

# DevOps



Fiche de formation





## DURÉE

*1 journée (7 heures)*



## NIVEAU

*Débutant/  
Intermédiaire*



## PRÉREQUIS

*Connaissances de base du  
développement logiciel*



## LIEU

*Partout en France*

### OBJECTIFS DE LA FORMATION

- Présenter les objectifs DevOps et ses avantages
- Proposer un aperçu des outils d'automatisation dans la plateforme RAGNARØKKR
- Comprendre et mettre en place des indicateurs de performance
- Aborder des exemples réels et analyser des cas pratiques

### PUBLIC VISÉ

- DSI
- développeurs
- testeurs
- architectes informatiques

### AVANTAGES DE LA FORMATION

- Formateurs expérimentés
- Formation certifiée Qualiopi
- Travaux pratiques
- Conseils adaptés à vos besoins

# Plan de la formation

## 1

### INTRODUCTION À DEVOPS

Vue d'ensemble de ce qu'est DevOps, ses avantages et son importance dans le développement logiciel.

- Votre formateur
- Objectif technique et organisationnel
- Le projet fil rouge



## 2

### CULTURE DEVOPS

Comprendre la philosophie de DevOps, les pratiques de travail collaboratif et les valeurs qui sous-tendent une culture DevOps.



## 3

### LE PROCESSUS DEVOPS (INFRASTRUCTURE AS CODE)

Apprendre comment les processus de développement et d'exploitation peuvent être alignés pour améliorer la livraison de logiciels.

- Concepts de l'infrastructure as code
- Pratiques de développement d'infrastructure as code

# 4

## AUTOMATISATION (INTÉGRATION CONTINUE ET LIVRAISON CONTINUE)

Comment automatiser les processus de développement, de test, de déploiement et d'exploitation pour accélérer la livraison de logiciels.

- **Configuration de l'environnement de CI/CD**

Expliquer comment mettre en place un environnement d'intégration continue et de livraison continue, y compris les serveurs d'intégration continue, les outils de compilation et de déploiement, et les outils de gestion de la configuration.

- **Automatisation des builds et des tests**

Comment automatiser le processus de compilation et d'exécution des tests unitaires et fonctionnels

- **Gestion des versions de code**

Expliquer comment utiliser les outils de contrôle de version tels que Git pour gérer les versions de code et les branches de développement.

- **Déploiement continu**

Comment automatiser le processus de déploiement en production en veillant à ce que le déploiement soit rapide, fiable et cohérent.



# 5

## GESTION DE LA CONFIGURATION

Comment gérer les versions de code, les configurations et les infrastructures pour faciliter le déploiement et la maintenance des logiciels.

### • Les principes de base de la gestion de la configuration

Expliquer les principes fondamentaux de la gestion de la configuration, tels que la gestion des versions, la gestion des branches, la gestion des conflits, la documentation, etc.

### • La gestion de la configuration pour le code source

Montrer comment utiliser les outils de gestion de la configuration pour gérer les modifications apportées au code source, telles que la création de branches, la fusion, la révision, etc.

### • La gestion de la configuration pour les configurations de serveur

Montrer comment utiliser les outils de gestion de la configuration pour gérer les configurations de serveur, telles que l'installation de logiciels, la configuration de fichiers, la gestion de clés et certificats, etc

### • L'intégration continue de la gestion de la configuration

Montrer comment intégrer la gestion de la configuration dans le processus d'intégration continue (CI), de sorte que chaque modification apportée aux ressources informatiques soit automatiquement suivie et intégrée dans le pipeline CI.



# 6

## MONITORING ET SURVEILLANCE

Comment surveiller les performances des applications, détecter les problèmes et y remédier rapidement.

- **Les principes de base du monitoring et du logging**

Collecte de données, la visualisation, l'alerting et l'analyse.

- **Les types de données collectées**

Les métriques de performance, les journaux d'événements, les traces d'exécution, les données de sécurité, etc.

- **La surveillance des applications et des services**

Comment surveiller les applications et les services déployés en production, en utilisant des outils de monitoring et de logging pour détecter rapidement les problèmes et effectuer des actions correctives.





# 7

## SÉCURITÉ DEVOPS

Comprendre les bonnes pratiques de sécurité pour les environnements DevOps, notamment la gestion des identités, l'audit et la conformité.

### • Sécurité dès la conception

La sécurité doit être prise en compte dès la conception du système. Les équipes DevOps doivent s'assurer que les fonctionnalités de sécurité sont intégrées dès le début, plutôt que d'être ajoutées à la fin du cycle de vie du développement.

### • Tests de sécurité automatisés

Les tests de sécurité doivent être automatisés tout au long du cycle de vie de développement, afin de détecter les vulnérabilités et les erreurs de sécurité le plus tôt possible.

### • Surveillance continue de la sécurité

Surveiller en permanence les systèmes en production pour détecter les menaces de sécurité en temps réel. Des outils de surveillance et de détection de menaces automatisés peuvent aider à identifier rapidement les problèmes de sécurité.

### • Gestion des identités et des accès

Mise en place des pratiques de gestion des identités et des accès pour garantir que seules les personnes autorisées ont accès aux ressources et aux données sensibles.

### • Gestion des vulnérabilités

Surveiller les vulnérabilités et les corriger rapidement pour éviter les attaques potentielles.





# Merci !

## Nous contacter



HOSTEUR

67 cours Mirabeau

13100 Aix-en-Provence



(+33) 9 80 08 26 80

