Lab: Load Balancer and Traffic Manager

All tasks in this lab are performed from the Azure portal (including a PowerShell Cloud Shell session) except for Exercise 1 Task 3, which includes steps performed from a Remote Desktop session to an Azure VM

Lab files:

- **Labfiles\Module_08\Load_Balancer_and_Traffic_Manager\az-101-03_01_azuredeploy.json**
- **Labfiles\Module_08\Load_Balancer_and_Traffic_Manager\az-101-03_01_1_azuredeploy.parameters.json**
- **Labfiles\Module_08\Load_Balancer_and_Traffic_Manager\az-101-03_01_2_azuredeploy.parameters.json**

Scenario

Adatum Corporation wants to implement Azure VM-hosted web workloads and facilitate their management for its subsidiary Contoso Corporation in a highly available manner by leveraging load balancing and Network Address Translation (NAT) features of Azure Load Balancer

Objectives

After completing this lab, you will be able to:

- Deploy Azure VMs by using Azure Resource Manager templates
- Implement Azure Load Balancing
- Implement Azure Traffic Manager load balancing

Exercise 0: Deploy Azure VMs by using Azure Resource Manager templates

The main tasks for this exercise are as follows:

1. Deploy management Azure VMs running Windows Server 2016 Datacenter with the Web Server (IIS) role installed into an availability set in the first Azure region by using an Azure Resource Manager template

2. Deploy management Azure VMs running Windows Server 2016 Datacenter with the Web Server (IIS) role installed into an availability set in the second Azure region by using an Azure Resource Manager template

Task 1: Deploy management Azure VMs running Windows Server 2016 Datacenter with the Web Server (IIS) role installed into an availability set in the first Azure region by using an Azure Resource Manager template

1. From the lab virtual machine, start Microsoft Edge, browse to the Azure portal at **http://portal.azure.com** and sign in by using a Microsoft account that has the Owner role in the target Azure subscription.
2. In the Azure portal, navigate to the **Create a resource** blade.
3. From the **Create a resource** blade, search Azure Marketplace for **Template deployment**.
4. Use the list of search results to navigate to the **Deploy a custom template** blade.
5. On the **Custom deployment** blade, click the **Build your own template in the editor** link. If you do not see this link, click **Edit template** instead.
6. From the **Edit template** blade, load the template file **Labfiles\Module_08\Load_Balancer_and_Traffic_Manager\az-101-03_01_azuredeploy.json**.

   > **Note**: Review the content of the template and note that it defines deployment of two Azure VMs hosting Windows Server 2016 Datacenter Core into an availability set.

7. Save the template and return to the **Custom deployment** blade.
8. From the **Custom deployment** blade, navigate to the **Edit parameters** blade.
9. From the **Edit parameters** blade, load the parameters file **Labfiles\Module_08\Load_Balancer_and_Traffic_Manager\az-101-03_01_1_azuredeploy.parameters.json**.
10. Save the parameters and return to the **Custom deployment** blade.
11. From the **Custom deployment** blade, initiate a template deployment with the following settings:
    - Subscription: the name of the subscription you intend to use in this lab
    - Resource group: the name of a new resource group **az1010301-RG**
    - Location: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Vm Name Prefix: **az1010301w-vm**
    - Nic Name Prefix: **az1010301w-nic**

- Image Publisher: **MicrosoftWindowsServer**
- Image Offer: **WindowsServer**
- Image SKU: **2016-Datacenter**
- Vm Size: use **Standard_DS1_v2** or **Standard_DS2_v2**, based on the instructor's recommendations
- Virtual Network Name: **az1010301-vnet**
- Address Prefix: **10.101.31.0/24**
- Virtual Network Resource Group: **az1010301-RG**
- Subnet0Name: **subnet0**
- Subnet0Prefix: **10.101.31.0/26**
- Availability Set Name: **az1010301w-avset**
- Network Security Group Name: **az1010301w-vm-nsg**
- Modules Url: **https://github.com/Azure/azure-quickstart-templates/raw/master/dsc-extension-iis-server-windows-vm/ContosoWebsite.ps1.zip**
- Configuration Function: **ContosoWebsite.ps1\ContosoWebsite**

> **Note**: To identify Azure regions where you can provision Azure VMs, refer to **https://azure.microsoft.com/en-us/regions/offers/**
>
> **Note**: Do not wait for the deployment to complete but proceed to the next task.

