

Designing and implementing Microsoft DevOps Solutions

RS : 5343

La certification comprend la conception et la mise en œuvre de stratégies de collaboration, de code, d'infrastructure, de contrôle de code, de contrôle de code, de sécurité, de conformité, d'intégration continue, de test, de livraison, de surveillance et de rétroaction.

Les étudiants de ce cours sont intéressés par la conception et la mise en œuvre de processus DevOps ou par la réussite à l'examen de certification Microsoft Azure DevOps Solutions.

Programme

Développer une stratégie d'instrumentation

- Concevoir et implémenter la journalisation
- Évaluer et configurer une infrastructure de journalisation
- Concevoir une stratégie d'agrégation et de stockage des journaux (par exemple, stockage Azure)
- Concevoir une stratégie d'agrégation et de requête des journaux (par exemple, Azure Monitor, Splunk, Exabeam, LogRhythm)
- Interroger les journaux Log Analytics à l'aide de requêtes Kusto (KQL) de base
- Gérer le contrôle d'accès aux journaux (centrés sur l'espace de travail/centrés sur les ressources)
- Intégrer l'analyse des incidents (App Center Crashes, Crashlytics)

Concevoir et mettre en œuvre la conception de télémétrie

- Mettre en œuvre le traçage distribué
- Inspecter les indicateurs de performance des applications
- Inspecter les indicateurs de performance de l'infrastructure
- Définir et mesurer les métriques clés (CPU, mémoire, disque, réseau)
- Mettre en œuvre des alertes sur les métriques clés (e-mail, SMS, webhooks, Teams/Slack)
- Intégrer l'analyse des utilisateurs (par exemple, les entonnoirs Application Insights, Visual Studio App Center, TestFlight, Google Analytics)

Intégrer des solutions de journalisation et de surveillance

- Configurer et intégrer la surveillance des conteneurs (Azure Monitor, Prometheus, etc.)
- Configurer et intégrer avec des outils de surveillance (Azure Monitor Application Insights, Dynatrace, New Relic, Nagios, Zabbix)
- Créer une boucle de rétroaction à partir d'outils de surveillance de plateforme (par exemple, extension Azure Diagnostics, agent Log Analytics, Journaux de plateforme Azure, Event Grid)
- Gérer le contrôle d'accès à la plateforme de surveillance

Développer une stratégie d'ingénierie de fiabilité de site (SRE)

Développer une stratégie d'alerte exploitable

- Identifier et recommander des métriques sur lesquelles baser les alertes
- Mettre en œuvre des alertes à l'aide de métriques appropriées
- Mettre en œuvre des alertes basées sur des messages de journal appropriés
- Implémenter des alertes basées sur des contrôles de santé des applications
- Analyser des combinaisons de métriques
- Développer un mécanisme de communication pour informer les utilisateurs des systèmes dégradés
- Mettre en œuvre des alertes pour les activités d'auto-réparation (par exemple, la mise à l'échelle, les basculements)

Microsoft



LE PUBLIC VISÉ :

- Ingénieur DevOps
- Professionnels de l'informatique qui souhaitent devenir Azure DevOps Engineer Expert
- Professionnels de l'informatique se préparant à l'examen AZ-400 de Microsoft

DURÉE :

- 5 jours soit en total 35 heures

NIVEAU :

- Avancé

TARIF :

- 3100€/personne

ELIGIBLE CPF :

- OUI sous le N° RS343

Concevoir une stratégie de prédiction des défaillances

- Analyser le comportement du système en ce qui concerne les conditions de charge et de défaillance
- Calculer quand un système échouera dans diverses conditions
- Mesurer les métriques de base pour tirer parti du système
- Tirer parti des seuils dynamiques et de détection intelligente d'Application Insights dans Azure Monitor

Implémenter un bilan de santé

- Analyser les dépendances du système pour déterminer quelle dépendance doit être incluse dans la vérification de l'état
- calculer des délais d'attente de réponse sains basés sur SLO pour l'approche de conception de services •
- Concevoir une approche pour les situations de santé partielles
- Désigner une approche de conception pour la récupération fragmentaire (par exemple, pour améliorer les stratégies objectives de temps de récupération)
- Intégrer le bilan de santé à l'environnement de calcul
- Mettre en œuvre différents types de contrôles d'intégrité (vivacité du conteneur, démarrage, arrêt)

Élaborer un plan de sécurité et de conformité

Concevoir une stratégie d'authentification et d'autorisation

- Concevoir une solution d'accès (Azure AD Privileged Identity Management (PIM), Azure AD Conditional Access, MFA, Azure AD B2B, etc.)
- Implémenter service principal et managed identity
- Concevoir une solution d'accès aux applications à l'aide d'Azure AD B2C
- Configurer les connexions de service

