**Challenge-1**

A 3-tier environment is a common setup. Use a tool of your choosing/familiarity create these resources.

Solution: -

Note: - Terraform is used for above problem in which this configuration file is creating these resources: -

**Resource Group**

**Network security group**

**Vnet**

**Availability set**

**Network interface card**

**Virtual machine**

**Load balancer**

**Virtual machine scale set**

#Creating azure resource group

resource "azurerm\_resource\_group" "rg" {

name **=** "rg1"

location **=** "West US"

}

#Creating azure Network Security Group

resource "azurerm\_network\_security\_group" "sg" {

name **=** "acceptanceTestSecurityGroup1"

location **=** azurerm\_resource\_group.**rg**.**location**

resource\_group\_name **=** azurerm\_resource\_group.**rg**.**name**

}

#Creating azure virtual network

resource "azurerm\_virtual\_network" vnet" {

name **=** "virtualNetwork1"

location **=** azurerm\_resource\_group.**rg**.**location**

resource\_group\_name **=** azurerm\_resource\_group.**rg**.**name**

address\_space **=** ["10.0.0.0/16"]

dns\_servers **=** ["10.0.0.4", "10.0.0.5"]

subnet {

name **=** "subnet1"

address\_prefix **=** "10.0.1.0/24"

}

subnet {

name **=** "subnet3"

address\_prefix **=** "10.0.3.0/24"

security\_group **=** azurerm\_network\_security\_group.**example**.**id**

}

}

# Create availabilityset for the vm.

resource "azurerm\_availability\_set" "vm" {

count = var.availability\_set == true ? 1 : 0

name = "set1”

location = var.location

resource\_group\_name = data.azurerm\_resource\_group.main\_rg.name

platform\_fault\_domain\_count = var.fault\_domain\_count

platform\_update\_domain\_count = var.update\_domain\_count

managed = true

}

# Create a network interface for the availability set VMs

resource "azurerm\_network\_interface" "nic" {

name = "nic1"

count = var.vm\_count

location = var.location

resource\_group\_name = data.azurerm\_resource\_group.main\_rg.name

ip\_configuration {

name = "internal"

subnet\_id = data.azurerm\_subnet.subnet.id

private\_ip\_address\_allocation = var.ip\_address

}

}

# Creating virtual machine

resource "azurerm\_virtual\_machine" "vm" {

count = var.vm\_count

name = " vm1"

location = var.location

resource\_group\_name = data.azurerm\_resource\_group.main\_rg.name

vm\_size = var.size

storage\_image\_reference {

id = var.image\_reference\_id

}

storage\_os\_disk {

name = "vm\_os1"

caching = "ReadWrite"

create\_option = "FromImage"

managed\_disk\_type = var.storage\_account\_type

}

os\_profile {

computer\_name = " vm\_os"

admin\_username = var.rootadmin\_login

admin\_password = var.rootadmin\_password

}

os\_profile\_linux\_config {

disable\_password\_authentication = false

}

boot\_diagnostics {

enabled = true

storage\_uri = var.boot\_diagnostics

}

}

# creating internal load balancer for the VMSS

resource "azurerm\_lb" "vmsslb" {

name = "ms${substr(var.appname,0,6)}${substr(var.environment,0,4)}a01-lb"

location = var.location

resource\_group\_name = data.azurerm\_resource\_group.main\_rg.name

frontend\_ip\_configuration {

name = var.frontendadd

private\_ip\_address\_allocation = var.ip\_address

subnet\_id = data.azurerm\_subnet.subnet.id

}

# Create a internal load balancer backend address pool

resource "azurerm\_lb\_backend\_address\_pool" "bpepool" {

resource\_group\_name = data.azurerm\_resource\_group.main\_rg.name

loadbalancer\_id = azurerm\_lb.vmsslb.id

name = "BackEndAddressPool"

}

# Create a windows virtual machine scaleset

resource "azurerm\_virtual\_machine\_scale\_set" "vmss" {

name = "vmss1”

location = var.location

resource\_group\_name = data.azurerm\_resource\_group.main\_rg.name

upgrade\_policy\_mode = "Manual"

sku {

name = var.vmss\_size

tier = "Standard"

capacity = var.instance\_count

}

storage\_profile\_image\_reference {

id = var.image\_reference\_id

}

storage\_profile\_os\_disk {

name = ""

caching = "ReadWrite"

create\_option = "FromImage"

managed\_disk\_type = var.replication\_type

}

storage\_profile\_data\_disk {

lun = 0

caching = "ReadWrite"

create\_option = "Empty"

disk\_size\_gb = var.disk\_size\_gb

}

os\_profile {

computer\_name\_prefix = "vmss"

admin\_username = var.rootadmin\_login

admin\_password = var.rootadmin\_password

}

os\_profile\_linux\_config {

disable\_password\_authentication = false

}

boot\_diagnostics {

enabled = true

storage\_uri = var.boot\_diagnostics

}

network\_profile {

name = "terraformnetworkprofile"

primary = true

ip\_configuration {

name = "internal"

subnet\_id = data.azurerm\_subnet.subnet.id

load\_balancer\_backend\_address\_pool\_ids = [azurerm\_lb\_backend\_address\_pool.bpepool.id]

primary = true

}

}