Lab 06 - Implement Traffic Management

Objectives

In this lab, we will:

* Task 1: Provision the lab environment
* Task 2: Configure the hub and spoke network topology
* Task 3: Test transitivity of virtual network peering
* Task 4: Configure routing in the hub and spoke topology
* Task 5: Implement Azure Load Balancer
* Task 6: Implement Azure Application Gateway

## Task 1: Provision the lab environment

In this task, we will deploy four virtual machines into the same Azure region. The first two will reside in a hub virtual network, while each of the remaining two will reside in a separate spoke virtual network.

For us to do this, first we need to sign in to the Azure portal. Then, in the Azure portal, we will open the **Azure Cloud Shell** and select **PowerShell**. There we will upload the two .json files, that were provided, in the Cloud shell directory. These files are: **az104-06-vms-loop-template.json** and **az104-06-vms-loop-parameters.json**.

After we upload the two files, we can start with the creation of the resource group that will be hosting the lab environment. We run the following code:

* $location = 'westus'

For the resource group name we run:

* $rgName = 'az104-06-rg1'

To create the resource group in our desired location:

* New-AzResourceGroup -Name $rgName -Location $location

To create the three virtual networks and four Azure VMs into them by using the template and parameter files we uploaded:

* New-AzResourceGroupDeployment `

-ResourceGroupName $rgName `

-TemplateFile $HOME/az104-06-vms-loop-template.json `

-TemplateParameterFile $HOME/az104-06-vms-loop-parameters.json

PS 
/home/ana> New-AzResourceGroupDep10yment 
-ResourceGroupName $rgName ' 
-TemplateFi1e $HOME/az1ø4-e6-vms-100p-template. json 
-TemplateparameterFiIe $HOME/az1e4-e6-vms-Ioop-parameters. 
DeploymentName 
ResourceGroupName 
ProvisioningState 
Timestamp 
Mode 
TemplateLink 
Parameters 
Outputs 
. az1ø4-e6-vms-100p-temp1ate 
. aziØ4-Ø6-rg1 
. Succeeded 
3/25/2e23 7:32:55 PM 
Incremental 
Name 
vmsize 
vmName 
vmCount 
adminusername 
adminpassword 
Type 
String 
String 
Int 
String 
Securestring 
Value 
"Standard D2s v3" 
" az1Ø4-e6-vm" 
4 
"Student" 
null 
DeploymentDebugLogLeve1 : 

Next, we run the following to install the Network Watcher extension on the Azure VMs deployed in the previous step.

* $rgName = 'az104-06-rg1'

$location = (Get-AzResourceGroup -ResourceGroupName $rgName).location

$vmNames = (Get-AzVM -ResourceGroupName $rgName).Name

foreach ($vmName in $vmNames) {

Set-AzVMExtension `

-ResourceGroupName $rgName `

-Location $location `

-VMName $vmName `

-Name 'networkWatcherAgent' `

-Publisher 'Microsoft.Azure.NetworkWatcher' `

-Type 'NetworkWatcherAgentWindows' `

-TypeHandlerVersion '1.4'

}

PS 
PS 
/home/ana> foreach ($vmName in $vmNames) { 
PS 
Set-AzVMExtension 
-ResourceGroupName $rgName 
-Location $location 
-VMName $vmName 
-Name ' networkWatcherAgent ' 
-publisher 'Microsoft .Azure. NetworkWatcher' 
-Type ' NetworkWatcherAgentWindows 
-TypeHand1erVersion '1.4' 
OK OK 
/home/ana> $rgName = 
/home/ana> $location = 
/home/ana> $vmNames 
(Get-AzResourceGroup -ResourceGroupName $rgName) . location 
(Get-AzVM -ResourceGroupName $rgName) . Name 
Requestld IsSuccessStatusCode StatusCode 
True 
True 
True 
True 
OK 
0K 
OK 
Reasonphrase 
OK 
OK 
OK 

## Task 2: Configure the hub and spoke network topology

In this task, we will configure local peering between the virtual networks we deployed in the previous tasks in order to create a hub and spoke network topology.

We review the virtual networks we created in the previous task. We get the Resource ID property from **az104-06-vnet2** and **az104-06-vnet3**. We will need this ID’s later on.

