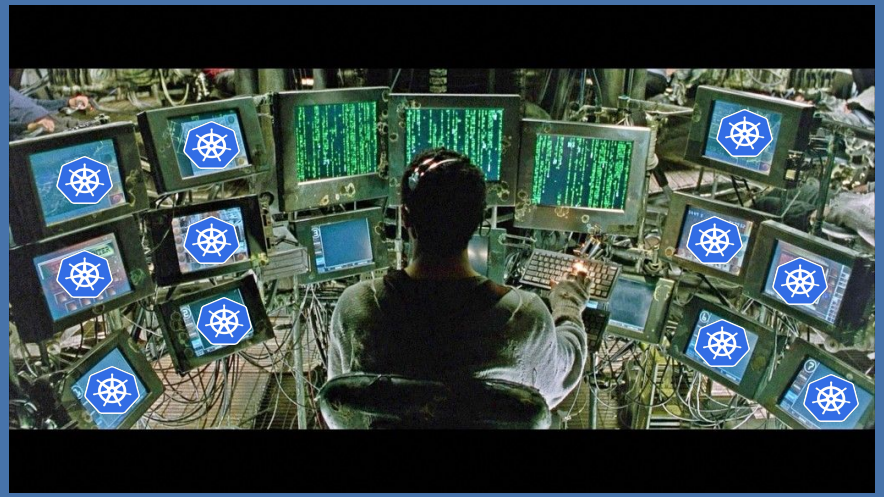
VOXXED DAYS
LUXEMBOURG

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

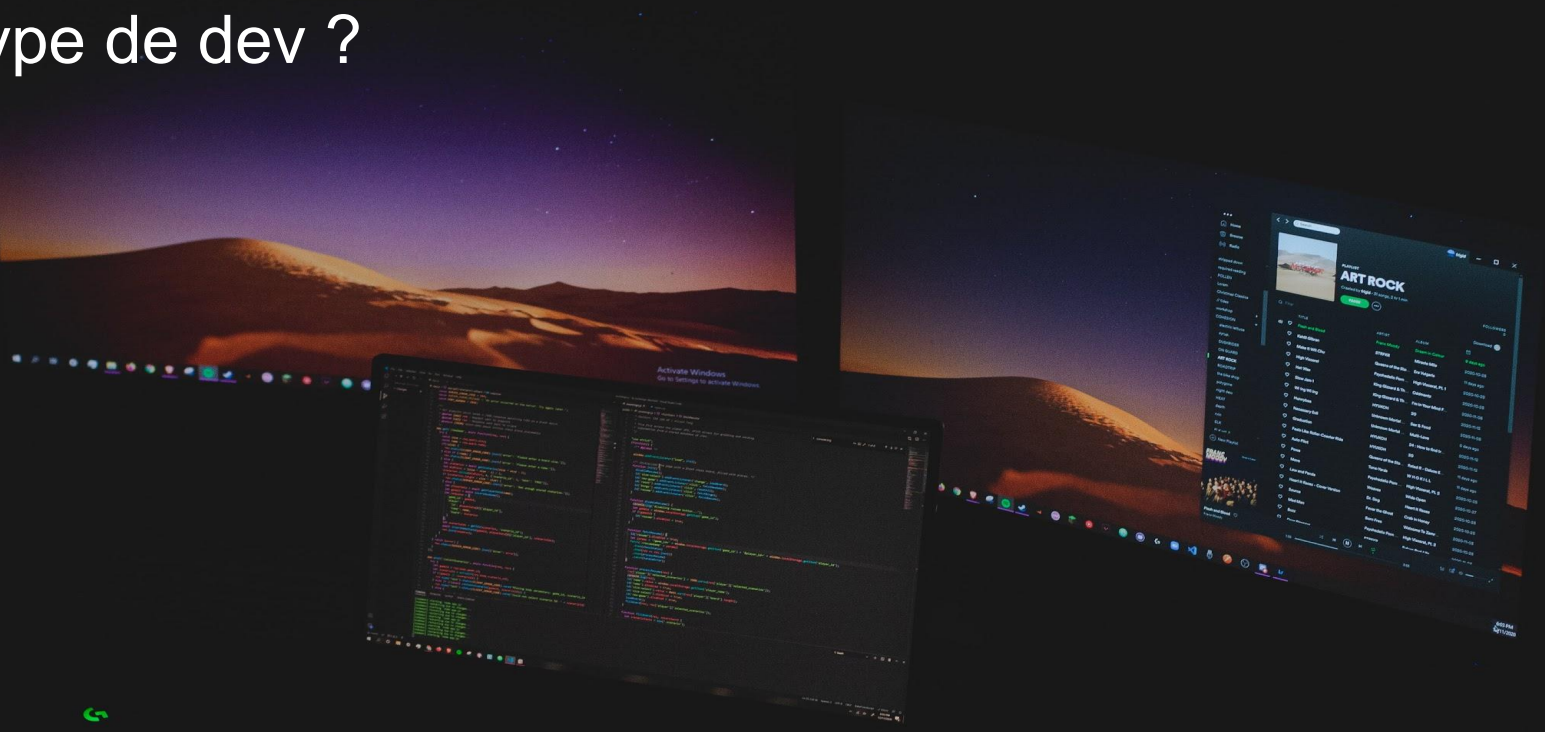



THANK YOU
memegenerator.net

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 ?

David Shultz

Disclaimer

➡️ Explications simplifiées pour que l’on parle le même langage

➡️ Je ne suis pas Ops / Expert Kubernetes

➡️ Je suis plutôt Dev

선인장


Emily Morter

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


YAML
LIGHTYEAR
MANIFESTS EVERYWHERE
imgflip.com

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.”

source

🔗 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 ressources definition (CRDs)

source

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)”

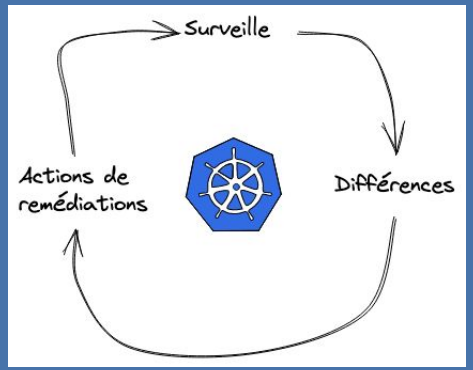
```yaml
apiVersion: apiextensions.k8s.io/v1
kind: CustomResourceDefinition
metadata:
  name: helloworldcustomresources.fr.wilda
spec:
  group: fr.wilda
  names:
    kind: HelloWorldCustomResource
    plural: helloworldcustomresources
    shortNames:
    - hw
    singular: helloworldcustomresource
  scope: Namespaced
  versions:
  - name: v1
    schema:
      openAPIV3Schema:
        properties:
          spec:
            properties:
              name:
                type: string
            type: object
          status:
            type: object
        type: object
    served: true
    storage: true
```

```yaml
apiVersion: "fr.wilda/v1"
kind: HelloWorldCustomResource
metadata:
  name: hello-world
  namespace: test-hw-crd
spec:
  name: stef
```

🔁 Boucle de réconciliation

➡️ Elément 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.”



