+33 (0)1 48 10 09 46
formation@ambient-it.net
140 Avenue Jean Lolive 93500 Pantin - France

Updated on 09/10/2025

Sign up

Bicep training: Infrastructure as code on Azure

3 days (21 hours)

Overview

Bicep is a declarative language developed by Microsoft Azure to simplify the creation and deployment of cloud infrastructures via ARM. More readable and typified than traditional ARM models, it enables clearer, modular and automated management of Infrastructure as Code (IaC).

Our Bicep - Infrastructure as Code on Azure training course will teach you how to design, deploy and industrialize your Azure architectures using Bicep. You'll discover how to integrate your templates into CI/CD pipelines, automate deployments and apply security and governance best practices.

At the end of this course, you'll be able to create professional Bicep templates, manage complex multi-environment deployments and integrate Bicep with your DevOps tools like Azure DevOps and GitHub Actions.

Like all our training courses, it is based on the latest stable version of Bicep and favors a practical, operational approach.

Objectives

- Design and structure modular Bicep templates.
- Automate deployments with CI/CD.
- Secure deployments with Key Vault and managed identities.
- Use what-if to validate and anticipate changes.

Target audience

- DevOps
- Cloud engineers
- Infrastructure architects

- System administrators
- Migration teams

Prerequisites

- Basic knowledge of Azure and its resource model
- Notions of Infrastructure as Code and use of the Azure CLI
- Experience with Git or a versioning tool

Bicep training program - Infrastructure as Code on Azure

[Day 1 - Morning]

Introduction to Bicep and IaC principles

- Introduction to Bicep and comparison with ARM Templates
- Understanding the fundamentals of Infrastructure as Code on Azure
- Discovering the Azure Resource Manager (ARM) model
- Structure of a Bicep file: parameters, variables, resources and outputs
- Key concepts: declarative, idempotency and deployment cycles
- Practical workshop: Creation and deployment of a first simple Bicep template.

[Day 1 - Afternoon]

Bicep working environment and syntax

- Installation and configuration of the Bicep CLI and VS Code
- Introduction to essential commands: bicep build, what-if, decompile
- Exploring Bicep syntax: parameters, variables, functions and conditions
- Azure resource management and cross-referencing
- Structuring a professional Bicep project
- Practical workshop: Creating and deploying a complete infrastructure from VS Code.

Variables, parameters and advanced functions

- Parameter types: strings, Booleans, objects and arrays
- Use of calculated variables and built-in functions
- Creating dynamic, reusable templates
- Code organization and modularization
- Best practices for naming and managing sensitive values
- Practical workshop: Developing a modular template with variables and functions.

[Day 2 - Morning]

Modularization and dependency management

- Module concepts and code sharing
- Creating interconnected modules: network, storage, calculation
- Using the Bicep Registry and community modules
- Managing dependencies and parameterization between modules
- Versioning and governance of Bicep modules
- Practical workshop: Building a complete modular architecture with Bicep.

[Day 2 - Afternoon]

Multi-scope deployments and security

- Deploying at resource group or subscription level
- Introduction to multi-environment and cross-tenant deployment
- Managed identity and secure access management
- Using Key Vault to manage secrets
- Best practices for secure deployment
- Practical workshop: Setting up a secure multi-environment template.

Error handling and deployment validation

- Using the what-if command to simulate changes
- Analyzing and resolving deployment errors
- Optimizing deployment performance and costs
- Logging, diagnostics and auditing in Azure
- Rollback and disaster recovery strategies
- Practical workshop: Complete deployment testing with validation and rollback.

[Day 3 - Morning]

CI/CD integration with Azure DevOps and GitHub Actions

- Introduction to CI/CD for Infrastructure as Code
- Creating an Azure DevOps deployment pipeline
- Integrating GitHub Actions to automate Bicep builds
- Testing, approvals and environment management
- Securing secrets and service connections
- Practical workshop: CI/CD pipeline automating a complete Bicep deployment.

[Day 3 - Afternoon]

Best practices, supervision and governance

- Structuring a Bicep project in production
- Monitor and audit deployments via Azure Monitor and Policy
- Implement tagging and budgeting strategies
- Integrate governance at enterprise level (SRE, FinOps, security)
- Feedback and review of common errors
- Practical workshop: Complete supervision and audit of a deployed Azure environment.

Complete case study and industrialization

- Case study: designing an end-to-end Azure infrastructure
- Deployment of a complete environment via modules and pipelines
- Compliance testing, validation and automation
- Optimization of code and maintenance costs
- Final review and certification
- Practical workshop: Independent implementation of a complete Bicep project.

Companies concerned

This course is aimed at both individuals and companies, large or small, wishing to train their teams in a new, advanced IT technology, or to acquire specific business knowledge or modern methods.

Positioning on entry to training

Positioning at the start of training complies with Qualiopi quality criteria. As soon as registration is finalized, the learner receives a self-assessment questionnaire which enables us to assess his or her estimated level of proficiency in different types of technology, as well as his or her expectations and personal objectives for the forthcoming training course, within the limits imposed by the selected format. This questionnaire also enables us to anticipate any connection or security difficulties within the company (intra-company or virtual classroom) which could be problematic for the follow-up and smooth running of the training session.

Teaching methods

Practical training: 60% hands-on, 40% theory. Training material distributed in digital format to all participants.

Organization

The course alternates theoretical input from the trainer, supported by examples, with brainstorming sessions and group work.

Validation

At the end of the session, a multiple-choice questionnaire verifies the correct acquisition of skills.

Certification

A certificate will be awarded to each trainee who has completed the entire course.