Step-01: Introduction

- Resources: Count Meta-Argument
- Understand Resource Meta-Argument `count`
- Also implement count and count index practically
- In general, 1 Azure VM Instance Resource in Terraform equals to 1 VM Instance in Real Azure Cloud
- 5 Azure VM Instance Resources = 5 Azure VM Instances in Azure Cloud
- With `Meta-Argument count` this is going to become super simple.

Step-02: Simple Example - Review terraform-manifests-v1

- Folder Path: terraform-manifests-v1
- c1-versions.tf
- c2-resource-group.tf

```
# Resource-1: Azure Resource Group
resource "azurerm_resource_group" "myrg" {
  name = "myrg-${count.index}"
  location = "East US"
  count = 3
}
```

Step-03: Execute Terraform Commands

```
# Change Directory
cd terraform-manifests-v1

# Terraform Initialize
terraform init

# Terraform Validate
terraform validate

# Terraform Plan
terraform plan

# Terraform Apply
terraform apply

# Terraform Destroy
terraform destroy -auto-approve

# Verify
1. We should see 3 Resource groups created.
2. Verify the count.index number for each resource group
```

Step-04: Review Terraform Configs V2

- **Usecase:** Create two Azure Linux VMs using Meta-Argument `count`

1. We need two Public IPs for two VMs
2. We need two Network Interfaces two VMs

- We are going to learn the following concepts over the process

- Learn about Terraform Console

- Learn about Terraform Length Function
- Learn about Terraform element Function
- Learn about Terarform Splat Expression
- **Folder Path:** terraform-manifests-v2
- c1-versions.tf: No changes
- c2-resource-group.tf: No changes
- c3-virtual-network.tf: Has changes for Network Interface
- c4-linux-virtual-machine.tf: Has changes

Step-05: terraform-manifests-v2 - c3-virtual-network.tf

- For Public IP resource add `count=2`

```
# Create Azure Public IP Address
resource "azurerm_public_ip" "mypublicip" {
  count = 2
  name                = "mypublicip-${count.index}"
  resource_group_name = azurerm_resource_group.myrg.name
  location            = azurerm_resource_group.myrg.location
  allocation_method   = "Static"
  domain_name_label = "app1-vm-${count.index}-${random_string.myrandom.id}"
}
```

Step-06: Understand about Splat Expression

- Terarform Splat Expression
- Terraform element Function

```
# Terraform console
terraform console
element(["atin", "gupta", "trainer"], 0)
element(["atin", "gupta", "trainer"], 1)
element(["atin", "gupta", "trainer"], 2)

# To get last element from list
length(["atin", "gupta", "trainer"])
element(["atin", "gupta", "trainer"], length(["atin", "gupta", "trainer"])-1)
```

Step-07: terraform-manifests-v2 - c3-virtual-network.tf

- For Network Interface resource add `count=2`
- Associate Public IP using `Element Function` and `Splat Expression`

```
# Create Network Interface
resource "azurerm_network_interface" "myvmnic" {
  count = 2
  name                = "vmnic-${count.index}"
  location            = azurerm_resource_group.myrg.location
  resource_group_name = azurerm_resource_group.myrg.name
  ip_configuration {
    name                          = "internal"
    subnet_id                     = azurerm_subnet.mysubnet.id
    private_ip_address_allocation = "Dynamic"
    public_ip_address_id = element(azurerm_public_ip.mypublicip[*].id, count.index)
  }
}
```

Step-08: c4-linux-virtual-machine.tf

- For Linux Virtual machine resource add `count=2`
- Associate Network interface to VM using `Element Function` and `Splat Expression`

```
# Resource: Azure Linux Virtual Machine
resource "azurerm_linux_virtual_machine" "mylinuxvm" {
  count = 2
  name                = "mylinuxvm-${count.index}"
  computer_name       = "devlinux-${count.index}" # Hostname of the VM
  resource_group_name = azurerm_resource_group.myrg.name
  location            = azurerm_resource_group.myrg.location
  size                = "Standard_DS1_v2"
  admin_username      = "azureuser"
  network_interface_ids = [element(azurerm_network_interface.myvmnic[*].id, count.index)]
  admin_ssh_key {
    username   = "azureuser"
    public_key = file("${path.module}/ssh-keys/terraform-azure.pub")
  }
  os_disk {
    name = "osdisk${count.index}"
    caching              = "ReadWrite"
    storage_account_type = "Standard_LRS"
    #disk_size_gb = 20
  }
  source_image_reference {
    publisher = "RedHat"
    offer     = "RHEL"
    sku       = "83-gen2"
    version   = "latest"
  }
  custom_data = filebase64("${path.module}/app-scripts/app1-cloud-init.txt")
}
```

Step-09: Execute Terraform Commands

```
# Change Directory
cd terraform-manifests-v2

# Terraform Initialize
terraform init

# Terraform Validate
terraform validate

# Terraform Plan
terraform plan

# Terraform Apply
terraform apply

# Verify
1. Azure Resource Group
2. Azure Virtual Network
3. Azure Subnet
4. Azure Public IP - 2 Resources created as specified in count
```

```
5. Azure Network Interface - 2 Resources created as specified in count
6. Azure Linux Virtual Machine - - 2 Resources created as specified in count

# Access Application
http://<PUBLIC_IP-1>
http://<PUBLIC_IP-2>
```

Step-10: Destroy Terraform Resources

```
# Destroy Terraform Resources
terraform destroy

# Remove Terraform Files
rm -rf .terraform*
rm -rf terraform.tfstate*
```

References

- Resources: Count Meta-Argument