

retengr

Architecture et DevOps

Durée : 5 jours (35h) – Réf : ARCDEV

Présentation

Il n'existe pas "d'architecture DevOps". En effet DevOps n'est pas une architecture, mais une approche qui vise à rapprocher les équipes de développement et d'exploitation afin d'améliorer la qualité des logiciels et réduire le temps de mise en production. En revanche, il existe des architectures, des process et des outils qui facilitent la mise en oeuvre de la pratique DevOps. Cette formation vous propose d'éclairer vos choix afin de tirer parti de l'approche DevOps. IaC, CI/CD, git, microservices, conteneurs, orchestration, Kubernetes, observabilité, ... sont autant de sujets qui seront abordés pour vous permettre de structurer votre activité et celle de vos collaborateurs afin de satisfaire aux objectifs du DevOps, à savoir : Améliorer la qualité des logiciels, réduire les délais de mise en production et faciliter la communication.

Méthodes pédagogiques

- La formation est composée de 60% de travaux pratiques (Mises en situation, débats, exercices).
- Une évaluation à chaud sera proposée au stagiaire à la fin du cours.
- Une feuille d'émargement par demi-journée de présence est fournie en fin de formation ainsi qu'une attestation de fin de formation si le stagiaire a bien assisté à la totalité de la session.

Objectifs

- Comprendre les principes fondamentaux de DevOps
- Concevoir une architecture microservices
- Définir une architecture CAAS (Container As A Service)
- Outiller la démarche devops
- Mettre en place une culture DevOps dans une organisation

Audience

- Développeurs
- Architectes
- Chefs de projet

Formateur

Le formateur est un expert du domaine qui intervient sur le sujet depuis plusieurs années en formation mais aussi en conseil.

Prérequis Positionnement

- Connaissance des principes de l'architecture logicielle
- Une connaissance minimale en java et linux
- Afin de valider les compétences et les pré-requis de chaque participant, en amont de la formation, le formateur organise un entretien téléphonique. Il confirme alors que le participant a le niveau nécessaire et que le contenu répond bien à ses attentes.

Programme

Les fondements de DevOps

- Contexte de l'apparition de DevOps
- Les promesses de DevOps
- Les principes fondamentaux de DevOps
- Quelle méthode pour mettre en oeuvre DevOps ? Lien avec l'agilité
- Comment implanter la culture DevOps dans une organisation ?

Exercices

- Débat sur les principes fondamentaux de DevOps
- Mise en situation sur l'implantation de la culture DevOps

Les outils dans le monde Devops

- Un écosystème très riche, et des outils pour chaque étape du cycle de vie du logiciel
- Quels critères pour choisir vos outils ?

- Les outils par domaine: Développement, gestion de code source, CI/CD, déploiement, environnement de run (Kubernetes) observabilité
- Gestion des compétences transverses

Exercices

- Echanges sur les outils utilisés par les participants
- Mise en place d'une plateforme minimale: Git, Gitlab, Docker, Kubernetes

Infrastructure as Code

- Comprendre l'approche de l'Infrastructure as Code (IaC)
- Différencier le provisioning d'infrastructure et le provisionning applicatif
- Appréhender les outils de provisionning d'infrastructure: Terraform, cloudformation, ...
- Appréhender les outils de provisionning applicatif: Ansible, puppet, chef...
- Automatiser le déploiement d'une infrastructure complète via un pipeline CI/CD

Exercices

- Mise en place d'une infrastructure via Terraform
- Provisionning d'une application via Ansible

Architecture microservices & Application Cloud Natives

- Présentation des microservices
- Microservices vs monolithes
- Scalabilité linéaire
- Applications cloud-native : les 12 facteurs
- Les patterns d'architecture microservices : API Gateway, Service Discovery, Circuit Breaker, Load Balancer

- Gestion de la montée en charge et tolérance aux pannes
- Cycle de vie : Développement, livraison et dédéploiement en continu
- Gouvernance, organisation des équipes de développement

Exercices

- Mise en place d'une application respectant une architecture microservices : analyse, packaging, déploiement