Task 2: Deploy management Azure VMs running Windows Server 2016 Datacenter with the Web Server (IIS) role installed into an availability set in the second Azure region by using an Azure Resource Manager template

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Template deployment**.
3. Use the list of search results to navigate to the **Deploy a custom template** blade.
4. On the **Custom deployment** blade, click the **Build your own template in the editor** link. If you do not see this link, click **Edit template** instead.
5. From the **Edit template** blade, load the template file **Labfiles\Module_08\Load_Balancer_and_Traffic_Manager\az-101-03_01_azuredeploy.json**.

> **Note**: This is the same template you used in the previous task. You will use it to deploy a pair of Azure VMs to the second region.

6. Save the template and return to the **Custom deployment** blade.
7. From the **Custom deployment** blade, navigate to the **Edit parameters** blade.

8. From the **Edit parameters** blade, load the parameters file **Labfiles\Module_08\Load_Balancer_and_Traffic_Manager\az-101-03_01_2_azuredeploy.parameters.json**.
9. Save the parameters and return to the **Custom deployment** blade.
10. From the **Custom deployment** blade, initiate a template deployment with the following settings:
    - Subscription: the name of the subscription you are using in this lab
    - Resource group: the name of a new resource group **az1010302-RG**
    - Location: the name of the Azure region different from the one you chose in the previous task and where you can provision Azure VMs
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Vm Name Prefix: **az1010302w-vm**
    - Nic Name Prefix: **az1010302w-nic**
    - Image Publisher: **MicrosoftWindowsServer**
    - Image Offer: **WindowsServer**
    - Image SKU: **2016-Datacenter**
    - Vm Size: use **Standard_DS1_v2** or **Standard_DS2_v2**, based on the instructor's recommendations
    - Virtual Network Name: **az1010302-vnet**
    - Address Prefix: **10.101.32.0/24**
    - Virtual Network Resource Group: **az1010302-RG**
    - Subnet0Name: **subnet0**
    - Subnet0Prefix: **10.101.32.0/26**
    - Availability Set Name: **az1010302w-avset**
    - Network Security Group Name: **az1010302w-vm-nsg**
    - Modules Url: **https://github.com/Azure/azure-quickstart-templates/raw/master/dsc-extension-iis-server-windows-vm/ContosoWebsite.ps1.zip**
    - Configuration Function: **ContosoWebsite.ps1\ContosoWebsite**

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

Result: After you completed this exercise, you have used Azure Resource Manager templates to initiate deployment of Azure VMs running Windows Server 2016 Datacenter with the Web Server (IIS) role installed into availability sets in two Azure regions.

Exercise 1: Implement Azure Load Balancing

The main tasks for this exercise are as follows:

1. Implement Azure load balancing rules in the first region.
2. Implement Azure load balancing rules in the second region.
3. Implement Azure NAT rules in the first region.
4. Implement Azure NAT rules in the second region.
5. Verify Azure load balancing and NAT rules

Task 1: Implement Azure load balancing rules in the first region

Note: Before you start this task, ensure that the template deployment you started in the first task of the previous exercise has completed.

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Load Balancer**.
3. Use the list of search results to navigate to the **Create load balancer** blade.
4. From the **Create load balancer** blade, create a new Azure Load Balancer with the following settings:
   - Subscription: the name of the subscription you are using in this lab
   - Resource group: **az1010301-RG**
   - Name: **az1010301w-lb**
   - Region: the name of the Azure region in which you deployed Azure VMs in the first task of the previous exercise
   - Type: **Public**
   - SKU: **Basic**
   - Public IP address: a new public IP address named **az1010301w-lb-pip**
   - Public IP address SKU: **Basic**
   - Assignment: **Dynamic**
   - Add a public IPv6 address: **No**
5. In the Azure portal, navigate to the blade of the newly deployed Azure load balancer **az1010301w-lb**.
6. From the **az1010301w-lb** blade, display the **az1010301w-lb - Backend pools** blade.
7. From the **az1010301w-lb - Backend pools** blade, add a backend pool with the following settings:
   - Name: **az1010301w-bepool**
   - Virtual network: **az1010301-vnet**
   - IP version: **IPv4**
   - Associated to: **Virtual machine**
   - Virtual machine: **az1010301w-vm0**
   - Network IP configuration: **az1010301w-nic0/ipconfig1 (10.101.31.4)**
   - Virtual machine: **az1010301w-vm1**

- Network IP configuration: **az1010301w-nic1/ipconfig1 (10.101.31.5)**

  > **Note**: It is possible that the IP addresses of the Azure VMs are assigned in the reverse order.
  >
  > **Note**: Wait for the operation to complete. This should take less than a minute.

