Lab: Deploy and Manage Virtual Machines

All tasks in this lab are performed from the Azure portal (including a PowerShell Cloud Shell session) except for Excercise 2 Task 2 and Exercise 2 Task 3, which include steps performed from a Remote Desktop session to an Azure VM

> ⓘNote
>
> **Note**: When not using Cloud Shell, the lab virtual machine must have Azure PowerShell module installed **https://docs.microsoft.com/en-us/powershell/azure/install-Az-ps**

Lab files:

- **Labfiles\Module_02\Deploy_and_Manage_Virtual_Machines\az-100-03_azuredeploy.json**
- **Labfiles\Module_02\Deploy_and_Manage_Virtual_Machines\az-100-03_azuredeploy.parameters.json**
- **Labfiles\Module_02\Deploy_and_Manage_Virtual_Machines\az-100-03_install_iis_vmss.zip**

Scenario

Adatum Corporation wants to implement its workloads by using Azure virtual machines (VMs) and Azure VM scale sets

Objectives

After completing this lab, you will be able to:

- Deploy Azure VMs by using the Azure portal, Azure PowerShell, and Azure Resource Manager templates
- Configure networking settings of Azure VMs running Windows and Linux operating systems
- Deploy and configure Azure VM scale sets

Exercise 1: Deploy Azure VMs by using the Azure portal, Azure PowerShell, and Azure Resource Manager templates

The main tasks for this exercise are as follows:

1. Deploy an Azure VM running Windows Server 2016 Datacenter into an availability set by using the Azure portal
2. Deploy an Azure VM running Windows Server 2016 Datacenter into the existing availability set by using Azure PowerShell
3. Deploy two Azure VMs running Linux into an availability set by using an Azure Resource Manager template

Task 1: Deploy an Azure VM running Windows Server 2016 Datacenter into an availability set by using the Azure portal

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 Azure subscription you intend to use in this lab.
2. In the Azure portal, navigate to the **Create a resource** blade.
3. From the **Create a resource** blade, search Azure Marketplace for **Windows Server**. Select **Windows Server** from the search results list.
4. On the Windows Server page, use the drop-down menu to select **[smalldisk] Windows Server 2016 Datacenter**, and then click **Create**.

- Subscription: the name of the subscription you are using in this lab
- Resource group: the name of a new resource group **az1000301-RG**
- Virtual machine name: **az1000301-vm0**
- Region: **(US) East US** (or a region closer to you)

  > ⓘNote
  >
  > **Note**: To identify Azure regions where you can provision Azure VMs, refer to **https://azure.microsoft.com/en-us/regions/offers/**

- Availability options: **Availability set**
- Availability set: Click **Create New**, and name the new availability set **az1000301-avset0** with **2** fault domains and **5** update domains. Click **OK**.
- Image: **[smalldisk] Windows Server 2016 Datacenter**
- Size: **Standard DS2_v2**
- Username: **Student**
- Password: **Pa55w.rd1234**
- Public inbound ports: **None**
- Already have a Windows license?: **No**

6. Click **Next: Disks >**.
   - OS disk type: **Standard HDD**
7. Click **Next: Networking >**.
8. On the Networking tab, click **Create new** under Virtual Network. Use the virtual network name already assigned by default and specify the following:
   - Virtual network address range: **10.103.0.0/16**
   - Subnet name: **subnet0**
   - Subnet address range: **10.103.0.0/24**
9. Click **OK**.
10. Leave all other default values, and click **Review + create**.
11. Click **Create**.

> ⓘNote
>
> **Note**: You will configure the network security group you create in this task in the second exercise of this lab

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

Task 2: Deploy an Azure VM running Windows Server 2016 Datacenter into the existing availability set by using Azure PowerShell

1. From the Azure Portal, start a PowerShell session in the Cloud Shell pane.

ⓘNote

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.

2. In the Cloud Shell pane, run the following command:

Code Copy

```
$vmName = 'az1000301-vm1'
$vmSize = 'Standard_DS2_v2'
```

ⓘNote

Note: This sets the values of variables designating the Azure VM name and its size

3. In the Cloud Shell pane, run the following commands:

Code Copy

```
$resourceGroup = Get-AzResourceGroup -Name 'az1000301-RG'
$location = $resourceGroup.Location
```

ⓘNote

Note: These commands set the values of variables designating the target resource group and its location

4. In the Cloud Shell pane, run the following commands:

Code Copy

```
$availabilitySet = Get-AzAvailabilitySet -ResourceGroupName $resourceGroup.ResourceGroupName
-Name 'az1000301-avset0'
$vnet = Get-AzVirtualNetwork -Name 'az1000301-RG-vnet' -ResourceGroupName
$resourceGroup.ResourceGroupName
$subnetid = (Get-AzVirtualNetworkSubnetConfig -Name 'subnet0' -VirtualNetwork $vnet).Id
```

ⓘNote

Note: These commands set the values of variables designating the availability set, virtual network, and subnet into which you will deploy the new Azure VM

5. In the Cloud Shell pane, run the following commands:

Code Copy

```
$nsg = New-AzNetworkSecurityGroup -ResourceGroupName $resourceGroup.ResourceGroupName -
Location $location -Name "$vmName-nsg"
$pip = New-AzPublicIpAddress -Name "$vmName-ip" -ResourceGroupName
$resourceGroup.ResourceGroupName -Location $location -AllocationMethod Dynamic
$nic = New-AzNetworkInterface -Name "$($vmName)$(Get-Random)" -ResourceGroupName
$resourceGroup.ResourceGroupName -Location $location -SubnetId $subnetid -PublicIpAddressId
$pip.Id -NetworkSecurityGroupId $nsg.Id
```

ⓘNote

Note: These commands create a new network security group, public IP address, and network interface that will be used by the new Azure VM

ⓘNote

Note: You will configure the network security group you create in this task in the second exercise of this lab

6. In the Cloud Shell pane, run the following commands:

Code Copy

```
$adminUsername = 'Student'
$adminPassword = 'Pa55w.rd1234'
$adminCreds = New-Object PSCredential $adminUsername, ($adminPassword | ConvertTo-SecureString -AsPlainText -Force)
```

ⓘNote

Note: These commands set the values of variables designating credentials of the local Administrator account of the new Azure VM

7. In the Cloud Shell pane, run the following commands:

Code Copy

```
$publisherName = 'MicrosoftWindowsServer'
$offerName = 'WindowsServer'
$skuName = '2016-Datacenter'
```

ⓘNote

Note: These commands set the values of variables designating the properties of the Azure Marketplace image that will be used to provision the new Azure VM

8. In the Cloud Shell pane, run the following command:

Code Copy

```
$osDiskType = (Get-AzDisk -ResourceGroupName $resourceGroup.ResourceGroupName)[0].Sku.Name
```

ⓘNote

Note: This command sets the values of a variable designating the operating system disk type of the new Azure VM

9. In the Cloud Shell pane, run the following commands:

Code Copy

```
$vmConfig = New-AzVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId
$availabilitySet.Id
Add-AzVMNetworkInterface -VM $vmConfig -Id $nic.Id
Set-AzVMOperatingSystem -VM $vmConfig -Windows -ComputerName $vmName -Credential $adminCreds
Set-AzVMSourceImage -VM $vmConfig -PublisherName $publisherName -Offer $offerName -Skus
$skuName -Version 'latest'
Set-AzVMOSDisk -VM $vmConfig -Name "$($vmName)_OsDisk_1_$(Get-Random)" -StorageAccountType
$osDiskType -CreateOption fromImage
Set-AzVMBootDiagnostic -VM $vmConfig -Disable
```

ⓘNote

Note: These commands set up the properties of the Azure VM configuration object that will be used to provision the new Azure VM, including the VM size, its availability set, network interface, computer name, local Administrator credentials, the source image, the operating system disk, and boot diagnostics settings.

10. In the Cloud Shell pane, run the following command:

Code Copy

```
New-AzVM -ResourceGroupName $resourceGroup.ResourceGroupName -Location $location -VM
$vmConfig
```

ⓘNote

Note: This command initiates deployment of the new Azure VM

ⓘNote

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

Task 3: Deploy two Azure VMs running Linux into an availability set 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**, and select **Template deployment (deploy using custom templates)**.

3. Click **Create**.

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_02\Deploy_and_Manage_Virtual_Machines\az-100-03_azuredeploy.json**.

ⓘNote

Note: Review the content of the template and note that it defines deployment of two Azure VMs hosting Linux Ubuntu into an availability set and into the existing virtual network **az1000301-vnet0**. This virtual network does not exist in your deployment. You will be changing the virtual network name in the parameters below.

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_02\Deploy_and_Manage_Virtual_Machines\az-100-03_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 **az1000302-RG**
    - Location: the same Azure region you chose earlier in this exercise
    - Vm Name Prefix: **az1000302-vm**
    - Nic Name Prefix: **az1000302-nic**
    - Pip Name Prefix: **az1000302-ip**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1000301-RG-vnet** *(change this value from the template default)*
    - Image Publisher: **Canonical**
    - Image Offer: **UbuntuServer**
    - Image SKU: **16.04.0-LTS**
    - Vm Size: **Standard_DS2_v2**

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

> ⓘNote
>
> **Result**: After you completed this exercise, you have deployed an Azure VM running Windows Server 2016 Datacenter into an availability set by using the Azure portal, deployed another Azure VM running Windows Server 2016 Datacenter into the same availability set by using Azure PowerShell, and deployed two Azure VMs running Linux Ubuntu into an availability set by using an Azure Resource Manager template.

> ⓘNote
>
> **Note**: You could certainly use a template to deploy two Azure VMs hosting Windows Server 2016 datacenter in a single task (just as this was done with two Azure VMs hosting Linux Ubuntu server). The reason for deploying these Azure VMs in two separate tasks was to give you the opportunity to become familiar with both the Azure portal and Azure PowerShell-based deployments.

Exercise 2: Configure networking settings of Azure VMs running Windows and Linux operating systems

The main tasks for this exercise are as follows:

1. Configure static private and public IP addresses of Azure VMs
2. Connect to an Azure VM running Windows Server 2016 Datacenter via a public IP address
3. Connect to an Azure VM running Linux Ubuntu Server via a private IP address

Task 1: Configure static private and public IP addresses of Azure VMs

2. From the **az1000301-vm0** blade, navigate to the **Networking** blade, displaying the configuration of the public IP address **az1000301-vm0-ip**, assigned to its network interface.

3. From the **Networking** blade, click the link representing the public IP address.

4. On the az1000301-vm0-ip blade, click **Configuration**.

5. Change the assignment of the public IP address to **Static**, and then click **Save**.

   > ⓘNote
   >
   > **Note**: Take a note of the public IP address assigned to the network interface of **az1000301-vm0**. You will need it later in this exercise.

6. In the Azure portal, navigate to the **az1000302-vm0** blade.

7. From the **az1000302-vm0** blade, display the **Networking** blade.

8. From the **az1000302-vm0 - Networking** blade, click the link representing the network interface.

9. From the blade displaying the properties of the network interface of **az1000302-vm0**, navigate to its **IP configurations** blade.

10. On the **IP configurations** blade, configure the **ipconfig1** private IP address to be static and set it to **10.103.0.100**, and then click **Save**.

    > ⓘNote
    >
    > **Note**: Changing the private IP address assignment requires restarting the Azure VM.

    > ⓘNote
    >
    > **Note**: It is possible to connect to Azure VMs via either statically or dynamically assigned public and private IP addresses. Choosing static IP assignment is commonly done in scenarios where these IP addresses are used in combination with IP filtering, routing, or if they are assigned to network interfaces of Azure VMs that function as DNS servers.

Task 2: Connect to an Azure VM running Windows Server 2016 Datacenter via a public IP address

1. In the Azure portal, navigate to the **az1000301-vm0** blade.

2. From the **az1000301-vm0** blade, navigate to the **Networking** blade.

3. On the **az1000301-vm0 - Networking** blade, review the inbound port rules of the network security group assigned to the network interface of **az1000301-vm0**.

   > ⓘNote
   >
   > **Note**: The default configuration consisting of built-in rules block inbound connections from the internet (including connections via the RDP port TCP 3389)

4. Add an inbound security rule to the existing network security group with the following settings:

   - Source: **Any**
   - Source port ranges: *****
   - Destination: **Any**
   - Destination port ranges: **3389**

- Action: **Allow**
- Priority: **100**
- Name: **AllowInternetRDPInBound**

5. In the Azure portal, display the **Overview** pane of the **az1000301-vm0** blade.
6. From the **Overview** pane of the **az1000301-vm0** blade, click **Connect** and generate an RDP file and use it to connect to **az1000301-vm0**.
7. When prompted, authenticate by specifying the following credentials:
   - User name: **Student**
   - Password: **Pa55w.rd1234**

Task 3: Connect to an Azure VM running Linux Ubuntu Server via a private IP address

1. Within the RDP session to **az1000301-vm0**, start **Command Prompt**.
2. From the Command Prompt, run the following:

   Code ⎘ Copy

   ```
   nslookup az1000302-vm0
   ```

3. Examine the output and note that the name resolves to the IP address you assigned in the first task of this exercise (**10.103.0.100**).

   > ⓘNote
   >
   > **Note**: This is expected. Azure provides built-in DNS name resolution within a virtual network.

4. Within the RDP session to **az1000301-vm0**, from Server Manager, click **Local Server**, then disable **IE Enhanced Security Configuration**.
5. Within the RDP session to **az1000301-vm0**, start Internet Explorer and download **putty.exe** from **https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html**
6. Use **putty.exe** to verify that you can successfully connect to **az1000302-vm0** on its private IP address via the **SSH** protocol (TCP 22).
7. When prompted, authenticate by specifying the following values:
   - User name: **Student**
   - Password: **Pa55w.rd1234**

   > ⓘNote
   >
   > **Note**: Both the username and password are case sensitive.

