

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

(Préparation certification Microsoft AZ-400)

Mise à jour : 05/06/2023

## 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 :

- Choisir le projet approprié
- Décrire les structures d'équipe
- Choisir les outils DevOps
- Planifier Agile avec les projets GitHub et Azure Boards
- Décrire les types de systèmes de contrôle de code source
- Utiliser Azure Repos et GitHub
- Structurer votre dépôt Git
- Gérer les branches et les workflows Git
- Collaborer avec des demandes de tirage (pull requests) dans Azure Repos
- Identifier la dette technique
- Explorer les hooks Git
- Planifier la promotion de l'inner source
- Gérer les référentiels Git
- Explorer Azure Pipelines
- Gérer les pools et les agents Azure Pipelines
- Décrire les pipelines et la concurrence
- Explorer l'intégration continue
- Implémenter une stratégie de pipeline
- Intégrer à Azure Pipelines
- Découvrir l'intégration continue avec GitHub Actions
- Concevoir une stratégie de build de conteneur
- Créer un pipeline de mise en production
- Explorer les recommandations de mise en production
- Provisionner et tester des environnements
- Gérer et moduler des tâches et des modèles
- Automatiser l'inspection de l'intégrité

- Introduire les modèles de déploiement
- Implémenter un déploiement bleu-vert et des bascules de fonctionnalité
- Implémenter des mises en production avec contrôle de validité et un dark launching
- Implémenter un test A/B et un déploiement avec exposition progressive
- Intégrer aux systèmes d'administration des identités
- Gérer les données de configuration d'application
- Explorer l'infrastructure en tant que code et la gestion de la configuration
- Créer des ressources Azure à partir de modèles Azure Resource Manager
- Créer des ressources Azure à l'aide d'Azure CLI
- Explorer Azure Automation avec DevOps
- Implémenter DSC (Desired State Configuration)
- Implémenter Bicep
- Implémenter un logiciel open source
- Analyser de composition logicielle
- Analyser via les Analyseurs statiques
- Utiliser OWASP et les analyseurs dynamiques
- Monitorer la sécurité et la gouvernance
- Explorer les dépendances de package
- Comprendre la gestion des packages
- Migrer des artefacts de consolidation et sécurisés
- Implémenter une stratégie de contrôle de version
- Découvrir les packages GitHub
- Implémenter des outils pour suivre l'utilisation et le flux
- Développer des tableaux de bord de supervision et d'état
- Partager des connaissances au sein des équipes
- Concevoir des processus pour automatiser l'analyse des applications
- Gérer les alertes, les rétrospectives irréprochables et une culture juste

## 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 :

### Choisir le projet approprié

- Comprendre les différents projets et systèmes pour guider la transition
- Sélectionner un projet pour démarrer la transformation DevOps
- Identifier les groupes pour réduire la résistance initiale
- Identifier les métriques du projet et les indicateurs de performance clés (KPI)

### Décrire les structures d'équipe

- Comprendre les pratiques agiles et les principes du développement agile
- Créer une équipe et une structure organisationnelle agile
- Identifier les membres idéaux de l'équipe DevOps
- Sélectionner et configurer des outils de collaboration

### Choisir les outils DevOps

- Concevoir une stratégie d'intégration d'outils
- Concevoir une stratégie de gestion des licences (exemple, utilisateurs Azure DevOps et GitHub)
- Concevoir une stratégie de traçabilité de bout en bout à partir des éléments de travail au logiciel en cours de travail
- Concevoir une stratégie d'authentification et d'accès
- Concevoir une stratégie pour l'intégration des ressources locales et Cloud

### Planifier Agile avec les projets GitHub et Azure Boards

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

### Décrire les types de systèmes de contrôle de code source

- Appliquer les pratiques de contrôle de code source dans votre processus de développement
- Expliquer les différences entre le contrôle de version centralisé et distribué
- Comprendre Git et TFVC
- Développer à l'aide de Git

