# Configuring and Managing Virtual Networks

# Lab: Configuring VNet peering and service chaining

## Scenario

ADatum Corporation wants to implement service chaining between Azure virtual networks in its Azure subscription.

## Objectives

After completing this lab, you will be able to:

- Deploy Azure VMs by using Azure Resource Manager templates.
- Configure VNet peering.
- Implement routing
- Validate service chaining

## Lab Setup

Estimated Time: 45 minutes

User Name: **Student**

Password: **Pa55w.rd**

# Exercise 1: Creating an Azure lab environment by using deployment templates

The main tasks for this exercise are as follows:

1. Create the first Azure virtual network environment by using an Azure Resource Manager template
2. Create the second Azure virtual network environment by using an Azure Resource Manager template

**Task 1: Create the first Azure virtual network environment by using an Azure Resource Manager template**

1. From the lab virtual machine, start Microsoft Edge and browse to the Azure portal at **http://portal.azure.com** and sign in by using the Microsoft account that has the Owner role in the target Azure subscription.
2. In the Azure portal, in the Microsoft Edge window, start a **Bash** session within the **Cloud Shell**.
3. If you are presented with the **You have no storage mounted** message, configure storage using the following settings:

   - Subsciption: the name of the target Azure subscription
   - Cloud Shell region: the name of the Azure region that is available in your subscription and which is closest to the lab location
   - Resource group: the name of a new resource group **az3000400-LabRG**
   - Storage account: a name of a new storage account
   - File share: a name of a new file share

4. From the Cloud Shell pane, create two resource groups by running (replace the <Azure region> placeholder with the name of the Azure region that is available in your subscription and which is closest to the lab location)

```
az group create --resource-group az3000401-LabRG --location <Azure region>
```

```
az group create --resource-group az3000402-LabRG --location <Azure region>
```

5. From the Cloud Shell pane, upload the first Azure Resource Manager template
   **\allfiles\AZ-300T02\Module_03\azuredeploy0401.json** into the home directory.

6. From the Cloud Shell pane, upload the parameter file
   **\allfiles\AZ-300T02\Module_03\azuredeploy04.parameters.json** into the home directory.

7. From the Cloud Shell pane, deploy the two Azure VMs hosting Windows Server 2016 Datacenter into the first
   virtual network by running:

```
az group deployment create --resource-group az3000401-LabRG --template-file
azuredeploy0401.json --parameters @azuredeploy04.parameters.json --no-wait
```

Note: Do not wait for the deployment to complete but proceed to the next task.

**Task 2: Create the second Azure virtual network environment by using an Azure Resource Manager template**

1. From the Cloud Shell pane, upload the second Azure Resource Manager template
   **\allfiles\AZ-300T02\Module_03\azuredeploy0402.json** into the home directory.

2. From the Cloud Shell pane, deploy an Azure VM hosting Windows Server 2016 Datacenter into the second
   virtual network by running:

```
az group deployment create --resource-group az3000402-LabRG --template-file
azuredeploy0402.json --parameters @azuredeploy04.parameters.json --no-wait
```

Note: The second template uses the same parameter file.

Note: Do not wait for the deployment to complete but proceed to the next exercise.

Result: After completing this exercise, you should have created two Azure virtual networks hosting Azure
VMs running Windows Server 2016 Datacenter.

# Exercise 2: Configuring VNet peering

The task for this exercise is as follows:

1. Configure VNet peering for both virtual networks

**Task 1: Configure VNet peering for both virtual networks**

1. In the Microsoft Edge window displaying the Azure portal, navigate to the **az3000401-vnet** virtual network blade.

2. From the **az3000401-vnet** blade, create a VNet peering with the following settings:

   ○ Name of the peering from the first virtual network to the second virtual network:
     **az3000401-vnet-to-az3000402-vnet**

   ○ Virtual network deployment model: **Resource manager**

   ○ Subscription: the name of the Azure subscription you are using for this lab

   ○ Virtual network: **az3000402-vnet**

   ○ Name of the peering from the second virtual network to the first virtual network:

**az3000402-vnet-to-az3000401-vnet**

- ○ Allow virtual network access from the first virtual network to the second virtual nework: **Enabled**
- ○ Allow virtual network access from the second virtual network to the first virtual nework: **Enabled**

- ○ Allow forwarded traffic from the first virtual network to the second virtual network: **Disabled**
- ○ Allow forwarded traffic from the second virtual network to the first virtual network: **Disabled**
- ○ Allow gateway transit: disabled

# Exercise 3: Implementing routing

The main tasks for this exercise are as follows:

1. Enable IP forwarding
2. Configure user defined routing
3. Configure routing on an Azure VM running Windows Server 2016

**Task 1: Enable IP forwarding**

1. In Microsoft Edge, navigate to the **az3000401-nic2** blade (the NIC of **az3000401-vm2**)
2. On the **az3000401-nic2** blade, modify the **IP configurations** by setting **IP forwarding** to **Enabled**.