8. Once you successfully authenticated, terminate the RDP session to **az1000301-vm0**.
9. On the lab virtual machine, in the Azure portal, navigate to the **az1000302-vm0** blade.
10. From the **az1000302-vm0** blade, navigate to the **Networking** blade.
11. On the **az1000302-vm0 - Networking** blade, review the inbound port rules of the network security group assigned to the network interface of **az1000301-vm0** to determine why your SSH connection via the private IP address was successsful.

ⓘNote

Note: The default configuration consisting of built-in rules allows inbound connections within the Azure virtual network environment (including connections via the SSH port TCP 22).

ⓘNote

Result: After you completed this exercise, you have configured static private and public IP addresses of Azure VMs, connected to an Azure VM running Windows Server 2016 Datacenter via a public IP address, and connect to an Azure VM running Linux Ubuntu Server via a private IP address

Exercise 3: Deploy and configure Azure VM scale sets

The main tasks for this exercise are as follows:

1. Identify an available DNS name for an Azure VM scale set deployment
2. Deploy an Azure VM scale set
3. Install IIS on a scale set VM by using DSC extensions

Task 1: Identify an available DNS name for an Azure VM scale set deployment

1. From the Azure Portal, start a PowerShell session in the Cloud Shell pane.
2. In the Cloud Shell pane, run the following command, substituting the placeholder <custom-label> with any string which is likely to be unique.

   Code Copy

   ```
   $rg = Get-AzResourceGroup -Name az1000301-RG
   Test-AzDnsAvailability -DomainNameLabel <custom-label> -Location $rg.Location
   ```

3. Verify that the command returned **True**. If not, rerun the same command with a different value of the <custom-label> until the command returns **True**.
4. Note the value of the <custom-label> that resulted in the successful outcome. You will need it in the next task

Task 2: Deploy an Azure VM scale set

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Virtual machine scale set**.
3. Use the list of search results to navigate to the **Create virtual machine scale set** blade.
4. Use the **Create virtual machine scale set** blade to deploy a virtual machine scale set with the following settings:
   - Virtual machine scale set name: **az1000303vmss0**
   - Operating system disk image: **Windows Server 2016 Datacenter**
   - Subscription: the name of the subscription you are using in this lab
   - Resource group: the name of a new resource group **az1000303-RG**
   - Location: the same Azure region you chose in the previous exercises of this lab
   - Availability zone: **None**

- Password: **Pa55w.rd1234**
- Instance count: **1**
- Instance size: **DS2 v2**
- Deploy as low priority: **No**
- Use managed disks: **Yes**
- Autoscale: **Disabled**
- Choose Load balancing options: **Load balancer**
- Public IP address name: **az1000303vmss0-ip**
- Domain name label: type in the value of the <custom-label> you identified in the previous task
- Virtual network: the name of a new virtual network **az1000303-vnet0** with the following settings:
    - Address range: **10.203.0.0/16**
    - Subnet name: **subnet0**
    - Subnet address range: **10.203.0.0/24**
- Public IP address per instance: **Off**
- Accelerated networking: **Off**
- NIC network security group: **Basic**
- Select inbound ports: **HTTP**
- Boot diagnostics: **Off**
- System assigned managed identity: **Off**

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

Task 3: Install IIS on a scale set VM by using DSC extensions

1. In the Azure portal, navigate to the **az1000303vmss0** blade.
2. From the **az1000303vmss0** blade, display its Extension blade.
3. From the **az1000303vmss0 - Extension** blade, add the **PowerShell Desired State Configuration** extension with the following settings:

    > ⓘNote
    >
    > **Note**: The DSC configuration module is available for upload from **Labfiles\Module_02\Deploy_and_Manage_Virtual_Machines\az-100-03_install_iis_vmss.zip**. The module contains the DSC configuration script that installs the Web Server (IIS) role.

    - Configuration Modules or Script: **“az-100-03_install_iis_vmss.zip”**
    - Module-qualified Name of Configuration: **az-100-03_install_iis_vmss.ps1\IISInstall**
    - Configuration Arguments: leave blank
    - Configuration Data PSD1 File: leave blank

   - Version: **2.76**
   - Auto Upgrade Minor Version: **Yes**

4. Navigate to the **az1000303vmss0 - Instances** blade and initiate the upgrade of the **az1000303vmss0_0** instance.

   > ⓘNote
   >
   > **Note**: The update will trigger application of the DSC configuration script. Wait for upgrade to complete. This should take about 5 minutes. You can monitor the progress from the **az1000303vmss0 - Instances** blade.

5. Once the upgrade completes, navigate to the **Overview** blade.
6. On the **az1000303vmss0-ip** blade, note the public IP address assigned to **az1000303vmss0**.
7. Start Microsoft Edge and navigate to the public IP address you identified in the previous step.
8. Verify that the browser displays the default IIS home page.

> ⓘNote
>
> **Result**: After you completed this exercise, you have identified an available DNS name for an Azure VM scale set deployment, deployed an Azure VM scale set, and installed IIS on a scale set VM by using the DSC extension.

Exercise 4: 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:

   Shell Copy

   ```
   az group list --query "[?starts_with(name,'az1000')].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

   Shell Copy

   ```
   az group list --query "[?starts_with(name,'az1000')].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.

> ⓘNote

MicrosoftLearning/AZ-103-MicrosoftAzureAdministrator

Lab: Implement and Manage Storage

All tasks in this lab are performed from the Azure portal (including a PowerShell Cloud Shell session) except for Exercise 2 Task 2, which includes steps performed from a Remote Desktop session to an Azure VM

> ⓘNote
>
> **Note**: When not using Cloud Shell, the lab virtual machine must have the Azure PowerShell 1.2.0 module (or newer) installed https://docs.microsoft.com/en-us/powershell/azure/install-az-ps

Lab files:

- **Labfiles\Module_03\Implement_and_Manage_Storage\az-100-02_azuredeploy.json**
- **Labfiles\Module_03\Implement_and_Manage_Storage\az-100-02_azuredeploy.parameters.json**

Scenario

Adatum Corporation wants to leverage Azure Storage for hosting its data

Objectives

After completing this lab, you will be able to:

- Deploy an Azure VM by using an Azure Resource Manager template
- Implement and use Azure Blob Storage
- Implement and use Azure File Storage

Exercise 0: Prepare the lab environment

The main tasks for this exercise are as follows:

1. Deploy an Azure VM by using an Azure Resource Manager template

Task 1: Deploy an Azure VM 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 Azure subscription you intend to use in this lab.
2. In the Azure portal, navigate to the **Subscriptions** blade.
3. From the **Subscriptions** blade, navigate to the blade displaying properties of your Azure subscription.
4. From the blade displaying the properties of your subscription, navigate to its **Resource providers** blade.
5. On the **Resource providers** blade, register the following resource providers (if these resource providers have not been yet registered):
   - Microsoft.Network
   - Microsoft.Compute
   - Microsoft.Storage

Note: This step registers the Azure Resource Manager Microsoft.Network, Microsoft.Compute, and Microsoft.Storage resource providers. This is a one-time operation (per subscription) required when using Azure Resource Manager templates to deploy resources managed by these resource providers (if these resource providers have not been yet registered).

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

3. Click **Create**.

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_03\Implement_and_Manage_Storage\az-100-02_azuredeploy.json**.

   > ⓘNote
   >
   > **Note**: Review the content of the template and note that it defines deployment of an Azure VM hosting Windows Server 2016 Datacenter.

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_03\Implement_and_Manage_Storage\az-100-02_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 **az1000201-RG**
    - Location: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs
    - Vm Size: **Standard_DS2_v2**
    - Vm Name: **az1000201-vm1**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1000201-vnet1**

    > ⓘNote
    >
    > **Note**: To identify Azure regions where you can provision Azure VMs, refer to **https://azure.microsoft.com/en-us/regions/offers/**

    > ⓘNote
    >
    > **Note**: Do not wait for the deployment to complete but proceed to the next exercise. You will use the virtual machine **az1000201-vm1** in the second exercise of this lab.

> ⓘNote
>
> **Result**: After you completed this exercise, you have initiated template deployment of an Azure VM **az1000201-vm1** that you will use in the second exercise of this lab.

Exercise 1: Implement and use Azure Blob Storage

The main tasks for this exercise are as follows:

2. Review configuration settings of Azure Storage accounts
3. Manage Azure Storage Blob Service
4. Copy a container and blobs between Azure Storage accounts
5. Use a Shared Access Signature (SAS) key to access a blob

Task 1: Create Azure Storage accounts

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Storage account**.
3. Use the list of search results to navigate to the **Create storage account** blade.
4. From the **Create storage account** blade, create a new storage account with the following settings:
    - Subscription: the same subscription you selected in the previous task
    - Resource group: the name of a new resource group **az1000202-RG**
    - Storage account name: any valid, unique name between 3 and 24 characters consisting of lowercase letters and digits
    - Location: the name of the Azure region which you selected in the previous task
    - Performance: **Standard**
    - Account kind: **Storage (general purpose v1)**
    - Replication: **Locally-redundant storage (LRS)**
5. Click **Review + create**, and then click **Create**.
6. Do not wait for the storage account to be provisioned but proceed to the next step.
7. In the Azure portal, navigate to the **Create a resource** blade.
8. From the **Create a resource** blade, search Azure Marketplace for **Storage account**.
9. Use the list of search results to navigate to the **Create storage account** blade.
10. From the **Create storage account** blade, create a new storage account with the following settings:
    - Subscription: the same subscription you selected in the previous task
    - Resource group: the name of a new resource group **az1000203-RG**
    - Storage account name: any valid, unique name between 3 and 24 characters consisting of lowercase letters and digits
    - Location: the name of an Azure region different from the one you chose when creating the first storage account
    - Performance: **Standard**
    - Account kind: **StorageV2 (general purpose v2)**
    - Access tier: **Hot**
    - Replication: **Geo-redundant storage (GRS)**
11. Click **Review + create**, then click **Create**.
12. Wait for the storage account to be provisioned. This should take less than a minute.

Task 2: Review configuration settings of Azure Storage accounts

2. With your storage account blade open, review the storage account configuration in the **Overview** section, including the performance, replication, and account kind settings.

3. Display the **Access keys** blade. Note that you have the option of copying the values of storage account name, as well as the values of key1 and key2. You also have the option to regenerate each of the keys.

4. Display the **Configuration** blade of the storage account.

5. On the **Configuration** blade, note that you have the option of performing an upgrade to **General Purpose v2** account, enforcing secure transfer, and changing the replication settings to either **Geo-redundant storage (GRS)** or **Read-access geo-redundant storage (RA-GRS)**. However, you cannot change the performance setting (this setting can only be assigned when the storage account is created).

6. Display the **Encryption** blade of the storage account. Note that encryption is enabled by default and that you have the option of using your own key.

   > ⓘNote
   >
   > **Note**: Do not change the configuration of the storage account.

7. In Azure Portal, navigate to the blade of the second storage account you created.

8. With your storage account blade open, review the storage account configuration in the **Overview** section, including the performance, replication, and account kind settings.

9. Display the **Configuration** blade of the storage account.

10. On the **Configuration** blade, note that you have the option of disabling the secure transfer requirement, setting the default access tier to **Cool**, and changing the replication settings to either **Locally-redundant storage (LRS)** or **Read-access geo-redundant storage (RA-GRS)**. In this case, you also cannot change the performance setting.

11. Display the **Encryption** blade of the storage account. Note that in this case encryption is also enabled by default and that you have the option of using your own key.

Task 3: Manage Azure Storage Blob Service

1. In the Azure portal, navigate to the **Blobs** blade of the first storage account you created.

2. From the **Blobs** blade of the first storage account, create a new container named **az1000202-container** with the **Public access level** set to **Private (no anonymous access)**.

3. From the **az1000202-container** blade, upload **Labfiles\Module_03\Implement_and_Manage_Storage\az-100-02_azuredeploy.json** and **Labfiles\Module_03\Implement_and_Manage_Storage\az-100-02_azuredeploy.parameters.json** into the container.

Task 4: Copy a container and blobs between Azure Storage accounts

1. From the Azure Portal, start a PowerShell session in the Cloud Shell pane.

   > ⓘNote
   >
   > **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.

2. In the Cloud Shell pane, run the following commands:

   Code Copy

```
$containerName = 'az1000202-container'
$storageAccount1Name = (Get-AzStorageAccount -ResourceGroupName 'az1000202-RG')[0].StorageAccountName
$storageAccount2Name = (Get-AzStorageAccount -ResourceGroupName 'az1000203-RG')[0].StorageAccountName
$storageAccount1Key1 = (Get-AzStorageAccountKey -ResourceGroupName 'az1000202-RG' -StorageAccountName $storageAccount1Name)[0].Value
$storageAccount2Key1 = (Get-AzStorageAccountKey -ResourceGroupName 'az1000203-RG' -StorageAccountName $storageAccount2Name)[0].Value
$context1 = New-AzStorageContext -StorageAccountName $storageAccount1Name -StorageAccountKey $storageAccount1Key1
$context2 = New-AzStorageContext -StorageAccountName $storageAccount2Name -StorageAccountKey $storageAccount2Key1
```

> ⓘNote
>
> **Note**: These commands set the values of variables representing the names of the blob container containing the blobs you uploaded in the previous task, the two storage accounts, their corresponding keys, and the corresponding security context for each. You will use these values to generate a SAS token to copy blobs between storage accounts by using the AZCopy command line utility.