Next, we will create Peerings for **az104-06-vnet01** using the following settings:

| Setting | Value |
| --- | --- |
| This virtual network: Peering link name | **az104-06-vnet01\_to\_az104-06-vnet2** |
| Traffic to remote virtual network | **Allow (default)** |
| Traffic forwarded from remote virtual network | **Block traffic that originates from outside this virtual network** |
| Virtual network gateway | **None (default)** |
| Remote virtual network: Peering link name | **az104-06-vnet2\_to\_az104-06-vnet01** |
| Virtual network deployment model | **Resource manager** |
| I know my resource ID | enabled |
| Resource ID | the value of resourceID parameter of **az104-06-vnet2** you recorded earlier in this task |
| Traffic to remote virtual network | **Allow (default)** |
| Traffic forwarded from remote virtual network | **Allow (default)** |
| Virtual network gateway | **None (default)** |

This step establishes two local peerings - one from az104-06-vnet01 to az104-06-vnet2 and the other from az104-06-vnet2 to az104-06-vnet01.

We create another Peering for **az104-06-vnet01** with following settings:

| Setting | Value |
| --- | --- |
| This virtual network: Peering link name | **az104-06-vnet01\_to\_az104-06-vnet3** |
| Traffic to remote virtual network | **Allow (default)** |
| Traffic forwarded from remote virtual network | **Block traffic that originates from outside this virtual network** |
| Virtual network gateway | **None (default)** |
| Remote virtual network: Peering link name | **az104-06-vnet3\_to\_az104-06-vnet01** |
| Virtual network deployment model | **Resource manager** |
| I know my resource ID | enabled |
| Resource ID | the value of resourceID parameter of **az104-06-vnet3** you recorded earlier in this task |
| Traffic to remote virtual network | **Allow (default)** |
| Traffic forwarded from remote virtual network | **Allow (default)** |
| Virtual network gateway | **None (default)** |

This step establishes two local peerings - one from az104-06-vnet01 to az104-06-vnet3 and the other from az104-06-vnet3 to az104-06-vnet01.

## Task 3: Test transitivity of virtual network peering

In this task, we will test transitivity of virtual network peering by using Network Watcher.

On the Network Watcher - Connection troubleshoot we initiate a check with the following settings:

| Setting | Value |
| --- | --- |
| Subscription | the name of the Azure subscription you are using in this lab |
| Resource group | **az104-06-rg1** |
| Source type | **Virtual machine** |
| Virtual machine | **az104-06-vm0** |
| Destination | **Specify manually** |
| URI, FQDN or IPv4 | **10.62.0.4** |
| Protocol | **TCP** |
| Destination Port | **3389** |

Here, the given IP Address is the private IP address of az104-06-vm2.

The screenshot is from the Connection troubleshoot for the above settings.

Diagnostic details 
azi04-06.vmo 
Diagnostic tests 
Test 
Connectivity Test 
NSG Outbound (fro.„ 
Next Hop (from sour.„ 
Hop by hop details 
Name 
Destination 
10.62.0.4 
Status 
Success 
Success 
Success 
Status 
Success 
Success 
Details 
Probes Sent: 66 ,Probes Failed: O 
Avg Latency: 1 ms 
Min Latency: I ms 
Min Latency: 3 ms 
Outbound communication from 
source is allowed 
Next Hop Type. VirtualNetworkPeering 
Route Table Id: System Route 
IP address 
10.60.0.4 
10.62.OA 
Next hop 
10.62.0.4 
None 
None 
None 
2 
azi04-06 
azi04 06 
•vmO 
nic2 

~ 小 90 0 e 
0 
0 山 90- 〕 e 
0 

We run another Network Watcher - Connection troubleshoot with the following settings:

| Setting | Value |
| --- | --- |
| Subscription | the name of the Azure subscription you are using in this lab |
| Resource group | **az104-06-rg1** |
| Source type | **Virtual machine** |
| Virtual machine | **az104-06-vm0** |
| Destination | **Specify manually** |
| URI, FQDN or IPv4 | **10.63.0.4** |
| Protocol | **TCP** |
| Destination Port | **3389** |

Here, the given IP Address is the private IP address of az104-06-vm3.

