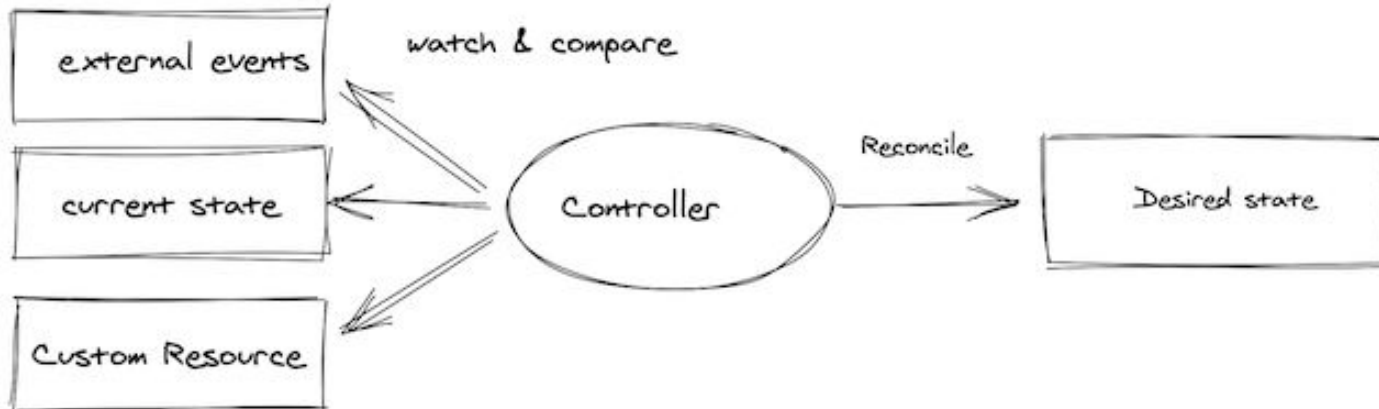
→ Élément central de la définition du [pattern contrôleur](#)

“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



Différents types d'opérateurs



👉 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

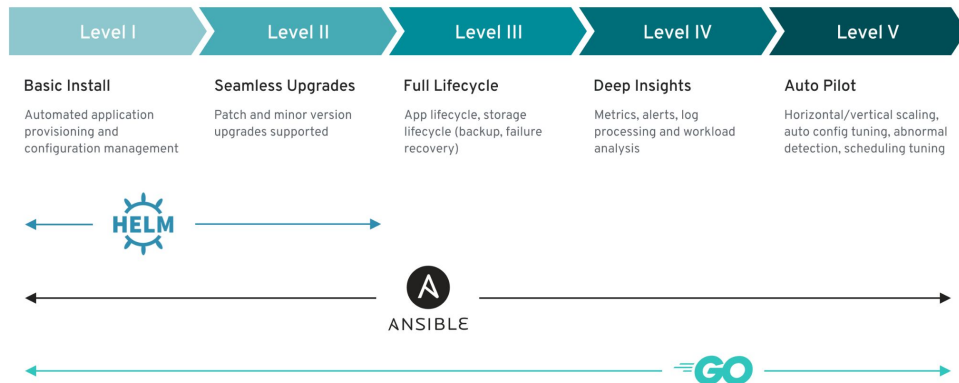
Frameworks

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](#)







Et en java ?



The screenshot shows the homepage of the Java Operator SDK. The top navigation bar is orange and contains the following links: HOME, DOCS, CODE OF CONDUCT, and RELEASES. There are also icons for Discord and GitHub. The main content area is orange and features the Java Operator SDK logo (a coffee cup) 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](#)

✨ 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

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”
- ➔ Packaging et déploiement dans Kubernetes



Bienvenue dans le monde réel

- Création de la CRD
- Gestion du déploiement d'un Nginx
- Gestion d'une suppression d'une ressource
- Packaging et déploiement



[source](#)

Ce qu'il faut retenir

- Projet jeune mais actif : v2.x
- 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 🥰

Stéphane Philippart

🛒 Baby DevRel@OVHCloud 🥑

🛒 Co-créateur de [TADx](#) (meetups Agile, Dev, DevOps)

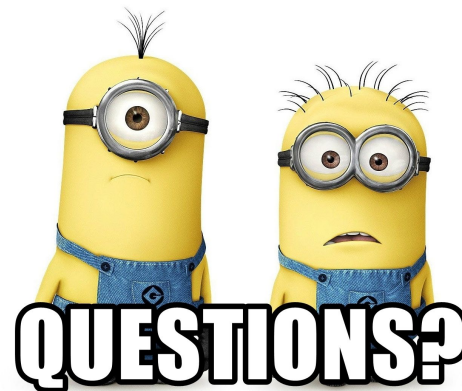
🔗 Démo : <https://github.com/philippart-s/cloud-sud-nginx>

🐦 [@wildagsx](#)

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

🐙 <https://github.com/philippart-s/>

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



<https://tiny.one/2aum28v2>

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://github.com/philippart-s/cloud-sud-hello-world>
- <https://github.com/philippart-s/cloud-sud-nginx>
- <https://github.com/fabric8io/kubernetes-client/blob/master/doc/CHEATSHEET.md>
- <https://stackoverflow.com/a/61437982>

