Terraform

Wednesday, 21 May 2025 10:43

1. Tf it's using HCL language
2. Provider available for all public clouds. It's easy to communicate with all
3. Agentless
    a. Not like puppet, chef
4. It's configuration tool , Eg:- not like update, modify application
5. Providers (written json or yaml)
    a. Azure:-
        i. Resource manager
    b. AWS
        i. Cloud formation
    c. GCP:-
        i. Cloud formation

Root Module:-

Main.tf (The configuration file that contains the code to create infra)

Variables.tf (The file that contains input parameter)

Output.tf (The file that stores output parameter)

Three imp commands

Tf init

Download the all dependencies and create folder name as **.terraform** in that folder contain all kind of dependencies

Tf plan

This is will tell to us, what are the resources are going to be created

Tf apply

This is will do execution job. Means will start running the job

Communication

There are certain way for communicating with azure cloud.

1. Authenticate using azure cli

CI\CD

1. Using managed identity
2. Using Service principle and client certificate
3. Using service principle and client secret

HCl

Arguments:-

Structure:-

1. Provider block;

```
Terraform{
    required_providers{
     azurerm = {
       source = "hashicorp/azurerm"
       version = "2.46.0"
     }
    }
}
```

2. Provider config block

```
#configure the microsoft Azure provider
provider"azurerm"{
 features{}
}
```

3. Variable block

```
variable"availabilty_zone_names"{
  type = list(string)
  default = ["us-west-la"]
}
```

4. Resource block

```
#create a resource group
resource"azurerm_resource_group""example"{
   name =  "example-resources"
   location = "WestEurope"
}
```

5. Data block

```
Data "azrerm_storage_account" "example"{
Name = "images"
Resource_group_name = "storage"

}
```

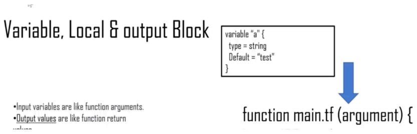
6. Output block

```
Output"storage_account_tier"{

     Value = data.azurerm_storage_account.example.aacount_tier

}
```

Terraform fmt

Checking syntax value correct or not

Variable, local & output Block



Variable, Local & output Block

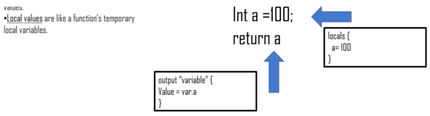
```
variable "a" {
 type = string
 Default = "test"
}
```

function main.tf (argument) {

- Input variables are like function arguments.
- Output values are like function return values

For building 4 resource

```
#Az provider
terraform {
  required_providers {
    azurerm = {
        source = "hashicorp/azurerm"
        version = "~3.0"
    }
  }
}
resource "azurerm_resource_group" "rg" {
    name = "samrsg"
    location = "India Central"

}
resource "azurerm_virtual_network" "vnet" {
    name                        = "samvnet"
    resource_group_name         = azurerm_resource_group.rg.name
    address_space               = ["10.0.0.0/16"]
    location                    = azurerm_resource_group.rg.location
}
resource "azurerm_subnet" "subnet" {
    name                        = "samsubnet"
    resource_group_name         = azurerm_resource_group.rg.name
    virtual_network_name        = azurerm_virtual_network.vnet.name
    address_prefixes            = ["10.0.1.0/24"]

}
resource "azurerm_network_security_group" "nsg" {
    name                        = "samnsg"
    resource_group_name         = azurerm_resource_group.rg.name
    location                    = azurerm_resource_group.rg.location
    security_rule {
        name                        = "SSH"
        priority                    = 1001
        direction                   = "Inbound"
        access                      = "Allow"
        protocol                    = "Tcp"
        source_port_range           = "*"
        destination_port_range      = "22"
        source_address_prefix       = "*"
        destination_address_prefix  = "*"
    }

}
resource "azurerm_network_interface_security_group_association"
"nic_nsg_association" {
    count                           = 4
    network_interface_id            =
azurerm_network_interface.nic[count.index].id
    network_security_group_id       = azurerm_network_security_group.nsg.id
}
resource "azurerm_network_interface" "nic" {
    count                           = 4 #for creating 4 vm nics
    name                            = "sam-nic-${count.index}"
    resource_group_name             = azurerm_resource_group.rg.name
    location                        = azurerm_resource_group.rg.location
ip_configuration{

    name                            = "internal"
    subnet_id                       = azurerm_subnet.subnet.id
    private_ip_address_allocation   = "Dynamic"
}
}
resource "tls_private_key" "ssh_key" {
    algorithm = "RSA"
    rsa_bits  = 2048
}
resource "azurerm_linux_virtual_machine" "vm" {
    count               = 4
    name                = "sam-vm_${count.index}"
    resource_group_name = azurerm_resource_group.rg.name
    location            = azurerm_resource_group.rg.location
    size                = "Standard_DS1_v2"
    admin_username      = "myadmin"
    disable_password_authentication = true
    network_interface_ids = [azurerm_network_interface.nic[count.index].id]
    os_disk {
        caching                 = "ReadWrite"
        storage_account_type    = "Standard_LRS"
    }
    source_image_refrence{
        publisher   = "Canonical"
        offer       = "ubuntuServer"
        sku         = "18.04-LTS"
        version     = "Latest"
    }
ssh_keys {
    key_data = tls_private_key.ssh_key.public_key_openssh
    username = "myadmin"
}
output "private_key" {
    value       = tls_private_key.ssh_key.private_key_pem
    sensitive   = true
}
}
```


values.
•Local values are like a function's temporary local variables.

Int a =100;
return a

```
locals {
  a= 100
}
```

```
output "variable" {
Value = var.a
}
```

A. Variable

Default value
Types;
primary types
String
Number
Bool
Complex types
Collection type; (allows multiple values of one other type to be grouped together as a single value)
List
Map
Set
Structural type. (allows multiple values of several distinct types to be grouped together as a single value
Object
tuple

Input can be define in main.tf or varibale.tf
-var to assign values
.tfvars

To assign the value into vaaribales
Environmental variable

Provider

The azure provider can be used to configure infra in MS Az using the ARM Api's

State in Terraform

Folder name **terraform.tfstate**

we can see the output and variables and main files
If we make the changes on current state file it'll create another that terraform.tfstate.backup

Creating resource group;

```
Resource "azurerm_resource_group" "example" {                #Resource is block and azurerm_resource_group is type
  name = "my-resources"                                      #argument
  location = "West Europe"                                   #argument
  depends_on
  count
  for_each
  provider
  lifecycle
}
```

Terraform destroy

Destroying the resources.

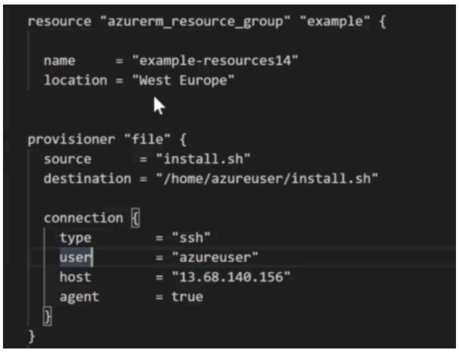
Provisioner

Local-exec

Execute the script on local machine

Remote-exec

Execute the script on remote machine



```
resource "azurerm_resource_group" "example" {

  name     = "example-resources14"
  location = "West Europe"


provisioner "file" {
  source      = "install.sh"
  destination = "/home/azureuser/install.sh"

  connection {
    type        = "ssh"
    user        = "azureuser"
    host        = "13.68.140.156"
    agent       = true
  }
}
```