Step-01: Introduction

- Understand about Output Values and implement them
- Query outputs using `terraform output`
- Understand about redacting secure attributes in output values
- Generate machine-readable output
- You can export both `Argument & Attribute References`
- Redact the sensitive outputs using `sensitve = true` in output block

Step-02: c1-versions.tf

```
# Terraform Block
terraform {
  required_version = ">= 1.0.0"
  required_providers {
    azurerm = {
      source = "hashicorp/azurerm"
      version = ">= 2.0"
    }
  }
}

# Provider Block
provider "azurerm" {
 features {}
}
```

Step-03: c2-variables.tf

```
# Input Variables

# 1. Business Unit Name
variable "business_unit" {
  description = "Business Unit Name"
  type = string
  default = "hr"
}
# 2. Environment Name
variable "environment" {
  description = "Environment Name"
  type = string
  default = "poc"
}
# 3. Resource Group Name
variable "resoure_group_name" {
  description = "Resource Group Name"
  type = string
  default = "myrg"
}
# 4. Resource Group Location
variable "resoure_group_location" {
  description = "Resource Group Location"
  type = string
  default = "East US"
}

# 5. Virtual Network Name
variable "virtual_network_name" {
  description = "Virtual Network Name"
  type = string
  default = "myvnet"
}
```

Step-04: c3-resource-group.tf

```
# Resource-1: Azure Resource Group
resource "azurerm_resource_group" "myrg" {
  name = "${var.business_unit}-${var.environment}-${var.resoure_group_name}"
  location = var.resoure_group_location
}
```

Step-05: c4-virtual-network.tf

```
# Create Virtual Network
resource "azurerm_virtual_network" "myvnet" {
  name                = "${var.business_unit}-${var.environment}-${var.virtual_network_name}"
  address_space       = ["10.0.0.0/16"]
  location            = azurerm_resource_group.myrg.location
  resource_group_name = azurerm_resource_group.myrg.name
}
```

Step-06: terraform.tfvars

```
business_unit = "it"
environment = "dev"
resoure_group_name = "rg"
virtual_network_name = "vnet"
```

Step-07: c5-outputs.tf

```
# 1. Output Values - Resource Group
output "resource_group_id" {
  description = "Resource Group ID"
  # Atrribute Reference
  value = azurerm_resource_group.myrg.id
}
output "resource_group_name" {
  description = "Resource Group name"
  # Argument Reference
  value = azurerm_resource_group.myrg.name
}

# 2. Output Values - Virtual Network
output "virtual_network_name" {
  description = "Virutal Network Name"
  value = azurerm_virtual_network.myvnet.name
}
```

Step-06: Execute Terraform Commands

```
# Initialize Terraform
terraform init

# Validate Terraform configuration files
terraform validate
```

```
# Format Terraform configuration files
terraform fmt

# Review the terraform plan
terraform plan

# Create Resources
terraform apply -auto-approve

# Observation
1. Review the outputs in CLI Output
```

Step-07: Query Terraform Outputs

- Terraform will load the project state in state file, so that using `terraform output` command we can query the state file.

```
# Terraform Output Commands
terraform output
terraform output resource_group_id
terraform output virtual_network_name
```

Step-08: Output Values - Suppressing Sensitive Values in Output

- We can redact the sensitive outputs using `sensitve = true` in output block
- This will only redact the cli output for terraform plan and apply
- When you query using `terraform output`, you will be able to fetch exact values from `terraform.tfstate` files
- Add `sensitve = true` for output `virtual_network_name`

```
# 2. Output Values - Virtual Network
output "virtual_network_name" {
  description = "Virutal Network Name"
  value = azurerm_virtual_network.myvnet.name
  sensitive = true
}
```

- Test the flow

```
# Terraform Apply
terraform apply -auto-approve
Observation:
1. You should see the value as sensitive

# Query using terraform output
terraform output virtual_network_name
Observation:
1. You should get non-redacted original value from terraform.tfstate file
```

Step-09: Generate machine-readable output

```
# Generate machine-readable output
terraform output -json
```

Step-10: Destroy Resources

```
# Destroy Resources
terraform destroy -auto-approve

# Clean-Up
rm -rf .terraform*
rm -rf terraform.tfstate*

# Comment sensitive=true
In c5-outputs.tf, roll back "sensitive=true"
```

References

- Terraform Output Values