**Task 2: Configure user defined routing**

1. In the Azure portal, create a new route table with the following settings:

   - ○ Name: **az3000402-rt1**
   - ○ Subscription: the name of the Azure subscription you use for this lab
   - ○ Resource group: **az3000402-LabRG**
   - ○ Location: the same Azure region in which you created the virtual networks
   - ○ Virtual network gateway route propagation: **Disabled**

   Once the creation of the route table has finished, click on **Go to resource**

2. In the Azure portal, on the route table az3000402-rt1 that was created on the previous step, click on **Routes** under **Settings** and add a route with the following settings:

   - ○ Route name: **custom-route-to-az3000401-vnet**
   - ○ Address prefix: **10.0.0.0/22**
   - ○ Next hop type: **Virtual appliance**
   - ○ Next hop address: **10.0.1.4**

3. In the Azure portal, associate the route table with the **subnet-1** of the **az3000402-vnet**.

**Task 3: Configure routing on an Azure VM running Windows Server 2016**

1. On the lab computer, from the Azure portal, start a Remote Desktop session to **az3000401-vm2** Azure VM.
2. When prompted to authenticate, specify the following credentials:

   - ○ User name: **Student**
   - ○ Password: **Pa55w.rd1234**

3. Once you are connected to az3000401-vm2 via the Remote Desktop session, from **Server Manager**, install the **Remote Access** server role with the **Routing** role service and all required features.

4. In the Remote Desktop session to az3000401-vm2, start the **Routing and Remote Access** console.

5. In the **Routing and Remote Access** console, right click under the name of the server az3000401-vm2 and select **Configure and Enable Routing and Remote Access** to run the **Routing and Remote Access Server Setup Wizard**.

6. In the **Routing and Remote Access Server Setup Wizard**, select **Custom configuration** under **Configuration** and enable **LAN routing**.

7. Start **Routing and Remote Access** service.

8. In the Remote Desktop session to az3000401-vm2, start the **Windows Firewall with Advanced Security** console and enable **File and Printer Sharing** (**Echo Request - ICMPv4-In**) inbound rule for all profiles.

> Result: After completing this exercise, you should have configured custom routing within the second virtual network.

# Exercise 4: Validating service chaining

The main tasks for this exercise are as follows:

1. Configure Windows Firewall with Advanced Security on an Azure VM
2. Test service chaining between peered virtual networks

**Task 1: Configure Windows Firewall with Advanced Security on the target Azure VM**

1. On the lab computer, from the Azure portal, start a Remote Desktop session to **az3000401-vm1** Azure VM.

2. When prompted to authenticate, specify the following credentials:

   ○ User name: **Student**

   ○ Password: **Pa55w.rd1234**

3. In the Remote Desktop session to az3000401-vm1, start the **Windows Firewall with Advanced Security** console and enable **File and Printer Sharing** (**Echo Request - ICMPv4-In**) inbound rule for all profiles.

**Task 2: Test service chaining between peered virtual networks**

1. On the lab computer, from the Azure portal, start a Remote Desktop session to **az3000402-vm1** Azure VM.

2. When prompted to authenticate, specify the following credentials:

   ○ User name: **Student**

   ○ Password: **Pa55w.rd1234**

3. Once you are connected to az3000402-vm1 via the Remote Desktop session, start **Windows PowerShell**.

4. In the **Windows PowerShell** window, run the following:

```
Test-NetConnection -ComputerName 10.0.0.4 -TraceRoute
```

5. Verify that test is successful and note that the connection was routed over 10.0.1.4

> Result: After completing this exercise, you should have validated service chaining between peered virtual

> networks.

# Exercise 5: Remove lab resources

**Task 1: Open Cloud Shell**

1. At the top of the portal, click the **Cloud Shell** icon to open the Cloud Shell pane.
2. If needed, switch to the Bash shell session by using the drop down list in the upper left corner of the Cloud Shell pane.
3. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to list all resource groups you created in this lab:

```
az group list --query "[?starts_with(name,'az30004')]".name --output tsv
```

4. Verify that the output contains only the resource groups you created in this lab. These groups will be deleted in the next task.

**Task 2: Delete resource groups**

1. At the **Cloud Shell** command prompt, type in the following command and press **Enter** to delete the resource groups you created in this lab

```
az group list --query "[?starts_with(name,'az30004')]".name --output tsv |
xargs -L1 bash -c 'az group delete --name $0 --no-wait --yes'
```

2. Close the **Cloud Shell** prompt at the bottom of the portal.

> Result: In this exercise, you removed the resources used in this lab.