The screenshot is from the Connection troubleshoot for the above settings.

Diagnostic details 
Source 
azl 04-06-vmO 
Diagnostic tests 
Connectivity Test 
NSG outbound (fro.- 
Next Hop (from sour... 
Hop by hop details 
Name 
azi04-06-vmo 
azi04-06.nic3 
Destination 
10.63.0A 
Success 
Success 
Success 
Success 
Success 
Details 
Probes Sent: 66 ,Probes Failed: O 
Avg Latency: 1 ms 
Min Latency: I ms 
Min Latency: 1 ms 
Outbound communication from 
source is allowed 
Next Hop Type: VirtualNetworkPeering 
Route Table Id: System Route 
Suggestions 
IP address 
10.60.0.4 
10.63.0.4 
Next hop 
10.6304 

az104-06-vmO 
az104-06-nic3 

In the above Connections troubleshoots, both checks were successful.

Next we will perform another Connection troubleshoot with the following settings:

| Setting | Value |
| --- | --- |
| Subscription | the name of the Azure subscription you are using in this lab |
| Resource group | **az104-06-rg1** |
| Source type | **Virtual machine** |
| Virtual machine | **az104-06-vm2** |
| Destination | **Specify manually** |
| URI, FQDN or IPv4 | **10.63.0.4** |
| Protocol | **TCP** |
| Destination Port | **3389** |

The screenshot from the above check:

Diagnostic details 
Source 
az 104-06-vm2 
Diagnostic tests 
Connectivity Test 
NSG Outbound (fro... 
Next Hop (from sour.„ 
Hop by hop details 
Name 
az 104-06-vm2 
Destination (10„. 
Destina tion 
10630.4 
Status 
Fail 
Fail 
Success 
Fail 
Info 
Details 
Probes Sent: O ,Probes Failed: O 
There are failed tests in the following 
NSGs: 
azi04-06-nsg2 
Next Hop Type: None 
Route Table 'd: System Route 
Suggest ions 
Go to VM > Update 
the networking rule 
Read docs 
None 
I P address 
1062.04 
10.6304 
Next hop 
10.6304 

o 
az104-06-vm2 
1062.04 
Destination 
10.63.0.4 

We can see that this check was not successful. This is like this because the two virtual networks are not peered with each other.

## Task 4: Configure routing in the hub and spoke topology

In this task, we will configure and test routing between the two spoke virtual networks by enabling IP forwarding on the network interface of the az104-06-vm0 virtual machine, enabling routing within its operating system, and configuring user-defined routes on the spoke virtual network.

First we enable IP forwarding for the az104-06-nic0 of the az104-06-vm0. This will enable the az104-06-vm0 to function as a router. After this we will configure the operating system az104-06-vm0 to support routing.

From az104-06-vm0 we go to RunPowerShellScript, and we run the following:

* Install-WindowsFeature RemoteAccess -IncludeManagementTools

After we get confirmation that the command completed successfully, we open again RunPowerShellScript and run the following commands:

* Install-WindowsFeature -Name Routing -IncludeManagementTools -IncludeAllSubFeature
* Install-WindowsFeature -Name "RSAT-RemoteAccess-Powershell"
* Install-RemoteAccess -VpnType RoutingOnly
* Get-NetAdapter | Set-NetIPInterface -Forwarding Enabled

We wait for the command to finish.

After the command finishes, we configure user defined routes on the spoke virtual networks. In Route tables we create new rout table with the following settings:

| Setting | Value |
| --- | --- |
| Subscription | the name of the Azure subscription you are using in this lab |
| Resource group | **az104-06-rg1** |
| Location | the name of the Azure region in which you created the virtual networks |
| Name | **az104-06-rt23** |
| Propagate gateway routes | **No** |

On the az104-06-rt23 route table, we add Routes and Subnets with the following settings:

* For the Routes:

| Setting | Value |
| --- | --- |
| Route name | **az104-06-route-vnet2-to-vnet3** |
| Address prefix destination | **IP Addresses** |
| Destination IP addresses/CIDR ranges | **10.63.0.0/20** |
| Next hop type | **Virtual appliance** |
| Next hop address | **10.60.0.4** |

* For the Subnets:

| Setting | Value |
| --- | --- |
| Virtual network | **az104-06-vnet2** |
| Subnet | **subnet0** |

After we create the Routes and Subnets for az104-06-rt23 we create new Rout tables with the following settings:

| Setting | Value |
| --- | --- |
| Subscription | the name of the Azure subscription you are using in this lab |
| Resource group | **az104-06-rg1** |
| Region | the name of the Azure region in which you created the virtual networks |
| Name | **az104-06-rt32** |
| Propagate gateway routes | **No** |

After this route is created, we create Routes and Subnets, like for the previous route.

