

Azure Infrastructure as Code with Terraform

What you'll learn

- How to create production-grade Azure Infrastructure Code using Terraform
- Azure Infrastructure as Code with Terraform
- Project-focused format so you know how to deploy resources in Azure

Duration

3 days

Requirements

We recommend that you have basic sysadmin skills to be successful in this course.

Software and Hardware requirements

High Speed internet connection and Azure account

Description

Learn how to create **production-grade** Azure **Infrastructure Code** using **Terraform** in this project-focused hands-on course. Follow along as you learn Terraform and put theory into practice by solving challenges.

We go over what Terraform is and why it's a highly sought-after skillset. We'll also go over how to set up Azure CloudShell so we can dive right into developing with Terraform in Azure straight away.

In the first section you'll learn:

- What is **Terraform**?
- Getting Started with **Azure CloudShell**
- How to **setup your projects** to follow along

Learn the fundamentals of Terraform by creating a configuration file and deploying resources into Azure. We'll also explore the most common ways to authenticate with Azure using the Azure Provider and discuss creating resource dependencies within the Terraform configuration.

Here's what you'll learn about **Terraform Configurations**:

- Terraform Configuration Files
- Authenticating with the Azure Provider
- Terraform Init, Plan, and Apply
- Using Interpolation within Terraform Configurations

Understanding how Terraform state works is a critical component for building Terraform solutions, which is why we'll be going deep into how the state file keeps track of changes made with Terraform.

Here's what you'll learn about **Terraform State**:

- Why does Terraform need to keep track of state
- Examine the inter-workings of the state file
- Storing the state file centrally using remote state
- Retrieve output from Terraform state

Variables allow Terraform modules to take in input making configuration files dynamic. Infrastructure code can be templated and re-used throughout different environments.

Here's what you'll learn about **Terraform Variables**:

- Create variable blocks in Terraform configurations
- Input values into Terraform configurations using several methods
- Learn about the common variable types

Terraform modules split up infrastructure into small, testable components that make it easy for infrastructure developers to introduce changes in small amounts. Learn how to easily create modules and use them with source control.

Here's what you'll learn about **Terraform Modules**:

- Create a module
- Pass data between modules
- Learn how to use modules from GitHub
- Learn about the Terraform Registry

HCL is a functional language that allows infrastructure developers to create loops and conditional logic within Terraform code. This allows a high levels of abstraction to be created for deploying infrastructure.

Here's what you'll learn about **HCL**:

- Functions, conditions, loops and expressions
- Dynamic blocks within Terraform resource blocks
- Creating resources that scale with count

Who this course is for:

- DevOps engineers curious about Infrastructure as Code.