**Terraform Modules - Notes**

**What is a Module in Terraform?**

A **module** in Terraform is a container for multiple resources that are used together. It acts as a **reusable package** of Terraform code.

Think of it like a function in traditional programming:

* **Input** → variables
* **Output** → values
* **Reusable logic** → a set of resources

**Why Use Modules?**

* **Reusability**: Write once, use many times (e.g., VMs, networks, key vaults).
* **Organization**: Keeps your Terraform code clean and manageable.
* **Scalability**: Helps manage large-scale infrastructure efficiently.

**Structure of a Terraform Module**

A typical module is organized in a folder with these files:

* main.tf – Resource definitions
* variables.tf – Input variables
* outputs.tf – Outputs from the module

**Example: Simple Key Vault Module**

**Folder Structure:**

terraform/

├── main.tf

├── variables.tf

└── outputs.tf

**main.tf**

resource "azurerm\_key\_vault" "example" {

name = var.key\_vault\_name

location = var.location

resource\_group\_name = var.resource\_group\_name

tenant\_id = var.tenant\_id

sku\_name = "standard"

access\_policy {

tenant\_id = var.tenant\_id

object\_id = var.object\_id

secret\_permissions = ["Get", "List"]

}

}

**variables.tf**

variable "key\_vault\_name" {}

variable "location" {}

variable "resource\_group\_name" {}

variable "tenant\_id" {}

variable "object\_id" {}

**outputs.tf**

output "key\_vault\_id" {

value = azurerm\_key\_vault.example.id

}

**How to Use This Module in Your Main Terraform Code**

module "keyvault" {

source = "./terraform"

key\_vault\_name = "examplekeyvault"

location = "West Europe"

resource\_group\_name = "example-resources"

tenant\_id = data.azurerm\_client\_config.current.tenant\_id

object\_id = data.azurerm\_client\_config.current.object\_id

}