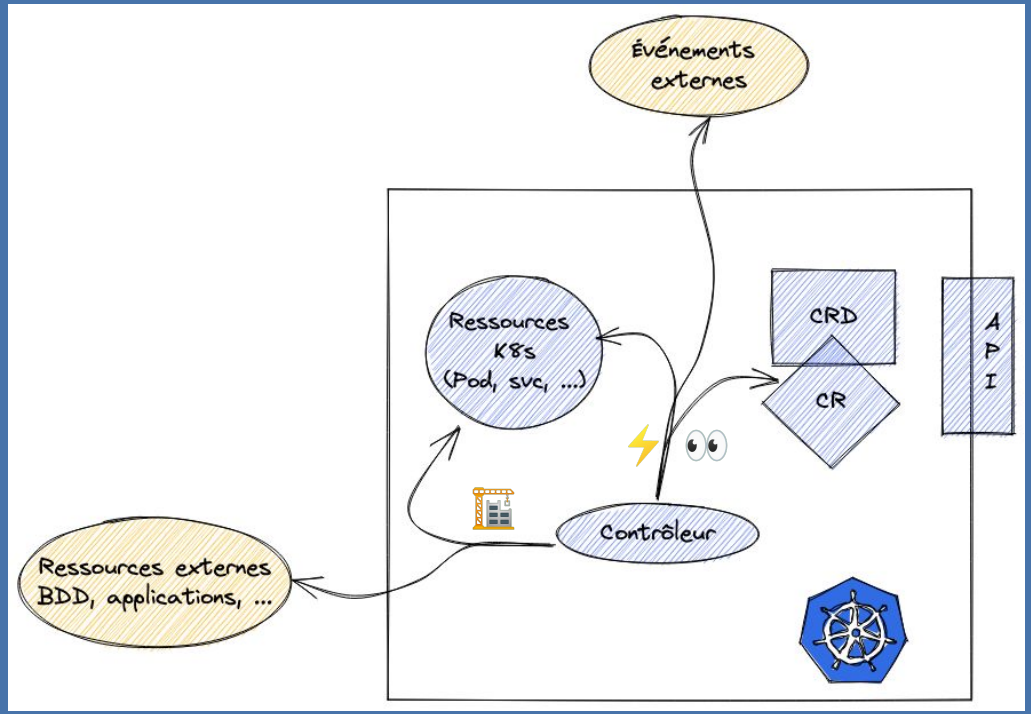
source

🎯 Un opérateur pour les gouverner tous

➡️ API + CRD (et CR) + Contrôleur

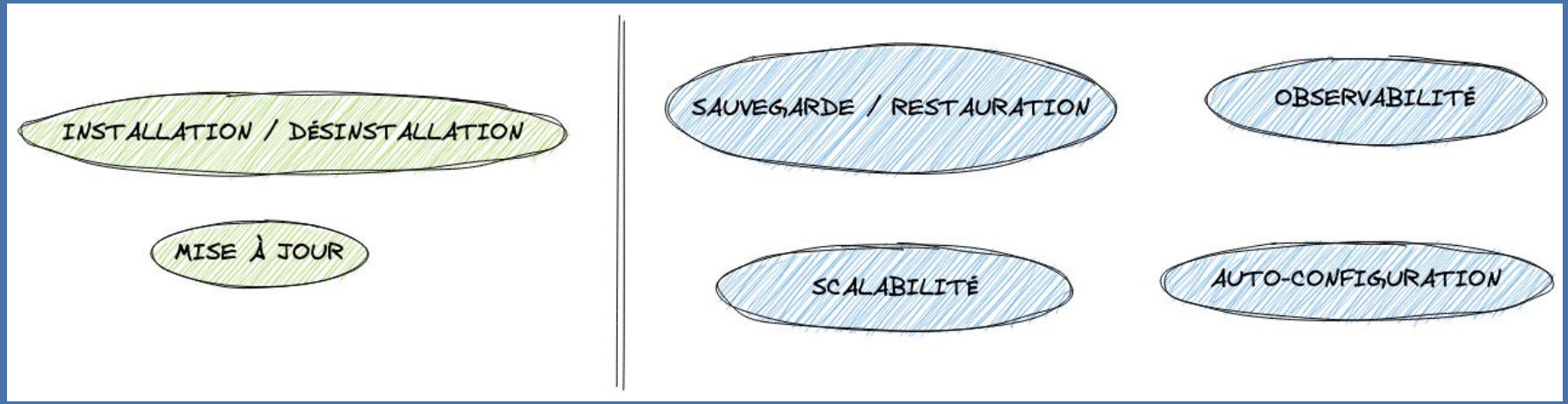
➡️ Gère une et une seule application (déployée ou non dans Kubernetes)