3. In the Cloud Shell pane, run the following command:

Code Copy

```
New-AzStorageContainer -Name $containerName -Context $context2 -Permission Off
```

> ⓘNote
>
> **Note**: This command creates a new container with the matching name in the second storage account

4. In the Cloud Shell pane, run the following commands:

Code Copy

```
$containerToken1 = New-AzStorageContainerSASToken -Context $context1 -ExpiryTime(get-date).AddHours(24) -FullUri -Name $containerName -Permission rwdl
$containerToken2 = New-AzStorageContainerSASToken -Context $context2 -ExpiryTime(get-date).AddHours(24) -FullUri -Name $containerName -Permission rwdl
```

> ⓘNote
>
> **Note**: These commands generate SAS keys that you will use in the next step to copy blobs between two containers.

5. In the Cloud Shell pane, run the following command:

Code Copy

```
azcopy cp $containerToken1 $containerToken2 --recursive=true
```

> ⓘNote
>
> **Note**: This command uses the AzCopy utility to copy the content of the container between the two storage accounts.

MicrosoftLearning/AZ-103-MicrosoftAzureAdministrator

7. Navigate to the **Blobs** blade of the second storage account and verify that it includes the entry representing the newly created **az1000202-container** and that the container includes two copied blobs.

Task 5: Use a Shared Access Signature (SAS) key to access a blob

1. From the **Blobs** blade of the second storage account, navigate to the container **az1000202-container**, and then open the **az-100-02_azuredeploy.json** blade.

2. On the **az-100-02_azuredeploy.json** blade, copy the value of the **URL** property.

3. Open another Microsoft Edge window and navigate to the URL you copied in the previous step.

   > ⓘNote
   >
   > **Note**: The browser will display the **ResourceNotFound**. This is expected since the container has the **Public access level** set to **Private (no anonymous access)**.

4. On the **az-100-02_azuredeploy.json** blade, generate a shared access signature (SAS) and the corresponding URL with the following settings:

   - Permissions: **Read**
   - Start date/time: specify the current date/time in your current time zone
   - Expiry date/time: specify the date/time 24 hours ahead of the current time
   - Allowed IP addresses: leave blank
   - Allowed protocols: **HTTP**
   - Signing key: **Key 1**

5. On the **az-100-02_azuredeploy.json** blade, copy **Blob SAS URL**.

6. From the previously opened Microsoft Edge window, navigate to the URL you copied in the previous step.

   > ⓘNote
   >
   > **Note**: This time, you will be prompted whether you want to open or save **az-100-02_azuredeploy.json**. This is expected as well, since this time you are no longer accessing the container anonymously, but instead you are using the newly generated SAS key, which is valid for the next 24 hours.

7. Close the Microsoft Edge window displaying the prompt.

> ⓘNote
>
> **Result**: After you completed this exercise, you have created two Azure Storage accounts, reviewed their configuration settings, created a blob container, uploaded blobs into the container, copied the container and blobs between the storage accounts, and used a SAS key to access one of the blobs.

Exercise 2: Implement and use Azure File Storage

The main tasks for this exercise are as follows:

1. Create an Azure File Service share

2. Map a drive to the Azure File Service share from an Azure VM

Task 1: Create an Azure File Service share

1. In the Azure portal, navigate to the blade displaying the properties of the second storage account you

2. From the storage account blade, display the properties of its File Service.
3. From the storage account **Files** blade, create a new file share with the following settings:
    - Name: **az10002share1**
    - Quota: **5 GB**

Task 2: Map a drive to the Azure File Service share from an Azure VM

> ⓘNote
>
> **Note**: Before you start this task, ensure that the template deployment you started in Exercise 0 has completed.

1. Navigate to the **az10002share1** blade and display the **Connect** blade.
2. From the **Connect** blade, copy into Clipboard the PowerShell commands that connect to the file share from a Windows computer.
3. In the Azure portal, navigate to the **az1000201-vm1** blade.
4. From the **az1000201-vm1** blade, connect to the Azure VM via the RDP protocol and, when prompted to sign in, provide the following credentials:
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
5. Within the RDP session, start a Windows PowerShell ISE session.
6. From the Windows PowerShell ISE session, open the script pane and paste into it the content of your local Clipboard.
7. Paste the script into the PowerShell ISE session, add `-Persist` at the end of the script, execute the script, and verify that its output confirms successful mapping of the Z: drive to the Azure Storage File Service share.
8. Start File Explorer, navigate to the Z: drive and create a folder named **Folder1**.
9. In the File Explorer window, navigate to **Folder1** and create a text document named **File1.txt**.

    > ⓘNote
    >
    > **Note**: Make sure that you take into account the default configuration of File Explorer that does not display known file extensions in order to avoid creating a file named **File1.txt.txt**.

10. From the PowerShell prompt, enter **Z:** to change the directory context to the mapped drive.
11. From the PowerShell prompt, enter **dir** to list the contents of the drive. You should see the directory that you created from File Explorer.
12. From the PowerShell prompt, enter **cd Folder1** to change directories to the folder. Run the **dir** command again to list the file contents.

> ⓘNote
>
> **Result**: After you completed this exercise, you have created an Azure File Service share, mapped a drive to the file share from an Azure VM, and used File Explorer from the Azure VM to create a folder and a file in the file share.

Exercise 3: Remove lab resources

MicrosoftLearning/AZ-103-MicrosoftAzureAdministrator

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:

   Shell Copy

   ```
   az group list --query "[?starts_with(name,'az1000')].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

   Shell Copy

   ```
   az group list --query "[?starts_with(name,'az1000')].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.

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

Lab: Configure Azure DNS

All tasks in this lab are performed from the Azure portal (including a PowerShell Cloud Shell session)

ⓘNote

Note: When not using Cloud Shell, the lab virtual machine must have the Azure PowerShell 1.2.0 module (or newer) installed https://docs.microsoft.com/en-us/powershell/azure/install-az-ps

Lab files:

- **Labfiles\Module_04\Configure_Azure_DNS\az-100-04b_01_azuredeploy.json**
- **Labfiles\Module_04\Configure_Azure_DNS\az-100-04b_02_azuredeploy.json**
- **Labfiles\Module_04\Configure_Azure_DNS\az-100-04_azuredeploy.parameters.json**

Scenario

Adatum Corporation wants to implement public and private DNS service in Azure without having to deploy its own DNS servers.

Objectives

After completing this lab, you will be able to:

- Configure Azure DNS for public domains
- Configure Azure DNS for private domains

Exercise 1: Configure Azure DNS for public domains

The main tasks for this exercise are as follows:

1. Create a public DNS zone
2. Create a DNS record in the public DNS zone
3. Validate Azure DNS-based name resolution for the public domain

Task 1: Create a public DNS zone

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 Azure subscription you intend to use in this lab.
2. In the Azure portal, navigate to the **Create a resource** blade.
3. From the **Create a resource** blade, search Azure Marketplace for **DNS zone**.
4. Select **DNS Zone**, and then click **Create**.
5. From the to **Create DNS zone** blade, create a new DNS zone with the following settings:
   - Subscription: the name of the Azure subscription you are using in this lab
   - Resource group: the name of a new resource group **az1000401b-RG**
   - Name: any unique, valid DNS domain name in the **.com** namespace
   - Resource group location: **East US** (or a supported region near you)

ⓘNote

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.

2. From your lab computer open a Powershell session, run the following in order to identify the public IP address of your lab computer:

Code Copy

```
Invoke-RestMethod http://ipinfo.io/json | Select-Object -ExpandProperty IP
```

ⓘNote

Note: Take a note of this IP address. You will use it later in this task.

3. In the Cloud Shell pane, run the following in order to create a public IP address resource:

Code Copy

```
$rg = Get-AzResourceGroup -Name az1000401b-RG

New-AzPublicIpAddress -ResourceGroupName $rg.ResourceGroupName -Sku Basic -AllocationMethod
Static -Name az1000401b-pip -Location $rg.Location
```

4. In the Azure portal, navigate to the **az1000401b-RG** resource group blade.
5. From the **az1000401b-RG** resource group blade, navigate to the blade displaying newly created public DNS zone.
6. From the DNS zone blade, click **+ Record set** to navigate to the **Add record set** blade
7. Create a DNS record with the following settings:
   - Name: **mylabvmpip**
   - Type: **A**
   - Alias record set: **No**
   - TTL: **1**
   - TTL unit: **Hours**
   - IP ADDRESS: the public IP address of your lab computer you identified earlier in this task
8. From the Overview blade, click **+ Record set**, and create another record with the following settings:
   - Name: **myazurepip**
   - Type: **A**
   - Alias record set: **Yes**
   - Alias type: **Azure resource**
   - Choose a subscription: the name of the Azure subscription you are using in this lab
   - Azure resource: **az1000401b-pip**
   - TTL: **1**

Task 3: Validate Azure DNS-based name resolution for the public domain

1. On the DNS zone blade, note the list of the name servers that host the zone you created. You will use the first of them named in the next step.

2. From the lab virtual machine, start Command Prompt and run the following to validate the name resolution of the two newly created DNS records (where <custom_DNS_domain> represents the custom DNS domain you created in the first task of this exercise and <name_server> represents the name of the DNS name server you identified in the previous step):

   Code Copy

   ```
   nslookup mylabvmpip.<custom_DNS_domain> <name_server>

   nslookup myazurepip.<custom_DNS_domain> <name_server>
   ```

3. Verify that the IP addresses returned match those you identified earlier in this task.

> ⓘNote
>
> **Result**: After you completed this exercise, you have created a public DNS zone, created a DNS record in the public DNS zone, and validated Azure DNS-based name resolution for the public domain.

Exercise 2: Configure Azure DNS for private domains

The main tasks for this exercise are as follows:

1. Provision a multi-virtual network environment

2. Create a private DNS zone

3. Deploy Azure VMs into virtual networks

4. Validate Azure DNS-based name reservation and resolution for the private domain

Task 1: Provision a multi-virtual network environment

1. From the Azure Portal, start a PowerShell session in the Cloud Shell.

2. In the Cloud Shell pane, run the following in order to create a resource group:

   Code Copy

   ```
   $rg1 = Get-AzResourceGroup -Name 'az1000401b-RG'

   $rg2 = New-AzResourceGroup -Name 'az1000402b-RG' -Location $rg1.Location
   ```

3. In the Cloud Shell pane, run the following in order to create two Azure virtual networks:

   Code Copy

   ```
   $subnet1 = New-AzVirtualNetworkSubnetConfig -Name subnet1 -AddressPrefix '10.104.0.0/24'

   $vnet1 = New-AzVirtualNetwork -ResourceGroupName $rg2.ResourceGroupName -Location
   $rg2.Location -Name az1000402b-vnet1 -AddressPrefix 10.104.0.0/16 -Subnet $subnet1

   $subnet2 = New-AzVirtualNetworkSubnetConfig -Name subnet1 -AddressPrefix '10.204.0.0/24'

   $vnet2 = New-AzVirtualNetwork -ResourceGroupName $rg2.ResourceGroupName -Location
   $rg2.Location -Name az1000402b-vnet2 -AddressPrefix 10.204.0.0/16 -Subnet $subnet2
   ```

Task 2: Create a private DNS zone

1. In the Cloud Shell pane, run the following in order to create a private DNS zone with the first virtual network supporting registration and the second virtual network supporting resolution:

   Code Copy

   ```
   $vnet1 = Get-AzVirtualNetwork -Name az1000402b-vnet1

   $vnet2 = Get-AzVirtualNetwork -name az1000402b-vnet2

   New-AzDnsZone -Name adatum.local -ResourceGroupName $rg2.ResourceGroupName -ZoneType Private -RegistrationVirtualNetworkId @($vnet1.Id) -ResolutionVirtualNetworkId @($vnet2.Id)
   ```

   > ⓘNote
   >
   > **Note**: Virtual networks that you assign to an Azure DNS zone cannot contain any resources.

2. In the Cloud Shell pane, run the following in order to verify that the private DNS zone was successfully created:

   Code Copy

   ```
   Get-AzDnsZone -ResourceGroupName $rg2.ResourceGroupName
   ```

Task 3: Deploy Azure VMs into virtual networks

1. In the Cloud Shell pane, upload **az-100-04b_01_azuredeploy.json**, **az-100-04b_02_azuredeploy.json**, and **az-100-04_azuredeploy.parameters.json** files.

2. In the Cloud Shell pane, run the following in order to deploy an Azure VM into the first virtual network:

   Code Copy

   ```
   cd $home

   New-AzResourceGroupDeployment -ResourceGroupName $rg2.ResourceGroupName -TemplateFile "./az-100-04b_01_azuredeploy.json" -TemplateParameterFile "./az-100-04_azuredeploy.parameters.json" -AsJob
   ```

3. In the Cloud Shell pane, run the following in order to deploy an Azure VM into the second virtual network:

   Code Copy

   ```
   New-AzResourceGroupDeployment -ResourceGroupName $rg2.ResourceGroupName -TemplateFile "./az-100-04b_02_azuredeploy.json" -TemplateParameterFile "./az-100-04_azuredeploy.parameters.json" -AsJob
   ```

   > ⓘNote
   >
   > **Note**: Wait for both deployments to complete before you proceed to the next task. You can identify the state of the jobs by running the `Get-Job` cmdlet in the Cloud Shell pane.

