Lab 06 - Implement Traffic Management
Student lab manual

Lab scenario

You were tasked with testing managing network traffic targeting Azure virtual machines in the hub and spoke network topology, which Contoso considers implementing in its Azure environment (instead of creating the mesh topology, which you tested in the previous lab). This testing needs to include implementing connectivity between spokes by relying on user defined routes that force traffic to flow via the hub, as well as traffic distribution across virtual machines by using layer 4 and layer 7 load balancers. For this purpose, you intend to use Azure Load Balancer (layer 4) and Azure Application Gateway (layer 7).

> **Note**: This lab, by default, requires total of 8 vCPUs available in the Standard_Dsv3 series in the region you choose for deployment, since it involves deployment of four Azure VMs of Standard_D2s_v3 SKU. If your students are using trial accounts, with the limit of 4 vCPUs, you can use a VM size that requires only one vCPU (such as Standard_B1s).

Objectives

In this lab, you 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

Estimated timing: 60 minutes

Instructions

Exercise 1

Task 1: Provision the lab environment

In this task, you 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.

1. Sign in to the Azure portal.
2. In the Azure portal, open the **Azure Cloud Shell** by clicking on the icon in the top right of the Azure Portal.
3. If prompted to select either **Bash** or **PowerShell**, select **PowerShell**.

   > **Note**: If this is the first time you are starting **Cloud Shell** and you are presented with the **You have no storage mounted** message, select the subscription you are using in this lab, and click **Create storage**.

4. In the toolbar of the Cloud Shell pane, click the **Upload/Download files** icon, in the drop-down menu, click **Upload** and upload the files **\Allfiles\Labs\06\az104-06-vms-loop-template.json** and **\Allfiles\Labs\06\az104-06-vms-loop-parameters.json** into the Cloud Shell home directory.
5. From the Cloud Shell pane, run the following to create the first resource group that will be hosting the lab environment (replace the `[Azure_region]` placeholder with the name of an Azure region where you intend to deploy Azure virtual machines)(you can use the "(Get-AzLocation).Location" cmdlet to get the region

list):

```
$location = '[Azure_region]'

$rgName = 'az104-06-rg1'

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

6. From the Cloud Shell pane, run the following to create the three virtual networks and four virtual machines into them by using the template and parameter files you uploaded:

