

Getting Started with Terraform for Google Cloud

This course provides an introduction to using Terraform for Google Cloud. It enables learners to describe how Terraform can be used to implement infrastructure as a code and to apply some of its key features and functionalities to create and manage Google Cloud infrastructure. Learners will get hands-on practice building Google Cloud resources using Terraform.

**DURATION**

1 day

**LEVEL**

100 - Introductory

**FORMAT**

Instructor-led

**SCALABLE**

1-16, scaled to 1000

What you'll learn

- Define the business need for infrastructure as code and the benefits of using it in your environment.
- Explain the features and functionalities of Terraform.
- Use Terraform resources, variables, and output values to create Google Cloud infrastructure resources.
- Use Terraform modules to build reusable configurations.
- Explain Terraform state and its importance.

Overview	5 modules
Who this course is for	• Cloud engineers, DevOps engineers, and individuals who want to start using Terraform to automate infrastructure provisioning with a focus on Google Cloud Platform
Products	• Advanced Compute Images • Advanced Networking • Storage Insights • Compute Engine • VPC Networking • Cloud Storage • Terraform • Cloud Foundation Toolkit
Prerequisites	• Complete Google Cloud Fundamentals: Core Infrastructure • Have basic programming skills and familiarity with using CLI • Have general familiarity with Google Cloud
Not covered	• Cloud Build

Module
01

Introduction to Terraform for Google Cloud

Topics

- Introduction to IaC
- What is infrastructure as code (IaC)?
- Problems IaC can solve
- Benefits of IaC
- Provisioning versus configuration
- Imperative versus declarative approach
- Introduction to Terraform
- Terraform overview
- Terraform features
- IaC configuration workflow
- Terraform use cases
- Using Terraform
- How to use Terraform
- Running Terraform in production

Objectives

- Define infrastructure as code.
- Explain the features and benefits of using Terraform.
- Explain the use case of Terraform for Google Cloud.
- Describe how to use Terraform for Google Cloud.

Activities

- 1 quiz
-

Module 02

Terms and concepts

Topics

- The Author phase
- Terraform Directory structure
- Introduction to HCL syntax
- Resources
- Variables
- State
- Modules
- Terraform commands
- terraform init
- terraform plan
- terraform apply
- terraform fmt
- terraform destroy
- Terraform Validator tool
- Introduction
- Why use the Terraform Validator tool
- Validation workflow
- Terraform Validator use cases

Objectives

- Explain the Terraform workflow.
- Create basic configuration files within Terraform.
- Explain the purpose of a few Terraform commands.
- Describe the Terraform Validator tool.
- Create, update, and destroy Google Cloud resources using Terraform.

Module
03

Writing Infrastructure Code for Google Cloud

Topics

- Introduction to Resources
- Resources overview
- Syntax
- Example
- Refer a resource attribute
- Considerations to define a resource block
- Meta-arguments for resources
- Resource dependencies
- Implicit dependency
- Explicit dependency
- Introduction to Variables
- Overview
- Syntax to declare a variable
- Syntax to reference and assign a value to a variable
- Variables best practices
- Introduction to output values
- Output values overview
- Best practices
- Terraform Registry and CFT
- Introduction to Terraform Registry
- Introduction to CFT

Objectives

- Declare the resources within Terraform.
- Explain implicit and explicit resource dependencies.
- Use variables and output values within the root configuration.
- Explain Terraform Registry and Cloud Foundation Toolkit.

Activities

- 1 lab
 - 1 quiz
-

Module 04

Organizing and Reusing Configuration with Terraform Modules

Topics

- Introduction to modules:
- Why are modules needed
- What is a module?
- Example
- Reusing configurations by using modules
- Module sources
- Calling a module into the source configuration
- Using variables to parameterize your configuration
- Pass resource attributes using output variables
- Module use cases, benefits, and best practices

Objectives

- Define Terraform modules.
- Use modules to reuse configurations.
- Use modules from the public registry.
- Use input variables to parameterize configurations.
- Use output values to access resource attributes outside the module.

Activities

- 1 lab
 - 1 quiz
-

Module 05

Introduction to Terraform State

Topics

- Introduction to Terraform state
- How information is stored in a Terraform state file
- Ways to save a state file
- Storing a state file in a Cloud Storage bucket
- Issues when storing the Terraform state locally
- Benefits of storing a state file in a Cloud Storage bucket
- Process of storing a Terraform state file remotely in a Cloud Storage bucket
- Terraform state best practices

Objectives

- Define Terraform state.
- List the benefits of storing the state file remotely.
- Explain how to store the Terraform state in a Cloud Storage bucket.
- Explain Terraform state best practices.

Activities

- 1 lab
- 1 quiz