Task 4: Validate Azure DNS-based name reservation and resolution for the private domain

1. In the Azure portal, navigate to the blade of the **az1000402b-vm2** Azure VM.

3. When prompted, authenticate by specifying the following credentials:
   - User name: **Student**
   - Password: **Pa55w.rd1234**
4. Within the Remote Desktop session to **az1000402b-vm2**, start a Command Prompt window and run the following:

   Code Copy

   ```
   nslookup az1000402b-vm1.adatum.local
   ```

5. Verify that the name is successfully resolved.
6. Switch back to the lab virtual machine and, in the Cloud Shell pane of the Azure portal window, run the following in order to create an additional DNS record in the private DNS zone:

   Code Copy

   ```
   New-AzDnsRecordSet -ResourceGroupName $rg2.ResourceGroupName -Name www -RecordType A -
   ZoneName adatum.local -Ttl 3600 -DnsRecords (New-AzDnsRecordConfig -IPv4Address "10.104.0.4")
   ```

7. Switch again to the Remote Desktop session to **az1000402b-vm2** and run the following from the Command Prompt window:

   Code Copy

   ```
   nslookup www.adatum.local
   ```

8. Verify that the name is successfully resolved.

ⓘNote

Result: After completing this exercise, you have provisioned a multi-virtual network environment, created a private DNS zone, deployed Azure VMs into virtual networks, and validated Azure DNS-based name reservation and resolution for the private domain

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:

   Shell Copy

   ```
   az group list --query "[?starts_with(name,'az1000')].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

   Shell Copy

   ```
   az group list --query "[?starts_with(name,'az1000')].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.

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

Lab: VNet Peering and Service Chaining

All tasks in this lab are performed from the Azure portal except for Exercise 2 Task 3, Exercise 3 Task 1, and Exercise 3 Task 2, which include steps performed from a Remote Desktop session to an Azure VM

Lab files:

- **Labfiles\Module_05\VNet_Peering_and_Service_Chaining\az-100-04_01_azuredeploy.json**
- **Labfiles\Module_05\VNet_Peering_and_Service_Chaining\az-100-04_02_azuredeploy.json**
- **Labfiles\Module_05\VNet_Peering_and_Service_Chaining\az-100-04_azuredeploy.parameters.json**

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:

- Create Azure virtual networks and deploy Azure VM by using Azure Resource Manager templates.
- Configure VNet peering.
- Implement custom routing
- Validate service chaining

Exercise 0: Prepare the Azure environment

The main tasks for this exercise are as follows:

1. Create the first virtual network hosting two Azure VMs by using an Azure Resource Manager template
2. Create the second virtual network in the same region hosting a single Azure VM by using an Azure Resource Manager template

Task 1: Create the first virtual network hosting two Azure VMs 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 Azure subscription you intend to use in this lab.
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 **Template deployment (deploy using custom templates)** blade, and then click **Create**.
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_05\VNet_Peering_and_Service_Chaining\az-100-04_01_azuredeploy.json**.

> ⓘNote
>
> **Note**: Review the content of the template and note that it defines deployment of an Azure VM hosting Windows Server 2016 Datacenter.

8. From the **Custom deployment** blade, navigate to the **Edit parameters** blade.

9. From the **Edit parameters** blade, load the parameters file **Labfiles\Module_05\VNet_Peering_and_Service_Chaining\az-100-04_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 are using in this lab
    - Resource group: the name of a new resource group **az1000401-RG**
    - Location: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs
    - Vm Size: **Standard_DS2_v2**
    - Vm1Name: **az1000401-vm1**
    - Vm2Name: **az1000401-vm2**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1000401-vnet1**

    ⓘNote

    Note: To identify Azure regions where you can provision Azure VMs, refer to https://azure.microsoft.com/en-us/regions/offers/

    ⓘNote

    Note: Do not wait for the deployment to complete but proceed to the next task. You will use the network and the virtual machines included in this deployment in the second exercise of this lab.

Task 2: Create the second virtual network in the same region hosting a single Azure VM 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 and select the **Template deployment (deploy using custom templates)** result, and then click **Create**.

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_05\VNet_Peering_and_Service_Chaining\az-100-04_02_azuredeploy.json**.

    ⓘNote

    Note: Review the content of the template and note that it defines deployment of an Azure VM hosting Windows Server 2016 Datacenter.

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_05\VNet_Peering_and_Service_Chaining\az-100-04_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 **az1000402-RG**
    - Location: the name of the Azure region which you selected in the previous task
    - Vm Size: **Standard_DS2_v2**
    - VmName: **az1000402-vm3**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1000402-vnet2**

    > ⓘNote
    >
    > **Note**: Do not wait for the deployment to complete but proceed to the next task. You will use the network and the virtual machines included in this deployment in the second exercise of this lab.

> ⓘNote
>
> **Result**: After you completed this exercise, you have created two Azure virtual networks and initiated deployments of three Azure VM by using Azure Resource Manager templates.

Exercise 1: Configure VNet peering

The main tasks for this exercise are as follows:

1. Configure VNet peering for the first virtual network

2. Configure VNet peering for the second virtual network

Task 1: Configure VNet peering for the first virtual network

1. In the Azure portal, navigate to the **az1000401-vnet1** virtual network blade.

2. From the **az1000401-vnet1** virtual network blade, display its **Peerings** blade.

3. From the **az1000401-vnet1 - Peerings** blade, click **+ Add** to create a VNet peering with the following settings:

    - Name: **az1000401-vnet1-to-az1000402-vnet2**
    - Virtual network deployment model: **Resource manager**
    - Subscription: the name of the Azure subscription you are using in this lab
    - Virtual network: **az1000402-vnet2**
    - Name of peering from az1000402-vnet2 to az1000401-vnet1: **az1000402-vnet2-to-az1000401-vnet1**
    - Allow virtual network access: **Enabled**
    - Allow forwarded traffic: **disabled**

> ⓘNote
>
> **Note**: Because you have administrative access to both virtual networks, the portal is configuring both directions (from vnet1 to vnet2, AND vnet2 to vnet1) in a single action. From the CLI, PowerShell, or REST API, these tasks must be performed independently.

Exercise 2: Implement custom routing

The main tasks for this exercise are as follows:

1. Enable IP forwarding for a network interface of an Azure VM
2. Configure user defined routing
3. Configure routing in an Azure VM running Windows Server 2016

Task 1: Enable IP forwarding for a network interface of an Azure VM

> ⓘNote
>
> **Note**: Before you start this task, ensure that the template deployments you started in Exercise 0 have completed.

1. In the Azure portal, navigate to the blade of the second Azure VM **az1000401-vm2**.
2. From the **az1000401-vm2** blade, display its **Networking** blade.
3. From the **az1000401-vm2 - Networking** blade, display the blade of the network adapter (**az1000401-nic2**) of the Azure VM.
4. From the **az1000401-nic2** blade, display its **IP configurations** blade.
5. From the **az1000401-nic2 - IP configurations** set IP forwarding to **Enabled**, and then click **Save**.

   > ⓘNote
   >
   > **Note**: The Azure VM **az1000401-vm2**, which network interface you configured in this task, will function as a router, facilitating service chaining between the two virtual networks.

Task 2: Configure user defined routing

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Route table**.
3. Select **Route table**, and then click **Create**.
4. From the **Create route table** blade, create a new route table with the following settings:
   - Name: **az1000402-rt1**
   - Subscription: the name of the Azure subscription you use for this lab
   - Resource group: **az1000402-RG**
   - Location: the same Azure region in which you created the virtual networks
   - Virtual network gateway route propagation: **Disabled**
5. In the Azure portal, navigate to the **az1000402-rt1** blade.
6. From the **az1000402-rt1** blade, display its **Routes** blade.

- Route name: **custom-route-to-az1000401-vnet1**
- Address prefix: **10.104.0.0/16**
- Next hop type: **Virtual appliance**
- Next hop address: **10.104.1.4**

> ⓘNote
>
> **Note**: **10.104.1.4** is the IP address of the network interface of **az1000401-vm2**, which will provide service chaining between the two virtual networks.

8. From the **az1000402-rt1** blade, display its **Subnets** blade.

9. From the **az1000402-rt1 - Subnets** blade, associate the route table **az1000402-rt1** with **subnet0** of **az1000402-vnet2**.

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

1. In the Azure portal, navigate to the blade of the **az1000401-vm2** Azure VM.

2. From the **Overview** pane of the **az1000401-vm2** blade, generate an RDP file and use it to connect to **az1000401-vm2**.

3. When prompted, authenticate by specifying the following credentials:

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

4. Within the Remote Desktop session to **az1000401-vm2**, from **Server Manager**, select **Manage** use the **Add Roles and Features Wizard**

5. Click **Next** twice, ensure **az1000401-vm2** is selected and click **Next**, select the **Remote Access** server role then click **Next** three times, Select the **Routing** role service, select **Add Features** and all required features. Select **Next** three times, click **Install**. Click **Close** when the installation is complete.

> ⓘNote
>
> **Note**: If you receive an error message **There may be a version mismatch between this computer and the destination server or VHD** once you select the **Remote Access** checkbox on the **Server Roles** page of the **Add Roles and Features Wizard**, clear the checkbox, click **Next**, click **Previous** and select the **Remote Access** checkbox again.

6. Within the Remote Desktop session to **az1000401-vm2**, from Server Manager, select **Tools** start the **Routing and Remote Access** console.

7. In the **Routing and Remote Access** console, right click on the server name and select **Configure and Enable Routing and Remote Access**, Select **Next** use the **Custom configuration** then **Next**, enable **LAN routing** then **Next**, click **Finish** and the click **Start Service**.
8. Within the Remote Desktop session to **az1000401-vm2**, start the **Windows Firewall with Advanced Security** console and enable **File and Printer Sharing (Echo Request - ICMPv4-In)** inbound rule for all profiles.

> ⓘNote
>
> **Result**: After completing this exercise, you have implemented custom routing between peered Azure virtual networks.

Exercise 3: Validating service chaining

The main tasks for this exercise are as follows:

2. Test service chaining between peered virtual networks

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

1. In the Azure portal, navigate to the blade of the **az1000401-vm1** Azure VM.

2. From the **Overview** pane of the **az1000401-vm1** blade, generate an RDP file and use it to connect to **az1000401-vm1**.

3. When prompted, authenticate by specifying the following credentials:

   - User name: **Student**

   - Password: **Pa55w.rd1234**

4. Within the Remote Desktop session to **az1000401-vm1**, open 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. In the Azure portal, navigate to the blade of the **az1000402-vm3** Azure VM.

2. From the **Overview** pane of the **az1000402-vm3** blade, generate an RDP file and use it to connect to **az1000402-vm3**.

3. When prompted, authenticate by specifying the following credentials:

   - User name: **Student**

   - Password: **Pa55w.rd1234**

4. Once you are connected to **az1-1000402-vm3** via the Remote Desktop session, start **Windows PowerShell**.

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

   Code Copy

   ```
   Test-NetConnection -ComputerName 10.104.0.4 -TraceRoute
   ```

   ⓘNote

   Note: **10.104.0.4** is the IP address of the network interface of the first Azure VM **az1000401-vm1**

6. Verify that test is successful and note that the connection was routed over **10.104.1.4**

   ⓘNote

   Note: Without custom routing in place, the traffic would flow directly between the two Azure VMs. **Result**: After you completed this exercise, you have validated service chaining between peered Azure virtual networks.

Exercise 4: 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

Shell Copy

```
az group list --query "[?starts_with(name,'az1000')].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

Shell Copy

```
az group list --query "[?starts_with(name,'az1000')].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.

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

Lab: Use Azure Network Watcher for monitoring and troubleshooting network connectivity

All tasks in this lab are performed from the Azure portal (including a PowerShell Cloud Shell session)

ⓘNote

Note: When not using Cloud Shell, the lab virtual machine must have the Azure PowerShell 1.2.0 module (or newer) installedhttps://docs.microsoft.com/en-us/powershell/azure/install-az-ps

Lab files:

- **Labfiles\Module_06\Network_Watcher\az-101-03b_01_azuredeploy.json**
- **Labfiles\Module_06\Network_Watcher\az-101-03b_02_azuredeploy.json**
- **Labfiles\Module_06\Network_Watcher\az-101-03b_01_azuredeploy.parameters.json**
- **Labfiles\Module_06\Network_Watcher\az-101-03b_02_azuredeploy.parameters.json**

Scenario

Adatum Corporation wants to monitor Azure virtual network connectivity by using Azure Network Watcher.

Objectives

After completing this lab, you will be able to:

- Deploy Azure VMs, Azure storage accounts, and Azure SQL Database instances by using Azure Resource Manager templates
- Use Azure Network Watcher to monitor network connectivity

Exercise 1: Prepare infrastructure for Azure Network Watcher-based monitoring

The main tasks for this exercise are as follows:

1. Deploy Azure VMs, an Azure Storage account, and an Azure SQL Database instance by using an Azure Resource Manager template
2. Enable Azure Network Watcher service
3. Establish peering between Azure virtual networks
4. Establish service endpoints to an Azure Storage account and Azure SQL Database instance

Task 1: Deploy Azure VMs, an Azure Storage account, and an Azure SQL Database instance by using Azure Resource Manager templates

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. In the list of results, click **Template deployment (deploy using custom templates)**, and then click **Create**.
5. On the **Custom deployment** blade, click the **Build your own template in the editor** link. If you do not

6. From the **Edit template** blade, load the template file **az-101-03b_01_azuredeploy.json**.

> ⓘNote
>
> **Note**: Review the content of the template and note that it defines deployment of an Azure VM, an Azure SQL Database, and an Azure Storage account.