➡️ Essaie de maintenir l'état de l'application avec ce qui est déclaré dans la CR



source

Différentes fonctionnalités d’un opérateur



source

☝ 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

source

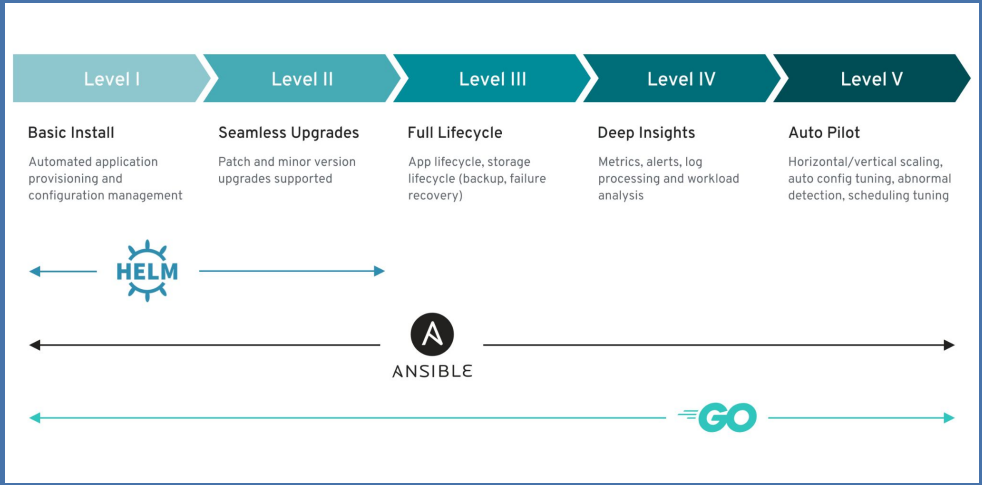
source

📦 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

- ➡️ Permet l’écriture en Helm, Ansible et Go
- ➡️ 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)