Settings for Routes:

| Setting | Value |
| --- | --- |
| Route name | **az104-06-route-vnet3-to-vnet2** |
| Address prefix destination | **IP Addresses** |
| Destination IP addresses/CIDR ranges | **10.62.0.0/20** |
| Next hop type | **Virtual appliance** |
| Next hop address | **10.60.0.4** |

Settings for Subnets:

| Setting | Value |
| --- | --- |
| Virtual network | **az104-06-vnet3** |
| Subnet | **subnet0** |

When we are finished with the creation of the az104-06-rt32 route, we head to Network Watcher - Connection troubleshoot to initiate a check with the following settings:

| Setting | Value |
| --- | --- |
| Subscription | the name of the Azure subscription you are using in this lab |
| Resource group | **az104-06-rg1** |
| Source type | **Virtual machine** |
| Virtual machine | **az104-06-vm2** |
| Destination | **Specify manually** |
| URI, FQDN or IPv4 | **10.63.0.4** |
| Protocol | **TCP** |
| Destination Port | **3389** |

When the check is finished it should say that it was successful.

Here is a screenshot of the check:

Source 
azi04-06.vm2 
Diagnostic tests 
Test 
Connectivity Test 
NSG Outbound (fro.„ 
Next Hop (from sour.„ 
Hop by hop details 
Desti nation 
10.63.0.4 
Success 
Success 
Success 
Success 
Success 
Success 
Details 
Probes Sent 66 ,Probes Failed: O 
Avg Latency. 2 ms 
Min Latency: 2 ms 
Min Latency: 4 ms 
Outbound communication from 
source is allowed 
Next Hop Type: VirtualAppliance 
Next Hop IP•. 10.60.0.4 
Sugge 
None 
None 
None 
Name 
azi04-06 
azi04-06 
azi04-06 
-vm2 
-nicO 
-nic3 
IP address 
10.62.0.4 
1060.04 
10.6304 
Next hop 
10.6004 
10.6304 

az104-0 & vm2 
az104•06•nicO 
dz104 • 0 & nic3 

## Task 5: Implement Azure Load Balancer

In this task, we will implement an Azure Load Balancer in front of the two Azure virtual machines in the hub virtual network.

In Azure portal we head to Load balancer, and we create new load balancer. For this load balancer we will use the followings settings:

| Setting | Value |
| --- | --- |
| Subscription | the name of the Azure subscription you are using in this lab |
| Resource group | **az104-06-rg4** (if necessary create) |
| Name | **az104-06-lb4** |
| Region | name of the Azure region into which you deployed all other resources in this lab |
| SKU | **Standard** |
| Type | **Public** |
| Tier | **Regional** |

Next, we configure Frontend IP with the following settings:

| Setting | Value |
| --- | --- |
| Name | **az104-06-pip4** |
| IP version | IPv4 |
| IP type | IP address |
| Public IP address | **Create new** |
| Availability zone | **No Zone** |

After this we go to Backend pools and put the following settings:

| Setting | Value |
| --- | --- |
| Name | **az104-06-lb4-be1** |
| Virtual network | **az104-06-vnet01** |
| Backend Pool Configuration | **NIC** |
| IP Version | **IPv4** |
| Click **Add** to add a virtual machine |  |
| az104-06-vm0 | **check the box** |
| az104-06-vm1 | **check the box** |

After this we go to Inbound rules with these settings:

| Setting | Value |
| --- | --- |
| Name | **az104-06-lb4-lbrule1** |
| IP Version | **IPv4** |
| Frontend IP Address | **az104-06-pip4** |
| Backend pool | **az104-06-lb4-be1** |
| Protocol | **TCP** |
| Port | **80** |
| Backend port | **80** |
| Health probe | **Create new** |
| Name | **az104-06-lb4-hp1** |
| Protocol | **TCP** |
| Port | **80** |
| Interval | **5** |
| Unhealthy threshold | **2** |
| Close the create health probe window | **OK** |
| Session persistence | **None** |
| Idle timeout (minutes) | **4** |
| TCP reset | **Disabled** |
| Floating IP | **Disabled** |
| Outbound source network address translation (SNAT) | **Recommended** |

After we finish with all this, we finally click on create to create the Load balancer.

