

Programme de formation AZ-400 : Designing and Implementing a Microsoft DevOps Solution

(Préparation certification Microsoft AZ-400)

Mise à jour : 22/12/2025

DESCRIPTION DE LA FORMATION :

Les professionnels Azure DevOps combinent opérationnels, processus et technologies pour fournir en permanence des produits et des services de valeur qui répondent aux besoins et objectifs des utilisateurs finaux. Cette formation vous apportera les connaissances suffisantes pour y parvenir.

OBJECTIFS PEDAGOGIQUES :

A l'issue de cette formation, les participants seront en capacité de :

- Développer le DevOps d'entreprise
- Implémenter CI avec Azure Pipelines et GitHub Actions
- Concevoir et implémenter une stratégie de mise en production
- Implémenter un déploiement continu sécurisé avec Azure Pipelines
- Gérer l'infrastructure en tant que code avec Azure et DSC
- Implémenter la sécurité et valider les bases de code pour la conformité
- Concevoir et implémenter une stratégie de gestion des dépendances
- Implémenter des commentaires continus

MÉTHODES & MODALITÉS PÉDAGOGIQUES :

- Cette formation sera principalement constituée de théorie et d'ateliers techniques qui permettront d'être rapidement opérationnel.
- Support : un support de cours officiel Microsoft sera remis aux participants au format électronique.
- Evaluation : Les acquis sont évalués tout au long de la formation par le formateur (Prérequis évalués avant la formation, questions régulières, travaux pratiques, QCM ou autres méthodes).
- Formateur : le tout animé par un consultant-formateur expérimenté, nourri d'une expérience terrain, et accrédité Microsoft Certified Trainer.
- Satisfaction : à l'issue de la formation, chaque participant répond à un questionnaire d'évaluation qui est ensuite analysé en vue de maintenir et d'améliorer la qualité de nos formations.
- Suivi : une feuille d'émargement par demi-journée de présence est signée par chacun des participants.
- Cette formation peut être dispensée en format inter-entreprises ou intra-entreprise sur demande et en mode présentiel comme en distanciel.

PROGRAMME DE FORMATION :

Découvrir de DevOps

- Définir devOps et principes fondamentaux
- Planifier la transformation organisationnelle
- Découvrir la plateforme Azure DevOps et GitHub
- Connaître les principes fondamentaux du contrôle de code source et git
- Connaître les stratégies de gestion des licences
- Optimiser le temps de cycle et apprentissage validé

Planifier Agile avec GitHub Projects et Azure Boards

- Décrire les projets GitHub et Azure Boards
- Lier Azure Boards et GitHub
- Configurer et gérer des projets et des tableaux GitHub
- Personnaliser les vues de projet

Concevoir et implémenter des stratégies de branche et des flux de travail

- Décrire les flux de travail de branchement Git
- Implémenter des branches de fonctionnalité
- Implémenter gitHub Flow
- Dupliquer un référentiel

Collaborer avec des demandes de tirage (pull requests) dans Azure Repos

- Mettre en œuvre des flux de travail de pull request de niveau entreprise pour une collaboration évolutive.
- Concevoir des processus de révision de code stratégiques qui améliorent la qualité et le transfert des connaissances
- Configurer des stratégies de branche avancées et des contrôles de gouvernance automatisés
- Orchestrer des modèles de collaboration entre équipes à l'aide des fonctionnalités d'intégration d'Azure Repos
- Établir des pistes d'audit et des infrastructures de conformité par le biais de la gouvernance des demandes de tirage

Explorer les hooks Git

- Créer des hooks Git qui vérifient automatiquement la qualité du code pour votre équipe
- Créer des hooks qui s'exécutent avant et après des validations pour vérifier votre code
- Créer des hooks de sécurité qui empêchent l'enregistrement des mots de passe et des secrets
- Connecter des hooks Git avec Azure DevOps et générer des pipelines
- Configurer des hooks Git pour que toute l'équipe utilise

Planifier la promotion de la source interne

- Utiliser Git pour favoriser le développement interne de sources au sein de votre organisation.
- Implémenter le workflow de duplication (fork) pour le développement collaboratif.
- Choisir entre les branches et les fourches pour différents scénarios.
- Partager efficacement du code entre les duplications.

Gérer et configurer des référentiels

- Travailler avec de grands dépôts Git.
- Utiliser VFS pour Git pour améliorer les performances.

