Infrastructure As Code with Bicep

Introduction

DevOps is a journey not a destination. Implementing Infrastructure-as-Code is one of the first steps you will need to take!

When implementing an application environment in the cloud, it is important to have a repeatable way to deploy the underlying infrastructure components as well as your software into the target environment. This includes resources such as:

- Virtual Networks, Network Security Groups (Firewalls), Public IPs, Virtual Machines, Storage (Disks)
- PaaS Services (Azure SQL, App Service, etc)
- Configuration Management (installing & configuring software on VMs)

The best way to make deployments repeatable is to define them with code, hence the term "Infrastructure as Code" (aka IAC). There are multiple technologies that enable you to achieve this. Some of these include:

- ARM Templates with JSON
- ARM Templates with Bicep
- PowerShell Desired State Configuration (DSC)
- HashiCorp's Terraform & Packer
- Ansible, Chef, Puppet, Salt Stack, and others

This hack is focused on using Bicep to implement your IaC. It does not mean this is the only way to implement IaC, it's just one of many ways. If you want to learn how to do IaC in Azure with other technologies, try one of our other IaC hacks for ARM Templates & PowerShell DSC, Terraform or Ansible.

Duration

3 days

Learning Objectives

This hack will help you learn:

- How ARM Templates with Bicep can be used to deploy Azure infrastructure

The challenges build upon each other incrementally. You will start by creating a basic Bicep template to get you familiar with the tools & syntax. Then you extend your template incrementally to deploy multiple infrastructure resources to Azure.

Challenges

- Challenge 00: Pre-Requisites - Ready, Set, Go!
  - Prepare your workstation to work with Azure
- Challenge 01: Basic Bicep
  - Develop a simple Bicep file that takes inputs to create an Azure Storage Account, and returns outputs
- Challenge 02: Bicep Expressions and Referencing resources
  - Learn Bicep expressions and referencing resources
- Challenge 03: Advanced Resource Declarations
  - Advanced resource declarations
- Challenge 04: Secret Values with Azure Key Vault
  - Learn how NOT to lose your job!
- Challenge 05: Deploy a Virtual Machine
  - Create complex deployment with multiple dependencies
- Challenge 06: Bicep Modules
  - Learn how create resusable modules for granular resource management
- Challenge 07: Configure VM to Run a Web Server
  - Learn about custom script extensions
- Challenge 08: Deploy a Virtual Machine Scale Set
  - Create complex deployment with Bicep using modules
- Challenge 09: Configure VM Scale Set to Run a Web Server
  - Learn about custom script extensions with VM Scale Sets
- Challenge 10: Configure VM Scale Set to Run a Web Server Using cloud-init
  - How cloud-init scripts can be run on a Virtual Machine Scale Set (VMSS)
- Challenge 11: Deploy Resources to Different Scopes
  - Learn how to deploy resources to different scopes
- Challenge 12: Deploy an Azure App Service
  - Learn how to an Azure App Service & deploy an app to it
- Challenge 13: Deploy an AKS cluster
  - Learn how to an AKS cluster & deploy an app to it

Prerequisites

You will want to prepare your machine with the following to help complete the Challenges for this hack:

- Windows Subsystem for Linux (Windows only) *Optional, but highly recommended.*
- Azure CLI
- PowerShell Cmdlets for Azure
- Visual Studio Code
- Bicep plugins for VS Code
    - Bicep VS Code Extension
    - Bicep CLI