```
New-AzResourceGroupDeployment `
   -ResourceGroupName $rgName `
   -TemplateFile $HOME/az104-06-vms-loop-template.json `
   -TemplateParameterFile $HOME/az104-06-vms-loop-parameters.json `
   -AsJob
```

> **Note**: Wait for the deployment to complete before proceeding to the next task. This should take about 5 minutes.

> **Note**: To verify the status of the deployment, you can examine the properties of the resource group you created in this task.

7. Close the Cloud Shell pane.

Task 2: Configure the hub and spoke network topology

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

1. In the Azure portal, search for and select **Virtual networks**.
2. Review the virtual networks you created in the previous task.

> **Note**: The template you used for deployment of the three virtual networks ensures that the IP address ranges of the three virtual networks do not overlap.

3. In the list of virtual networks, click **az104-06-vnet01**.
4. On the **az104-06-vnet01** virtual network blade, in the **Settings** section, click **Peerings** and then click **+ Add**.
5. Add a peering with the following settings (leave others with their default values) and click Add:

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**

Setting	Value
Virtual network deployment model	**Resource manager**
Subscription	the name of the Azure subscription you are using in this lab
Virtual network	**az104-06-vnet2**
Traffic to remote virtual network	**Allow (default)**
Traffic forwarded from remote virtual network	**Allow (default)**
Virtual network gateway	**None (default)**

> **Note**: Wait for the operation to complete.

> **Note**: 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.

> **Note**: **Allow forwarded traffic** needs to be enabled in order to facilitate routing between spoke virtual networks, which you will implement later in this lab.

6. On the **az104-06-vnet01** virtual network blade, in the **Settings** section, click **Peerings** and then click **+ Add**.

7. Add a peering with the following settings (leave others with their default values) and click Add:

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**
Subscription	the name of the Azure subscription you are using in this lab
Virtual network	**az104-06-vnet3**
Traffic to remote virtual network	**Allow (default)**
Traffic forwarded from remote virtual network	**Allow (default)**
Virtual network gateway	**None (default)**

> **Note**: 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. This completes setting up the hub and spoke topology (with two spoke virtual networks).

> **Note**: **Allow forwarded traffic** needs to be enabled in order to facilitate routing between spoke virtual networks, which you will implement later in this lab.

Task 3: Test transitivity of virtual network peering

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

1. In the Azure portal, search for and select **Network Watcher**.
2. On the **Network Watcher** blade, expand the listing of Azure regions and verify that the service is enabled in the Azure into which you deployed resources in the first task of this lab.
3. On the **Network Watcher** blade, navigate to the **Connection troubleshoot**.
4. On the **Network Watcher - Connection troubleshoot** blade, initiate a check with the following settings (leave others with their default values):

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**

> **Note**: **10.62.0.4** represents the private IP address of **az104-06-vm2**

5. Click **Check** and wait until results of the connectivity check are returned. Verify that the status is **Reachable**. Review the network path and note that the connection was direct, with no intermediate hops in between the VMs.

> **Note**: This is expected, since the hub virtual network is peered directly with the first spoke virtual network.

> **Note**: The initial check can take about 2 minutes because it requires installation of the Network Watcher Agent virtual machine extension on **az104-06-vm0**.

6. On the **Network Watcher - Connection troubleshoot** blade, initiate a check with the following settings (leave others with their default values):

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**

Setting	Value
Destination	**Specify manually**
URI, FQDN or IPv4	**10.63.0.4**
Protocol	**TCP**
Destination Port	**3389**

> **Note**: **10.63.0.4** represents the private IP address of **az104-06-vm3**

7. Click **Check** and wait until results of the connectivity check are returned. Verify that the status is **Reachable**. Review the network path and note that the connection was direct, with no intermediate hops in between the VMs.

> **Note**: This is expected, since the hub virtual network is peered directly with the second spoke virtual network.

8. On the **Network Watcher - Connection troubleshoot** blade, initiate a check with the following settings (leave others with their default values):

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**

9. Click **Check** and wait until results of the connectivity check are returned. Note that the status is **Unreachable**.

> **Note**: This is expected, since the two spoke virtual networks are not peered with each other (virtual network peering is not transitive).

Task 4: Configure routing in the hub and spoke topology

In this task, you 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.

1. In the Azure portal, search and select **Virtual machines**.
2. On the **Virtual machines** blade, in the list of virtual machines, click **az104-06-vm0**.
3. On the **az104-06-vm0** virtual machine blade, in the **Settings** section, click **Networking**.
4. Click the **az104-06-nic0** link next to the **Network interface** label, and then, on the **az104-06-nic0** network interface blade, in the **Settings** section, click **IP configurations**.

5. Set **IP forwarding** to **Enabled** and save the change.

> **Note**: This setting is required in order for **az104-06-vm0** to function as a router, which will route traffic between two spoke virtual networks.

> **Note**: Now you need to configure operating system of the **az104-06-vm0** virtual machine to support routing.

6. In the Azure portal, navigate back to the **az104-06-vm0** Azure virtual machine blade and click **Overview**.
7. On the **az104-06-vm0** blade, in the **Operations** section, click **Run command**, and, in the list of commands, click **RunPowerShellScript**.
8. On the **Run Command Script** blade, type the following and click **Run** to install the Remote Access Windows Server role.

Code Copy

```
Install-WindowsFeature RemoteAccess -IncludeManagementTools
```

> **Note**: Wait for the confirmation that the command completed successfully.

9. On the **Run Command Script** blade, type the following and click **Run** to install the Routing role service.

Code Copy

```
Install-WindowsFeature -Name Routing -IncludeManagementTools -IncludeAllSubFeature

Install-WindowsFeature -Name "RSAT-RemoteAccess-Powershell"