- Utiliser le stockage de fichiers volumineux Git (LFS) pour les fichiers volumineux.
- Nettoyer les données du référentiel.
- Gérer et automatiser les notes de publication avec GitHub.

Identifier la dette technique

- Trouver et gérer la dette technique dans votre code.
- Utiliser des outils de qualité du code pour améliorer vos logiciels.
- Planifier des révisions de code efficaces pour votre équipe.
- Comprendre la complexité et les métriques de qualité.

Explorer Azure Pipelines

- Découvrir ce qu'est Azure Pipelines et comment il aide votre équipe.
- Expliquer les composants clés et comment ils fonctionnent ensemble.
- Choisir l'approche d'automatisation appropriée pour vos projets.
- Utiliser la terminologie Azure Pipelines essentielle avec confiance.

Gérer les agents et pools Azure Pipeline

- Choisir entre les agents hébergés par Microsoft et auto-hébergés.
- Installer et configurez les agents Azure Pipelines.
- Configurer des pools d'agents.
- Sécuriser les agents et les pools.
- Explorer la communication à déployer à l'aide d'Azure Pipelines.

Décrire les pipelines et la concurrence

- Utiliser et estimer les travaux parallèles.
- Utiliser Azure Pipelines pour des projets open source ou privés.
- Utiliser le Concepteur visuel.
- Collaborer avec Azure Pipelines et YAML.

Concevoir et implémenter une stratégie de pipeline

- Concevoir et implémenter des stratégies de construction efficaces
- Configurer et gérer les demandes de l'agent
- Implémenter des builds multiconfiguration et multi-agent
- Intégrer des référentiels GitHub à Azure Pipelines
- Concevoir des stratégies de test complètes
- Implémenter la couverture du code dans les flux de travail de pipeline
- Utiliser différents types de contrôle de code source disponibles dans Azure Pipelines

Intégrer à Azure Pipelines

- Comprendre l'anatomie et la structure avancées d'Azure Pipelines
- Créer et utiliser des modèles de pipeline pour la réutilisation du code
- Utiliser des ressources et des dépendances YAML
- Implémenter des flux de travail multi-référentiels dans vos pipelines
- Appliquer des modèles d'intégration de pipeline modernes et des bonnes pratiques

Découvrir l'intégration continue avec GitHub Actions

- Concevoir et implémenter des workflows d'intégration continue avec GitHub Actions
- Configurer des variables d'environnement et gérer les données de flux de travail en toute sécurité

- Partager des artefacts entre des travaux et automatiser la gestion des mises en production avec des balises Git
- Créer et gérer des secrets chiffrés pour les opérations CI/CD sécurisées
- Appliquer les meilleures pratiques du secteur pour les pipelines d'automatisation gérables et évolutifs

Concevoir une stratégie de construction de conteneur

- Concevoir des stratégies de construction de conteneurs efficaces
- Utiliser des conteneurs Docker et des fichiers Docker
- Implémenter des builds multi-phases et des meilleures pratiques
- Déployer des conteneurs sur des services Azure

Créer un pipeline de mise en production

- Expliquer la terminologie et les concepts de gestion des versions d'Azure Pipelines.
- Décrire les tâches de génération et de mise en production et leurs fonctionnalités de déploiement.
- Implémenter des travaux de mise en production et configurer des flux de travail de déploiement.

Explorer les recommandations de stratégie de mise en production

- Analyser les points clés à prendre en compte pour concevoir des stratégies de mise en production efficaces.
- Définir les composants du pipeline de déploiement et configurer les sources d'artefact.
- Créer des flux de travail d'approbation de mise en production complets.
- Implémenter et configurer des portes de mise en production pour l'assurance qualité.

Configurer et approvisionner des environnements

- Provisionner et configurer des environnements cibles pour les flux de travail de déploiement.
- Déployer en toute sécurité sur des environnements à l'aide de connexions de service.
- Configurer l'automatisation des tests fonctionnels et implémenter des tests de disponibilité.
- Configurer une infrastructure de test complète pour l'assurance qualité.

Gérer et modulariser des tâches et des modèles

- Créer et gérer des groupes de tâches pour les composants de pipeline réutilisables.
- Implémenter des groupes de variables pour la gestion centralisée de la configuration.
- Configurer les variables de mise en production et les variables d'étape pour les déploiements spécifiques à l'environnement.
- Développer des tâches de build et de mise en production personnalisées pour les exigences de pipeline spécialisées.