8. From the **az1010301w-lb - Backend pools** blade, display the **az1010301w-lb - Health probes** blade.
9. From the **az1010301w-lb - Health probes** blade, add a health probe with the following settings:
   - Name: **az1010301w-healthprobe**
   - Protocol: **TCP**
   - Port: **80**
   - Interval: **5** seconds
   - Unhealthy threshold: **2** consecutive failures

   > **Note**: Wait for the operation to complete. This should take less than a minute.

10. From the **az1010301w-lb - Health probes** blade, display the **az1010301w-lb - Load balancing rules** blade.
11. From the **az1010301w-lb - Load balancing rules** blade, add a load balancing rule with the following settings:
    - Name: **az1010301w-lbrule01**
    - IP Version: **IPv4**
    - Frontend IP address: **LoadBalancerFrontEnd**
    - Protocol: **TCP**
    - Port: **80**
    - Backend port: **80**
    - Backend pool: **az1010301w-bepool (2 virtual machines)**
    - Health probe: **az1010301w-healthprobe (TCP:80)**
    - Session persistence: **None**
    - Idle timeout (minutes): **4**
    - Floating IP (direct server return): **Disabled**

Task 2: Implement Azure load balancing rules in the second region

> **Note**: Before you start this task, ensure that the template deployment you started in the second task of the previous exercise has completed.

1. In the Azure portal, navigate to the **Create a resource** blade.

2. From the **Create a resource** blade, search Azure Marketplace for **Load Balancer**.
3. Use the list of search results to navigate to the **Create load balancer** blade.
4. From the **Create load balancer** blade, create a new Azure Load Balancer with the following settings:
   - Subscription: the name of the subscription you are using in this lab
   - Resource group: **az1010302-RG**
   - Name: **az1010302w-lb**
   - Region: the name of the Azure region in which you deployed Azure VMs in the second task of the previous exercise
   - Type: **Public**
   - SKU: **Basic**
   - Public IP address: a new public IP address named **az1010302w-lb-pip**
   - Public IP address SKU: **Basic**
   - Assignment: **Dynamic**
   - Add a public IPv6 address: **No**
5. In the Azure portal, navigate to the blade of the newly deployed Azure load balancer **az1010302w-lb**.
6. From the **az1010302w-lb** blade, display the **az1010302w-lb - Backend pools** blade.
7. From the **az1010302w-lb - Backend pools** blade, add a backend pool with the following settings:
   - Name: **az1010302w-bepool**
   - Virtual network: **az1010302-vnet**
   - IP version: **IPv4**
   - Associated to: **Virtual machine**
   - Virtual machine: **az1010302w-vm0**
   - Network IP configuration: **az1010302w-nic0/ipconfig1 (10.101.32.4)**
   - Virtual machine: **az1010302w-vm1**
   - Network IP configuration: **az1010302w-nic1/ipconfig1 (10.101.32.5)**

   > **Note**: It is possible that the IP addresses of the Azure VMs are assigned in the reverse order.
   >
   > **Note**: Wait for the operation to complete. This should take less than a minute.

8. From the **az1010302w-lb - Backend pools** blade, display the **az1010302w-lb - Health probes** blade.
9. From the **az1010302w-lb - Health probes** blade, add a health probe with the following settings:
   - Name: **az1010302w-healthprobe**
   - Protocol: **TCP**

- Port: **80**
- Interval: **5** seconds
- Unhealthy threshold: **2** consecutive failures

> **Note**: Wait for the operation to complete. This should take less than a minute.

10. From the **az1010302w-lb - Health probes** blade, display the **az1010302w-lb - Load balancing rules** blade.
11. From the **az1010302w-lb - Load balancing rules** blade, add a load balancing rule with the following settings:
    - Name: **az1010302w-lbrule01**
    - IP Version: **IPv4**
    - Frontend IP address: **LoadBalancerFrontEnd**
    - Protocol: **TCP**
    - Port: **80**
    - Backend port: **80**
    - Backend pool: **az1010302w-bepool (2 virtual machines)**
    - Health probe: **az1010302w-healthprobe (TCP:80)**
    - Session persistence: **None**
    - Idle timeout (minutes): **4**
    - Floating IP (direct server return): **Disabled**

Task 3: Implement Azure NAT rules in the first region

1. In the Azure portal, navigate to the blade of the Azure load balancer **az1010301w-lb**.
2. From the **az1010301w-lb** blade, display the **az1010301w-lb - Inbound NAT rules** blade.

> **Note**: The NAT functionality does not rely on health probes.

