

DevOps, démarche et outils

2 j (14 heures)

Ref : DDO

Public

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

Pré-requis

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

Moyens pédagogiques

Modalité : Formation présentielle ou Formation distancielle (classe virtuelle) - Inter / Intra - Groupes de 4 à 12 stagiaires

Méthodes : Présentation des concepts, discussion technique, démonstrations, exercices et TP

Matériel :

Présentiel : Un poste informatique par stagiaire connecté à internet, à une imprimante en réseau et au réseau informatique,

Les salles sont équipées d'un tableau interactif ou d'un vidéoprojecteur et d'un paperboard

Distanciel : Aelion met à disposition de chaque stagiaire

- Un PC équipé des outils et logiciels nécessaires à la formation auquel vous accédez via un outil de prise en main à distance

- Un accès à un outil de classe virtuelle (Meet)

Support de formation : Un support de formation sera remis à chaque stagiaire en fin de formation : plateforme collaborative intégrant le code source des exercices réalisés en formation, webographie, mémos

Modalités de suivi et d'évaluation

Questionnaire d'évaluation des pré-requis, suivi des connaissances tout au long de la formation par des questionnements, exercices..., Evaluation des acquis en fin de formation

Questionnaire d'évaluation de la satisfaction en fin de stage, feuille de présence émarginée par demi-journée par les stagiaires et le formateur, Attestation de fin de formation

DevOps propose une démarche et une philosophie pour améliorer la qualité des services IT en associant les équipes de développement et d'exploitation ainsi que leurs compétences. DevOps est un ensemble de pratiques qui met l'accent sur la collaboration et la communication entre les développeurs de logiciels et les professionnels des opérations informatiques, en automatisant le processus de livraison de logiciels et les changements d'infrastructure. Le terme DevOps est né de l'union du « development » et des « opérations » dont l'objectif est de favoriser une meilleure communication entre les deux équipes.

Le mouvement DevOps vise à créer une culture et un environnement dans lesquels la conception, les tests et la diffusion de logiciels peuvent être réalisés rapidement, fréquemment et efficacement. Ce n'est pas seulement une méthodologie, c'est une véritable philosophie de travail.

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

Action collective OPCO ATLAS - [Inscription CampusAtlas](#)

Objectifs

- 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 œuvre de la démarche, incluant l'anticipation des risques

Programme détaillé

DEMARCHE, ACTEURS ET ENJEUX DEVOPS

- Cycle de vie complet d'une application (ALM - Application Lifecycle Management) et identification des problématiques qualités (tests, sécurité, etc.) et organisationnelles (agile, time to market, etc.)
- Manifeste et culture DevOps
- Patterns et anti patterns DevOps
- Technologies et pratiques liées au mouvement DevOps (Infra-as-code, Continuous Intégration, Continuous Deployment/Delivery)

CONNAITRE LES ELEMENTS CONSTITUTIFS D'UNE USINE LOGICIELLE DEVOPS

- Identifier les outils nécessaires à la fabrique logicielle pour produire des livrables de qualité
- Gestion du code avec Git et les branches
- Tests et déploiement automatisés :
- Identifier et définir les typologies de tests
- Intégration et déploiement continue avec des pipelines CI/CD

IDENTIFIER LES PRINCIPAUX OUTILS DE GESTION DE CONFIGURATION

- Introduction aux problématiques de gestion de configuration, de provisioning et d'orchestration
- Infrastructure as Code
- Infrastructure As A Service (ou Container as a service)
- Exemples d'outils
- Gestion du code source avec une application Git : 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...

Exemple de code utilisant Terraform ou Kubernetes mettre en oeuvre une application logicielle "Cloud-ready"

IDENTIFIER LES ETAPES DE MISE EN ŒUVRE DE LA DEMARCHE DEVOPS, INCLUANT L'ANTICIPATION DES RISQUES

Définir l'organisation d'un projet DevOps pour réussir sa mise en œuvre

- Considérations organisationnelles : la conduite du changement et l'implication de tous
 - Comment évoluer vers une organisation DevOps en intégrant l'infrastructure existante ?
 - Quelles actions peuvent faciliter l'adoption et/ou la montée en compétences des équipes Dev et Ops.
 - Cibler les parties à automatiser (notamment parmi les tests, l'Intégration, la livraison et déploiement continus)
 - La place des technologies Cloud et la virtualisation des environnements
 - Surveillance, feedback et optimisation : l'amélioration continu
-