Programme de formation

DevOps et Multi-Cloud

Décrire les origines, les principes, et les avantages du framework DevOps
Mettre en oeuvre l'automatisation (IaC, CI/CD, CaaS) dans un environnement Cloud
Identifier les outils d'automatisation et d'orchestration adaptés à la gestion de vos applications dans le Cloud
Lister les bénéfices des technologies d'intégration et de déploiement continus
Déployer et surveiller des applications sur des Clouds publics.

Prérequis

avoir les connaissances équivalentes. Avoir connaissance des concepts du Cloud, ainsi que les bases de l'un des trois grands "Cloud Providers" suivants : AWS, Azure ou GCP. Disposer de solides compétences en système / réseau et en Linux / Unix.

Durée

5 jours

Public

Architectes systèmes et réseaux, architectes logiciel, développeurs, chefs de projets, ou ingénieurs systèmes débutant dans DevOps

Moyens et méthodes pédagogiques

- La formation alterne entre présentations des concepts théoriques et mises en application à travers d'ateliers et exercices pratiques (hors formation de type séminaire).
- Les participants bénéficient des retours d'expérience terrains du formateur ou de la formatrice
- Un support de cours numérique est fourni aux stagiaires

Modalités d'évaluation

- En amont de la session de formation, un questionnaire d'auto-positionnement est remis aux participants, afin qu'ils situent leurs connaissances et compétences déjà acquises par rapport au thème de la formation (variable selon la formation suivie).
- En cours de formation, l'évaluation se fait sous forme d'ateliers, exercices et travaux pratiques de validation, de retour d'observation et/ou de partage d'expérience.
- En fin de session, le formateur évalue les compétences et connaissances acquises par les apprenants grâce à un questionnaire reprenant les mêmes éléments que l'auto-positionnement, permettant ainsi une analyse détaillée de leur progression.

Programme de formation

Jour 1

Origine et définition de la culture DevOps

Origine DevOps
-Les principes DevOps
-Les enjeux liés au Cloud
-Les impacts organisationnels

Pourquoi l'arrivée du DevOps ?
-Les bonnes pratiques issues du développement logiciel
-L'IaC (Infrastructure as Code)
-Le développement logiciel (Agilité, 12 facteurs, micro-services)
-Chaîne de production logicielle
-L'intégration, le déploiement et la livraison continus

Chaîne de développement logiciel

Gestion des sources
-Objectifs des outils de VCS / SCM (Version Control System / System Control Management)
-Terminologie (branches, tags, commits...)
-Comment gérer les branches ?
-Mettre en place un workflow de collaboration
-Exemple avec Git
Exemples de travaux pratiques (à titre indicatif)
-Mise en oeuvre d'une solution de VCS avec GitLab ou GitHub, et manipulations
-Création de branche, merge, pull request, commit...

Outils de build
-Objectifs des outils de build logiciel
-Terminologie (artefacts, formats...)
-Comment gérer ses builds ?
-Tests, qualité, mesures...
-Mettre en place un outil de build avec SonarQube
Exemples de travaux pratiques (à titre indicatif)
-Mise en oeuvre de SonarQube
-Lancement de build logiciel
-Rapport de build

Jour 2

CI/CD (Continuous Integration / Continuous Delivery)
-Objectifs des outils de CI/CD
-Concepts (pipeline, testing, historique, logs...)
-Architecture maîtres / esclaves
-Comment construire ses pipelines CI/CD ?
-Analyse du code, tests, build, livraison
-Mettre en place un outil de CI/CD avec Jenkins, GitHub Actions et GitLab CI/CD
Exemples de travaux pratiques (à titre indicatif)
-Mise en oeuvre de Jenkins, GitHub Actions, ou GitLab CI/CD
-Création de pipeline

IaC (Infrastructure as Code)

Automatisation
-Objectifs des outils d'automatisation
-Provisioning et infrastructure immutable
-Présentation des outils du marché
-Exemple avec Ansible
Exemples de travaux pratiques (à titre indicatif)
-Mise en oeuvre d'Ansible et déploiement automatisé d'un serveur
-Automatisation de l'installation de serveurs dans une architecture multi-tiers

Orchestration
-Objectifs des outils d'orchestration
-Provisioning d'architectures complexes
-Présentation des outils du marché
-Prise en main de Terraform
-Faire du Multi-Cloud avec du code Terraform multi-provider
-Exemple avec Terraform
Exemples de travaux pratiques (à titre indicatif)
-Mise en oeuvre de Terraform
-Déploiement d'une application multi-tiers sur 2 Clouds publics

Jour 3

CaaS (Container as a Service)

Conteneurisation
-Origine et apports des conteneurs avec Docker
-Concepts et architecture Docker
-Outils Docker : CLI, Dockerfile, Docker Compose
Exemples de travaux pratiques (à titre indicatif)
-Mise en oeuvre de Docker
--Découverte de la CLI
--Création de Dockerfile
--Construction d'images Docker
--Optimisation des images Docker avec le multi-stage
-Création d'images d'une application multi-tiers
--Provisioning avec Docker Compose

Intégration Docker et CI/CD
-Apport des conteneurs dans une chaîne de développement logiciel
-Intégration de Docker à Jenkins, gestion des esclaves
-Pilotage des conteneurs de Docker build depuis Jenkins
-Gestion des artefacts sous forme d'images Docker
Exemples de travaux pratiques (à titre indicatif)
-Mise en oeuvre de Docker dans un pipeline CI/CD
-Création d'esclaves Docker dans Jenkins
-Build d'images Docker depuis le pipeline Jenkins
-Déploiement automatisé avec Docker Compose depuis Jenkins

Jour 4

Orchestration de containers CaaS
-Objectifs des outils d'orchestration de containers
-Présentation des outils du marché : Docker Swarm, Kubernetes...
-Architecture de Kubernetes
-Présentation des ressources Kubernetes
-Présentation des concepts de mises à jour et de scalabilité des applications en micro-services
-Utilisation de Kubernetes avec les registres de conteneurs publics et privés (GitHub, ACR...)
Exemples de travaux pratiques (à titre indicatif)
-Mise en oeuvre de Kubernetes en local
--Création des manifests de déploiement de l'application multi-tiers
--Mise à l'échelle de l'application
--Mise à jour de l'application
-Déploiement de l'application multi-tiers en Multi-Cloud
--Provisioning d'un cluster Kubernetes dans 2 Clouds publics
--Déploiement de l'application multi-tiers précédente dans les 2 Clouds publics
-Automatisation
--Provisioning automatisé d'un cluster Kubernetes avec Terraform
--Monitoring du cluster K8s avec des alertes

Jour 5

Gestion de solutions Multi-Cloud et hybrides

Projection de ressources on-premises et hybrides
-Principe et avantages des solutions Multi-Cloud
-Gestion centralisée des solutions Multi-Cloud
--Simplicité, économie, cohérence, automatisation
-Solutions du marché : Azure Arc, AWS Hybrid Cloud
Exemples de travaux pratiques (à titre indicatif)
-Activation dans Azure Arc ou AWS Hybrid Cloud de ressources (serveurs, clusters K8s, data services...) externes (i.e. sur un autre Cloud, ou on-premise) en vue de leur gestion et surveillance centralisées