Concevoir une stratégie de gestion des informations sensibles

- Évaluer et configurer la solution de coffre (Azure Key Vault, Hashicorp Vault)
- Gérer les certificats de sécurité
- Concevoir une stratégie de stockage et de récupération des secrets (secrets KeyVault, secrets GitHub, secrets Azure Pipelines)
- Formuler un plan de déploiement de fichiers secrets dans le cadre d'une version

Développer la sécurité et la conformité

- Automatiser l'analyse des dépendances pour la sécurité (analyse des conteneurs, OWASP)
- Automatiser l'analyse des dépendances pour la conformité (licences : MIT, GPL)
- Évaluer et signaler les risques
- Concevoir une solution de conformité du code source (par exemple, analyse du code GitHub, analyse secrète GitHub, analyses basées sur des pipelines, hooks Git, SonarQube, Dependabot, etc.)

Concevoir des mécanismes d'application de la gouvernance

- Implémenter des stratégies Azure pour appliquer les exigences organisationnelles
- Mettre en œuvre l'analyse des conteneurs (par exemple, analyse statique, logiciels malveillants, extraction de crypto)
- Concevoir et implémenter des tâches Azure Container Registry
- Concevoir une stratégie de rupture de verre pour répondre aux incidents de sécurité

Gérer le contrôle de code source

Développer une stratégie de contrôle de code source moderne

- Intégrer/migrer des systèmes de contrôle de code source disparates (par exemple, GitHub, Azure Repos)
- Désigner des stratégies d'authentification
- Désigner une approche de conception pour la gestion de fichiers binaires volumineux (par exemple, Git LFS)
- Désigner une approche de conception pour référentiel croisé le partage (par exemple, les sous-modules Git, les packages)
- mettre en œuvre une approche de conception de hooks de flux de travail pour des révisions de code efficaces (par exemple, affectations de révision de code GitHub, rappels de planification, Pull Analytics)

Planifier et mettre en œuvre des stratégies de branchement pour le code source

- Définir des directives Pull Requests (PR) pour appliquer la corrélation des éléments de travail
- Implémenter des restrictions de fusion de branche (par exemple, stratégies de branche, protections de branche, manuel)
- Définir la stratégie de branche (par exemple, basée sur la jonction, branche de fonctionnalité, branche de version, flux GitHub)
- Implémenter un flux de travail PR (révisions de code, approbations)
- Appliquer l'analyse statique du code pour la cohérence de la qualité du code sur PR

Conception et Mise en œuvre de solutions DevOps Microsoft (AZ 400)

Configurer les référentiels

- Configurer les autorisations dans le référentiel de contrôle de code source
- Organiser le référentiel avec git-tags
- Planifier la gestion des référentiels surdimensionnés
- Planifier la récupération de contenu dans tous les états du référentiel
- Purger les données du contrôle de code source

Intégrer le contrôle de code source avec les outils

- Intégrer GitHub aux pipelines DevOps
- Intégrer GitHub avec les solutions de gestion des identités (Azure AD)
- Concevoir GitOps
- Concevoir ChatOps
- Intégrer les artefacts de contrôle de code source pour la consommation humaine (par exemple, Git changelog)
- Intégrer GitHub Codespaces

Faciliter la communication et la collaboration

- Communiquer les informations de déploiement et de publication avec les parties prenantes de l'entreprise
- Créer des tableaux de bord combinant des tableaux de bord, des pipelines (tableaux de bord personnalisés sur Azure DevOps)
- Concevoir une stratégie de communication de gestion des coûts
- Intégrer le pipeline de publication avec le suivi des éléments de travail (par exemple, AZ DevOps, Jira, ServiceNow)
- Intégrer GitHub en tant que référentiel avec Azure Boards
- Communiquer l'analyse des utilisateurs

Générer un processus DevOps documentation

- Concevoir un processus d'intégration pour les nouveaux employés
- Évaluer et documenter les dépendances externes (par exemple, intégrations, packages)
- Évaluer et documenter les artefacts (version, notes de publication)

Automatiser la communication avec les membres de l'équipe

- Intégrer des outils de surveillance aux plates-formes de communication (par exemple, Teams, Slack, tableaux de bord)
- Notifier les parties prenantes sur les indicateurs clés, les alertes, la gravité à l'aide de plateformes de communication et de gestion de projet (par exemple, e-mail, SMS, Slack, Teams, ServiceNow, etc.)
- Intégrer la génération et la publication avec les plates-formes de communication (par exemple, échec de la génération, échec de la publication)
- Intégrer les approbations de demande d'attraction GitHub via des applications mobiles

Définir et mettre en œuvre une intégration continue

Automatisation de la conception et de la construction

- Intégrer le pipeline de construction à des outils externes (par exemple, analyse de la dépendance et de la sécurité, couverture du code)
- Mettre en œuvre des critères de qualité (par exemple, couverture du code, internationalisation, examen par les pairs)
- concevoir une stratégie de test (par exemple, intégration, chargement, fuzz, API, chaos)
- intégrer plusieurs outils (par exemple, GitHub Actions, Azure Pipeline, Jenkins)