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 **az-101-03b_01_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 **az1010301b-RG**
    - Location: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs and Azure SQL Database
    - Vm Size: **Standard_DS2_v2**
    - Vm Name: **az1010301b-vm1**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1010301b-vnet1**
    - Sql Login Name: **Student**
    - Sql Login Password: **Pa55w.rd1234**
    - Database Name: **az1010301b-db1**
    - Sku Name: **Basic**
    - Sku Tier: **Basic**

> ⓘNote
>
> **Note**: To identify VM sizes available in your subscription in a given region, run the following from Cloud Shell and review the values in the **Restriction** column (where <location> represents the target Azure region):

Code Copy

```
Get-AzComputeResourceSku | where {$_.Locations -icontains "<location>"} | Where-Object
{($_.ResourceType -ilike "virtualMachines")}
```

> ⓘNote
>
> **Note**: To identify whether you can provision Azure SQL Database in a given region, run the following from Cloud Shell and ensure that the resulting **Status** is set to **Available** (where <location> represents the target Azure region):

Code Copy

```
Get-AzSqlCapability -LocationName <regionname>
```

> ⓘNote
>
> **Note**: Do not wait for the deployment to complete but proceed to the next step.

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

13. From the **Create a resource** blade, search Azure Marketplace for **Template deployment**.

14. In the results, click **Template deployment (deploy using custom templates)**, and then click **Create**.

15. 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.

16. From the **Edit template** blade, load the template file **az-101-03b_02_azuredeploy.json**.

> ⓘNote
>
> **Note**: Review the content of the template and note that it defines deployment of an Azure VM.

17. Save the template and return to the **Custom deployment** blade.

18. From the **Custom deployment** blade, navigate to the **Edit parameters** blade.

19. From the **Edit parameters** blade, load the parameters file **az-101-03b_02_azuredeploy.parameters.json**.

20. Save the parameters and return to the **Custom deployment** blade.

21. 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 **az1010302b-RG**
    - Location: the name of an Azure region where you can provision Azure VMs, but which is **different** from the one you selected during previous deployment,
    - Vm Size: **Standard_DS2_v2**
    - Vm Name: **az1010302b-vm2**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1010302b-vnet2**

> ⓘNote
>
> **Note**: Make sure to choose a different Azure region for this deployment

> ⓘNote
>
> **Note**: Do not wait for the deployment to complete but proceed to the next step.

Task 2: Enable Azure Network Watcher service

1. In the Azure portal, use the search text box on the **All services** blade to navigate to the **Network Watcher** blade.

2. On the **Network Watcher** blade, verify that Network Watcher is enabled in both Azure regions into which you deployed resources in the previous task and, if not, enable it.

Task 3: Establish peering between Azure virtual networks

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

1. In the Azure portal, navigate to the **az1010301b-vnet1** virtual network blade.
2. From the **az1010301b-vnet1** virtual network blade, display the **az1010301b-vnet1 - Peerings** blade.
3. From the **az1010301b-vnet1 - Peerings** blade, create a VNet peering with the following settings:
   - Name: **az1010301b-vnet1-to-az1010302b-vnet2**
   - Virtual network deployment model: **Resource manager**
   - Subscription: the name of the Azure subscription you are using in this lab
   - Virtual network: **az1010302b-vnet2**
   - Name of peering from az1010302b-vnet2 to az1010301b-vnet1: **az1010302b-vnet2-to-az1010301b-vnet1**
   - Allow virtual network access: **Enabled**
   - Allow forwarded traffic: **disabled**
   - Allow gateway transit: disabled

   > ⓘNote
   >
   > **Note**: The Azure portal allows you to configure both directions of the peering simultaneously. When using other management tools, each direction must be configured independently.

Task 4: Establish service endpoints to an Azure Storage account and Azure SQL Database instance

1. In the Azure portal, navigate to the **az1010301b-vnet1** virtual network blade.
2. From the **az1010301b-vnet1** virtual network blade, display the **Service endpoints** blade.
3. From the **az1010301b-vnet1 - Service endpoints** blade, add a service endpoint with the following settings:
   - Service: **Microsoft.Storage**
   - Subnets: **subnet0**
4. Repeat the step to create a second service endpoint:
   - Service: **Microsoft.Sql**
   - Subnets: **subnet0**
5. In the Azure portal, navigate to the **az1010301b-RG** resource group blade.
6. From the **az1010301b-RG** resource group blade, navigate to the blade of the storage account included in the resource group.
7. From the storage account blade, navigate to its **Firewalls and virtual networks** blade.
8. From the **Firewalls and virtual networks** blade of the storage account, configure the following settings:
   - Allow access from: **Selected networks**
   - Virtual networks:

        - SUBNET: **subnet0**
    - Firewall:
        - ADDRESS RANGE: none
    - Exceptions:
        - Allow trusted Microsoft services to access this storage account: **Enabled**
        - Allow read access to storage logging from any network: **Disabled**
        - Allow read access to storage metrics from any network: **Disabled**

9. In the Azure portal, navigate to the **az1010301b-RG** resource group blade.
10. From the **az1010301b-RG** resource group blade, navigate to the **az1010301b** Azure SQL Server blade.
11. From the Azure SQL Server blade, navigate to its server's **Firewalls and virtual networks** blade.
12. From the **Firewalls and virtual networks** blade of the Azure SQL Database server, configure the following settings:
    - Allow access to Azure services: **ON**
    - No firewall rules configured
    - Virtual networks:
        - Name: **az1010301b-vnet1**
        - Subscription: the name of the subscription you are using in this lab
        - Virtual network: **az1010301b-vnet1**
        - Subnet name: **subnet0/ 10.203.0.0/24**

> ⓘNote
>
> **Result**: After you completed this exercise, you have deployed Azure VMs, an Azure Storage account, and an Azure SQL Database instance by using Azure Resource Manager templates, enabled Azure Network Watcher service, established global peering between Azure virtual networks, and established service endpoints to an Azure Storage account and Azure SQL Database instance.

Exercise 2: Use Azure Network Watcher to monitor network connectivity

The main tasks for this exercise are as follows:

1. Test network connectivity to an Azure VM via virtual network peering by using Network Watcher
2. Test network connectivity to an Azure Storage account by using Network Watcher
3. Test network connectivity to an Azure SQL Database by using Network Watcher

Task 1: Test network connectivity to an Azure VM via virtual network peering by using Network Watcher

1. In the Azure portal, navigate to the **Network Watcher** blade.
2. From the **Network Watcher** blade, navigate to the **Connection troubleshoot**.
3. On the **Network Watcher - Connection troubleshoot** blade, initiate a check with the following settings:
    - Source:
        - Subscription: the name of the Azure subscription you are using in this lab
        - Resource group: **az1010301b-RG**

    - Virtual machine: **az1010301b-vm1**
  - Destination: **Specify manually**
    - URI, FQDN or IPv4: **10.203.16.4**

    > ⓘNote
    >
    > **Note**: **10.203.16.4** is the private IP address of the second Azure VM az1010302b-vm1 which you deployed to another Azure region

  - Probe Settings:
    - Protocol: **TCP**
    - Destination port: **3389**
  - Advanced settings:
    - Source port: blank

4. Wait until results of the connectivity check are returned and 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
   >
   > **Note**: If this is the first time you are using Network Watcher, the check can take up to 5 minutes.

Task 2: Test network connectivity to an Azure Storage account by using Network Watcher

1. From the Azure Portal, start a PowerShell session in the Cloud Shell.

   > ⓘNote
   >
   > **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.

2. In the Cloud Shell pane, run the following command to identify the IP address of the blob service endpoint of the Azure Storage account you provisioned in the previous exercise:

   Code Copy

   ```
   [System.Net.Dns]::GetHostAddresses($(Get-AzStorageAccount -ResourceGroupName 'az1010301b-RG')
   [0].StorageAccountName + '.blob.core.windows.net').IPAddressToString
   ```

3. Note the resulting string and, from the **Network Watcher - Connection troubleshoot** blade, initiate a check with the following settings:
   - Source:
     - Subscription: the name of the Azure subscription you are using in this lab
     - Resource group: **az1010301b-RG**
     - Source type: **Virtual machine**
     - Virtual machine: **az1010301b-vm1**
   - Destination: **Specify manually**

- Probe Settings:
    - Protocol: **TCP**
    - Destination port: **443**
- Advanced settings:
    - Source port: blank

4. Wait until results of the connectivity check are returned and 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, with minimal latency.

    > ⓘNote
    >
    > **Note**: The connection takes place over the service endpoint you created in the previous exercise. To verify this, you will use the **Next hop** tool of Network Watcher.

5. From the **Network Watcher - Connection troubleshoot** blade, navigate to the **Network Watcher - Next hop** blade and test next hop with the following settings:
    - Subscription: the name of the Azure subscription you are using in this lab
    - Resource group: **az1010301b-RG**
    - Virtual machine: **az1010301b-vm1**
    - Network interface: **az1010301b-nic1**
    - Source IP address: **10.203.0.4**
    - Destination IP address: the IP address of the blob service endpoint of the storage account you identified earlier in this task
6. Verify that the result identifies the next hop type as **VirtualNetworkServiceEndpoint**
7. From the **Network Watcher - Connection troubleshoot** blade, initiate a check with the following settings:
    - Source:
        - Subscription: the name of the Azure subscription you are using in this lab
        - Resource group: **az1010302b-RG**
        - Source type: **Virtual machine**
        - Virtual machine: **az1010302b-vm2**
    - Destination: **Specify manually**
        - URI, FQDN or IPv4: the IP address of the blob service endpoint of the storage account you identified earlier in this task
    - Probe Settings:
        - Protocol: **TCP**
        - Destination port: **443**
    - Advanced settings:
        - Source port: blank
8. Wait until results of the connectivity check are returned and verify that the status is **Reachable**.

MicrosoftLearning/AZ-103-MicrosoftAzureAdministrator

> ⓘNote
>
> **Note**: The connection is successful, however it is established over Internet. To verify this, you will use again the **Next hop** tool of Network Watcher.

9. From the **Network Watcher - Connection troubleshoot** blade, navigate to the **Network Watcher - Next hop** blade and test next hop with the following settings:
    - Subscription: the name of the Azure subscription you are using in this lab
    - Resource group: **az1010302b-RG**
    - Virtual machine: **az1010302b-vm2**
    - Network interface: **az1010302b-nic1**
    - Source IP address: **10.203.16.4**
    - Destination IP address: the IP address of the blob service endpoint of the storage account you identified earlier in this task
10. Verify that the result identifies the next hop type as **Internet**

Task 3: Test network connectivity to an Azure SQL Database by using Network Watcher

1. From the Azure Portal, start a PowerShell session in the Cloud Shell.
2. In the Cloud Shell pane, run the following command to identify the IP address of the Azure SQL Database server you provisioned in the previous exercise:

```
[System.Net.Dns]::GetHostAddresses($(Get-AzSqlServer -ResourceGroupName 'az1010301b-RG')
[0].FullyQualifiedDomainName).IPAddressToString
```

3. Note the resulting string and, from the **Network Watcher - Connection troubleshoot** blade, initiate a check with the following settings:
    - Source:
        - Subscription: the name of the Azure subscription you are using in this lab
        - Resource group: **az1010301b-RG**
        - Source type: **Virtual machine**
        - Virtual machine: **az1010301b-vm1**
    - Destination: **Specify manually**
        - URI, FQDN or IPv4: the IP address of the Azure SQL Database server you identified in the previous step of this task
    - Probe Settings:
        - Protocol: **TCP**
        - Destination port: **1433**
    - Advanced settings:
        - Source port: blank
4. Wait until results of the connectivity check are returned and 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, with

> ⓘNote
>
> **Note**: The connection takes place over the service endpoint you created in the previous exercise. To verify this, you will use the **Next hop** tool of Network Watcher.

5. From the **Network Watcher - Connection troubleshoot** blade, navigate to the **Network Watcher - Next hop** blade and test next hop with the following settings:
    - Subscription: the name of the Azure subscription you are using in this lab
    - Resource group: **az1010301b-RG**
    - Virtual machine: **az1010301b-vm1**
    - Network interface: **az1010301b-nic1**
    - Source IP address: **10.203.0.4**
    - Destination IP address: the IP address of the Azure SQL Database server you identified earlier in this task
6. Verify that the result identifies the next hop type as **VirtualNetworkServiceEndpoint**
7. From the **Network Watcher - Connection troubleshoot** blade, initiate a check with the following settings:
    - Source:
        - Subscription: the name of the Azure subscription you are using in this lab
        - Resource group: **az1010302b-RG**
        - Source type: **Virtual machine**
        - Virtual machine: **az1010302b-vm2**
    - Destination: **Specify manually**
        - URI, FQDN or IPv4: the IP address of the Azure SQL Database server you identified earlier in this task
    - Probe Settings:
        - Protocol: **TCP**
        - Destination port: **1433**
    - Advanced settings:
        - Source port: blank
8. Wait until results of the connectivity check are returned and verify that the status is **Reachable**.

> ⓘNote
>
> **Note**: The connection is successful, however it is established over Internet. To verify this, you will use again the **Next hop** tool of Network Watcher.

9. From the **Network Watcher - Connection troubleshoot** blade, navigate to the **Network Watcher - Next hop** blade and test next hop with the following settings:
    - Subscription: the name of the Azure subscription you are using in this lab
    - Resource group: **az1010302b-RG**
    - Virtual machine: **az1010302b-vm2**

   - Source IP address: **10.203.16.4**
   - Destination IP address: the IP address of the Azure SQL Database server you identified earlier in this task

10. Verify that the result identifies the next hop type as **Internet**

> ⓘNote
>
> **Result**: After you completed this exercise, you have used Azure Network Watcher to test network connectivity to an Azure VM via virtual network peering, network connectivity to Azure Storage, and network connectivity to Azure SQL Database.

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:

   Shell Copy

   ```
   az group list --query "[?starts_with(name,'az1010')].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

   Shell Copy

   ```
   az group list --query "[?starts_with(name,'az1010')].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.