From Frontend IP configuration from the Load Balancer resource page we copy the IP address. In the web browser we navigate to this IP address (in our case 13.86.211.56). We should see messages Hello World from az104-06-vm0 or Hello World from az104-06-vm1.

We should refresh the window more often to verify the message changes to the other virtual machine or open a new browser window in InPrivate mode. This demonstrates the load balancer rotating through the virtual machines.

Here are screenshots displaying the two messages.

C A Not secure | 13.86211.56 
Hello World from azl 04-06-vml 

A Notsecure | 13.86.211.56 
Hello World from az 104-06-vmO 

## Task 6: Implement Azure Application Gateway

In this task, we will implement an Azure Application Gateway in front of the two Azure virtual machines in the spoke virtual networks.

In Virtual networks for az104-06-vnet01 we add Subnet with these settings:

| Setting | Value |
| --- | --- |
| Name | **subnet-papaws** |
| Subnet address range | **10.60.3.224/27** |

Next, we create Application Gateways. In the Basic tab we specify the following settings:

| Setting | Value |
| --- | --- |
| Subscription | the name of the Azure subscription you are using in this lab |
| Resource group | **az104-06-rg5** (create new) |
| Application gateway name | **az104-06-appgw5** |
| Region | name of the Azure region into which you deployed all other resources in this lab |
| Tier | **Standard V2** |
| Enable autoscaling | **No** |
| Instance count | **2** |
| Availability zone | **None** |
| HTTP2 | **Disabled** |
| Virtual network | **az104-06-vnet01** |
| Subnet | **subnet-appgw (10.60.3.224/27)** |

On Frontends tab we specify the following settings:

| Setting | Value |
| --- | --- |
| Frontend IP address type | **Public** |
| Public IP address | **Add new** |
| Name | **az104-06-pip5** |
| Availability zone | **None** |

On Backends tab we add a backend pool. Settings for the pool:

| Setting | Value |
| --- | --- |
| Name | **az104-06-appgw5-be1** |
| Add backend pool without targets | **No** |
| IP address or FQDN | **10.62.0.4** |
| IP address or FQDN | **10.63.0.4** |

The targets represent the private IP addresses of virtual machines in the spoke virtual networks az104-06-vm2 and az104-06-vm3.

On the Configuration tab, we add a routing rule. Settings for the rule:

| Setting | Value |
| --- | --- |
| Rule name | **az104-06-appgw5-rl1** |
| Priority | **10** |
| Listener name | **az104-06-appgw5-rl1l1** |
| Frontend IP | **Public** |
| Protocol | **HTTP** |
| Port | **80** |
| Listener type | **Basic** |
| Error page url | **No** |

After this we switch to the Backend targets tab and give the following settings:

| Setting | Value |
| --- | --- |
| Target type | **Backend pool** |
| Backend target | **az104-06-appgw5-be1** |
| Backend settings | **Add new** |
| Backend settings name | **az104-06-appgw5-http1** |
| Backend protocol | **HTTP** |
| Backend port | **80** |
| Additional settings | **take the defaults** |
| Host name | **take the defaults** |

After all settings are done we go to Review + Create tab and click create.

On Application Gateways we find and click az104-06-appgw5 and from there we copy the value of the Frontend public IP address (in our case 20.228.82.143). In another browser window we navigate to the IP address. The window should display the messages **Hello World from az104-06-vm2** or **Hello World from az104-06-vm3.** We refresh the window to verify the message changes to the other virtual machine.

Here are screenshots from the message in the two virtual machines.

C A Not secure | 20.22882.143 
Hello World from az104-06-vm2 

C A Not secure | 20.228.82.143 
Hello World from azl 04-06-vm3 