Automatiser l'inspection de la santé

- Mettre en place des systèmes de monitoring automatisés pour la surveillance continue et la détection proactive des problèmes.
- Créer et configurer des systèmes de notification pilotés par les événements pour les commentaires de pipeline en temps réel.
- Configurer des infrastructures de notification complètes sur les plateformes Azure DevOps et GitHub.
- Développer des hooks de service pour la supervision automatisée de pipelines et les flux de travail d'intégration.
- Faire la distinction entre les artefacts de mise en production et les processus de mise en production tout en implémentant des mécanismes de contrôle de qualité.

- Évaluer et sélectionner les outils de gestion des mises en production appropriés pour les exigences organisationnelles.

Découvrir des modèles de déploiement

- Analyser les architectures de modèle de déploiement et les stratégies d'implémentation
- Évaluer les principes d'architecture des microservices et la conception du système distribué
- Comparer les modèles de déploiement classiques et modernes pour des stratégies de mise en production optimales
- Concevoir des architectures de déploiement évolutives alignées sur les exigences organisationnelles

Implémenter un déploiement bleu-vert et des bascules de fonctionnalité

- Analyser des stratégies de déploiement avancées et des modèles de livraison progressifs
- Implémenter des architectures de déploiement bleu-vert pour les versions sans temps d'arrêt
- Configurer et gérer les emplacements de déploiement Azure pour l'isolation de l'environnement
- Concevoir et implémenter des systèmes de bascule de fonctionnalités pour le contrôle dynamique des fonctionnalités

Implémenter des mises en production avec contrôle de validité et un dark launching

- Analyser les stratégies de déploiement progressif et les modèles de déploiement incrémentiel
- Implémenter des architectures de mise en production du contrôle de validité avec une migration progressive du trafic
- Configurer Azure Traffic Manager pour la distribution intelligente du trafic
- Concevoir des stratégies de lancement sombres pour la validation des fonctionnalités masquées

Implémenter des tests A/B et un déploiement d'exposition progressive

- Concevoir des stratégies de déploiement progressif avec une cohorte d'utilisateurs ciblés par phases
- Implémenter des frameworks de test A/B pour la validation des fonctionnalités pilotées par les données
- Configurer des architectures de déploiement en anneau pour les déploiements CI/CD contrôlés
- Analyser les compromis de stratégie de déploiement et sélectionner des modèles optimaux

Intégrer à des systèmes de gestion des identités

- Intégrer Azure DevOps à des architectures de gestion des identités d'ID Microsoft Entra
- Implémenter l'authentification unique GitHub (SSO) et des stratégies d'authentification multifacteur
- Concevoir et provisionner des principaux de service pour les scénarios d'identité de charge de travail
- Configurer des identités managées pour l'authentification des ressources Azure sans mot de passe

Gérer les données de configuration d'application

- Concevoir des architectures de configuration externalisées implémentant la séparation des principes des préoccupations
- Implémenter Azure App Configuration avec des stratégies de gestion et d'étiquetage de paires clé-valeur
- Intégrer Azure Key Vault à Azure Pipelines pour une gestion sécurisée des secrets
- Configurer des indicateurs de fonctionnalité et une configuration dynamique pour les déploiements de fonctionnalités progressifs

- Gérer les secrets, les jetons et les certificats en suivant les meilleures pratiques de sécurité DevOps
- Implémenter des modèles de magasin de configuration externe pour les applications distribuées

Explorer l'infrastructure en tant que gestion du code et de la configuration

- Implémenter des stratégies d'automatisation du déploiement d'environnement à l'aide de l'infrastructure en tant que code
- Concevoir des approches de gestion de la configuration de l'environnement avec contrôle de version
- Différencier les méthodologies de configuration impératives et déclaratives
- Appliquer des principes de configuration idempotents garantissant des opérations d'infrastructure cohérentes

Créer des ressources Azure à l'aide de modèles Azure Resource Manager

- Créer et déployer des ressources Azure à l'aide de modèles ARM et de la syntaxe Bicep.
- Comprendre les composants de modèle ARM, notamment les paramètres, les variables, les fonctions, les ressources et les sorties.
- Gérer les dépendances de ressources à l'aide de l'élément dependsOn et des fonctions de référence.
- Organiser et modulariser des modèles à l'aide de modèles liés et imbriqués pour une réutilisation.
- Gérer en toute sécurité les secrets et les valeurs sensibles via l'intégration d'Azure Key Vault.
- Choisir les modes de déploiement appropriés (incrémentiel et terminé) pour différents scénarios.

