Case Study Code/Script

Contents:

* Create a Resource Group and 1 VNET with 1 subnet and a security group with rules and 2 VMs in an Availability set and an attached Load Balancer with Load Balancing rule and Inbound NAT (SEA Region)
* Create 1 subnet inside an existing VNET and a security group with rules and 1 VM (SEA Region):
* Create a Resource Group and a VNET with 1 subnet and a security group with rules and 1 VM (EUS Region):
* Peering VNET Regions and test peering connection (EUS - SEA region)
* Create a Storage Account for EUS Resource Group
* Create a Storage Account for SEA Resource Group
* Get Storage Account Key
* Create Backup vault and enable backup for VM in EUS Region
* Create Backup vault and enable backup for 2 VMs in SEA Region
* Create a user id for subscription with Virtual Machine Administrator Login role
* Create a user id for EUS region with Backup contributor role
* Create Alert for VMs

1. Create a Resource Group and 1 VNET with 1 subnet and a security group with rules and 2 VMs in an Availability set and an attached Load Balancer with Load Balancing rule and Inbound NAT (SEA Region):

**PowerShell Script:**

#Create Resource Group

$RGName=Read-Host "Please type Resource group name"

$location=Read-Host "Please type location name"

$rg=New-AzResourceGroup -Name $RGName -Location $location

#Create Public Ip Address

$publicip = New-AzPublicIpAddress -ResourceGroupName $RGName -Name "corplbpub-ipp" -Location $location -AllocationMethod Static -Sku Standard

#CReate Load Balancer

$frontend = New-AzLoadBalancerFrontendIpConfig -Name "CorpLBFrontEnd" `

-PublicIpAddress $publicip

$backendAddressPool = New-AzLoadBalancerBackendAddressPoolConfig -Name "CorpLbBpool"

$probe = New-AzLoadBalancerProbeConfig -Name "CorpLBHProbe" `

-Protocol "http" -Port 80 -IntervalInSeconds 15 -ProbeCount 2 `

-RequestPath "index.html"

$inboundNatRule1 = New-AzLoadBalancerInboundNatRuleConfig -Name "CorpLBinboundNatRule1" `

-FrontendIPConfiguration $frontend -Protocol Tcp `

-FrontendPort 5580 -BackendPort 3389 -IdleTimeoutInMinutes 4

$inboundNatRule2 = New-AzLoadBalancerInboundNatRuleConfig -Name "CorpLBinboundNatRule2" `

-FrontendIPConfiguration $frontend -Protocol Tcp `

-FrontendPort 5581 -BackendPort 3389 -IdleTimeoutInMinutes 4

$lbrule = New-AzLoadBalancerRuleConfig -Name "CorpLBruleName" `

-FrontendIPConfiguration $frontend -BackendAddressPool $backendAddressPool `

-Probe $probe -Protocol "Tcp" -FrontendPort 80 `

-BackendPort 80 -IdleTimeoutInMinutes 4 -LoadDistribution SourceIP -DisableOutboundSNAT

$lb = New-AzLoadBalancer -Name "CorpLoadBalancer" -ResourceGroupName $RGName `

-Location $location -FrontendIpConfiguration $frontend `

-BackendAddressPool $backendAddressPool -Probe $probe `

-InboundNatRule $inboundNatRule1,$inboundNatRule2 `

-LoadBalancingRule $lbrule -Sku Standard

#CreateVNET and Subnet

$subnetConfig = New-AzVirtualNetworkSubnetConfig `

-Name "Corp\_Subnet" `

-AddressPrefix 10.10.1.0/24

$vnet = New-AzVirtualNetwork `

-ResourceGroupName $RGName `

-Location $location `

-Name "Corp\_Vnet" `

-AddressPrefix 10.10.0.0/16 `

-Subnet $subnetConfig

#Create Network Security Group

$nsgrule1= New-AzNetworkSecurityRuleConfig -Name corprule `

-Description corpnsginboundrule `

-Protocol TCP `

-SourcePortRange \* `

-DestinationPortRange 80,3389 `

-Access Allow -Priority 1000 `

-Direction Inbound `

-SourceAddressPrefix \* `

-DestinationAddressPrefix 10.10.1.0/24

$nsgrule2= New-AzNetworkSecurityRuleConfig -Name jumprule `

-Description jumpnsginboundrule `

-SourceAddressPrefix 10.10.2.0/24 `

-DestinationAddressPrefix 10.10.1.0/24 `

-Protocol TCP `

-SourcePortRange \* `

-DestinationPortRange 5985 `

-Access Allow -Priority 1010 `

-Direction Inbound

$corpnsg=New-AzNetworkSecurityGroup -Name Corp\_NSG -ResourceGroupName $RGName -Location $location -SecurityRules $nsgrule1,$nsgrule2