Git

- A quoi sert Git ?
- Les ops utilisent-ils Git ?
- Fonctionnement de Git
- Les branches : master, develop, feature, release, hotfix
- Les stratégies de Gitflow : Github flow, Gitlab flow, trunk based development
- Merge request, pull request Tags,
- releases

Exercices

- Création d'un dépôt Git local
- Création de branches, de tags

CI/CD

- Présentation des concepts de CI/CD
- cD : Delivery ou Deployment ?
- Les outils de CI/CD : Jenkins, Github Action, GitlabCI, CircleCI, TravisCI, ...
- Les étapes d'un pipeline CI/CD : Build, Test, Package, Deliver, Deploy

- Les bonnes pratiques : Tests, Sécurité, Qualité, Performance
- Les outils de qualité : SonarQube, Checkmarx, ...
- Versionnement automatique avec semantic versionning

Exercices

- Mise en place d'un pipeline CI/CD avec GitlabCI
- Démonstration : Intégration de Semantic Versionning

Conteneurisation & Docker

- Qu'est ce qu'un conteneur ?
- Docker: Présentation, installation, utilisation
- Les images Docker : Création, partage, utilisation
- Définition des rôles dans la conception des images Docker: Ops, Dev, Sec
- Les conteneurs Docker : Création, démarrage, arrêt, suppression
- Les réseaux Docker : Création, utilisation
- Les registres Docker : DockerHub, Gitlab Registry, AWS ECR, Azure ACR

Exercices

- Utilisation d'un conteneur Docker m partir d'une image existante
- Création d'une image Docker
- Livraison d'une image et réutilisation

Orchestration de conteneurs : Kubernetes

- Présentation de Kubernetes
- Architecture
- Les ressources Kubernetes : Pod, Deployment, Service, Ingress, StatefulSet, ...
- Les outils de déploiement : Helm, Kustomize, ...

Exercices

- Déploiement d'une application dans Kubernetes
- Créer des services
- Utiliser des volumes avec vos Pod
- Création d'un chart Helm pour une application complète

GITOPS

- A quoi sert GitOps ?
- Git comme référentiel de configuration
- Les différents outils de GitOps: ArgoCD, FluxCD, ...
- Les bonnes pratiques: Déploiement, Rollback, Monitoring
- Rapport de GitOps avec IaC

Exercices

- Mise en place d'un pipeline GitOps avec ArgoCD

La sécurité dans le monde DevOps : DevSecOps

- De quelle sécurité parle-t-on ?
- Les bonnes pratiques: Sécurité des conteneurs, Sécurité des images, Sécurité des déploiements
- Recherche des vulnérabilités (CVE) dès votre cycle de développement puis dans vos images Docker
- La gestion des certificats dans Kubernetes avec cert-manager
- Les outils: Clair, Anchore, ...
- Sécurisation des flux dans Kubernetes: les Networks Policies
- Mise en place d'un "Kubernetes Policy Engine" : OPA, Kyverno

Exercices

- Mise en place d'une politique de sécurité avec Kyverno
- Démonstration : cert-manager

Observabilité

- Présentation des concepts d'observabilité
- Les 3 piliers : Logs, Métriques, Traces
- Les outils : ELK, Prometheus, Grafana, Jaeger, Zipkin, ...

Exercices

- Mise en place d'une solution d'observabilité des métriques dans Kubernetes avec Prometheus et Grafana
- Option : Mise en place d'opentelemetry pour la collecte des traces

Modalités et délais d'accès à la formation

Les inscriptions sont possibles jusqu'à 48 heures ouvrées avant le début de la formation, en interentreprises, dans la limite des places disponibles. Pour les formations organisées en intra entreprise, la liste des participants peut être modifiée jusqu'à 24h ouvrées avant le début de la formation.

Accessibilité

RETENGR facilite l'accessibilité de ses formations. Cette formation est accessible aux personnes en situation de handicap. Si vous avez un besoin d'accès spécifique, contactez Céline BOURREIL (celine.bourreil@retengr.com) qui étudiera avec Handifeel's (notre référent handicap) votre demande et vous proposera les meilleures solutions