Concevoir une stratégie de gestion de package

- Recommander des outils de gestion de package (par exemple, GitHub Packages, Azure Artifacts, Azure Automation Runbooks Gallery, Nuget, Jfrog, Artifactory)
- Concevoir une implémentation Azure Artifacts, y compris une stratégie de gestion des versions de conception de flux liés pour les ressources de code (par exemple, SemVer, basé sur la date)
- Planifier l'évaluation, la mise à jour et le rapport des dépendances des packages (GitHub Automated Security Updates, NuKeeper, GreenKeeper)
- Concevoir une stratégie de gestion des versions pour les packages (par exemple, SemVer, basé sur la date)

Concevoir une stratégie de gestion des versions pour les artefacts de déploiement

Concevoir une stratégie de gestion de l'infrastructure applicative

- Évaluer un mécanisme de gestion de la configuration pour l'infrastructure des applications
- Définir et appliquer la configuration d'état souhaitée pour les environnements

Mettre en œuvre une conception de stratégie de build

- Implémenter une infrastructure d'agent de build (y compris le coût, la sélection d'outils, les licences, la maintenabilité)
- Développer et implémenter des règles de déclenchement de build
- Développer des pipelines de build
- Concevoir l'orchestration de build (produits composés de plusieurs builds)
- Intégrer la configuration dans le processus de build
- Développer des scénarios de génération complexes (par exemple, agents conteneurisés, hybrides, GPU)

Maintenir la stratégie de build

- Surveiller l'intégrité du pipeline (taux de défaillance, durée, tests floconneux)
- Optimiser la build (coût, temps, performances, fiabilité)
- Analyser la charge CI pour déterminer la configuration et la capacité de l'agent de build

Concevoir un processus de normalisation des builds dans toute l'organisation

- Gérer les agents de build auto-hébergés (modèles de VM, conteneurisation, etc.)
- Créer des sous-systèmes de build réutilisables (modèles YAML, groupes de tâches, groupes de variables, etc.)

Définir et mettre en œuvre une stratégie de livraison continue et de gestion des versions

- Développer des scripts et des modèles de déploiement
- Recommander une solution de déploiement (par exemple, GitHub Actions, Azure Pipelines, Jenkins, CircleCI, etc.)
- Concevoir et implémenter l'infrastructure en tant que code (ARM, Terraform, PowerShell, CLI)
- Développer le processus de déploiement d'applications (conteneur, binaire, scripts)
- Développer le processus de déploiement de base de données (migrations, déplacement de données, ETL)
- Intégrer la gestion de la configuration dans le cadre du processus de publication développer des déploiements complexes (IoT, Azure IoT Edge, mobile, App Center, DR, multi région, CDN, cloud souverain, Azure Stack, etc.)

Mettre en œuvre une solution d'automatisation de l'orchestration

- Combiner des cibles de publication en fonction du livrable de la version (par exemple, infrastructure, code, ressources, etc.)
- Concevoir le pipeline de publication pour garantir un ordre fiable des déploiements de dépendances
- Organiser les configurations et les processus de publication partagée (modèles YAML, groupes de variables, configuration de l'application Azure)
- Concevoir et implémenter des portes de publication et des processus d'approbation

Planifier la stratégie de l'environnement de déploiement

- Concevoir une stratégie de publication (blue/green, canary, ring)
- Implémenter la stratégie de publication (en utilisant des emplacements de déploiement, des configurations d'équilibrage de charge, Azure Traffic Manager, feature toggle, etc.)
- Sélectionner la solution d'état souhaitée appropriée pour un environnement de déploiement (PowerShell DSC, Chef, Puppet, etc.)
- Planifier la réduction des temps d'arrêt pendant les déploiements (échange VIP, équilibreur de charge, déploiements continus, etc.)
- Concevoir un plan de chemin d'accès au correctif pour répondre aux correctifs de code de haute priorité

Objectifs pédagogiques

Cette formation fournit les connaissances et les compétences nécessaires pour concevoir et implémenter les processus et les pratiques DevOps. Les professionnels DevOps sont connus pour rationaliser la livraison de produits par l'automatisation, optimiser les pratiques et améliorer la collaboration et la communication. Ces personnes sont responsables de la planification et de l'exécution de stratégies pour le code d'infrastructure et d'application qui permet les différentes approches d'ingénierie logicielle telles que l'intégration continue, la livraison continue, la surveillance continue, les tests continus et la rétroaction.

