Lab 8: Using Variables in Terraform (Single File VM Deployment)

Author: Dr. Sandeep Kumar Sharma
Level: Beginner
Platform: Ubuntu Linux + Microsoft Azure
Prerequisite: Lab 1 to Lab 7

Introduction to Variables in Terraform

What is a Variable?

A **variable** is a container that stores a value.

In simple words: - Variable = Value holder

Example: - Name = Sandeep
Here, `Name` is variable and `Sandeep` is value.

Why Variables are Important in Terraform

Without variables: - Values are fixed (hardcoded) - Code cannot be reused - Changes require editing code again and again

With variables: - Code becomes reusable - Configuration becomes flexible - Environment changes become easy - Infrastructure becomes dynamic

---

Variable Concept in Terraform

Terraform uses variables to: - Avoid hardcoding values - Make code reusable - Support multiple environments - Improve maintainability

---

Types of Variables in Terraform (Basic Level)

- String
- Number
- Boolean
- List

- Map

(In this lab, we will use only **string** variables)

How Variables Work in Terraform

Flow:

Declare Variable → Define Value → Use Variable in Resource

Hands-On Lab

Objective

Create one Linux Virtual Machine using **only variables** in a **single Terraform file**.

Step 1: Go to Terraform Directory

```
cd terraform-azure-lab
```

Step 2: Create One Terraform File

```
touch vm_with_variables.tf
```

Step 3: Open File

```
nano vm_with_variables.tf
```

Step 4: Complete Single File Configuration

Paste **everything** below in one file:

```
########################################
# Variables Declaration
########################################

variable "rg_name" {
  description = "Resource Group name"
  type        = string
  default     = "rg-terraform-lab"
}

variable "location" {
  description = "Azure region"
  type        = string
  default     = "East US"
}

variable "vnet_name" {
  description = "Virtual Network name"
  type        = string
  default     = "vnet-variable-lab"
}

variable "subnet_name" {
  description = "Subnet name"
  type        = string
  default     = "subnet-variable-lab"
}

variable "nsg_name" {
  description = "Network Security Group name"
  type        = string
  default     = "nsg-variable-lab"
}

variable "public_ip_name" {
  description = "Public IP name"
  type        = string
  default     = "pip-variable-lab"
}

variable "nic_name" {
  description = "NIC name"
  type        = string
  default     = "nic-variable-lab"
}

variable "vm_name" {
```

```
  description = "Virtual Machine name"
  type        = string
  default     = "Sandeep-machine-variable"
}

variable "admin_username" {
  description = "VM admin username"
  type        = string
  default     = "azureuser"
}

variable "vm_size" {
  description = "VM size"
  type        = string
  default     = "Standard_B1s"
}

########################################
# Provider
########################################

terraform {
  required_providers {
    azurerm = {
      source  = "hashicorp/azurerm"
      version = "~> 3.0"
    }
  }
}

provider "azurerm" {
  features {}
}

########################################
# Resource Group
########################################

resource "azurerm_resource_group" "rg" {
  name     = var.rg_name
  location = var.location
}

########################################
# Virtual Network
########################################

resource "azurerm_virtual_network" "vnet" {
```

```
  name                = var.vnet_name
  address_space       = ["10.10.0.0/16"]
  location            = var.location
  resource_group_name = var.rg_name
}

########################################
# Subnet
########################################

resource "azurerm_subnet" "subnet" {
  name                 = var.subnet_name
  resource_group_name  = var.rg_name
  virtual_network_name = azurerm_virtual_network.vnet.name
  address_prefixes     = ["10.10.1.0/24"]
}

########################################
# Network Security Group
########################################

resource "azurerm_network_security_group" "nsg" {
  name                = var.nsg_name
  location            = var.location
  resource_group_name = var.rg_name
}

########################################
# Public IP
########################################

resource "azurerm_public_ip" "pip" {
  name                = var.public_ip_name
  location            = var.location
  resource_group_name = var.rg_name
  allocation_method   = "Static"
}

########################################
# Network Interface
########################################

resource "azurerm_network_interface" "nic" {
  name                = var.nic_name
  location            = var.location
  resource_group_name = var.rg_name

  ip_configuration {
```

```
    name                          = "internal"
    subnet_id                     = azurerm_subnet.subnet.id
    private_ip_address_allocation = "Dynamic"
    public_ip_address_id          = azurerm_public_ip.pip.id
  }
}

########################################
# Linux Virtual Machine
########################################

resource "azurerm_linux_virtual_machine" "vm" {
  name                = var.vm_name
  resource_group_name = var.rg_name
  location            = var.location
  size                = var.vm_size
  admin_username      = var.admin_username

  network_interface_ids = [
    azurerm_network_interface.nic.id
  ]

  admin_ssh_key {
    username   = var.admin_username
    public_key = file("~/.ssh/id_rsa.pub")
  }

  os_disk {
    caching              = "ReadWrite"
    storage_account_type = "Standard_LRS"
  }

  source_image_reference {
    publisher = "Canonical"
    offer     = "0001-com-ubuntu-server-focal"
    sku       = "20_04-lts"
    version   = "latest"
  }
}
```

Step 5: Initialize Terraform

```
terraform init
```

Step 6: Plan

```
terraform plan
```

Step 7: Apply

```
terraform apply
```

Type:

```
yes
```

Step 8: Verify VM

VM Name:

```
Sandeep-machine-variable
```

Step 9: Cleanup

```
terraform destroy
```

Type:

```
yes
```