**Terraform**

1. Terraform is an IAC tool, used to automate various infrastructure tasks.
2. We can automate the cloud resources instead of creating manually.
3. Terraform is a tool, command line program you run to define and make changes in your infrastructure.
4. It’s open source.

**Terraform lifecycle:**

1.**Terraform init** initializes the (local) Terraform environment. Usually executed only once per session.  
2. **Terraform plan** compares the Terraform state with the as-is state in the cloud, build and display an  
execution plan. This does not change the deployment (read-only).  
3. **Terraform apply** executes the plan. This potentially changes the deployment.  
4. **Terraform destroy** deletes all resources that are governed by this specific terraform environment.

**Install terraform in Linux:**

* wget <https://releases.hashicorp.com/terraform/1.3.5/terraform_1.3.5_linux_amd64.zip>

**unzip the downloaded file.**

* sudo unzip terraform\_1.3.5\_linux\_amd64.zip -d /usr/local/bin

**To verify if the Terraform directory is in the selected location, use the list command.**

* ls -l /usr/local/bin

**Confirm the installation is successful by running a terraform command.**

* terraform -version.

**VM and Cluster Creation:**

**Main.tf**

resource "azurerm\_virtual\_network" "example" {  
  name                = var.resource\_vnet  
  address\_space       = ["10.0.0.0/16"]  
  location            = var.resource\_group\_location  
  resource\_group\_name = var.resource\_groupname  
}

resource "azurerm\_subnet" "example" {  
  name                 = var.resource\_subnet  
  resource\_group\_name  = var.resource\_groupname  
  virtual\_network\_name = azurerm\_virtual\_network.example.name  
  address\_prefixes     = ["10.0.2.0/24"]  
}

resource "azurerm\_network\_interface" "example" {  
  name                = var.resource\_nic  
  location            = var.resource\_group\_location  
  resource\_group\_name = var.resource\_groupname

  ip\_configuration {  
    name                          = "internal"  
    subnet\_id                     = azurerm\_subnet.example.id  
    private\_ip\_address\_allocation = "Dynamic"  
  }  
}

resource "azurerm\_windows\_virtual\_machine" "example" {  
  name                = var.resource\_virtual\_machine  
  resource\_group\_name = var.resource\_groupname  
  location            = var.resource\_group\_location  
  size                = var.resource\_size  
  admin\_username      = var.resource\_adminusername  
  admin\_password      = var.resource\_adminpassword  
  network\_interface\_ids = [  
    azurerm\_network\_interface.example.id,  
  ]

  os\_disk {  
    caching              = "ReadWrite"  
    storage\_account\_type = "Standard\_LRS"  
  }

  source\_image\_reference {  
    publisher = "MicrosoftWindowsServer"  
    offer     = "WindowsServer"  
    sku       = "2016-Datacenter"  
    version   = "latest"  
  }  
}

resource "azurerm\_kubernetes\_cluster" "cluster"{  
    name = var.resource\_aks  
    location = var.resource\_group\_location  
    resource\_group\_name = var.resource\_groupname  
    dns\_prefix = "terrsform-kubernetes"  
    default\_node\_pool {  
        name = "default"  
        node\_count = "2"  
        vm\_size = var.resource\_size  
    }  
    identity {

        type = "SystemAssigned"

    }

}

**Variables.tf**

variable "resource\_virtual\_machine" {

  default     = "example-machine"  
  description = "adds VM"  
}

variable "resource\_group\_location" {  
  default     = "eastus"  
  description = "Location of the resource group."  
}

variable "resource\_groupname" {  
  default     = "test-rs-santhil"  
  description = "Resource group name that is unique in your Azure subscription."  
}

variable "resource\_aks {

  default     = "terraform-kubernetes"  
  description = "adds kubernetes service with forthmentioned name"  
}

variable "resource\_subnet" {

  default     = "internal"  
  description = "adds subnet"  
}

variable "resource\_vnet" {

  default     = "example-network"  
  description = "adds vnet"  
}

variable "resource\_nic" {

  default     = "example-nic"  
  description = "adds network interface"  
}

variable "resource\_size" {

  default     = "Standard\_F2"  
  description = "adds size to vm/aks"  
}

variable "resource\_adminusername" {

default     = "adminuser"  
  description = "adds admin name"  
}

variable "resource\_adminpassword" {

  default     = "P@$$w0rd1234!"  
  description = "adds admin password"  
}

**Commands to Run terraform scripts:**

* terraform init -upgrade.
* terraform plan -out main.tfplan.
* terraform apply main.tfplan.
* terraform destroy.
* az aks get-credentials --admin --name terraform-kubernetes --resource-group test-rs-santhil.