

Mis à jour le 07/10/2025

S'inscrire

Formation Bicep : Infrastructure as code sur Azure

3 jours (21 heures)

Présentation

Bicep est un langage déclaratif développé par Microsoft Azure pour simplifier la création et le déploiement d'infrastructures cloud via ARM. Plus lisible et typé que les modèles ARM traditionnels, il permet une gestion plus claire, modulaire et automatisée de l'Infrastructure as Code (IaC).

Notre formation Bicep – Infrastructure as Code sur Azure vous apprendra à concevoir, déployer et industrialiser vos architectures Azure à l'aide de Bicep. Vous découvrirez comment intégrer vos templates dans des pipelines CI/CD, automatiser les déploiements et appliquer les meilleures pratiques de sécurité et de gouvernance.

À l'issue de cette formation, vous serez en mesure de créer des templates Bicep professionnels, de gérer des déploiements complexes multi-environnements et d'intégrer Bicep à vos outils DevOps comme Azure DevOps et GitHub Actions.

Comme toutes nos formations, elle s'appuie sur [la dernière version stable de Bicep](#) et privilégie une approche pratique et opérationnelle.

Objectifs

- Concevoir et structurer des templates Bicep modulaires.
- Automatiser les déploiements avec CI/CD.
- Sécuriser les déploiements avec Key Vault et les identités managées.
- Utiliser what-if pour valider et anticiper les changements.

Public visé

- DevOps
- Ingénieurs Cloud
- Architectes d'infrastructure

- Administrateurs système
- Équipes en migration

Pré-requis

- Connaissances de base d'Azure et de son modèle de ressources
- Notions d'Infrastructure as Code et usage de la CLI Azure
- Pratique de Git ou d'un outil de versioning

Programme de formation Bicep – Infrastructure as Code sur Azure

[Jour 1 - Matin]

Introduction à Bicep et principes de l'IaC

- Présentation de Bicep et comparaison avec les ARM Templates
- Comprendre les fondements de l'Infrastructure as Code sur Azure
- Découverte du modèle Azure Resource Manager (ARM)
- Structure d'un fichier Bicep : paramètres, variables, ressources et outputs
- Concepts clés : déclaratif, idempotence et cycles de déploiement
- Atelier pratique : Création et déploiement d'un premier template Bicep simple.

[Jour 1 - Après-midi]

Environnement de travail et syntaxe Bicep

- Installation et configuration de la CLI Bicep et de VS Code
- Présentation des commandes essentielles : bicep build, what-if, decompile
- Exploration de la syntaxe Bicep : paramètres, variables, fonctions et conditions
- Gestion des ressources Azure et références croisées
- Structuration d'un projet Bicep professionnel
- Atelier pratique : Créer et déployer une infrastructure complète depuis VS Code.

Variables, paramètres et fonctions avancées

- Types de paramètres : chaînes, booléens, objets et tableaux
- Utilisation des variables calculées et fonctions intégrées
- Création de templates dynamiques et réutilisables
- Organisation du code et modularisation
- Bonnes pratiques de nommage et de gestion des valeurs sensibles
- Atelier pratique : Développer un template modulaire avec variables et fonctions.

[Jour 2 - Matin]

Modularisation et gestion des dépendances

- Concepts de modules et partage de code
- Création de modules interconnectés : réseau, stockage, calcul
- Utilisation du Bicep Registry et modules communautaires
- Gestion des dépendances et du paramétrage entre modules
- Versionnage et gouvernance des modules Bicep
- Atelier pratique : Construire une architecture modulaire complète avec Bicep.

[Jour 2 - Après-midi]

Déploiements multi-scope et sécurité

- Déployer à l'échelle d'un resource group ou d'une subscription
- Introduction au déploiement multi-environnements et cross-tenant
- Gestion des identités managées et des accès sécurisés
- Utilisation de Key Vault pour la gestion des secrets
- Bonnes pratiques pour un déploiement sécurisé
- Atelier pratique : Mise en place d'un template multi-environnements sécurisé.

Gestion des erreurs et validation des déploiements

- Utilisation de la commande what-if pour simuler les changements
- Analyse et résolution des erreurs de déploiement
- Optimisation des performances et du coût des déploiements
- Logs, diagnostics et audit dans Azure
- Rollback et stratégies de reprise sur incident
- Atelier pratique : Test complet d'un déploiement avec validation et rollback.

[Jour 3 - Matin]

Intégration CI/CD avec Azure DevOps et GitHub Actions

- Introduction à la CI/CD pour l'Infrastructure as Code
- Création d'un pipeline de déploiement Azure DevOps
- Intégration de GitHub Actions pour automatiser les builds Bicep
- Tests, approbations et gestion des environnements
- Sécurisation des secrets et des connexions de service
- Atelier pratique : Pipeline CI/CD automatisant un déploiement Bicep complet.

[Jour 3 - Après-midi]

Bonnes pratiques, supervision et gouvernance

- Structurer un projet Bicep en production
- Surveiller et auditer les déploiements via Azure Monitor et Policy
- Implémenter des stratégies de tagging et de budget
- Intégrer la gouvernance au niveau entreprise (SRE, FinOps, sécurité)
- Retour d'expérience et revue des erreurs courantes
- Atelier pratique : Supervision complète et audit d'un environnement Azure déployé.

Étude de cas complète et industrialisation

- Étude de cas : conception d'une infrastructure Azure de bout en bout
- Déploiement d'un environnement complet via modules et pipelines
- Tests de conformité, validation et automatisation
- Optimisation du code et du coût de maintenance
- Revue finale et certification des acquis
- Atelier pratique : Réalisation d'un projet complet Bicep en autonomie.

Sociétés concernées

Cette formation s'adresse à la fois aux particuliers ainsi qu'aux entreprises, petites ou grandes, souhaitant former ses équipes à une nouvelle technologie informatique avancée ou bien à acquérir des connaissances métiers spécifiques ou des méthodes modernes.

Positionnement à l'entrée en formation

Le positionnement à l'entrée en formation respecte les critères qualité Qualiopi. Dès son inscription définitive, l'apprenant reçoit un questionnaire d'auto-évaluation nous permettant d'apprécier son niveau estimé sur différents types de technologies, ses attentes et objectifs personnels quant à la formation à venir, dans les limites imposées par le format sélectionné. Ce questionnaire nous permet également d'anticiper certaines difficultés de connexion ou de sécurité interne en entreprise (intraentreprise ou classe virtuelle) qui pourraient être problématiques pour le suivi et le bon déroulement de la session de formation.

Méthodes pédagogiques

Stage Pratique : 60% Pratique, 40% Théorie. Support de la formation distribué au format numérique à tous les participants.

Organisation

Le cours alterne les apports théoriques du formateur soutenus par des exemples et des séances de réflexions, et de travail en groupe.

Validation

À la fin de la session, un questionnaire à choix multiples permet de vérifier l'acquisition correcte des compétences.

Sanction

Une attestation sera remise à chaque stagiaire qui aura suivi la totalité de la formation.