Créer des ressources Azure à l'aide d'Azure CLI

- Installer Azure CLI : configurer Azure CLI sur des systèmes d'exploitation Linux, macOS et Windows.
- S'authentifier auprès d'Azure : connectez-vous à Azure à l'aide de différentes méthodes, notamment la connexion interactive, les identités managées et les principaux de service.
- Comprendre la syntaxe CLI : Découvrez la structure de commandes Azure CLI, notamment les groupes de référence, les commandes, les paramètres et les valeurs.
- Créer des ressources de manière interactive : exécutez des commandes Azure CLI sur le terminal pour créer des groupes de ressources et des comptes de stockage.
- Automatiser avec des scripts : écrire et exécuter des scripts Bash qui tirent parti d'Azure CLI pour la gestion automatisée des ressources.
- Résoudre efficacement les problèmes : utilisez le paramètre --debug et comprenez les différences de langage de script pour résoudre les problèmes courants.

Explorer Azure Automation avec DevOps

- Créer et configurer des comptes Azure Automation pour gérer les ressources cloud.
- Implémenter et gérer des runbooks à l'aide de PowerShell, de Flux de travail PowerShell, de Python et d'éditeurs graphiques.
- Configurer des webhooks pour déclencher des runbooks à partir de systèmes et de services externes.
- Intégrer Azure Automation à GitHub et Azure DevOps pour le contrôle de version et CI/CD.
- Créer et exécuter des flux de travail PowerShell avec des points de contrôle et un traitement parallèle.
- Planifier et implémenter la gestion hybride à l'aide d'Azure Automation dans les ressources cloud et locales.

Implémenter la configuration de l'état souhaité (DSC)

- Comprendre et gérer la dérive de configuration dans votre infrastructure.
- Implémenter la configuration d'état souhaité (DSC) avec PowerShell.
- Décrire Azure Automation State Configuration et ses composants.
- Créer et gérer des fichiers de configuration DSC.
- Implémenter l'automatisation DSC et Linux sur Azure.
- Planifier la gestion hybride avec DSC.

Implémenter Bicep

- Comprendre ce que Bicep est et ses avantages pour l'infrastructure en tant que code.
- Installer Bicep et configurez une expérience de création fluide avec Visual Studio Code.
- Créer des modèles Bicep pour définir des ressources Azure.
- Comprendre la structure, la syntaxe, les paramètres, les variables et les modules de fichier Bicep.
- Déployer des fichiers Bicep à l'aide d'Azure CLI et de Cloud Shell.
- Déployer des modèles Bicep à partir d'Azure Pipelines pour un déploiement continu.
- Déployer des modèles Bicep à partir de flux de travail GitHub pour les déploiements automatisés.
- Découvrir comment Bicep s'intègre aux modèles Azure Resource Manager.

Découvrir de Secure DevOps

- Identifier et comprendre les attaques par injection SQL et leur impact sur la sécurité des applications.
- Expliquer les principes DevSecOps et comment la sécurité s'intègre tout au long du cycle de vie du développement.
- Implémenter la validation de sécurité à des points clés dans votre pipeline DevOps.
- Effectuer une modélisation des menaces pour identifier et hiérarchiser les risques de sécurité.
- Utiliser GitHub CodeQL pour l'analyse automatisée de la sécurité et la détection des vulnérabilités.

Implémenter un logiciel open source

- Découvrir comment les logiciels modernes sont créés à l'aide de composants open source et des avantages qu'ils offrent.
- Expliquer les préoccupations de l'entreprise concernant les logiciels open source, notamment les vulnérabilités de sécurité, la conformité des licences et les risques liés à la chaîne logistique.
- Décrire les licences open source courantes, notamment MIT, Apache, PG, BSD et leurs principales caractéristiques.
- Évaluer les implications de licence pour le développement de logiciels commerciaux et comprendre la compatibilité des licences.
- Implémenter des stratégies de gestion des composants open source, notamment la gestion des stocks, l'analyse des vulnérabilités et la conformité des licences.

Analyser de la composition logicielle

- Comprendre l'analyse de composition logicielle (SCA) et pourquoi il est essentiel de gérer les dépendances open source en toute sécurité.
- Inspecter et valider les bases de code pour la conformité des licences et les vulnérabilités de sécurité à l'aide d'outils automatisés.
- Implémenter GitHub Dependabot pour détecter automatiquement les dépendances vulnérables et créer des demandes de tirage pour les mises à jour de sécurité.
- Intégrer des vérifications d'analyse de composition logicielle dans Azure Pipelines pour analyser les dépendances pendant les processus de génération et de déploiement.

