

Formation & Certification • AZ-400

Conception et implémentation de solutions Microsoft DevOps

La formation "Conception et implémentation de solutions Microsoft DevOps" forme les participants à intégrer les pratiques DevOps avec Azure, automatiser le cycle de vie des applications, concevoir des pipelines CI/CD, gérer Git, superviser l'infrastructure et assurer une livraison continue de qualité.

Learn • Practice • Certify

DevOps • CI/CD • Azure

Distributeur officiel Certiport

Durée 4 jours (28 h)	Examen AZ-400
Modalité Distanciel Présentiel sur demande	Niveau Intermédiaire DevOps / Cloud

INSCRIPTION / RÉSERVATION



Je m'inscris
maintenant

- **Learn** : Comprendre les concepts DevOps et les outils Azure DevOps/GitHub pour automatiser le cycle de vie.
- **Practice** : Mettre en œuvre des scénarios de gestion de code, CI/CD, tests automatisés et infrastructure as code.
- **Certify** : Se préparer efficacement à la certification AZ-400 grâce à un alignement complet sur les objectifs de l'examen,

OBJECTIFS PÉDAGOGIQUES

- Intégrer DevOps pour améliorer la collaboration entre développement et exploitation.
- Concevoir et implémenter des pipelines CI/CD robustes avec Azure DevOps ou GitHub Actions.
- Gérer le contrôle de version avec Git et définir des stratégies de branche adaptées.
- Déployer et superviser des applications et infrastructures Azure selon les bonnes pratiques.
- Se préparer à la certification AZ-400 en couvrant tous les objectifs officiels.

PUBLIC CIBLE

- Ingénieurs DevOps.
- Développeurs de logiciels/Administrateurs de systèmes et d'infrastructures/Architectes de solutions cloud/Chefs de projets IT avec une orientation DevOps.

PRÉREQUIS

- Bonne compréhension des concepts de développement logiciel et d'administration système.
- Connaissance de base d'Azure et des services cloud.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Concevoir et implémenter des processus et des communications

Concevoir et implémenter la traçabilité et le flux de travail

- Concevoir et implémenter une structure pour le flux de travail, y compris GitHub Flow.
- Concevoir et implémenter une stratégie pour les cycles de commentaires, y compris les notifications et les problèmes GitHub.
- Concevoir et implémenter l'intégration pour le suivi du travail, y compris les projets GitHub, Azure Boards et les dépôts.
- Concevoir et implémenter la traçabilité des sources, des bogues et de la qualité.

Concevoir et implémenter des métriques et des tableaux de bord DevOps

- Concevoir et implémenter des métriques et des requêtes appropriées pour DevOps.
- Concevoir et implémenter un tableau de bord incluant les durées des cycles, le temps de récupération et les délais.
- Concevoir et implémenter des métriques et requêtes pour la planification, le développement, les tests, la sécurité, la livraison et les opérations.

Configurer la collaboration et la communication

- Documenter un projet en configurant des wikis et des diagrammes de processus (Markdown, Mermaid).
- Configurer la documentation de mise en production, y compris les notes de version et la documentation d'API.
- Automatiser la création de la documentation à partir de l'historique Git.
- Configurer l'intégration via des webhooks.
- Configurer l'intégration entre Azure Boards et les référentiels GitHub.
- Configurer l'intégration entre GitHub ou Azure DevOps et Microsoft Teams.

Concevoir et implémenter une stratégie de contrôle de code source

Concevoir et implémenter des stratégies de branche

- Concevoir des stratégies de branche, y compris trunk-based, branche de fonctionnalité et branche de mise en production

PROGRAMME DE LA FORMATION – DÉTAILLÉ

.Concevoir et implémenter un workflow de pull request avec stratégies et protections de branche.

- Implémenter des restrictions de fusion de branches via politiques et protections.

Configurer et gérer des dépôts Git

- Concevoir une stratégie pour la gestion des grands fichiers (Git LFS, git-fat).
- Concevoir une stratégie de mise à l'échelle et d'optimisation de dépôt (partage inter-dépôt, scalar).
- Configurer les autorisations du dépôt de contrôle de code source.
- Configurer des étiquettes pour organiser le dépôt.
- Récupérer des données spécifiques via des commandes Git.
- Supprimer des données sensibles du contrôle de code source.

Concevoir et implémenter des pipelines de build et de mise en production

Concevoir et implémenter une stratégie de gestion des packages

- Recommander des outils de gestion de packages (GitHub Packages, Azure Artifacts).
- Concevoir et implémenter des flux et vues de package pour des packages locaux et en amont.
- Concevoir et implémenter une stratégie de version des dépendances (SemVer, CalVer).
- Concevoir et implémenter une stratégie de versioning pour les artefacts de pipeline.

Concevoir et implémenter une stratégie de test pour les pipelines

- Concevoir des barrières de qualité et de mise en production (sécurité, gouvernance).
- Concevoir une stratégie de test complète : tests locaux, unitaires, intégration, charge.
- Implémenter des tests dans le pipeline (tâches de test, agents, intégration des résultats).
- Implémenter l'analyse de la couverture de code.