#Create Network Interface

for ($i=1; $i -le 2; $i++)

{

New-AzNetworkInterface `

-ResourceGroupName $RGName `

-Name Server$i `

-Location $location `

-NetworkSecurityGroup $corpnsg `

-Subnet $vnet.Subnets[0] `

-LoadBalancerBackendAddressPool $lb.BackendAddressPools[0]

}

#Create Availability Set

$aset=New-AzAvailabilitySet `

-Location $location `

-Name "CorpASet" `

-ResourceGroupName $RGName `

-Sku aligned `

-PlatformFaultDomainCount 2 `

-PlatformUpdateDomainCount 3

#Pass credentials

$cred=Get-Credential -Message "Please enter VM credential:"

#Create VM

for($i=1; $i -le 2; $i++){

$corpvm=New-AzVM -ResourceGroupName $RGName `

-Location $location `

-Name Server$i `

-Credential $cred `

-VirtualNetworkName $vnet.Name `

-SubnetName 'Corp\_Subnet' `

-AvailabilitySetName $aset.Name `

-image MicrosoftWindowsServer:WindowsServer:2016-Datacenter:latest `

-Size 'Standard\_DS1\_v2'

}

**User Input:**

Resource Group Name and Location Name.

1. Create 1 subnet inside an existing VNET and a security group with rules and 1 VM (SEA Region):

**PowerShell Script:**

$location='southeastasia'

$RGName='SEA-RG'

#Get VNET from ResourceGroup

$vnet=Get-AzVirtualNetwork -Name Corp\_Vnet -ResourceGroupName $RGName

#Create Subnet

$subnet= Add-AzVirtualNetworkSubnetConfig -Name 'Jump\_Subnet' `

-VirtualNetwork $vnet -AddressPrefix 10.10.2.0/24

$vnet | Set-AzVirtualNetwork

#Create network security group

$nsgrule= New-AzNetworkSecurityRuleConfig -Name rdprule `

-Description rdpinboundrule `

-Protocol TCP `

-SourcePortRange \* `

-DestinationPortRange 3389 `

-Access Allow -Priority 1000 `

-Direction Inbound `

-SourceAddressPrefix \* `

-DestinationAddressPrefix 10.10.2.0/24

$jumpnsg=New-AzNetworkSecurityGroup -Name Jump\_NSG `

-ResourceGroupName $rg.ResourceGroupName `

-Location $location -SecurityRules $nsgrule

#Create public ip address

$ip = @{

Name = 'jumppub-ip'

ResourceGroupName = $rg.ResourceGroupName

Location = $location

Sku = 'Standard'

AllocationMethod = 'Static'

IpAddressVersion = 'IPv4'

}

#Pass Credentials for JumpServer

$cred=Get-Credential -Message "Please enter VM credential:"

#Craete VM

$jumpvm=New-AzVM -ResourceGroupName $RGName `

-Location $location `

-Name JumpServer `

-Credential $cred `

-VirtualNetworkName $vnet.Name `

-SubnetName 'Jump\_Subnet' `

-PublicIpAddressName $ip.Name `

-SecurityGroupName $jumpnsg.Name `

-image MicrosoftWindowsServer:WindowsServer:2016-Datacenter:latest `

-Size 'Standard\_DS1\_v2'

