



## DevOps : démarche et outils

LM003

Durée: 2 jours

### Public :

Développeurs, Architectes, Chef de projet, Responsable et acteur de la DSI, décideur...

### Objectifs :

A l'issue de la formation, le stagiaire sera capable de comprendre les concepts et les principaux outils d'une démarche DevOps.

### Connaissances préalables nécessaires :

avoir une connaissance des services IT. Une expérience sur un projet informatique est souhaitable (développement, intégration, exploitation...).

### Objectifs pédagogiques :

Comprendre les principes et les bénéfices du mouvement DevOps  
Appréhender les enjeux du DevOps dans une organisation  
Connaître les éléments constitutifs d'une usine logicielle DevOps  
Identifier les principaux outils de gestion de configuration  
Identifier les étapes de mise en oeuvre de la démarche, incluant l'anticipation des risques

### Programme :

#### Comprendre les principes et les bénéfices du mouvement DevOps

Bref historique du mouvement DevOps  
Principes fondamentaux : les cinq piliers CALMS (Culture, Automation, Lean, Measurement, Sharing)  
Les principales raisons de la mise en place d'une démarche DevOps  
Identifier l'ensemble des parties prenantes : clients, business (métiers), équipe de développement (Dev) et équipe d'exploitation (Ops)

#### Processus et bonnes pratiques à implémenter

Les procédés clés de la DSI DevOps et différences avec la DSI traditionnelle  
Considérations organisationnelles : la conduite du changement et l'implication de tous  
Automatiser tout ce qui peut l'être ?  
Intégration, livraison et déploiement continus  
La place des technologies Cloud et la virtualisation des environnements  
Les tests logiciels dans une démarche DevOps  
Surveillance, feedback et optimisation : l'amélioration continue



# Phirio

---

## Connaître les éléments constitutifs d'une usine logicielle DevOps

---

La chaîne d'outils DevOps : caractéristiques, contextes d'utilisation et démonstrations  
L'Open Source comme fondement de la culture DevOps

---

## Identifier les principaux outils de gestion de configuration

---

Gestion du code source : Git, Subversion, GitLab, GitHub et Bitbucket  
Intégration continue et déploiement continu (CI/CD) : Jenkins, Gitlab-CI, Bamboo, TeamCity, Travis, Maven, Gradle...  
Automatisation des tests logiciels : Selenium, Cucumber, Jasmine, Appium...  
Fournisseurs pour l'hébergement Cloud : Azure, AWS, GCP, OpenStack  
Gestion des conteneurs : la suite d'outils Docker et l'orchestration avec Kubernetes, les alternatives Mesos et Docker-Swarm  
Automatisation et gestion des configurations : Ansible, Chef, Puppet, SaltStack...  
L'Infrastructure-as-code avec Terraform (ou Ansible)  
Supervision et monitoring en continu : la stack ELK (ElasticSearch, Logstash, Kibana), Prometheus, Grafana, Zabbix, Nagios, EON...  
Outils de gestion de projet : JIRA, Trello, Slack, Hipchat...