PROGRAMME DE LA FORMATION – DÉTAILLÉ

Concevoir et implémenter des pipelines

- Sélectionner la solution d'automatisation (GitHub Actions, Azure Pipelines).
- Concevoir l'infrastructure d'exécution (runners GitHub, agents Azure DevOps).
- Concevoir et implémenter l'intégration entre dépôts GitHub et Azure Pipelines.
- Développer des règles de déclenchement de pipeline.
- Développer des pipelines en YAML.
- Concevoir l'ordre d'exécution des jobs (parallélisme, multi-phases).
- Développer des scénarios complexes (hybrides, modèles de VM, agents auto-hébergés).
- Créer des éléments réutilisables (modèles YAML, groupes de tâches, variables et groupes de variables).
- Concevoir et implémenter des vérifications et approbations avec des environnements YAML.

Concevoir et implémenter des stratégies de déploiement

Concevoir et implémenter des déploiements

- Concevoir une stratégie de déploiement (blue/green, canary, feature flags, tests A/B).
- Concevoir un pipeline garantissant l'ordre fiable des dépendances.
- Réduire les temps d'indisponibilité (VIP swap, load balancing, déploiements progressifs, slots).
- Concevoir un plan de correctif pour les patches à priorité élevée.
- Concevoir et implémenter une stratégie de résilience pour les déploiements.
- Implémenter les indicateurs de fonctionnalités via Azure App Configuration.
- Implémenter le déploiement d'applications via conteneurs, binaires et scripts.
- Implémenter des tâches de base de données dans les déploiements.

Concevoir et implémenter l'infrastructure en tant que code (IaC)

Définir et implémenter une stratégie IaC

- Recommander une technologie de gestion de configuration pour l'infrastructure applicative.
- Définir une stratégie IaC (contrôle de code source, tests et déploiements automatisés).



PROGRAMME DE LA FORMATION – DÉTAILLÉ

- Concevoir et implémenter la configuration d'état souhaité (ARM, Bicep, Azure Automation, Automanage).
- Concevoir et implémenter des environnements de déploiement Azure pour des déploiements à la demande.

Gérer les pipelines DevOps

- Superviser l'intégrité des pipelines (taux d'échec, durée, fiabilité des tests).
- Optimiser les pipelines en termes de coûts, de performances et de fiabilité.
- Concevoir et implémenter une stratégie de rétention pour les artefacts et dépendances.
- Migrer des pipelines classiques vers des pipelines YAML dans Azure Pipelines.

Développer un plan de sécurité et de conformité DevOps

Concevoir et implémenter des méthodes d'authentification et d'autorisation

- Choisir entre principaux de service et identités managées.
- Implémenter et gérer l'authentification GitHub (Apps, GITHUB_TOKEN, PAT).
- Configurer des projets et équipes dans Azure DevOps.
- Implémenter et gérer des connexions de service et PAT Azure DevOps.
- Concevoir et implémenter des autorisations et rôles dans GitHub et Azure DevOps.
- Recommander des niveaux d'accès adaptés (stakeholders, collaborateurs externes).
- Configurer des projets et équipes dans Azure DevOps.

Gérer les informations sensibles et l'automatisation de la sécurité

- Implémenter et gérer des secrets, clés et certificats via Azure Key Vault.
- Implémenter et gérer des secrets dans GitHub Actions et Azure Pipelines.
- Concevoir une stratégie de gestion des fichiers sensibles (fichiers sécurisés Azure Pipelines).
- Concevoir des pipelines pour empêcher la fuite d'informations sensibles



PROGRAMME DE LA FORMATION – DÉTAILLÉ

Automatiser l'analyse de la sécurité et de la conformité

- Concevoir une stratégie d'analyse de sécurité (dépendances, code, secrets, licences).
- Configurer la sécurité DevOps dans Microsoft Defender for Cloud.
- Configurer GitHub Advanced Security et l'intégrer à Microsoft Defender for Cloud.
- Automatiser l'analyse des conteneurs (images, CodeQL en conteneur).
- Automatiser l'analyse open source (Dependabot : licences, vulnérabilités, mises à jour).

Implémenter une stratégie d'instrumentation et de supervision

Configurer l'observabilité pour les environnements DevOps

- Configurer la supervision pour un environnement DevOps.
- Configurer Azure Monitor et Log Analytics pour les outils DevOps.
- Configurer la collecte de télémétrie (Application Insights, VM Insights, Container Insights, Storage & Network Insights).
- Configurer la supervision dans GitHub (insights et graphiques).
- Configurer des alertes pour les événements de GitHub Actions et Azure Pipelines.

Analyser les métriques et journaux

- Inspecter les indicateurs de performance de l'infrastructure (CPU, mémoire, disque, réseau).
- Analyser les métriques à partir des données de télémétrie (utilisation et performance applicative).
- Inspecter le suivi distribué via Application Insights.
- Interroger les journaux avec des requêtes Kusto (KQL) de base.

