

Formation Conception et implémentation de solutions Microsoft DevOps (AZ-400)

Cette formation vise l'acquisition des connaissances et compétences clés pour concevoir et implémenter les processus et les pratiques DevOps. Les apprenants seront formés sur la manière de structurer les opérations DevOps, l'utilisation efficace de Git, sa mise à l'échelle pour une entreprise, les artefacts, la gestion des dépendances, des secrets, l'intégration continue ou encore l'implémentation d'une stratégie de mise en production. La formation prépare également au passage de l'examen de certification AZ-400, Designing and Implementing Microsoft DevOps solutions (en ligne et en différé).

Durée

4 jours

Objectifs pédagogiques

- ❖ Décrire les avantages et la mise en œuvre d'un contrôle de code source
- ❖ Expliquer et mettre en œuvre l'intégration et le déploiement continus
- ❖ Décrire les tâches de génération, de livraison et de déploiement
- ❖ Dimensionner Git pour un environnement DevOps d'entreprise
- ❖ Configurer l'intégration continue avec Azure DevOps
- ❖ Développer et mettre en œuvre une stratégie de génération de code pour les conteneurs
- ❖ Intégrer des tests automatiques pour assurer la qualité du code
- ❖ Gérer la sécurité et la conformité dans les projets DevOps

Public

ingénieurs devops, développeurs

Prérequis

Connaissances de base sur Azure, le contrôle de version, les principes de base du développement logiciel et l'Agilité.

Programme de formation

Phase d'inclusion

Accueil des participants, présentation des objectifs et contextes professionnels de chacun.

Introduction à la formation sur les solutions DevOps de Microsoft

Présentation de DevOps : définitions, équipes, objectifs partagés et délais
Choisir le projet approprié : projets Greenfield et Brownfield, systèmes d'enregistrement et d'engagement, résistance au changement et KPI
Pratiques et principes de développement agile
Structure organisationnelle, collaboration et outils
Les outils Azure DevOps et GitHub : autorisation et accès, migration, stratégie de gestion des licences
Tableaux de projet GitHub et Azure Boards
Avantages et meilleures pratiques du contrôle de version, utilisation de Git
Azure Repos et GitHub

Git et DevOps

Structurer un dépôt Git
Gérer les branches et les différents workflows
Collaboration et pull requests dans Azure Repos
La dette technique, la complexité, les métriques de qualité et les plans pour des revues de code efficaces
Les hooks de Gut et leur utilisation dans le processus de développement
Inner source et duplication (fork)
Gérer les référentiels Git

Intégration continue avec Azure Pipelines et GitHub Actions

Le concept de pipelines dans Azure DevOps
Gestion des pools et agents
Comprendre les travaux parallèles
Les 4 piliers de l'intégration continue
Propriétés de builds
Activer l'intégration continue avec Azure Pipelines
Stratégies de pipeline
Introduction aux actions GitHub

Concevoir une stratégie de build de conteneur

Concevoir et implémenter une stratégie de mise en production

Introduction à la livraison continue
Créer un pipeline de versions
Explorer les recommandations de version
Environnements de mise à disposition et de test
Gérer et modulariser les tâches et les modèles
Automatiser l'inspection de la santé

Implémenter un déploiement continu sécurisé avec Azure Pipelines

Introduction aux modèles de déploiement
Implémenter le déploiement bleu-vert et les bascules de fonctionnalités
Implémenter les versions Canary et le lancement sombre
Mettre en œuvre des tests A/B et un déploiement progressif de l'exposition
Intégration aux systèmes de gestion des identités
Gérer les données de configuration des applications

Infrastructure as Code avec Azure

Explorez l'infrastructure as code et la gestion de la configuration
Créer des ressources Azure à l'aide des modèles Azure Resource Manager
Créer des ressources Azure à l'aide d'Azure CLI
Explorez Azure Automation avec DevOps
Implémenter la configuration de l'état souhaité (DSC)
Implémenter Bicep

Stratégie de gestion des dépendances

Dépendances de package
Gestion des packages
Migrer des artefacts de consolidation et sécurisés
Implémenter une stratégie de contrôle de version
Les packages GitHub : visibilité, publication, installation, suppression&...

Commentaires continus

Mettre en œuvre des outils pour suivre
l'utilisation et le flux
Développer des tableaux de bord de
surveillance et d'état
Partager les connaissances au sein des
équipes
Concevoir des processus pour
automatiser l'analyse des applications
Gérez les alertes, les rétrospectives
irréprochables et une culture juste

Sécurité et validation du code

Introduction au Secure DevOps
Implémenter un logiciel open source
Analyse de la composition logicielle
Analyseurs statiques
OWASP et analyseurs dynamiques
Surveillance de la sécurité et gouvernance

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.
- **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 cohérence avec les objectifs pédagogiques visés.
- **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.