

Développez avec une boucle de retour rapide sur votre cluster Kubernetes

Jeff Maury (@jeffmaury)
Red Hat



La problématique

- Eviter le syndrome 'Works on my machine'
- Architecture micro-services
- Utiliser un cluster pendant le dev
- Comment propager les changements du code ?

Solution sans outils

- Construire l'image Docker dans la chaîne de build (FMP, JIB)
- Pousser l'image sur la registry du cluster
- Redéploiement

Solution OpenShift

- Build dans le cluster
 - À partir d'une image donnée (Java, Node,...)
 - A partir d'un repo Git
 - A partir d'un tarball
- Le résultat est une nouvelle image
- Hooks à partir du repo Git

Solution OpenShift (2)

- Le redéploiement est automatique
 - ImageStreams:
 - Détecte automatique le changement d'une image
 - OpenShift va redéployer automatiquement s'il détecte que l'ImageStream a été modifié

Solution Odo

- OpenShift Do (odo)
 - Outil CLI simplifiant les concepts Kubernetes / OpenShift
 - Orienté pour les applications micro services

Solution Odo (2)

- Concepts
 - Service: service provisionné par l'infrastructure (ex base de donnée)
 - Composant: service applicatif
 - Stockage: stockage partagé
 - Application: collection des 3 concepts précédants

Solution Odo (3)

- Supervisord
 - Processus supervisor injecté dans le container
 - Détecte les changements de fichiers et lance un build S2I puis relance le processus du container
 - Utilisé en conjonction avec `odo push` ou `odo watch`

Workflow odo

- Odo create: création du composant
- Odo push: lance le build -> l'application est alors disponible
 - Pour Maven, on fait un build complete (téléchargement des dépendances)
- Modification de code
- Odo push
 - Pour Maven, on ne fait que la compilation et l'assemblage car les dépendances sont toujours là

Workflow odo (2)

- Odo create: création du composant
- Odo watch: lance le build -> l'application est alors disponible
 - Pour Maven, on fait un build complete (téléchargement des dépendances)
- Modification de code
 - Pour Maven, on ne fait que la compilation et l'assemblage car les dépendances sont toujours là

VSCode

- OpenShift Connector extension (disponible sur la marketplace)
- OpenShift Java Extension Pack (bientôt disponible)
 - OpenShift Connector extension
 - Java extensions (language, debugger, SpringBoot)



Demo: VSCode

VSCode

- Création d'une application
- Création d'un service (Postgresql)
- Création d'un composant (SpringBoot)
- Modification de code

IntelliJ

- OpenShift Connector plugins (bientôt disponible sur la plugins.jetbrains.com)





Demo: IntelliJ

IntelliJ

- Modification de code en continu

JBoss Tools

- OpenShift plugins disponibles sur la marketplace Eclipse
- De nombreux autres plugins (Wildfly/EAP, Hibernate, JAX-RS,...)





Demo: JBoss Tools



JBoss Tools

- Live debugging



Liens utiles

- Odo
 - github.com/redhat-developer/odo
- VSCode OpenShift Connector
 - marketplace.visualstudio.com/items?itemName=redhat.vscode-openshift-connector
- JBoss Tools
 - tools.jboss.org



Merci

- Questions / réponses
- On recherche des retours
 - N'hésitez pas