- Examiner et configurer les outils SCA, notamment Mend (WhiteSource), Snyk, OWASP Dependency-Check et les sources en amont d'Azure Artifacts.
- Automatiser l'analyse des images conteneur pour détecter les vulnérabilités dans les images de base et les dépendances d'application.
- Interpréter les alertes des outils d'analyse et hiérarchiser la correction en fonction de la gravité, de l'exploitabilité et de l'impact métier.

Surveiller et gouverner de la sécurité

- Implémenter les meilleures pratiques de sécurité du pipeline et sécurisez les flux de travail DevOps.
- Configurer Microsoft Defender pour cloud pour la protection contre les menaces et la surveillance de la conformité.
- Créer et gérer des stratégies Azure pour la gouvernance et l'application de la conformité.
- Comprendre les initiatives de stratégie, les verrous de ressources et les infrastructures de gouvernance.
- Déployer la détection des menaces Microsoft Defender pour Identity.
- Intégrer GitHub Advanced Security à Microsoft Defender pour Cloud.
- Configurer les fonctionnalités de Sécurité avancée GitHub, notamment l'analyse du code, l'analyse des secrets et l'analyse des dépendances.

Explorer les dépendances de package

- Définir la stratégie de gestion des dépendances : comprenez les principes et les avantages de la gestion des dépendances dans le développement de logiciels modernes.
- Identifier les dépendances : reconnaître et classer les dépendances dans vos projets et solutions.
- Décrire les éléments et la composantisation de la gestion des dépendances : découvrez comment décomposer les systèmes en composants gérables.
- Analyser votre base de code pour les dépendances : utilisez des outils et techniques pour découvrir et analyser les dépendances dans votre code source.

Comprendre la gestion des packages

- Comprendre les concepts des paquets et de la gestion des paquets
- Implémenter et gérer des flux de paquets avec des sources publiques et privées
- Explorer les gestionnaires de flux de package et les options d'hébergement
- Consommer des packages à partir de différentes sources dans vos projets
- Créer et publier des packages sur Azure Artifacts et d'autres flux

Migrer, consolider et sécuriser des artefacts

- Identifier les référentiels d'artefacts existants au sein de votre organisation et évaluer les besoins de migration
- Migrer et intégrer des référentiels d'artefacts dans Azure Artifacts à l'aide de stratégies appropriées
- Sécuriser les flux de package avec des paramètres de contrôle d'accès et de visibilité appropriés
- Implémenter des rôles pour contrôler l'accès au flux à différents niveaux (Lecteur, Collaborateur, Contributeur, Propriétaire)
- Configurer des autorisations granulaires pour les utilisateurs, les équipes et les principaux de service
- Configurer des mécanismes d'authentification, notamment des jetons d'accès personnels et des fournisseurs d'informations d'identification

Implémenter une stratégie de contrôle de version

- Comprendre l'importance du contrôle de version pour la gestion des packages et le suivi des dépendances
- Implémenter le contrôle de version sémantique (SemVer 2.0) pour la numérotation de version cohérente
- Utiliser les vues Azure Artifacts (@Local, @Prerelease, @Release)) pour gérer la qualité des packages
- Promouvoir des packages par le biais de différentes perspectives en fonction des niveaux de qualité
- Envoyer (push) des packages à partir de pipelines de build avec les numéros de version appropriés
- Appliquer les meilleures pratiques pour les stratégies de contrôle de version dans les flux de travail DevOps

Découvrir des packages GitHub

- Publier des packages sur GitHub Packages à partir de référentiels et de flux de travail CI/CD
- Installer des packages à partir de packages GitHub dans des environnements de développement et de production
- Supprimer et restaurer des packages à l'aide de l'interface web et de l'API GitHub
- Configurer le contrôle d'accès et la visibilité pour les packages publics, privés et internes
- S'authentifier auprès de packages GitHub à l'aide de jetons d'accès personnels (PAT)
- Comprendre les registres de packages pris en charge (npm, Docker, NuGet, Maven, RubyGems)

Implémenter des outils pour suivre l'utilisation et le flux

- Implémenter des outils pour suivre les données de commentaires et de télémétrie à partir d'applications
- Planifier des stratégies de surveillance continue dans les environnements de développement et de production
- Implémenter Application Insights pour une surveillance complète des performances des applications
- Utiliser le langage de requête Kusto (KQL) pour interroger et analyser des données de télémétrie
- Concevoir et implémenter des métriques et des requêtes pour des insights actionnables
- Surveiller les performances des applications à l'aide d'Azure Monitor et de Log Analytics
- Comprendre les flux de travail de développement de boucle interne et de boucle externe