Install-RemoteAccess -VpnType RoutingOnly

Get-NetAdapter | Set-NetIPInterface -Forwarding Enabled
```

> **Note**: Wait for the confirmation that the command completed successfully.

> **Note**: Now you need to create and configure user defined routes on the spoke virtual networks.

10. In the Azure portal, search and select **Route tables** and, on the **Route tables** blade, click **+ Add**.
11. Create a route table with the following settings (leave others with their default values):

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**

12. Click **Review and Create**. Let validation occur, and click **Create** to submit your deployment.

> **Note**: Wait for the route table to be created. This should take about 3 minutes.

13. Back on the **Route tables** blade, click **Refresh** and then click **az104-06-rt23**.
14. On the **az104-06-rt23** route table blade, in the **Settings** section, click **Routes**, and then click **+ Add**.
15. Add a new route with the following settings (leave others with their default values):

Setting	Value
Route name	**az104-06-route-vnet2-to-vnet3**
Address prefix	**10.63.0.0/20**
Next hop type	**Virtual appliance**
Next hop address	**10.60.0.4**

16. Click **OK**
17. Back on the **az104-06-rt23** route table blade, in the **Settings** section, click **Subnets**, and then click **+ Associate**.
18. Associate the route table **az104-06-rt23** with the following subnet:

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

19. Click **OK**
20. Navigate back to **Route tables** blade and click **+ Add**.
21. Create a route table with the following settings (leave others with their default values):

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**

22. Click Review and Create. Let validation occur, and hit Create to submit your deployment.

> **Note**: Wait for the route table to be created. This should take about 3 minutes.

23. Back on the **Route tables** blade, click **Refresh** and then click **az104-06-rt32**.
24. On the **az104-06-rt32** route table blade, in the **Settings** section, click **Routes**, and then click **+ Add**.
25. Add a new route with the following settings:

Setting	Value
Route name	**az104-06-route-vnet3-to-vnet2**
Address prefix	**10.62.0.0/20**

Setting	Value
Next hop type	**Virtual appliance**
Next hop address	**10.60.0.4**

26. Click **OK**

27. Back on the **az104-06-rt32** route table blade, in the **Settings** section, click **Subnets**, and then click **+ Associate**.

28. Associate the route table **az104-06-rt32** with the following subnet:

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

1.Click **OK**

1. In the Azure portal, navigate back to the **Network Watcher - Connection troubleshoot** blade.

2. On the **Network Watcher - Connection troubleshoot** blade, initiate a check with the following settings (leave others with their default values):

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**

3. Click **Check** and wait until results of the connectivity check are returned. Verify that the status is **Reachable**. Review the network path and note that the traffic was routed via **10.60.0.4**, assigned to the **az104-06-nic0** network adapter. If status is **Unreachable**, you should restart az104-06-vm0.

> **Note**: This is expected, since the traffic between spoke virtual networks is now routed via the virtual machine located in the hub virtual network, which functions as a router.

> **Note**: You can use **Network Watcher** to view topology of the network.

Task 5: Implement Azure Load Balancer

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

1. In the Azure portal, search and select **Load balancers** and, on the **Load balancers** blade, click **+ Add**.

2. Create a load balancer with the following settings (leave others with their default values):

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	the name of a new resource group **az104-06-rg4**
Name	**az104-06-lb4**
Region	name of the Azure region into which you deployed all other resources in this lab
Type	**Public**
SKU	**Standard**
Public IP address	**Create new**
Public IP address name	**az104-06-pip4**
Availability zone	**No Zone**
Add a public IPv6 address	**No**

3. Click Review and Create. Let validation occur, and hit Create to submit your deployment.

> **Note**: Wait for the Azure load balancer to be provisioned. This should take about 2 minutes.

4. On the deployment blade, click **Go to resource**.
5. On the **az104-06-lb4** load balancer blade, in the **Settings** section, click **Backend pools**, and click **+ Add**.
6. Add a backend pool with the following settings (leave others with their default values):

Setting	Value
Name	**az104-06-lb4-be1**
Virtual network	**az104-06-vnet01**
IP version	**IPv4**
Virtual machine	**az104-06-vm0**
Virtual machine IP address	**ipconfig1 (10.60.0.4)**
Virtual machine	**az104-06-vm1**
Virtual machine IP address	**ipconfig1 (10.60.1.4)**

7. Click **Add**
8. Wait for the backend pool to be created, in the **Settings** section, click **Health probes**, and then click **+ Add**.
9. Add a health probe with the following settings:

Setting	Value
Name	**az104-06-lb4-hp1**
Protocol	**TCP**
Port	**80**
Interval	**5**

Setting	Value
Unhealthy threshold	**2**

10. Click **OK**
11. Wait for the health probe to be created, in the **Settings** section, click **Load balancing rules**, and then click **+ Add**.
12. Add a load balancing rule with the following settings (leave others with their default values):

Setting	Value
Name	**az104-06-lb4-lbrule1**
IP Version	**IPv4**
Protocol	**TCP**
Port	**80**
Backend port	**80**
Backend pool	**az104-06-lb4-be1**
Health probe	**az104-06-lb4-hp1**
Session persistence	**None**
Idle timeout (minutes)	**4**
TCP reset	**Disabled**
Floating IP (direct server return)	**Disabled**

13. Click **OK**
14. Wait for the load balancing rule to be created, click **Overview**, and note the value of the **Public IP address**.