### Utiliser Azure Repos et GitHub

- Décrire Azure Repos et GitHub
- Migrer de TFVC vers Git
- Utiliser GitHub Codespaces

### Structurer votre dépôt Git

- Comprendre et structurer des dépôts Git
- Implémenter des monorepos ou des dépôts multiples
- Implémenter un journal des modifications

### Gérer les branches et les workflows Git

- Décrire les workflows de branchement Git
- Implémenter des branches de fonctionnalité
- Implémenter le flux GitHub

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

- Tirer profit des demandes de tirage à des fins de collaboration et de révision du code
- Envoyer des commentaires à l'aide de demandes de tirage
- Configurer des stratégies de branche
- Utiliser GitHub mobile pour les approbations de demandes de tirage

#### **Identifier la dette technique**

- Identifier et gérer la dette technique
- Intégrer les outils de qualité du code
- Planifier les révisions du code
- Décrire la complexité et les métriques de qualité

#### **Explorer les hooks Git**

- Comprendre ce que sont les hooks Git et identifier quand les utiliser
- Implémenter des hooks Git pour l'automatisation
- Expliquer le comportement des hooks Git

#### **Planifier la promotion de l'inner source**

- Utiliser Git pour promouvoir l'inner source dans l'organisation
- Implémenter un workflow de duplication
- Choisir entre des branches et des fourches
- Partager du code entre des fourches

#### **Gérer les référentiels Git**

- Comprendre les grands dépôts Git
- Expliquer VFS pour Git
- Utiliser le stockage de grands fichiers Git (LFS)
- Vider les données des dépôts
- Gérer et automatiser les notes de publication avec GitHub

#### **Explorer Azure Pipelines**

- Décrire Azure Pipelines, expliquer son rôle et celui de ses composants
- Déterminer la responsabilité de l'automatisation du pipeline
- Comprendre les termes clés d'Azure Pipelines

#### **Gérer les pools et les agents Azure Pipelines**

- Choisir entre des agents hébergés par Microsoft et autohébergés
- Installer et configurer des agents Azure Pipelines
- Configurer et sécuriser des pools d'agents
- Explorer la communication en vue d'un déploiement avec Azure Pipelines

#### **Décrire les pipelines et la concurrence**

- Utiliser et estimer les travaux parallèles
- Utiliser Azure Pipelines pour les projets open source ou privés
- Utiliser le concepteur visuel
- Utiliser Azure Pipelines et YAML

#### **Explorer l'intégration continue**

- Implémenter l'intégration continue à l'aide d'Azure Pipelines.

- Expliquer l'importance et les avantages de l'intégration continue.
- Décrire les propriétés de build.

#### **Implémenter une stratégie de pipeline**

- Définir une stratégie de build
- Expliquer et configurer les demandes
- Implémenter des builds avec plusieurs agents
- Utiliser différents types de contrôle de code source disponibles dans Azure Pipelines

#### **Intégrer à Azure Pipelines**

- Décrire l'anatomie et la structure avancées d'Azure Pipelines
- Détailler les ressources YAML et les modèles
- Implémenter et utiliser plusieurs dépôts

#### **Découvrir l'intégration continue avec GitHub Actions**

- Implémenter l'intégration continue avec GitHub Actions
- Utiliser des variables d'environnement
- Partager des artefacts entre travaux et utiliser des étiquettes Git
- Créer et gérer des secrets

#### **Concevoir une stratégie de build de conteneur**

- Concevoir une stratégie de conteneur
- Utiliser des conteneurs Docker
- Créer un Azure Container Registry
- Expliquer les microservices et les conteneurs Docker

#### **Créer un pipeline de mise en production**

- Expliquer la terminologie utilisée dans Azure DevOps et d'autres outils de gestion des mises en production
- Décrire une tâche de build et de mise en production, ce qu'elles peuvent faire et donner des exemples de tâche de déploiement disponibles
- Implémenter des travaux de mise en production