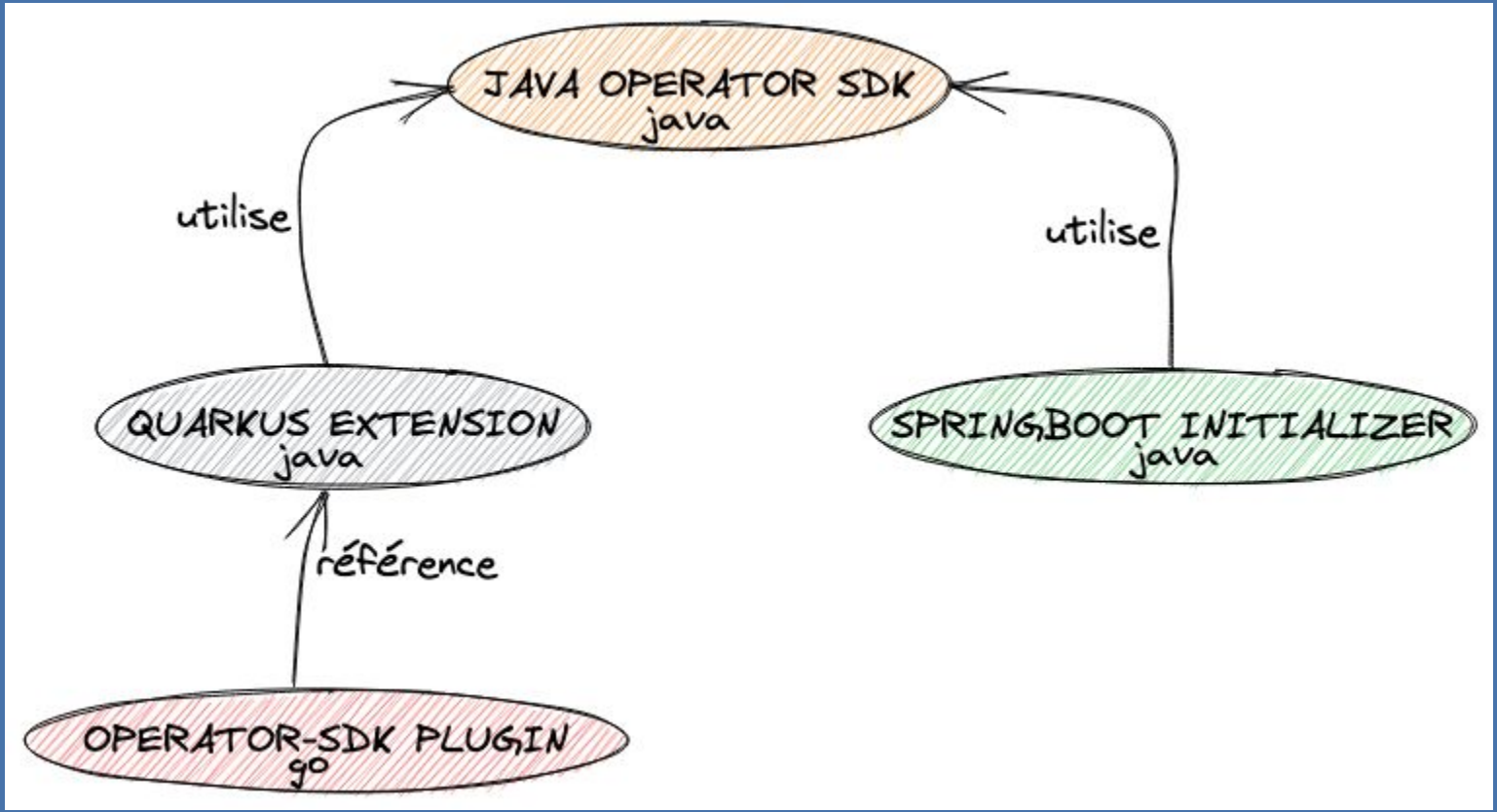
☕ Et en java ?