3. From the **az1010301w-lb - Inbound NAT rules** blade, add the first inbound NAT rule with the following settings:
    - Name: **az1010301w-vm0-RDP**
    - Frontend IP address: **LoadBalancerFrontEnd**
    - IP Version: **IPv4**
    - Service: **Custom**
    - Protocol: **TCP**
    - Port: **33890**
    - Target virtual machine: **az1010301w-vm0**

- Network IP configuration: **ipconfig1 (10.101.31.4)** or **ipconfig1 (10.101.31.5)**
- Port mapping: **Custom**
- Floating IP (direct server return): **Disabled**
- Target port: **3389**

> **Note**: Wait for the operation to complete. This should take less than a minute.

4. From the **az1010301w-lb - Inbound NAT rules** blade, add the second inbound NAT rule with the following settings:
   - Name: **az1010301w-vm1-RDP**
   - Frontend IP address: **LoadBalancerFrontEnd**
   - IP Version: **IPv4**
   - Service: **Custom**
   - Protocol: **TCP**
   - Port: **33891**
   - Target virtual machine: **az1010301w-vm1**
   - Network IP configuration: **ipconfig1 (10.101.31.4)** or **ipconfig1 (10.101.31.5)**
   - Port mapping: **Custom**
   - Floating IP (direct server return): **Disabled**
   - Target port: **3389**

   > **Note**: Wait for the operation to complete. This should take less than a minute.

Task 4: Implement Azure NAT rules in the second region

1. In the Azure portal, navigate to the blade of the Azure load balancer **az1010302w-lb**.
2. From the **az1010302w-lb** blade, display the **az1010302w-lb - Inbound NAT rules** blade.
3. From the **az1010302w-lb - Inbound NAT rules** blade, add the first inbound NAT rule with the following settings:
   - Name: **az1010302w-vm0-RDP**
   - Frontend IP address: **LoadBalancedFrontEnd**
   - IP Version: **IPv4**
   - Service: **Custom**
   - Protocol: **TCP**
   - Port: **33890**
   - Target virtual machine: **az1010302w-vm0**

   - Network IP configuration: **ipconfig1 (10.101.32.4)** or **ipconfig1 (10.101.32.5)**
   - Port mapping: **Custom**
   - Floating IP (direct server return): **Disabled**
   - Target port: **3389**

   > **Note**: Wait for the operation to complete. This should take less than a minute.

4. From the **az1010302w-lb - Inbound NAT rules** blade, add the second inbound NAT rule with the following settings:
   - Name: **az1010302w-vm1-RDP**
   - Frontend IP address: **LoadBalancedFrontEnd**
   - IP Version: **IPv4**
   - Service: **Custom**
   - Protocol: **TCP**
   - Port: **33891**
   - Target virtual machine: **az1010302w-vm1**
   - Network IP configuration: **ipconfig1 (10.101.32.4)** or **ipconfig1 (10.101.32.5)**
   - Port mapping: **Custom**
   - Floating IP (direct server return): **Disabled**
   - Target port: **3389**

   > **Note**: Wait for the operation to complete. This should take less than a minute.

Task 5: Verify Azure load balancing and NAT rules.

1. In the Azure portal, navigate to the blade of the Azure load balancer **az1010301w-lb**.
2. On the **az1010301w-lb** blade, identify the public IP address assigned to the load balancer frontend.
3. In the Microsoft Edge window, open a new tab and browse to the IP address you identified in the previous step.
4. Verify that the tab displays the default Internet Information Services home page.
5. Close the browser tab displaying the default Internet Information Services home page.
6. In the Azure portal, navigate to the blade of the Azure load balancer **az1010301w-lb**.
7. On the **az1010301w-lb** blade, identify the public IP address assigned to the load balancer frontend.

8. From the lab virtual machine, run the following command, after replacing the <az1010301w-lb_public_IP< placeholder with the IP address you identified in the previous task:

   CodeCopy

```
mstsc /v:<az1010301w-lb_public_IP>:33890
```

> **Note**: This command initiates a Remote Desktop session to the **az1010301w-vm0** Azure VM by using the **az1010301w-vm0-RDP** NAT rule you created in the previous task.

9. When prompted to sign in, provide the following credentials:
   - Admin Username: **Student**
   - Admin Password: **Pa55w.rd1234**
10. Once you sign in, from the command prompt, run the following command:

    CodeCopy

```
hostname
```

11. Review the output and verify that you are actually connected to the **az1010301w-vm0** Azure VM.

> **Note**: Repeat the same tests for the second region.

Result: After you completed this exercise, you have implemented load balancing rules and NAT rules of Azure in two Azure regions and verified load balancing rules and NAT rules of Azure load balancers in the first region.