15. Start another browser window and navigate to the IP address you identified in the previous step.
16. Verify that the browser window displays the message **Hello World from az104-06-vm0** or **Hello World from az104-06-vm1**.
17. Open another browser window but this time by using InPrivate mode and verify whether the target vm changes (as indicated by the message).

> **Note**: You might need to refresh the browser window or open it again by using InPrivate mode.

Task 6: Implement Azure Application Gateway

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

1. In the Azure portal, search and select **Virtual networks**.
2. On the **Virtual networks** blade, in the list of virtual networks, click **az104-06-vnet01**.
3. On the **az104-06-vnet01** virtual network blade, in the **Settings** section, click **Subnets**, and then click **+ Subnet**.
4. Add a subnet with the following settings (leave others with their default values):

Setting	Value

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

5. Click **Save**

> **Note**: This subnet will be used by the Azure Application Gateway instances, which you will deploy later in this task. The Application Gateway requires a dedicated subnet of /27 or larger size.

6. In the Azure portal, search and select **Application Gateways** and, on the **Application Gateways** blade, click **+ Add**.

7. On the **Basics** tab of the **Create an application gateway** blade, specify the following settings (leave others with their default values):

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	the name of a new resource group **az104-06-rg5**
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**
HTTP2	**Disabled**
Virtual network	**az104-06-vnet01**
Subnet	**subnet-appgw**

8. Click **Next: Frontends >** and, on the **Frontends** tab of the **Create an application gateway** blade, specify the following settings (leave others with their default values):

Setting	Value
Frontend IP address type	**Public**
Firewall public IP address	the name of a new public ip address **az104-06-pip5**

9. Click **Next: Backends >**, on the **Backends** tab of the **Create an application gateway** blade, click **Add a backend pool**, and, on the **Add a backend pool** blade, specify the following settings (leave others with their default values):

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

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

10. Click **Add**, click **Next: Configuration >** and, on the **Configuration** tab of the **Create an application gateway** blade, click **+ Add a routing rule**.
11. On the **Add a routing rule** blade, on the **Listener** tab, specify the following settings:

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

12. Switch to the **Backend targets** tab of the **Add a routing rule** blade and specify the following settings (leave others with their default values):

Setting	Value
Target type	**Backend pool**
Backend target	**az104-06-appgw5-be1**

13. Click **Add new** under to the **HTTP setting** text box, and, on the **Add an HTTP setting** blade, specify the following settings (leave others with their default values):

Setting	Value
HTTP setting	**az104-06-appgw5-http1**
Backend protocol	**HTTP**
Backend port	**80**
Cookie-based affinity	**Disable**
Connection draining	**Disable**
Request time-out (seconds)	**20**

14. Click **Add** on the **Add an HTTP setting** blade, and back on the **Add a routing rule** blade, click **Add**.
15. Click **Next: Tags >**, followed by **Next: Review + create >** and then click **Create**.

> **Note**: Wait for the Application Gateway instance to be created. This might take about 8 minutes.

16. In the Azure portal, search and select **Application Gateways** and, on the **Application Gateways** blade, click **az104-06-appgw5**.
17. On the **az104-06-appgw5** Application Gateway blade, note the value of the **Frontend public IP address**.
18. Start another browser window and navigate to the IP address you identified in the previous step.

19. Verify that the browser window displays the message **Hello World from az104-06-vm2** or **Hello World from az104-06-vm3**.

20. Open another browser window but this time by using InPrivate mode and verify whether the target vm changes (based on the message displayed on the web page).

> **Note**: You might need to refresh the browser window or open it again by using InPrivate mode.

> **Note**: Targeting virtual machines on multiple virtual networks is not a common configuration, but it is meant to illustrate the point that Application Gateway is capable of targeting virtual machines on multiple virtual networks (as well as endpoints in other Azure regions or even outside of Azure), unlike Azure Load Balancer, which load balances across virtual machines in the same virtual network.

Clean up resources

> **Note**: Remember to remove any newly created Azure resources that you no longer use. Removing unused resources ensures you will not see unexpected charges.

1. In the Azure portal, open the **PowerShell** session within the **Cloud Shell** pane.

2. List all resource groups created throughout the labs of this module by running the following command:

```
Code                                                    Copy

Get-AzResourceGroup -Name 'az104-06*'
```

3. Delete all resource groups you created throughout the labs of this module by running the following command:

```
Code                                                    Copy

Get-AzResourceGroup -Name 'az104-06*' | Remove-AzResourceGroup -Force -AsJob
```

> **Note**: The command executes asynchronously (as determined by the -AsJob parameter), so while you will be able to run another PowerShell command immediately afterwards within the same PowerShell session, it will take a few minutes before the resource groups are actually removed.

Review

In this lab, you have:

- Provisioned the lab environment
- Configured the hub and spoke network topology
- Tested 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