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

MicrosoftLearning/AZ-103-MicrosoftAzureAdministrator

Lab: Implement Azure Site Recovery between Azure regions

All tasks in this lab are performed from the Azure portal

Lab files:

- **Labfiles\Module_07\Azure_Site_Recovery_Between_Regions\az-101-01_azuredeploy.json**
- **Labfiles\Module_07\Azure_Site_Recovery_Between_Regions\az-101-01_azuredeploy.parameters.json**

Scenario

Adatum Corporation wants to implement Azure Site Recovery to facilitate migration and protection of Azure VMs between regions

Objectives

After completing this lab, you will be able to:

- Implement Azure Site Recovery Vault
- Configure replication of Azure VMs between Azure regions by using Azure Site Recovery

Exercise 1: Implement prerequisites for migration of Azure VMs by using Azure Site Recovery

The main tasks for this exercise are as follows:

1. Deploy an Azure VM to be migrated by using an Azure Resource Manager template
2. Create an Azure Recovery Services vault

Task 1: Deploy an Azure VM to be migrated 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 Azure subscription you intend to use in this lab.
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_07\Azure_Site_Recovery_Between_Regions\az-101-01_azuredeploy.json**.

   > ⓘNote
   >
   > **Note**: Review the content of the template and note that it defines deployment of an Azure VM hosting Windows Server 2016 Datacenter.

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_07\Azure_Site_Recovery_Between_Regions\az-101-01_azuredeploy.parameters.json**.

11. 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 **az1010101-RG**
    - Location: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs
    - Vm Name: **az1010101-vm**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Image Publisher: **MicrosoftWindowsServer**
    - Image Offer: **WindowsServer**
    - Image SKU: **2016-Datacenter-Server-Core-smalldisk**
    - Vm Size: **Standard_DS1_v2**

    > ⓘNote
    >
    > **Note**: To identify Azure regions where you can provision Azure VMs, refer to **https://azure.microsoft.com/en-us/regions/offers/**

    > ⓘNote
    >
    > **Note**: Do not wait for the deployment to complete but proceed to the next task. You will use the virtual machine **az1010101-vm** in the second exercise of this lab.

Task 2: Implement an Azure Site Recovery vault

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Backup and Site Recovery**.
3. Use the list of search results to navigate to the **Recovery Services vault** blade.
4. Use the **Recovery Services vault** blade, to create a Site Recovery vault with the following settings:

    - Name: **vaultaz1010102**
    - Subscription: the same Azure subscription you used in the previous task of this exercise
    - Resource group: the name of a new resource group **az1010102-RG**
    - Location: the name of an Azure region that is available in your subscription and which is different from the region you deployed the Azure VM in the previous task of this exercise

> ⓘNote
>
> **Result**: After you completed this exercise, you have initiated deployment of an Azure VM by using an Azure Resource Manager template and created an Azure Site Recovery vault that will be used to replicate content of the Azure VM disk files.

Exercise 2: Migrate an Azure VM between Azure regions by using Azure Site Recovery

The main tasks for this exercise are as follows:

1. Configure Azure VM replication

Task 1: Configure Azure VM replication

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

1. In the Azure portal, navigate to the blade of the newly provisioned Azure Recovery Services vault **vaultaz1010102**.
2. From the **vaultaz1010102** blade, configure the following replication settings:
   - Source: **Azure**
   - Source location: the same Azure region into which you deployed the Azure VM in the previous exercise of this lab
   - Azure virtual machine deployment model: **Resource Manager**
   - Source subscription: the same Azure subscription you used in the previous exercise of this lab
   - Source resource group: **az1010101-RG**
   - Virtual machines: **az1010101-vm**
   - Target location: the name of an **Azure region** that is available in your subscription and which is **different from the region you deployed an Azure VM** in the previous task. If possible, use the same Azure region into which you deployed the Azure Site Recovery vault.
   - Target resource group: **(new) az1010101-RG-asr**
   - Target virtual network: **(new) az1010101-vnet-asr**
   - Cache storage account: accept the default setting
   - Replica managed disks: **(new) 1 premium disk(s), 0 standard disk(s)**
   - Target availability sets: **Not Applicable**
   - Replication policy: the name of a new replication policy **12-hour-retention-policy**
   - Recovery point retention: **12 Hours**
   - App consistent snapshot frequency: **6 Hours**
   - Multi-VM consistency: **No**
3. From the **Configure settings** blade, initiate creation of target resources and wait until you are redirected to the **Enable replication** blade.
4. From the **Enable replication** blade, enable the replication.

Task 2: Review Azure VM replication settings

1. In the Azure portal, navigate to the **vaultaz1010102 - Replicated items** blade.
2. On the **vaultaz1010102 - Replicated items** blade, ensure that there is an entry representing the **az1010101-vm** Azure VM and verify that its **REPLICATION HEALTH** is **Healthy** and that its **STATUS** is **Enabling protection**.
3. From the **vaultaz1010102 - Replicated items** blade, display the replicated item blade of the **az1010101-vm** Azure VM.
4. On the **az1010101-vm** replicated item blade, review the **Health and status**, **Failover readiness**, **Latest recovery points**, and **Infrastructure view** sections. Note the **Failover** and **Test Failover** toolbar icons.

MicrosoftLearning/AZ-103-MicrosoftAzureAdministrator

> ⓘNote
>
> **Note**: The remaining steps of this task are optional and not graded.

5. If time permits, wait until the replication status changes to **100% synchronized**. This might take additional 90 minutes.

6. Examine the values of **RPO**, as well as **Crash-consistent** and **App-consistent** recovery points.

7. Perform a test failover to the **az1010101-vnet-asr** virtual network.

> ⓘNote
>
> **Result**: After you completed this exercise, you have configured replication of an Azure VM and reviewed Azure VM replication settings.

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:

   Shell

   ```shell
   az group list --query "[?starts_with(name,'az10101')].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

   Shell

   ```shell
   az group list --query "[?starts_with(name,'az10101')].name" --output tsv | xargs -L1 bash -c 'az group delete --name $0 --no-wait --yes'
   ```

ⓘNote

Note: If you encounter an error similar to "…cannot perform delete operation because following scope(s) are locked…" then you need to run the following steps to remove the lock on the resource that prevents its deletion:

Shell Copy

```shell
lockedresource=$(az resource list --resource-group az1010101-RG-asr --resource-type Microsoft.Compute/disks --query "[?starts_with(name,'az10101')].name" --output tsv)
az disk revoke-access -n $lockedresource --resource-group az1010101-RG-asr
lockid=$(az lock show --name ASR-Lock --resource-group az1010101-RG-asr --resource-type Microsoft.Compute/disks --resource-name $lockedresource --output tsv --query id)
az lock delete --ids $lockid
az group list --query "[?starts_with(name,'az10101')].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.

ⓘNote

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

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
>
> **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: **Standard_D2s_v3**
    - 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
    >
    > **Note**: To identify Azure regions where you can provision Azure VMs, refer to **https://azure.microsoft.com/en-us/regions/offers/**

    > ⓘNote
    >
    > **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

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
   >
   > **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: **Standard_D2s_v3**
    - 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-**

- Configuration Function: **ContosoWebsite.ps1\ContosoWebsite**

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

> ⓘNote
>
> **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
>
> **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:
    - Name: **az1010301w-lb**
    - Type: **Public**
    - SKU: **Basic**
    - Public IP address: a new public IP address named **az1010301w-lb-pip**
    - Assignment: **Dynamic**
    - Subscription: the name of the subscription you are using in this lab
    - Resource group: **az1010301-RG**
    - Location: the name of the Azure region in which you deployed Azure VMs in the first task of the previous exercise
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:

- IP version: **IPv4**
- Associated to: **Availability set**
- Availability set: **az1010301w-avset**
- 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
>
> **Note**: It is possible that the IP addresses of the Azure VMs are assigned in the reverse order.

> ⓘNote
>
> **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
>
> **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**

- Floating IP (direct server return): **Disabled**

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

> ⓘNote
>
> **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:
    - Name: **az1010302w-lb**
    - Type: **Public**
    - SKU: **Basic**
    - Public IP address: a new public IP address named **az1010302w-lb-pip**
    - Assignment: **Dynamic**
    - Subscription: the name of the subscription you are using in this lab
    - Resource group: **az1010302-RG**
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**
    - IP version: **IPv4**
    - Associated to: **Availability set**
    - Availability set: **az1010302w-avset**
    - 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
    >
    > **Note**: It is possible that the IP addresses of the Azure VMs are assigned in the reverse order.

    > ⓘNote
    >
    > **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:

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

> ⓘNote
>
> **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
>
> **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**

- Port mapping: **Custom**
- Floating IP (direct server return): **Disabled**
- Target port: **3389**

> ⓘNote
>
> **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
>
> **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**

- Target port: **3389**

> ⓘNote
>
> **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
>
> **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:

Code Copy

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

> ⓘNote
>
> **Note**: This command initiates a Remote Desktop session to the **az1010301w-vm0** Azure VM by using the **az1010301w-**

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:

    Code Copy

    ```
    hostname
    ```

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

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

> ⓘNote
>
> **Result**: After you completed this exercise, you have implemented and verified load balancing rules and NAT rules of Azure load balancers in both regions.

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
>
> **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
   >
   > **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

> ⓘNote
>
> **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
   >
   > **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**

   - 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
   >
   > **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:

   Code Copy

   ```
   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:

   Code Copy

   ```
   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.

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

MicrosoftLearning/AZ-103-MicrosoftAzureAdministrator

Lab: Implement Directory Synchronization

All tasks in this lab are performed from the Azure portal (including a PowerShell Cloud Shell session) except for Exercise 3 Task 1, Exercise 3 Task 2, and Exercise 3 Task 3, which include steps performed from a Remote Desktop session to an Azure VM

ⓘNote

Note: When not using Cloud Shell, the lab virtual machine must have the Azure PowerShell 1.2.0 module (or newer) installed https://docs.microsoft.com/en-us/powershell/azure/install-az-ps

Lab files: none

Scenario

Adatum Corporation wants to integrate its Active Directory with Azure Active Directory

Objectives

After completing this lab, you will be able to:

- Deploy an Azure VM hosting an Active Directory domain controller
- Create and configure an Azure Active Directory tenant
- Synchronize Active Directory forest with an Azure Active Directory tenant

Exercise 1: Deploy an Azure VM hosting an Active Directory domain controller

The main tasks for this exercise are as follows:

1. Identify an available DNS name for an Azure VM deployment
2. Deploy an Azure VM hosting an Active Directory domain controller by using an Azure Resource Manager template

Task 1: Identify an available DNS name for an Azure VM deployment

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 Azure subscription you intend to use in this lab and is a Global Administrator of the Azure AD tenant associated with that subscription.
2. From the Azure Portal, start a PowerShell session in the Cloud Shell pane.

   ⓘNote

   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.

3. In the Cloud Shell pane, run the following command, substituting the placeholder <custom-label> with any string which is likely to be unique and the placeholder <location> with the name of the Azure region into which you want to deploy the Azure VM that will host an Active Directory domain controller.

   ⓘNote

   Note: To identify Azure regions where you can provision Azure VMs, refer to **https://azure.microsoft.com/en-us/regions/offers/**

```
Test-AzDnsAvailability -DomainNameLabel <custom-label> -Location '<location>'
```

4. Verify that the command returned **True**. If not, rerun the same command with a different value of the <custom-label> until the command returns **True**.

5. Note the value of the <custom-label> that resulted in the successful outcome. You will need it in the next task

Task 2: Deploy an Azure VM hosting an Active Directory domain controller by using an Azure Resource Manager template

1. From the lab virtual machine, start another instance of Microsoft Edge, browse to the GitHub Azure QuickStart Templates page at **https://github.com/Azure/azure-quickstart-templates**.

2. On the Azure Quickstart Templates page, click **active-directory-new-domain**.

3. On the **Create a new Windows VM and create a new AD Forest, Domain and DC** page, right-click **Deploy to Azure**, and click **Open in new tab**.

4. On the **Create an Azure VM with a new AD Forest** 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 **az1000501-RG**
   - Location: the name of the Azure region which you used in the previous task
   - Admin Username: **Student**
   - Admin Password: **Pa55w.rd1234**
   - Domain Name: **adatum.com**
   - Dns Prefix: the <custom-label> you identified in the previous task
   - VM Size: **Standard_D2s_v3**
   - _artifacts Location: accept the default value
   - _artifacts Location Sas Token: leave blank
   - Location: accept the default value

   > ⓘNote
   >
   > **Note**: Do not wait for the deployment to complete but proceed to the next exercise. You will use the virtual machine deployed in this task in the third exercise of this lab.

> ⓘNote
>
> **Result**: After you completed this exercise, you have initiated deployment of an Azure VM that will host an Active Directory domain controller by using an Azure Resource Manager template

Exercise 2: Create and configure an Azure Active Directory tenant

The main tasks for this exercise are as follows:

1. Create an Azure Active Directory (AD) tenant

2. Add a custom DNS name to the new Azure AD tenant

Task 1: Create an Azure Active Directory (AD) tenant

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Azure Active Directory**.
3. Use the list of search results to navigate to the **Create directory** blade.
4. From the **Create directory** blade, create a new Azure AD tenant with the following settings:

- Organization name: **AdatumSync**
- Initial domain name: a unique name consisting of a combination of letters and digits.
- Country or region: **United States**

> ⓘNote
>
> **Note**: The green check mark in the **Initial domain name** text box will indicate whether the domain name you typed in is valid and unique.

Task 2: Add a custom DNS name to the new Azure AD tenant

1. In the Azure portal, set the **Directory + subscription** filter to the newly created Azure AD tenant.

   > ⓘNote
   >
   > **Note**: The **Directory + subscription** filter appears to the left of the notification icon in the toolbar of the Azure portal

   > ⓘNote
   >
   > **Note**: You might need to refresh the browser window if the **AdatumSync** entry does not appear in the **Directory + subscription** filter list.

