

Fundamentals of Bicep Training

Unit 1

- **What is infrastructure as code?**
- **Defining Infrastructure as code**
- **Why use Infrastructure as code?**

- **Increase confidence**
 - Integration with current processes
 - Consistency
 - Automated scanning
 - Secret management
 - Access control
 - Avoid configuration drift

- **Manage multiple environments**
 - Provision new environments
 - Non-production environments
 - Disaster recovery

- **Better understand your cloud resources**
 - Audit trail
 - Documentation
 - Unified system
 - Better understanding of cloud infrastructure

Imperative and declarative code

Unit 2

What is Azure Resource Manager?

Azure Resource Manager concepts

Terminology

- Resource
- Resource group
- Subscription
- Management group:
- Azure Resource Manager template

Benefits

Operations: Control plane and data plane

Why use ARM templates?

- Repeatable results
- Orchestration
- Preview
- Testing and Validation
- Modularity
- CI/CD integration
- Extensibility

JSON and Bicep templates

Unit 3

What is Bicep?

Benefits of Bicep

- Simpler syntax:
- Modules
- Automatic dependency management
- Type validation and IntelliSense

Unit 4

How Bicep works

Bicep deployment

Comparing JSON and Bicep

Unit 5

When to use Bicep

Is Bicep the right tool?

When is Bicep the right tool?

- Azure-native
- Azure integration
- Azure support
- No state management
- Easy transition from JSON

When is Bicep not the right tool?

- Existing tool set:
- Multicloud
- **What is the main goal?**
- After completing this module, you'll be able to determine whether infrastructure as code is the right approach and tool for your organization.

Intermediate Bicep

Unit 1

Understand Azure resources

- Recourse providers
- Resource Types
- Recourse IDs

Unit 2

Define child resources

- How are child resources defined?
- Nested Resources
- Parent Property
- Construct the resource name
- Child Resource IDs
- Exercises on how to define child resources

Unit 3

Define extension resources

- How are extension resources defined?
- Extension resource IDs

Unit 4

Work with existing resources

- Refer to existing resources
- Refer to resources outside the resource group
- Refer to an existing resources properties
- Exercise on Deploy extension resources and use existing resources

What is the main goal?

By the end of this module, you'll be able to create Bicep templates that include child and extension resources. You'll also be able to create Bicep templates that refer to preexisting Azure resources that have been created outside the template file.

Advanced Bicep

Unit 1

Understand Deployment Scoped

- The Azure resource hierarchy
- Subscription-scoped resources
- Tenant-scoped resources
- Resource IDs

Unit 2

Deploy templates at various scopes

- Specify the target scope for a Bicep file
- Create resource groups
- Submit a deployment
- Exercise – deploy a subscription-scoped resource

Unit 3

Deploy resources to multiple scopes

- Specify the scope for a module
- Deploy across multiple resource groups
- Specify the scope for a single resource
- Create a Management group and subscription hierarchy
- Exercise – Deploy resources to multiple scopes by using modules
- Exercise – Deploy resources to a management group

What is the main goal?

By the end of this module, you'll be able to create Bicep templates that you can deploy at the subscription, management group, and tenant scopes. You'll also be able to create and use Bicep files to deploy across multiple scopes in a single deployment.