



Activate CI/CD with YML and Bicep

Duration: 3 days – Remote | Focus Area: Infrastructure as Code | Difficulty: 300 - Advanced

Intended Audience

Primary Audience:

- DevOps / Cloud Engineers
- Cloud Solution Architects
- Infrastructure / Platform Engineers

Secondary Audience:

- Software Engineers
- Site Reliability Engineer

Overview/Objectives

In this workshop, you will learn how to use Azure DevOps to automate common tasks like deploying resources in Azure using Infrastructure as Code, and then how to automate the build, test, and deployment of your application code. You will learn how to create a reusable pipeline using YML and Bicep, and how to use triggers, variables, and templates to make your pipeline more flexible, and add the ability to deploy to multiple environments. You will also learn how to create reusable libraries of reusable Bicep and YML templates.

Key Takeaways

By the end of this workshop, you will have a solid understanding of how to use Azure DevOps to automate your deployment and development process and how to create reusable pipeline components that can be easily adapted to different projects and environments.

Agenda

Bicep Presentation ~4 hours

- What is Infrastructure as Code?
- Tooling with Visual Studio Code
- Introduction to ARM Templates
- What is Bicep?
 - Build Reusable Bicep using Parameters
 - Using Conditions and Loops in Bicep
 - Using Modules in a Bicep Template
 - Migrate Resources & ARM to Bicep
 - Reusable Bicep with Template Libraries
- Deployment Scopes
- Bicep Tips and Tricks

Bicep Hands-On Labs ~4 hours

YML Presentation ~4 hours

- Intro to YML Pipelines
- AzDO Pipeline Agents
- AzDO Service Connections
- Pipeline Basics
 - Using Templates in Pipelines
 - Using Variables in Pipelines
 - Pipelines Triggers
- Reusable YML with Template Libraries
- Advanced Pipeline Topics
- Deploying Bicep in a Pipeline
- Pipeline Reports

YML Hands-On Labs ~4 hours

Course Details

IaC Overview

- Infrastructure as Code – Overview
- Benefits
- Challenges

Tooling with Visual Studio Code

- VS Code Tooling overview
- Version Control overview

Introduction to Bicep

- What is Azure Resource Manager?
- What is Bicep?
- Building a Bicep Template

Bicep Parameters

- Understand Parameters
- Parameter Files
- Secure Parameters

Conditions and Loops in Bicep

- Conditional deployments
- Deploy resources using loops
- Control loop execution and nest loops
- Using variable and output loops

Bicep Modules

- Create and use Bicep modules
- Add parameters and outputs to modules
- Local Modules
- Remote modules - private Registry (ACR)
- Remote modules - public registry (AVM)

Migrate Resources to Bicep

- Migrating ARM templates to Bicep
- Converting Azure Portal resources to Bicep
- Testing Bicep modules and reviewing changes using What-If

Azure Template Specs (Optional)

- What are Template Specs?
- Comparison on Template Specs to Bicep Registry
- Creating and Using Template Specs

Deploy to Subscriptions & Mgmt. Groups

- Understanding deployment scopes
- Reviewing deployment commands

Bicep Tips and Tricks

YML: Before You Start

- Setting up an Azure DevOps Project
- Visual Studio Code
- Creating a Custom Build Agent (optional)

YML: Introduction To Pipelines

- Create a YAML-Pipeline via GUI
- Use the assistant to add tasks
- Extend the pipeline with variables

YML: Tasks, Jobs, Stages and Dependencies

- Separating tasks into different jobs
- Adding dependencies between jobs
- Splitting our pipeline into stages
- Dependencies between stages
- Approvals

YML: Working with Templates

- Load steps from templates
- Reusing stages with templates

YML: Triggers

- Working with triggers and branches
- Schedule Triggers
- Pull Request Triggers

YML: Variables and Parameters

- Using Parameters
- Using Local Variables
- Variable Template Files
- Predefined Variables
- Creating Variables in Code
- Using Variable Groups
- Loading Values from Key Vault

YML: Multiple Environments and Stages

- Creating AzDO Environments
- Deploying to Multiple Environments with Dependencies

YML: Reusable Libraries

- Creating a Template Repository

YML: Splitting CI and CD

- Creating Build pipeline and save artifacts
- Creating Release Pipeline and using a pre-built artifact

Prerequisites

Cloud Environment

- Azure Sandbox Environment (Subscription or Resource Group) with Contributor level Access
- Azure Sandbox DevOps Organization and Project
- Cloud Shell

Local Environment

- Admin privileges on local machine to install CLI tools such as Azure CLI and Bicep
- Visual Studio Code and Git
- Azure CLI – with Bicep installed
- AZ PowerShell Modules installed
- Modern Browser

For more information

Contact your Microsoft Account Representative for further details.