2. In the Azure portal, navigate to the **AdatumSync - Overview** blade.
3. From the **AdatumSync - Overview** blade, display the **AdatumSync - Custom domain names** blade.
4. On the **AdatumSync - Custom domain names** blade, identify the primary, default DNS domain name associated with the Azure AD tenant. Note its value - you will need it in the next task.
5. From the **AdatumSync - Custom domain names** blade, add the **adatum.com** custom domain.
6. On the **adatum.com** blade, review the information necessary to perform verification of the Azure AD domain name.

   > ⓘNote
   >
   > **Note**: You will not be able to complete the validation process because you do not own the **adatum.com** DNS domain name. This will not prevent you from synchronizing the **adatum.com** Active Directory domain with the Azure AD tenant. You will use for this purpose the default primary DNS name of the Azure AD tenant (the name ending with the **onmicrosoft.com** suffix), which you identified earlier in this task. However, keep in mind that, as a result, the DNS domain name of the Active Directory domain and the DNS name of the Azure AD tenant will differ. This means that Adatum users will need to use different names when signing in to the Active Directory domain and when signing in to Azure AD tenant.

Task 3: Create an Azure AD user with the Global Administrator role

1. In the Azure portal, navigate to the **Users - All users** blade of the **AdatumSync** Azure AD tenant.

   - User name: **syncadmin@**** where ** represents the default primary DNS domain name you identified in the previous task. Take a note of this user name. You will need it later in this lab.

   - Name: **syncadmin**

   - Password: select the checkbox **Show Password** and note the string appearing in the **Password** text box. You will need it later in this task.

   - Groups: **0 groups selected**

   - Directory role: click “User” and select **Global administrator**

   > ⓘNote
   >
   > **Note**: An Azure AD user with the Global Administrator role is required in order to implement Azure AD Connect.

3. Open an InPrivate Microsoft Edge window.

4. In the new browser window, navigate to the Azure portal and sign in using the **syncadmin** user account. When prompted, change the password to a new value.

   > ⓘNote
   >
   > **Note**: You will need to provide the fully qualified name of the **syncadmin** user account, including the Azure AD tenant DNS domain name.

5. Sign out as **syncadmin** and close the InPrivate browser window.

> ⓘNote
>
> **Result**: After you completed this exercise, you have created an Azure AD tenant, added a custom DNS name to the new Azure AD tenant, and created an Azure AD user with the Global Administrator role.

Exercise 3: Synchronize Active Directory forest with an Azure Active Directory tenant

The main tasks for this exercise are as follows:

1. Configure Active Directory in preparation for directory synchronization

2. Install Azure AD Connect

3. Verify directory synchronization

Task 1: Configure Active Directory in preparation for directory synchronization

> ⓘNote
>
> **Note**: Before you start this task, ensure that the template deployment you started in Exercise 1 has completed.

1. In the Azure portal, set the **Directory + subscription** filter back to the Azure AD tenant associated with the Azure subscription you used in the first exercise of this lab.

   > ⓘNote
   >
   > **Note**: The **Directory + subscription** filter appears to the left of the notification icon in the toolbar of the Azure portal

2. In the Azure portal, navigate to the **adVM** blade, displaying the properties of the Azure VM hosting an Active Directory domain controller that you deployed in the first exercise of this lab.

4. When prompted, authenticate by specifying the following credentials:
    - User name: **Student**
    - Password: **Pa55w.rd1234**
5. Within the Remote Desktop session to **adVM**, open the **Active Directory Administrative Center**.
6. From **Active Directory Administrative Center**, create a root level organizational unit named **ToSync**.
7. From **Active Directory Administrative Center**, in the organizational unit **ToSync**, create a new user account with the following settings:
    - Full name: **aduser1**
    - User UPN logon: **aduser1@adatum.com**
    - User SamAccountName logon: **adatum\aduser1**
    - Password: **Pa55w.rd1234**
    - Other password options: **Password never expires**

Task 2: Install Azure AD Connect

1. Within the RDP session to **adVM**, from Server Manager, disable temporarily **IE Enhanced Security Configuration**.
2. Within the RDP session to **adVM**, start Internet Explorer and download **Azure AD Connect** from [**https://www.microsoft.com/en-us/download/details.aspx?id=47594**](https://www.microsoft.com/en-us/download/details.aspx?id=47594)
3. Start **Microsoft Azure Active Directory Connect** wizard, accept the licensing terms, and, on the **Express Settings** page, select the **Customize** option.
4. On the **Install required components** page, leave all optional configuration options deselected and start the installation.
5. On the **User sign-in** page, ensure that only the **Password Hash Synchronization** is enabled.
6. When prompted to connect to Azure AD, authenticate by using the credentials of the **syncadmin** account you created in the previous exercise.
7. When prompted to connect your directories, add the **adatum.com** forest, choose the option to **Create new AD account**, and authenticate by using the following credentials:
    - User name: **ADATUM\Student**
    - Password: **Pa55w.rd1234**
8. On the **Azure AD sign-in configuration** page, note the warning stating **Users will not be able to sign-in to Azure AD with on-premises credentials if the UPN suffix does not match a verified domain name** and enable the checkbox **Continue without matching all UPN suffixes to verified domain**.

    > ⓘNote
    >
    > **Note**: As explained earlier, this is expected, since you could not verify the custom Azure AD DNS domain **adatum.com**.

9. On the **Domain and OU filtering** page, ensure that only the **ToSync** OU is selected.
10. On the **Uniquely identifying your users** page, accept the default settings.
11. On the **Filter users and devices** page, accept the default settings.
12. On the **Optional features** page, accept the default settings.
13. On the **Ready to configure** page, ensure that the **Start the synchronization process when**

> ⓘNote
>
> **Note**: Installation should take about 2 minutes.

14. Close the Microsoft Azure Active Directory Connect window once the configuration is completed.

Task 3: Verify directory synchronization

1. Within the RDP session to **adVM**, start Internet Explorer, browse to the Azure portal at http://portal.azure.com and sign in by using the **syncadmin** account that you created in the previous exercise.

2. In the Azure portal, navigate to the **AdatumSync - Overview** blade.

3. From the **AdatumSync - Overview** blade, display the **Users - All users** blade of the AdatumSync Azure AD tenant.

4. On the **Users - All users** blade, note that the list of user objects includes the **aduser1** account, with the **Windows Server AD** appearing in the **SOURCE** column.

5. From the **Users - All users** blade, display the **aduser1 - Profile** blade. Note that the **Department** attribute is not set.

6. Within the RDP session to **adVM**, switch to the **Active Directory Administrative Center**, open the window displaying properties of the **aduser1** user account, and set the value of its **Department** attribute to **Sales**.

7. Within the RDP session to **adVM**, start **Windows PowerShell** as Administrator.

8. From the Windows PowerShell prompt, start Azure AD Connect delta synchronization by running the following:

   Code

   ```
   Import-Module -Name 'C:\Program Files\Microsoft Azure AD Sync\Bin\ADSync\ADSync.psd1'

   Start-ADSyncSyncCycle -PolicyType Delta
   ```

9. Within the RDP session to **adVM**, switch to the Internet Explorer window displaying the Azure portal.

10. In the Azure portal, navigate back to the **Users - All users** blade and refresh the page.

11. From the **Users - All users** blade, display the **aduser1 - Profile** blade. Note that the **Department** attribute is now set to **Sales**.

    > ⓘNote
    >
    > **Note**: You might need to wait for another minute and refresh the page again if the **Department** attribute remains not set.

> ⓘNote
>
> **Result**: After you completed this exercise, you have configured Active Directory in preparation for directory synchronization, installed Azure AD Connect, and verified directory synchronization.

Exercise 4: Remove lab resources

Task 1: Open Cloud Shell

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:

   Shell Copy

   ```
   az group list --query "[?starts_with(name,'az1000')].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

   Shell Copy

   ```
   az group list --query "[?starts_with(name,'az1000')].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.

Task 3: Delete the Azure AD tenant.

1. Start Windows PowerShell as Administrator on the lab VM.

2. From the Windows PowerShell console on the lab VM, install the MsOnline PowerShell module by running the following (when prompted, in the NuGet provider is required to continue dialog box, click **Yes**):

   Code Copy

   ```
   Install-Module MsOnline -Force
   ```

3. From the Windows PowerShell console on the lab VM, connect to the AdatumSync Azure AD tenant by running the following (when prompted, sign in with the SyncAdmin credentials):

   Code Copy

   ```
   Connect-MsolService
   ```

4. From the Windows PowerShell console on the lab VM, disable the Azure AD Connect synchronization by running the following:

   Code Copy

   ```
   Set-MsolDirSyncEnabled -EnableDirSync $false -Force
   ```

5. From the Windows PowerShell console on the lab VM, verify that the operation was successful by running the following:

   Code Copy

   ```
   (Get-MSOLCompanyInformation).DirectorySynchronizationEnabled
   ```

6. On the lab VM, sign out from the Azure portal and close the Microsoft Edge window.

7. From the lab VM, start Microsoft Edge, navigate to the Azure portal, and sign in by using the SyncAdmin

8. In the Azure portal, navigate to the **Users - All users** blade of the AdatumSync Azure AD tenant and delete all users with the exception of the AdatumSync account.

> ⓘNote
>
> **Note**: You might need to wait a few hours before you can complete this step.

1. Navigate to the AdatumSync - Overview blade and click **Properties**.
2. On the **Properties** blade of Azure Active Directory click **Yes** in the **Access management for Azure resource** section and then click **Save**.
3. Sign out from the Azure portal and sign back in by using the SyncAdmin credentials.
4. Navigate to the **AdatumSync - Overview** blade and delete the Azure AD tenant by clicking **Delete directory**.
5. On the **Delete directory 'AdatumSync'?** blade, click **Delete**.

> ⓘNote
>
> **Note**: For any additional information regarding this task, refer to https://docs.microsoft.com/en-us/azure/active-directory/users-groups-roles/directory-delete-howto

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

Lab: Azure AD Identity Protection

All tasks in this lab are performed from the Azure portal, except for steps in Exercise 2 performed within a Remote Desktop session to an Azure VM.

Lab files:

- **Labfiles\Module_10\Azure_AD_Identity_Protection\az-101-04b_azuredeploy.json**
- **Labfiles\Module_10\Azure_AD_Identity_Protection\az-101-04b_azuredeploy.parameters.json**

Scenario

Adatum Corporation wants to take advantage of Azure AD Premium features for Identity Protection.

Objectives

After completing this lab, you will be able to:

- Deploy an Azure VM by using an Azure Resource Manager template
- Implement Azure MFA
- Implement Azure AD Identity Protection

Exercise 0: Prepare the lab environment

The main tasks for this exercise are as follows:

1. Deploy an Azure VM by using an Azure Resource Manager template

Task 1: Deploy an Azure VM 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 Azure subscription you intend to use in this lab.
2. In the Azure portal, navigate to the **New** blade.
3. From the **New** blade, search Azure Marketplace for **Template deployment**.
4. Use the list of search results to navigate to the **Custom deployment** blade.
5. On the **Custom deployment** blade, select the **Build your own template in the editor**.
6. From the **Edit template** blade, load the template file **az-101-04b_azuredeploy.json**.

   ⓘNote

   Note: Review the content of the template and note that it defines deployment of an Azure VM hosting Windows Server 2016 Datacenter.

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 **az-101-04b_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 are using in this lab

- Location: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs
- Vm Size: **Standard_DS1_v2**
- Vm Name: **az1010401b-vm1**
- Admin Username: **Student**
- Admin Password: **Pa55w.rd1234**
- Virtual Network Name: **az1010401b-vnet1**

> ⓘNote
>
> **Note**: To identify Azure regions where you can provision Azure VMs, refer to **https://azure.microsoft.com/en-us/regions/offers/**

> ⓘNote
>
> **Note**: Do not wait for the deployment to complete but proceed to the next exercise. You will use the virtual machine included in this deployment in the last exercise of this lab.

> ⓘNote
>
> **Result**: After you completed this exercise, you have initiated a template deployment of an Azure VM **az1010401b-vm1** that you will use in the next exercise of this lab.

Exercise 1: Implement Azure MFA

The main tasks for this exercise are as follows:

1. Create a new Azure AD tenant
2. Activate Azure AD Premium v2 trial
3. Create Azure AD users and groups
4. Assign Azure AD Premium v2 licenses to Azure AD users
5. Configure Azure MFA settings, including fraud alert, trusted IPs, and app passwords
6. Validate MFA configuration

Task 1: Create a new Azure AD tenant

1. In the Azure portal, navigate to the **New** blade.
2. From the **New** blade, search Azure Marketplace for **Azure Active Directory**.
3. Use the list of search results to navigate to the **Create directory** blade.
4. From the **Create directory** blade, create a new Azure AD tenant with the following settings:

- Organization name: **AdatumLab101-4b**
- Initial domain name: a unique name consisting of a combination of letters and digits.
- Country or region: **United States**

> ⓘNote
>
> **Note**: Take a note of the initial domain name. You will need it later in this lab.

Task 2: Activate Azure AD Premium v2 trial

1. In the Azure portal, set the **Directory + subscription** filter to the newly created Azure AD tenant.

   > ⓘNote
   >
   > **Note**: The **Directory + subscription** filter appears to the right of the Cloud Shell icon in the toolbar of the Azure portal

   > ⓘNote
   >
   > **Note**: You might need to refresh the browser window if the **AdatumLab101-4b** entry does not appear in the **Directory + subscription** filter list.