Exercise 2: Implement Azure Traffic Manager load balancing

The main tasks for this exercise are as follows:

1. Assign DNS names to public IP addresses of Azure load balancers
2. Implement Azure Traffic Manager load balancing
3. Verify Azure Traffic Manager load balancing

Task 1: Assign DNS names to public IP addresses of Azure load balancers

Note: This task is necessary because each Traffic Manager endpoint must have a DNS name assigned.

1. In the Azure portal, navigate to the blade of the public IP address resource associated with the Azure load balancer in the first region named **az1010301w-lb-pip**.
2. From the **az1010301w-lb-pip** blade, display its **Configuration** blade.
3. From the **az1010301w-lb-pip - Configuration** blade set the **DNS name label** of the public IP address to a unique value.

   Note: The green check mark in the **DNS name label (optional)** text box will indicate whether the name you typed in is valid and unique.

4. Navigate to the blade of the public IP address resource associated with the Azure load balancer in the second region named **az1010302w-lb-pip**.
5. From the **az1010302w-lb-pip** blade, display its **Configuration** blade.
6. From the **az1010302w-lb-pip - Configuration** blade set the **DNS name label** of the public IP address to a unique value.

   Note: The green check mark in the **DNS name label (optional)** text box will indicate whether the name you typed in is valid and unique.

Task 2: Implement Azure Traffic Manager load balancing

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Traffic Manager profile**.
3. Use the list of search results to navigate to the **Create Traffic Manager profile** blade.
4. From the **Create Traffic Manager profile** blade, create a new Azure Traffic Manager profile with the following settings:
   - Name: a globally unique name in the trafficmanager.net DNS namespace
   - Routing method: **Weighted**
   - Subscription: the name of the subscription you are using in this lab
   - Resource group: the name of a new resource group **az1010303-RG**
   - Location: either of the Azure regions you used earlier in this lab
5. In the Azure portal, navigate to the blade of the newly provisioned Traffic Manager profile.
6. From the Traffic Manager profile blade, display its **Configuration** blade and review the configuration settings.

   Note: The default TTL of the Traffic Manager profile DNS records is 60 seconds

7. From the Traffic Manager profile blade, display its **Endpoints** blade.
8. From the **Endpoints** blade, add the first endpoint with the following settings:
   - Type: **Azure endpoint**
   - Name: **az1010301w-lb-pip**
   - Target resource type: **Public IP address**
   - Target resource: **az1010301w-lb-pip**
   - Weight: **100**
   - Custom Header settings: leave blank
   - Add as disabled: leave blank
9. From the **Endpoints** blade, add the second endpoint with the following settings:
   - Type: **Azure endpoint**
   - Name: **az1010302w-lb-pip**
   - Target resource type: **Public IP address**
   - Target resource: **az1010302w-lb-pip**
   - Weight: **100**
   - Custom Header settings: leave blank
   - Add as disabled: leave blank
10. On the **Endpoints** blade, examine the entries in the **MONITORING STATUS** column for both endpoints. Wait until both are listed as **Online** before you proceed to the next task.

Task 3: Verify Azure Traffic Manager load balancing

1. From the **Endpoints** blade, switch to the **Overview** section of the Traffic Manager profile blade.
2. Note the DNS name assigned to the Traffic Manager profile (the string following the **http://** prefix).
3. From the Azure Portal, start a PowerShell session in the Cloud Shell pane.

   > **Note**: If this is the first time you are launching the Cloud Shell in the current Azure subscription, you will be asked to create an Azure file share to persist Cloud Shell files. If so, accept the defaults, which will result in creation of a storage account in an automatically generated resource group.

4. In the Cloud Shell pane, run the following command, replacing the <TM_DNS_name< placeholder with the value of the DNS name assigned to the Traffic Manager profile you identified in the previous task:

   CodeCopy

```
nslookup <TM_DNS_name>
```

5. Review the output and note the **Name** entry. This should match the DNS name of the one of the Traffic Manager profile endpoints you created in the previous task.
6. Wait for at least 60 seconds and run the same command again:

   CodeCopy

```
nslookup <TM_DNS_name>
```

7. Review the output and note the **Name** entry. This time, the entry should match the DNS name of the other Traffic Manager profile endpoint you created in the previous task.

Result: After you completed this exercise, you have implemented and verified Azure Traffic Manager load balancing

Exercise 3: 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. At the Cloud Shell interface, select **Bash**.
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:

   ShellCopy

```
az group list --query "[?starts_with(name,'az101030')].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

   ShellCopy

```
az group list --query "[?starts_with(name,'az101030')].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.