A l'issue de la formation les candidats sont en mesure de :

- Apprendre à planifier DevOps
- Utiliser le contrôle des sources
- Mettre à l'échelle Git pour une entreprise
- Consolider les artéfacts
- Concevoir une stratégie de gestion des dépendances
- Gérer des secrets, implémenter l'intégration continue
- Implémenter une stratégie de construction de conteneur
- Concevoir une stratégie de publication
- Concevoir un flux de travail de gestion de publication
- Implémenter un schéma de déploiement et optimiser les mécanismes de commentaires.

Méthode et 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 en français sera remis aux participants au format électronique via la plateforme

Evaluation :

les acquis sont évalués tout au long de la formation et en fin de formation par le formateur (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. Les appréciations que vous avez formulées font l'objet d'un enregistrement et d'une analyse qualitative de la formation et du formateur. ITsystem formation dispose d'un processus qualité qui prend en considération les retours des participants afin d'être proactif quant à la solution corrective adaptée.

Nous veillons à ce que tous les objectifs de l'examen soient couverts en profondeur afin que vous soyez prêt pour toute question de l'examen. Nos tests pratiques sont rédigés par des experts de l'industrie en la matière. Ils travaillent en étroite collaboration avec les fournisseurs de certification pour comprendre les objectifs de l'examen, participer aux tests bêta et passer l'examen eux-mêmes avant de créer de nouveaux tests pratiques

- Suivi : une feuille d'émargement par demi-journée de présence est signée par chacun des participants.
- Les simulations en ligne basées sur la performance offrent une expérience pratique de l'environnement de travail
- Les questions sont similaires aux questions d'examen afin que vous testiez votre connaissance des objectifs de l'examen
- Des explications détaillées pour les réponses correctes et distrayantes renforcent le matériel
- Le mode étude couvre tous les objectifs en veillant à ce que les sujets soient couverts
- Le mode de certification (chronométré) prépare les étudiants aux conditions de passage des examens
- Des rapports de score instantanés et approfondis vous indiquent exactement les domaines sur lesquels vous concentrer.
- Cette formation peut être dispensée en mode présentiel comme en distanciel.
- Elle prend en charge les compétences ci-dessous ; le pourcentage indique le poids relatif du module dans l'examen global. Plus vous vous concentrez sur des modules avec un pourcentage plus élevé, plus vous obtiendrez probablement plus de notes à l'examen.

Cet examen mesure votre capacité à accomplir les tâches techniques suivantes :

- Développer une stratégie d'instrumentation (5 à 10%)
- Développer une stratégie d'ingénierie SRE (Site Reliability Engineering) (5 à 10%),
- Développer un plan de sécurité et de conformité (10 à 15%)
- Gérer le contrôle des sources (10 à 15%)
- Faciliter la communication et la collaboration (10 à 15%)
- Définir et implémenter l'intégration continue (20 à 25%)
- Définir et implémenter une livraison continue et une stratégie de gestion des publications (10 à 15%).

Pour qui ?

Les étudiants de ce cours sont intéressés par la conception et la mise en œuvre de processus DevOps ou par la réussite à l'examen de certification Microsoft Azure DevOps Solutions.

Le public visé :

- Ingénieur Devops
- Professionnels de l'informatique qui souhaitent devenir Azure DevOps Engineer Expert
- Professionnels de l'informatique se préparant à l'examen AZ-400 de Microsoft

Conception et Mise en œuvre de solutions DevOps Microsoft (AZ 400)

Pré-requis

Il n'y a donc aucune condition préalable pour passer l'examen de certification Azure DevOps.

Les étudiants qui réussissent auront une connaissance et une compréhension préalables de :

- Les concepts du Cloud computing, y compris une compréhension des mises en œuvre de PaaS, SaaS et IaaS.
- L'administration et le développement de Azure avec une expertise avérée dans au moins un de ces domaines.
- Contrôle de version, développement logiciel Agile et principes de base du développement logiciel. Il serait utile d'avoir une expérience dans une organisation qui fournit des logiciels.

Cependant, **si vous souhaitez obtenir le badge Microsoft Certified: DevOps Engineer Expert, vous devez également passer l'examen AZ-104 ou AZ-204.**

Accessibilité

Il est possible de vous inscrire jusqu'à 2 jours ouvrés avant le début de la formation, sous condition de places disponibles et de réception du devis signé.

Il est aussi possible – sur demande – d'adapter des moyens de la prestation pour les personnes en situation de handicap en fonction du type de handicap.

Le centre de formation ITsystem Formation est situé au :

Grand Paris au
21 rue Jean Rostand
91898 ORSAY

Vous pouvez facilement y accéder par les transports en commun suivants :

RER B Le guichet BUS 11 et BUS 7

En voiture : prendre la N118, sortie 9 Centre universitaire Grandes écoles

Pré certification

Cette formation ouvre la voie à la certification « **Conception et Mise en œuvre des solutions DevOps Microsoft AZ 400** »