🔗 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

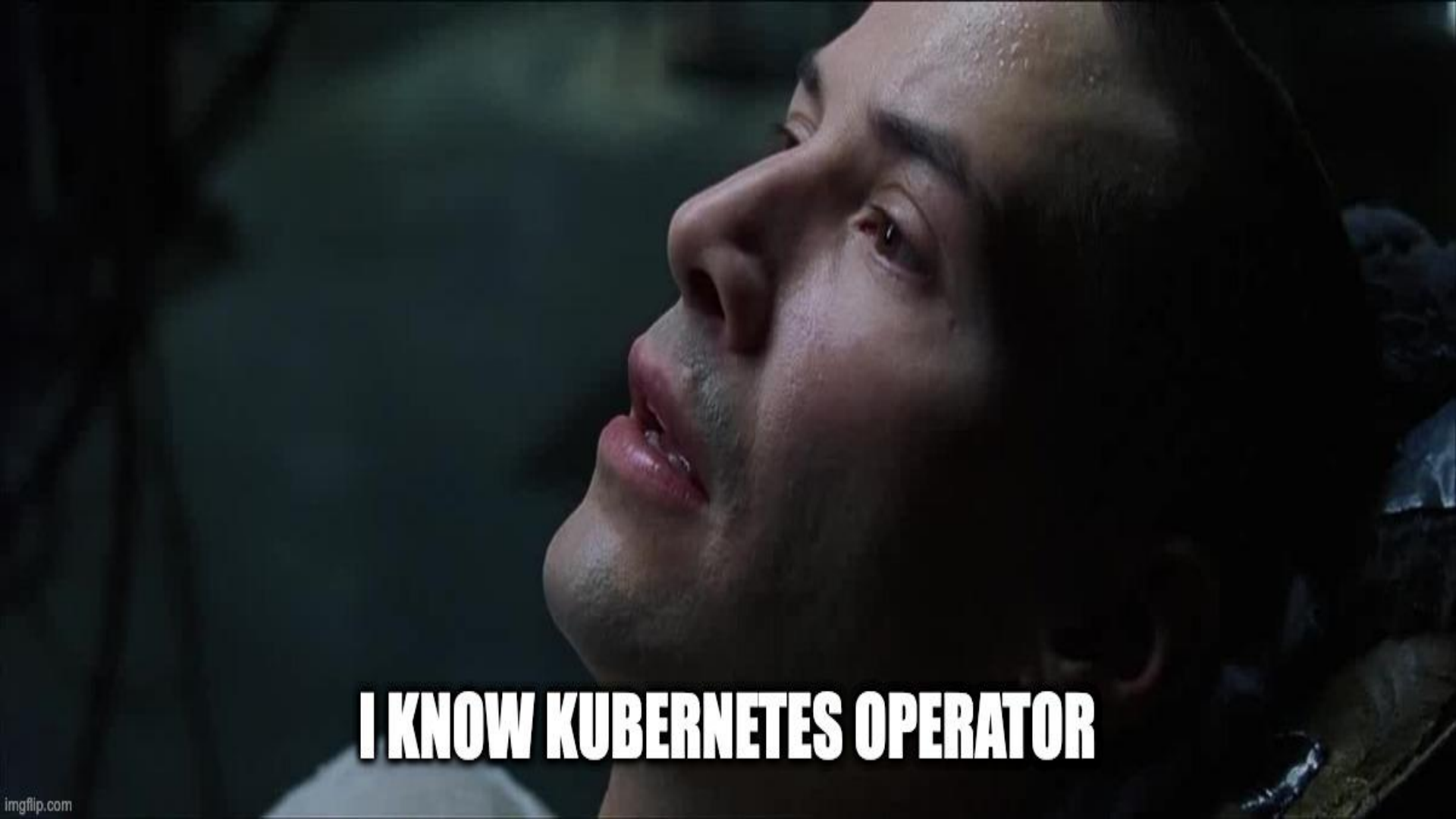
👋 Hello world !!

