

Kubernetes : déployer des applications en mode GitOps avec ArgoCD

DESCRIPTION

Grâce aux outils de l'infrastructure-as-code, il est maintenant facile de versionner et reproduire des déploiements applicatifs. Cependant, à cause d'une action manuelle ou d'un code mal exécuté /non idempotent, nous ne sommes pas à l'abri de divergences entre l'état désiré et l'état actuel d'un environnement, qui sont sources d'instabilité et rendent difficile la maintenance. En effet, traquer la source de désynchronisation peut s'avérer chronophage.

L'approche déclarative de Kubernetes et la philosophie GitOps permettent désormais de comparer les différences d'état entre ce qui est désiré et ce qui est actuellement déployé. Selon les principes GitOps, l'unique source de vérité devient le dépôt de code Git. Par ailleurs, si des divergences existent, un outil doit être capable de superviser et reporter celles-ci et au besoin re-synchroniser automatiquement le système. Les opérations manuelles de déploiement sont proscrites et doivent être pilotées par Git (PR, merge, etc.).

Cette formation s'adresse aux développeurs et aux ops souhaitant piloter le déploiement de leurs applications selon l'approche GitOps. A travers des exercices concrets, vous apprendrez à packager le déploiement de votre application sous la forme d'un chart Helm. Vous aurez à "versionner" celle-ci dans Git, puis à installer et configurer ArgoCD pour synchroniser ce dépôt de code et déployer votre application sur votre cluster Kubernetes de manière automatisée après chaque commit. Vous découvrirez les différents types de synchronisation pour sécuriser vos déploiements et faire des restaurations en cas de désastre. Vous aborderez aussi les différents patterns pour gérer les secrets dans le cadre GitOps, et vous manipulerez Sealed Secrets et Kubeseal pour pouvoir "versionner" des secrets dans Git. Enfin, vous apprendrez à monitorer vos déploiements.

OBJECTIFS PEDAGOGIQUES

- Découvrir les concepts GitOps
- Appréhender les outils adaptés à la philosophie GitOps
- Packager le déploiement d'une application sous la forme d'un chart Helm

Stage pratique Opérations

Code :
GITOP

Durée :
1 jour(s) (7,00 heures)

Exposés : **45 %**
Cas pratiques : **40 %**
Echanges d'expérience : **15 %**

Inter-entreprises :
Prochaines sessions
disponibles [sur notre site web](#).
Tarif : 950,00 € HT / participant

Intra-entreprise :
Tarifs et dates sur demande.

- Installer et configurer ArgoCD
- Synchroniser un dépôt de code Git pour déployer une application avec ArgoCD
- Explorer les différents types de synchronisation pour automatiser et sécuriser ses déploiements
- Gérer les secrets en mode GitOps avec Sealed Secrets et Kubeseal
- S'approprier les bonnes pratiques GitOps

PUBLIC CIBLE

Architectes, ops et développeurs qui utilisent déjà Kubernetes et souhaitent découvrir et mettre en place les pratiques GitOps pour améliorer les processus et la sécurisation des déploiements de leurs applications.

PRE-REQUIS

- Connaissance de l'utilisation standard et du déploiement d'applications dans Kubernetes (manipulation des ressources classiques).
- Avoir suivi la formation "Déployer vos applications en production avec Docker et Kubernetes" (DKUSR) est préférable.
- Connaissance de l'approche CI/CD.

METHODE PEDAGOGIQUE

Formation avec apports théoriques, échanges sur les contextes des participants et retours d'expérience pratique des formateurs, complétés de travaux pratiques et de mises en situation.

PROFIL DES INTERVENANTS

MODALITÉS D'ÉVALUATION ET FORMALISATION À L'ISSUE DE LA FORMATION

L'évaluation des acquis se fait tout au long de la session au travers des ateliers et des mises en pratique. Une évaluation à chaud sur la satisfaction des stagiaires est réalisée systématiquement en fin de session et une attestation de formation est délivrée aux participants mentionnant les objectifs de la formation, la nature, le programme et la durée de l'action de formation ainsi que la formalisation des acquis.

PROGRAMME PEDAGOGIQUE DETAILLE

Le concept GitOps

- Qu'est-ce que GitOps ? “Push” vs “Pull” ?
- Enjeux du GitOps
- Outils et modèle d'architecture pour déployer des applications en mode GitOps

Packager le déploiement d'une application avec Helm

- Introduction à Helm
- Création d'un chart Helm
- Publication d'un chart dans un repository Helm
- Mise en pratique : “Packager le déploiement d'une application cloud native sous la forme d'un chart Helm”
- Aller plus loin avec Helm : syntaxe avancée, hooks et tests

Synchroniser un dépôt de code Git pour déployer une application avec ArgoCD

- Présentation d'ArgoCD
- Installation et configuration d'ArgoCD
- Mise en pratique : “Installation et configuration d'ArgoCD avec son chart Helm”
- Les différents modes de synchronisation entre ArgoCD et un dépôt de code Git
- Restauration en cas d'échec d'une mise en production
- Mise en pratique : “Synchroniser un dépôt de code Git pour déployer avec ArgoCD l'application précédemment packagée, suivre le déploiement d'une mise à jour et restaurer l'application dans un état fonctionnel”
- Gérer l'ordonnancement d'un déploiement applicatif avec ArgoCD
- Mise en pratique : “Gérer l'ordonnancement d'un déploiement applicatif avec ArgoCD”
- Intégration d'ArgoCD dans une pipeline CI/CD
- Patterns de déploiement avancé : review apps, multi-clusters, gestion d'Argocd en mode GitOps, etc.
- Limites des pratiques GitOps avec ArgoCD

Gérer les secrets en mode GitOps

- Les différents patterns de gestion de secrets
- Présentation de Sealed Secrets et Kubeseal
- Mise en pratique : “Installation de Sealed Secrets et création de secrets avec Kubeseal”
- Récupération des secrets en cas de désastre

- Limite du couple Sealed Secrets/Kubeseal

Bonnes pratiques GitOps

- Gestion des dépôts de code Git
- Monitoring des déploiements applicatifs
- Mise en pratique : “Installer la stack de monitoring Prometheus/Grafana en mode GitOps et déployer un dashboard pour suivre les déploiements”