Développer des tableaux de bord de surveillance et d'état

- Configurer une surveillance complète dans GitHub avec des badges d'état, des insights et des analyses Actions
- Créer et personnaliser des tableaux de bord Azure avec des visualisations épinglées à partir de plusieurs services Azure
- Développer des classeurs Azure Monitor interactifs avec des paramètres, des requêtes et des visualisations personnalisées
- Implémenter des tableaux de bord Power BI pour l'analytique DevOps avancée et l'aide à la décision
- Créer des applications de supervision personnalisées à l'aide des API REST Azure DevOps et des kits SDK Application Insights
- Surveiller les métriques de santé du pipeline, notamment les taux d'échec, la durée d'exécution et la détection de tests instables

- Optimiser les pipelines pour les coûts, le temps, les performances et la fiabilité à l'aide de stratégies pilotées par les données
- Implémenter l'optimisation de la concurrence du pipeline pour équilibrer les performances du débit avec les coûts d'infrastructure
- Concevoir des stratégies de visualisation efficaces qui communiquent l'intégrité du système à diverses parties prenantes
- Établir des alertes et des systèmes de notification en fonction des métriques de tableau de bord et des violations de seuil

Partager des connaissances au sein des équipes

- Implémenter des stratégies de partage des connaissances pour préserver l'apprentissage organisationnel et réduire les erreurs répétées
- Créer et gérer des wikis de projet Azure DevOps avec des autorisations et un contrôle d'accès appropriés
- Écrire une documentation technique à l'aide de la syntaxe Markdown et GitHub Flavored Markdown (GFM)
- Créer des diagrammes dans la documentation à l'aide de la syntaxe Mermaid pour les organigrammes, les diagrammes de séquence et les graphiques
- Intégrer GitHub à Microsoft Teams pour les notifications, les discussions sur les pull requests et la gestion des tickets.
- Intégrer Azure DevOps Boards, Repos et Pipelines à Microsoft Teams pour une collaboration unifiée
- Configurer des abonnements, des filtres et des notifications pour les éléments de travail, les pull requests et les événements de pipeline
- Établir des pratiques de documentation qui prennent en charge l'intégration d'équipe, la communication standard et la rétention des connaissances

Processus de conception pour automatiser l'analytique des applications

- Concevoir des processus d'analytique des applications automatisées à l'aide d'Azure Application Insights.
- Implémenter des stratégies de recherche augmentée pour une réponse rapide aux incidents et une analyse de la cause racine.
- Intégrer la collecte de données de télémétrie entre les applications et l'infrastructure.
- Sélectionner et configurer des outils de surveillance avec des fonctionnalités avancées pour la surveillance continue des applications.
- Connecter les outils ITSM aux services de supervision Azure pour simplifier la gestion des incidents.

Gérer les alertes, les rétrospectives sans blâme et une culture juste

- Configurer la détection intelligente Azure Application Insights et interpréter les notifications de dégradation des performances.
- Diagnostiquer et améliorer les performances des applications à l'aide des données de télémétrie et de profileur.
- Réduire la fatigue des alertes en distinguant les alertes actionnables du bruit sans signification.
- Mener des rétrospectives sans blâme qui se concentrent sur les améliorations du système plutôt que sur la responsabilité individuelle.
- Construire une culture juste où les ingénieurs apprennent à partir d'erreurs sans crainte de punition.

PRÉREQUIS :

Pour suivre cette formation, vous devez avoir suivi le cours « AZ-900 Azure Fundamentals » et, en fonction de votre projet ou votre rôle, le cours « AZ-104 : Azure Administrator » ou « AZ-204 : Developing Solutions for Microsoft Azure » pour suivre ce cours ou avoir un niveau d'expérience Azure équivalent.

Un niveau d'anglais B1 est recommandé, retrouvez les niveaux sur ce lien : [Classification des niveaux de langue](#)

PRE-CERTIFICATION :

Cette formation ouvre la voie à la certification Microsoft « AZ-400 Microsoft Azure DevOps Solutions ».

DUREE : 4 jours (28 heures)

INTERLOCUTEURS : Administrateurs, développeurs, Ingénieurs DevOps

NIVEAU : Expert