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

💊 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

source


I KNOW KUBERNETES OPERATOR
imgflip.com

💡 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

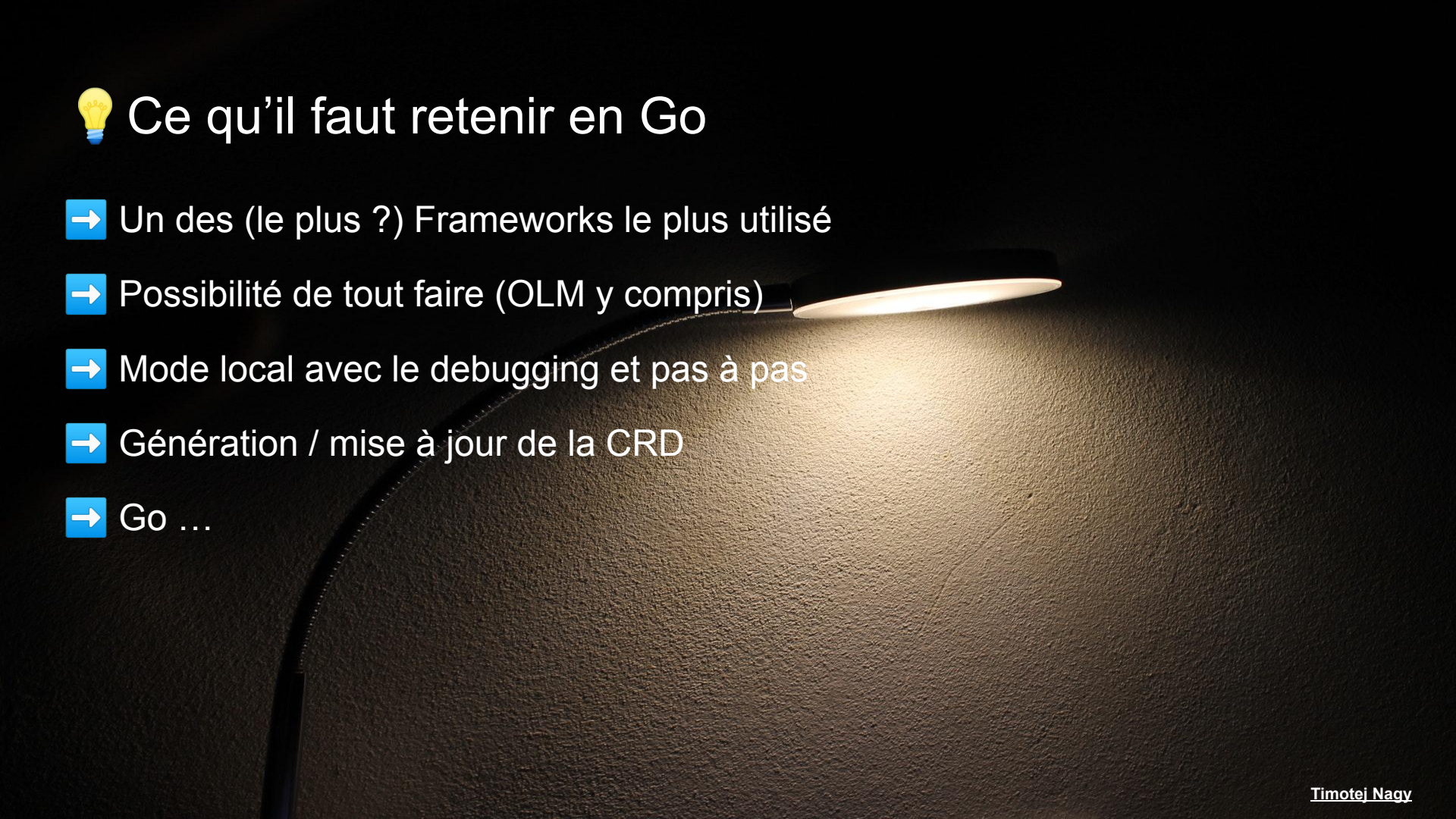
➡️ Mode local avec le debugging et pas à pas

➡️ 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 …
Timotej Nagy

💡 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 : controleur custom en Go

Brian McGowan

Bare Metal Cloud

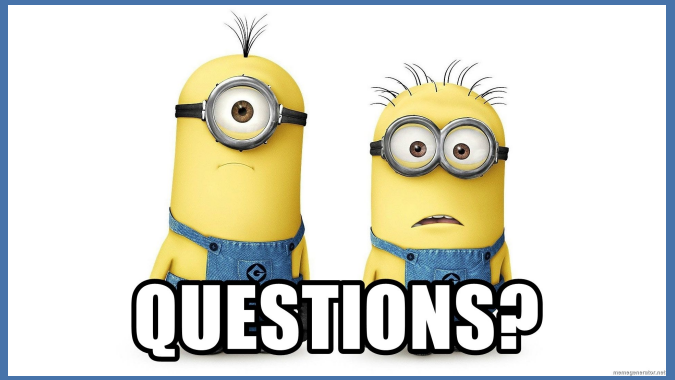
Hosted Private Cloud

Public Cloud

Web Cloud

Télécom

Stéphane Philippart



🏷️ Baby DevRel@OVHCloud 🥑 🦄 🎉

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

🔗 Démos :

- https://github.com/philippart-s/voxxed-days-helm-operator
- https://github.com/philippart-s/voxxed-days-go-operator
- https://github.com/philippart-s/voxxed-days-java-operator

🐦 @wildagsx

📝 https://philippart-s.github.io/blog

🐙 https://github.com/philippart-s/

🌍 https://www.linkedin.com/in/philippartstephane/

https://tinyurl.com/h3de23rm

Liens

© ArtVstudio

- 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://sdk.operatorframework.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://github.com/fabric8io/kubernetes-client/blob/master/doc/CHEATSHEET.md
- https://stackoverflow.com/a/61437982