#### **Explorer les recommandations de mise en production**

- Expliquer les éléments à prendre en compte lors de la conception d'une stratégie de mise en production
- Définir les composants d'un pipeline de mise en production et utiliser des sources d'artefact
- Créer un plan d'approbation de mise en production
- Implémenter des portes de mise en production

#### **Provisionner et tester des environnements**

- Approvisionner et configurer un environnement cible
- Déployer dans un environnement en toute sécurité à l'aide d'une connexion de service
- Configurer l'automatisation des tests fonctionnels et exécuter des tests de disponibilité
- Configurer l'infrastructure de test

#### **Gérer et moduler des tâches et des modèles**

- Utiliser et gérer des groupes de tâches et de variables

- Utiliser des variables de mise en production et des variables de phase dans votre pipeline de mise en production
- Utiliser des variables dans les pipelines de mise en production

#### **Automatiser l'inspection de l'intégrité**

- Implémenter l'inspection automatisée de l'intégrité
- Créer et configurer des événements
- Configurer des notifications dans Azure DevOps et GitHub
- Créer des crochets de service pour surveiller le pipeline
- Classifier une mise en production par rapport à un processus de mise en production et esquisser la façon de contrôler la qualité des deux
- Choisir un outil de gestion des mises en production

#### **Introduire les modèles de déploiement**

- Décrire les modèles de déploiement classiques et modernes
- Expliquer l'architecture des microservices
- Planifier et concevoir votre architecture

#### **Implémenter un déploiement bleu-vert et des bascules de fonctionnalité**

- Expliquer les stratégies de déploiement
- Implémenter un déploiement bleu-vert
- Comprendre les emplacements de déploiement
- Implémenter et gérer des bascules de fonctionnalité

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

- Décrire les stratégies de déploiement
- Implémenter une mise en production avec contrôle de validité
- Expliquer un gestionnaire de trafic
- Comprendre le dark launching

#### **Implémenter un test A/B et un déploiement avec exposition progressive**

- Implémenter un déploiement avec exposition progressive
- Implémenter un test A/B
- Implémenter la CI/CD avec des boucles de déploiement
- Identifier la meilleure stratégie de déploiement

#### **Intégrer aux systèmes d'administration des identités**

- Intégrer Azure DevOps aux systèmes d'administration des identités
- Intégrer GitHub à l'authentification unique (SSO)
- Comprendre et créer un principal de service
- Créer des identités de service managé

#### **Gérer les données de configuration d'application**

- Repenser les données de configuration d'application
- Comprendre la séparation des préoccupations
- Intégrer Azure Key Vault à Azure Pipelines
- Gérer les secrets, les jetons et les certificats
- Décrire Azure App Configuration
- Comprendre les paires clé-valeur
- Comprendre la gestion des fonctionnalités de configuration d'application

- Implémenter la configuration d'application

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

- Comprendre comment déployer votre environnement.
- Planifier la configuration de votre environnement.
- Choisir entre une configuration impérative et une configuration déclarative.
- Expliquer ce qu'est la configuration idempotent

#### **Créer des ressources Azure à partir de modèles Azure Resource Manager**

- Créer des ressources Azure à partir de modèles Azure Resource Manager
- Gérer les dépendances et les secrets dans les modèles
- Organiser et modulariser des modèles

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

- Créer des ressources Azure à l'aide d'Azure CLI
- Utiliser Azure CLI
- Exécuter des modèles en utilisant Azure CLI

#### **Explorer Azure Automation avec DevOps**

- Implémenter l'automatisation avec Azure DevOps
- Créer et gérer des runbooks
- Créer des webhooks
- Créer et exécuter d'un runbook de workflows et de workflows PowerShell