1. Create a Resource Group and a VNET with 1 subnet and a security group with rules and 1 VM (EUS Region):

**Terraform Script:**

* **providers.tf File:**

**terraform {**

**required\_providers {**

**azurerm = {**

**source = "hashicorp/azurerm"**

**version = "=2.46.0"**

**}**

**}**

**}**

**provider "azurerm" {**

**features {}**

**}**

* **variables.tf File:**

**variable "region" {**

**default = "eastus"**

**}**

* **main.tf File:**

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

**name = "EUS-RG"**

**location = var.region**

**}**

**resource "azurerm\_virtual\_network" "rg" {**

**name = "Branch\_Vnet"**

**address\_space = ["10.20.0.0/16"]**

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

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

**}**

**resource "azurerm\_subnet" "rg" {**

**name = "Brunch\_Subnet"**

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

**virtual\_network\_name = azurerm\_virtual\_network.rg.name**

**address\_prefixes = ["10.20.1.0/24"]**

**}**

**resource "azurerm\_public\_ip" "pubip" {**

**name = "branchpub-ip"**

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

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

**allocation\_method = "Static"**

**}**

**resource "azurerm\_network\_interface" "rg" {**

**name = "BranchVm-NIC"**

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

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

**ip\_configuration {**

**name = "internal"**

**subnet\_id = azurerm\_subnet.rg.id**

**private\_ip\_address\_allocation = "Dynamic"**

**public\_ip\_address\_id = azurerm\_public\_ip.pubip.id**

**}**

**}**

**resource "azurerm\_network\_security\_group" "myterraformnsg" {**

**name = "BranchNsg"**

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

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

**security\_rule {**

**name = "PortRule"**

**priority = 1000**

**direction = "Inbound"**

**access = "Allow"**

**protocol = "Tcp"**

**source\_port\_range = "\*"**

**destination\_port\_range = "80"**

**source\_address\_prefix = "\*"**

**destination\_address\_prefix = "10.20.1.0/24"**

**}**

**}**

**resource "azurerm\_network\_interface\_security\_group\_association" "nsg-nic" {**

**network\_interface\_id = azurerm\_network\_interface.rg.id**

**network\_security\_group\_id = azurerm\_network\_security\_group.myterraformnsg.id**

**}**

**resource "azurerm\_windows\_virtual\_machine" "winvm" {**

**name = "Server11"**

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

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

**network\_interface\_ids = [azurerm\_network\_interface.rg.id]**

**size = "Standard\_D2s\_v3"**

**os\_disk {**

**name = "OsDisk"**

**caching = "ReadWrite"**

**storage\_account\_type = "Standard\_LRS"**

**}**

**source\_image\_reference {**

**publisher = "MicrosoftWindowsServer"**

**offer = "WindowsServer"**

**sku = "2016-Datacenter"**

**version = "latest"**

**}**

**computer\_name = "Server11"**

**admin\_username = "vmadmin"**

**admin\_password = "Password@12345"**

**}**

* **output.tf File:**

**output "PublicIP Address of Server11" {**

**value = azurerm\_public\_ip.pubip.ip\_address**

**description = "PublicIP Address of Server11"**

**}**

1. Peering VNET Regions and test peering connection (EUS - SEA region):

**PowerShell Script:**

#Get Vnet

$corp\_vnet=Get-AzVirtualNetwork -Name Corp\_Vnet -ResourceGroupName SEA-RG

$branch\_vnet=Get-AzVirtualNetwork -Name Branch\_Vnet -ResourceGroupName EUS-RG

#Peering Vnet

Add-AzVirtualNetworkPeering `

-Name Corp\_Vnet-Branch\_Vnet `

-VirtualNetwork $corp\_vnet `

-RemoteVirtualNetworkId $branch\_vnet.Id

Add-AzVirtualNetworkPeering `

-Name Branch\_Vnet-Corp\_Vnet `

-VirtualNetwork $branch\_vnet `

-RemoteVirtualNetworkId $corp\_vnet.Id

#Check peering state

Get-AzVirtualNetworkPeering `

-ResourceGroupName SEA-RG `

-VirtualNetworkName Corp\_Vnet `

| Select PeeringState

1. Create a Storage Account for EUS Resource Group:

**Terraform Script:**

* **providers.tf File:**

**terraform {**

**required\_providers {**

**azurerm = {**

**source = "hashicorp/azurerm"**

**version = "=2.46.0"**

**}**

**}**

**}**

**provider "azurerm" {**

**features {}**

**}**

* **variables.tf File:**

**variable "region" {**

**default = "eastus"**

**}**

**variable "rgroup" {**

**default = "EUS-RG"**

**}**

**variable "sku" {**

**default = "ZRS"**

**}**

**variable "name" {**

**default = "bramchstg"**

**}**

* **main.tf File:**

**resource "azurerm\_storage\_account" "storageaccount" {**

**name = var.name**

**resource\_group\_name = var.rgroup**

**location = var.region**

**account\_tier = "Standard"**

**account\_replication\_type = var.sku**

**}**

* **output.tf File:**

