Terraform Modules: Overview & Best Practices

# What is a Terraform Module?

A Terraform module is a container for multiple resources that are used together. A module can be thought of as a reusable component that can be called multiple times throughout your configuration. Every Terraform configuration file (.tf) is part of a module, even if it's just a single file.

# Why Are Terraform Modules Required?

Modules are required to promote code reuse, improve maintainability, and simplify complex infrastructure deployments. They help break down large configurations into smaller, manageable, and reusable units.

## Benefits of Using Terraform Modules

• Reusability: Define infrastructure once and use it in multiple places.  
• Maintainability: Update logic in one place and apply it everywhere.  
• Organization: Split complex infrastructure into logical units.  
• Collaboration: Teams can work on modules independently.

# Example: Using a Terraform Module to Create a GCP VM

Root Configuration (`main.tf`):

module "vm\_instance" {  
 source = "./modules/compute\_instance"  
 instance\_name = "dev-vm"  
 machine\_type = "e2-medium"  
 zone = "asia-south1-a"  
 project\_id = "my-gcp-project-id"  
 image\_family = "debian-11"  
 image\_project = "debian-cloud"  
}

Module Configuration (`modules/compute\_instance/main.tf`):

resource "google\_compute\_instance" "default" {  
 name = var.instance\_name  
 machine\_type = var.machine\_type  
 zone = var.zone  
 project = var.project\_id  
  
 boot\_disk {  
 initialize\_params {  
 image = "${var.image\_project}/${var.image\_family}"  
 }  
 }  
  
 network\_interface {  
 network = "default"  
 access\_config {}  
 }  
}

Module Variables (`modules/compute\_instance/variables.tf`):

variable "instance\_name" {}  
variable "machine\_type" {}  
variable "zone" {}  
variable "project\_id" {}  
variable "image\_family" {}  
variable "image\_project" {}

# Conclusion

Terraform modules help standardize and simplify your infrastructure code by encapsulating logic in reusable units. This promotes best practices, reduces duplication, and enables clean, scalable configurations.