#### **Implémenter DSC (Desired State Configuration)**

- Implémenter DSC (Desired State Configuration)
- Décrire Azure Automation State Configuration
- Implémenter l'automatisation Linux et DSC sur Azure
- Planifier la gestion hybride

#### **Implémenter Bicep**

- Découvrir et installer Bicep, créer une expérience de création fluide
- Utiliser Bicep pour déployer des ressources sur Azure
- Déployer des fichiers Bicep dans Cloud Shell et Visual Studio Code

#### **Implémenter un logiciel open source**

- Implémenter un logiciel open source
- Expliquer les préoccupations de l'entreprise pour les composants Open source
- Décrire les licences Open source
- Comprendre les implications et les évaluations de la licence

#### **Analyser de composition logicielle**

- Inspecter et valider la conformité des bases de code
- Intégrer des outils de sécurité comme WhiteSource avec Azure DevOps
- Implémenter de la validation de sécurité du pipeline
- Interpréter des alertes des outils d'analyse
- Configurer les alertes et la sécurité de GitHub Dependabot

#### **Analyser via les Analyseurs statiques**

- Utiliser SonarCloud
- Utiliser de CodeQL dans GitHub
- Interpréter des alertes des outils d'analyse

#### **Utiliser OWASP et les analyseurs dynamiques**

- Implémenter des pratiques de codage de sécurité OWASP
- Comprendre la conformité pour les bases de code

#### **Monitorer la sécurité et la gouvernance**

- Configurer Microsoft Defender pour le cloud
- Comprendre les stratégies Azure
- Décrire les initiatives, les verrous de ressources et les modèles Azure
- Travailler avec Microsoft Defender pour Identity

#### **Explorer les dépendances de package**

- Définir une stratégie de gestion des dépendances
- Identifier les dépendances
- Décrire les éléments et la composantisation de la gestion des dépendances
- Analyser votre codebase à la recherche de dépendances

#### **Comprendre la gestion des packages**

- Implémenter la gestion des packages
- Gérer le flux de package
- Utiliser et créer des packages
- Publier des packages

#### **Migrer des artefacts de consolidation et sécurisés**

- Identifier les dépôts d'artefacts
- Migrer et intégrer des dépôts d'artefacts
- Sécuriser les flux de packages
- Comprendre les rôles, les autorisations et l'authentification

#### **Implémenter une stratégie de contrôle de version**

- Implémenter une stratégie de contrôle de version
- Promouvoir des packages
- Envoyer des packages à partir du pipeline
- Décrire la sémantique et explorer les meilleures pratiques pour le contrôle de version

#### **Découvrir les packages GitHub**

- Installer et publier des packages
- Supprimer et restaurer des packages
- Configurer le contrôle d'accès et la visibilité

#### **Implémenter des outils pour suivre l'utilisation et le flux**

- Implémenter des outils pour suivre les commentaires
- Planifier une surveillance continue
- Implémenter Application Insights
- Utiliser Kusto Query Language (KQL)

### **Développer des tableaux de bord de supervision et d'état**

- Configurer des tableaux de bord Azure
- Utiliser le Concepteur de vues dans Azure Monitor
- Créer des classeurs Azure Monitor
- Superviser avec Power BI

### **Partager des connaissances au sein des équipes**

- Partager des connaissances avec des équipes de développement
- Utiliser des wikis Azure DevOps
- Intégrer à Azure Boards

### **Concevoir des processus pour automatiser l'analyse des applications**

- Automatiser l'analytique des applications
- Assister DevOps avec des réponses rapides et une recherche augmentée
- Intégrer la télémétrie
- Implémenter des outils et des technologies de supervision

### **Gérer les alertes, les rétrospectives irréprochables et une culture juste**

- Effectuer des rétrospectives irréprochables et créez une culture juste
- Améliorer les performances des applications
- Expliquer la dégradation du temps de réponse du serveur
- Réduire les alertes sans signification et sans action

## **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 requis, 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** : Intermédiaire