**output "detailsofdeploymnt" {**

**value = azurerm\_storage\_account.storageaccount.primary\_blob\_endpoint**

**description = "storage account endpont URL"**

**}**

1. Create a Storage Account for SEA Resource Group:

**Terraform Script:**

* **providers.tf File:**

**terraform {**

**required\_providers {**

**azurerm = {**

**source = "hashicorp/azurerm"**

**version = "=2.46.0"**

**}**

**}**

**}**

**provider "azurerm" {**

**features {}**

**}**

* **variables.tf File:**

**variable "region" {**

**default = "southeastasia"**

**}**

**variable "rgroup" {**

**default = "SEA-RG"**

**}**

**variable "sku" {**

**default = "GRS"**

**}**

**variable "name" {**

**default = "corpstg"**

**}**

* **main.tf File:**

**resource "azurerm\_storage\_account" "storageaccount" {**

**name = var.name**

**resource\_group\_name = var.rgroup**

**location = var.region**

**account\_tier = "Standard"**

**account\_replication\_type = var.sku**

**}**

* **output.tf File:**

**output "detailsofdeploymnt" {**

**value = azurerm\_storage\_account.storageaccount.primary\_blob\_endpoint**

**description = "storage account endpont URL"**

**}**

1. Get Storage Account Key:

**PowerShell Script:**

$RG\_name = Read-Host "Please enter Resource group name"

$Storage\_name = Read-Host "Please enter storage account name"

Get-AzStorageAccountKey -ResourceGroupName $RG\_name -AccountName $Storage\_name |Format-List

**User Input:**

Resource Group name and Storage account name.

1. Create Backup vault and enable backup for VM in EUS Region:

**Azure CLI Command:**

**az backup vault create --resource-group EUS-RG --name branchbackupvault --location eastus**

**az backup vault backup-properties set --name branchbackupvault --resource-group EUS-RG --backup-storage-redundancy "GeoRedundant"**

**az backup protection enable-for-vm --resource-group EUS-RG --vault-name branchbackupvault --vm server11 --policy-name DefaultPolicy**

1. Create Backup vault and enable backup for 2 VMs in SEA Region:

**Azure CLI Command:**

**az backup vault create --resource-group SEA-RG --name corpbackupvault --location southeastasia**

**az backup vault backup-properties set --name corpbackupvault --resource-group SEA-RG --backup-storage-redundancy "GeoRedundant"**

**az backup protection enable-for-vm --resource-group SEA-RG --vault-name corpbackupvault --vm server1 --policy-name DefaultPolicy**

**az backup protection enable-for-vm --resource-group SEA-RG --vault-name corpbackupvault --vm server2 --policy-name DefaultPolicy**

1. Create a user id for subscription with VMAdmin Login role:

**PowerShell Script:**

$userid= Read-Host "Please enter the userid"

$Password= Read-host "Please enter the password" | ConvertTo-SecureString -AsPlainText -Force

New-AzADUser -DisplayName $userid -UserPrincipalName $userid@anirbanroyaz104outlook.onmicrosoft.com -Password $Password -MailNickname $userid

New-AzRoleAssignment -SignInName $userid@anirbanroyaz104outlook.onmicrosoft.com `

-RoleDefinitionName "Virtual Machine Administrator Login" `

-Scope "/subscriptions/ea090fd4-85b2-4ac7-bdde-480b9c95be42"

**User Input:**

User id and Password.

1. Create a user id for EUS region with Backup contributor role:

**PowerShell Script:**

$userid= Read-Host "Please enter the userid"

$Password= Read-host "Please enter the password" | ConvertTo-SecureString -AsPlainText -Force

New-AzADUser -DisplayName $userid -UserPrincipalName $userid@anirbanroyaz104outlook.onmicrosoft.com -Password $Password -MailNickname $userid

New-AzRoleAssignment -SignInName $userid@anirbanroyaz104outlook.onmicrosoft.com `

-RoleDefinitionName "Backup Contributor" `

-Scope "/subscriptions/ea090fd4-85b2-4ac7-bdde-480b9c95be42/resourceGroups/EUS-RG"

**User Input:**

User id and Password.

1. Create Alert for VMs:

**PowerShell Script:**

$Criteria=New-AzMetricAlertRuleV2Criteria -MetricName "Percentage CPU" -MetricNameSpace "Microsoft.Compute/virtualMachines" -TimeAggregation Average -Operator GreaterThan -Threshold 80

Add-AzMetricAlertRuleV2 -Name CorpVMAlert -ResourceGroupName SEA-RG `

-WindowSize 00:05:00 -Frequency 00:05:00 `

-TargetResourceScope '/subscriptions/ea090fd4-85b2-4ac7-bdde-480b9c95be42/resourceGroups/SEA-RG/providers/Microsoft.Compute/virtualMachines/Server1','/subscriptions/ea090fd4-85b2-4ac7-bdde-480b9c95be42/resourceGroups/SEA-RG/providers/Microsoft.Compute/virtualMachines/Server2' `

-TargetResourceType 'Microsoft.Compute/virtualMachines' `

-TargetResourceRegion 'southeastasia' `

-Condition $Criteria -Severity 2