2. In the Azure portal, navigate to the **AdatumLab101-4b - Overview** blade.

3. From the **AdatumLab101-4b - Overview** blade, navigate to the **Licenses - Overview** blade.

4. From the **Licenses - Overview** blade, navigate to the **Licenses - All products** blade.

5. From the **Licenses - All products** blade, navigate to the **Activate** blade and activate **Azure AD Premium P2** free trial.

Task 3: Create Azure AD users and groups.

1. In the Azure portal, navigate to the **Users - All users** blade of the AdatumLab101-4b Azure AD tenant.

2. From the **Users - All users** blade, create a new user with the following settings:

   - Name: **aaduser1**

   - User name: **aaduser1@<DNS-domain-name>.onmicrosoft.com** where <DNS-domain-name> represents the initial domain name you specified in the first task of this exercise.

   > ⓘNote
   >
   > **Note**: Take a note of this user name. You will need it later in this lab.

   - Profile: **Default**

   - Properties: **Default**

   - Groups: **0 groups selected**

   - Directory role: **Global administrator**

   - Password: select the checkbox **Show Password** and note the string appearing in the **Password** text box. You will need it later in this lab.

3. From the **Users - All users** blade, create a new user with the following settings:

   - Name: **aaduser2**

   - User name: **aaduser2@<DNS-domain-name>.onmicrosoft.com** where <DNS-domain-name> represents the initial domain name you specified in the first task of this exercise.

> ⓘNote
>
> **Note**: Take a note of this user name. You will need it later in this lab.

- Profile: **Default**
- Properties: **Default**
- Groups: **0 groups selected**
- Directory role: **User**
- Password: select the checkbox **Show Password** and note the string appearing in the **Password** text box. You will need it later in this lab.

Task 4: Assign Azure AD Premium v2 licenses to Azure AD users

> ⓘNote
>
> **Note**: In order to assign Azure AD Premium v2 licenses to Azure AD users, you first have to set their location attribute.

1. From the **Users - All users** blade, navigate to the **aaduser1 - Profile** blade and set the **Usage location** to **United States**.
2. From the **aaduser1 - Profile** blade, navigate to the **aaduser1 - Licenses** blade and assign to the user an Azure Active Directory Premium P2 license with all licensing options enabled.
3. Return to the **Users - All users** blade, navigate to the **aaduser2 - Profile** blade, and set the **Usage location** to **United States**.
4. From the **aaduser2 - Profile** blade, navigate to the **aaduser2 - Licenses** blade and assign to the user an Azure Active Directory Premium P2 license with all licensing options enabled.
5. Return to the **Users - All users** blade, navigate to the Profile entry of your user account and set the **Usage location** to **United States**.
6. Navigate to **Licenses** blade of your user account and assign to it an Azure Active Directory Premium P2 license with all licensing options enabled.
7. Sign out from the portal and sign back in using the same account you are using for this lab.

   > ⓘNote
   >
   > **Note**: This step is necessary in order for the license assignment to take effect.

Task 5: Configure Azure MFA settings.

1. In the Azure portal, navigate to the **Users - All users** blade of the AdatumLab101-4b Azure AD tenant.
2. From the **Users - All users** blade of the AdatumLab101-4b Azure AD tenant, use the **Multi-Factor Authentication** link to open the **multi-factor authentication** portal.
3. On the **multi-factor authentication** portal, display to the **service settings** tab, review its settings, and the **verification options**, including **Text message to phone**, **Notification through mobile app**, and **Verification code from mobile app or hardware token** are enabled.
4. On the **multi-factor authentication** portal, switch to the **users** tab, select **aaduser1** entry, and enable its multi-factor authentication status.
5. On the **multi-factor authentication** portal, note that the multi-factor authentication status of **aaduser1** changed to **Enabled** and that, once you select the user entry again, you have the option of changing it to

> ⓘNote
>
> **Note**: Changing the user status from enabled to enforced impacts only legacy, Azure AD integrated apps which do not support Azure MFA and, once the status changes to enforced, require the use of app passwords.

6. On the **multi-factor authentication** portal, with the **aaduser1** entry selected, display the **Manage user settings** window and review its options, including:
   - Require selected users to provide contact methods again
   - Delete all existing app passwords generated by the selected users
   - Restore multi-factor authentication on all remembered devices
7. Do not make any changes to user settings and switch back to the Azure portal.
8. From the **Users - All users** blade of the AdatumLab101-4b Azure AD tenant, navigate to the **AdatumLab101-4b - Overview** blade.
9. From the **AdatumLab101-4b - Overview** blade, navigate to the **AdatumLab101-4b - MFA** blade.
10. From the **AdatumLab101-4b - MFA** blade, navigate to the **Multi-Factor Authentication - Fraud alert** blade and configure the following settings:
    - Allow users to submit fraud alerts: **On**
    - Automatically block users who report fraud: **On**
    - Code to report fraud during initial greeting: **0**

Task 6: Validate MFA configuration

1. Open an InPrivate Microsoft Edge window.
2. In the new browser window, navigate to the Azure portal and sign in using the **aaduser1** user account. When prompted, change the password to a new value.

   > ⓘNote
   >
   > **Note**: You will need to provide a fully qualified name of the **aaduser1** user account, including the Azure AD tenant DNS domain name, as noted earlier in this lab.

3. When prompted with the **More information required** message, continue to the **Additional security verification** page.
4. On the **How should we contact you?** page, note that you need to set up one of the following options:
   - **Authentication phone**
   - **Mobile app**
5. Select the **Authentication phone** option with the **Send me a code by text message** method.
6. Complete the verification and note the automatically generated app password.
7. When prompted, change the password from the one generated when you created the **aaduser1** account.
8. Verify that you successfully signed in to the Azure portal.
9. Sign out as **aaduser1** and close the InPrivate browser window.

Exercise 2: Implement Azure AD Identity Protection:

The main tasks for this exercise are as follows:

2. Configure user risk policy
3. Configure sign-in risk policy
4. Validate Azure AD Identity Protection configuration by simulating risk events

Task 1: Enable Azure AD Identity Protection

1. From the lab virtual machine, start Microsoft Edge, browse to the Azure portal at http://portal.azure.com and sign in by using the Microsoft account you used to create the **AdatumLab101-4b** Azure AD tenant.

   > ⓘNote
   >
   > **Note**: Ensure that you are signed-in to the **AdatumLab101-4b** Azure AD tenant. You can use the **Directory + subscription** filter to switch between Azure AD tenants.

2. In the Azure portal, navigate to the **New** blade.
3. From the **New** blade, search Azure Marketplace for **Azure AD Identity Protection**.
4. Select the **Azure AD Identity Protection** in the list of search results and proceed to create an instance of **Azure AD Identity Protection** associated with the **AdatumLab101-4b** Azure AD tenant.
5. In the Azure portal, navigate to the **All services** blade and use the search filter to display the **Azure AD Identity Protection** blade.

Task 2: Configure user risk policy

1. From the **Azure AD Identity Protection** blade, navigate to the **Azure AD Identity Protection - User risk policy** blade
2. On the **Azure AD Identity Protection - User risk policy** blade, configure the **User risk remediation policy** with the following settings:
   - Assignments:
     - Users: **All users** (be sure to exclude the current admin account to avoid getting locked out of the tenant)
     - Conditions:
       - User risk: **Medium and above**
   - Controls:
     - Access: **Allow access**
     - **Require password change**
   - Enforce Policy: **On**

Task 3: Configure sign-in risk policy

1. From the **Azure AD Identity Protection - User risk policy** blade, navigate to the **Azure AD Identity Protection - Sign-in risk policy** blade
2. On the **Azure AD Identity Protection - Sign-in risk policy** blade, configure the **Sign-in risk remediation policy** with the following settings:
   - Assignments:
     - Users: **All users**
     - Conditions:

- Controls:
  - Access: **Allow access**
  - **Require multi-factor authentication**
- Enforce Policy: **On**

Task 4: Validate Azure AD Identity Protection configuration by simulating risk events

> ⓘNote
>
> **Note**: Before you start this task, ensure that the template deployment you started in Exercise 0 has completed.

1. In the Azure portal, set the **Directory + subscription** filter to the default Azure AD tenant.
2. In the Azure portal, navigate to the **az1010401b-vm1** blade.
3. From the **az1010401b-vm1** blade, connect to the Azure VM via Remote Desktop session and, when prompted to sign in, provide the following credentials:
   - Admin Username: **Student**
   - Admin Password: **Pa55w.rd1234**
4. Within the Remote Desktop session, in Server Manager, click **Local Server** and then click **IE Enhanced Security Configuration**.
5. In the **Internet Explorer Enhanced Security Configuration** dialog box, set both options to **Off** and click **OK**.
6. Within the Remote Desktop session, open an InPrivate Internet Explorer window.
7. In the new browser window, navigate to the ToR Browser Project at **https://www.torproject.org/projects/torbrowser.html.en**, download the ToR Browser, and install it with the default options.
8. Once the installation completes, start the ToR Browser, use the **Connect** option on the initial page, and navigate to the Application Access Panel at **https://myapps.microsoft.com**
9. When prompted, sign in with the **aaduser2** account you created in the previous exercise.
10. You will be presented with the message **Your sign-in was blocked**. This is expected, since this account is not configured with multi-factor authentiation, which is required due to increased sign-in risk associated with the use of ToR Browser.
11. Use the **Sign out and sign in with a different account option** to sign in as **aaduser1** account you created and configured for multi-factor authentication in the previous exercise.
12. This time, you will be presented with the **Suspicious activity detected** message. Again, this is expected, since this account is configured with multi-factor authentiation. Considering the increased sign-in risk associated with the use of ToR Browser, you will have to use multi-factor authentication, according to the sign-in risk policy you configured in the previous task.
13. Use the **Verify** option and specify whether you want to verify your identity via text or a call.
14. Complete the verification and ensure that you successfully signed in to the Application Access Panel.
15. Sign out as **aaduser1** and close the ToR Browser window.
16. Start Internet Explorer, browse to the Azure portal at **http://portal.azure.com** and sign in by using the Microsoft account you used to create the **AdatumLab101-4b** Azure AD tenant.
17. In the Azure portal, navigate to the **Azure AD Identity Protection - Risk events** blade and note that the entry representing **Sign-in from anonymous IP address**.

18. From the **Azure AD Identity Protection - Risk events** blade, navigate to the **Azure AD Identity Protection - Users flagged for risk** blade and note the entry representing **aaduser2**.

> ⓘNote
>
> **Result**: After you completed this exercise, you have enabled Azure AD Identity Protection, configured user risk policy and sign-in risk policy, as well as validated Azure AD Identity Protection configuration by simulating risk events

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:

   Shell Copy

   ```
   az group list --query "[?starts_with(name,'az1010')].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

   Shell Copy

   ```
   az group list --query "[?starts_with(name,'az1010')].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.

> ⓘNote
>
> **Note**: To remove the Azure AD tenant you created in this lab, follow https://docs.microsoft.com/en-us/azure/active-directory/users-groups-roles/directory-delete-howto

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

MicrosoftLearning/AZ-103-MicrosoftAzureAdministrator

Lab: Self-Service Password Reset

All tasks in this lab are performed from the Azure portal

Lab files: none

Scenario

Adatum Corporation wants to take advantage of Azure AD Premium features

Objectives

After completing this lab, you will be able to:

- Manage Azure AD users and groups
- Manage Azure AD-integrated SaaS applications

Exercise 1: Manage Azure AD users and groups

The main tasks for this exercise are as follows:

1. Create a new Azure AD tenant
2. Activate Azure AD Premium v2 trial
3. Create and configure Azure AD users
4. Assign Azure AD Premium v2 licenses to Azure AD users
5. Manage Azure AD group membership
6. Configure self-service password reset functionality
7. Validate self-service password reset functionality

Task 1: Create a new Azure AD tenant

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 Azure subscription you intend to use in this lab.
2. In the Azure portal, navigate to the **New** blade.
3. From the **New** blade, search Azure Marketplace for **Azure Active Directory**.
4. Use the list of search results to navigate to the **Create directory** blade.
5. From the **Create directory** blade, create a new Azure AD tenant with the following settings:

- Organization name: **AdatumLab100-5b**
- Initial domain name: a unique name consisting of a combination of letters and digits.
- Country or region: **United States**

> ⓘNote
>
> **Note**: Take a note of the initial domain name. You will need it later in this lab.

Task 2: Activate Azure AD Premium v2 trial

1. In the Azure portal, set the **Directory + subscription** filter to the newly created Azure AD tenant.

> ⓘNote
>
> **Note**: The **Directory + subscription** filter appears to the right of the Cloud Shell icon in the toolbar of the Azure portal

> ⓘNote
>
> **Note**: You might need to refresh the browser window if the **AdatumLab100-5b** entry does not appear in the **Directory + subscription** filter list.

2. In the Azure portal, navigate to the **AdatumLab100-5b - Overview** blade.
3. From the **AdatumLab100-5b - Overview** blade, navigate to the **Licenses - Overview** blade.
4. From the **Licenses - Overview** blade, navigate to the **Products** blade.
5. From the **Products** blade, navigate to the **Activate** blade and activate **Azure AD Premium P2** free trial.

Task 3: Create and configure Azure AD users

1. In the Azure portal, navigate to the **Users - All users** blade of the AdatumLab100-5b Azure AD tenant.
2. From the **Users - All users** blade, create a new user with the following settings:
    - Name: **aaduser1**
    - User name: **aaduser1@<DNS-domain-name>.onmicrosoft.com** where <DNS-domain-name> represents the initial domain name you specified in the first task of this exercise.

    > ⓘNote
    >
    > **Note**: Take a note of this user name. You will need it later in this lab.

    - Profile:
        - Department: **Sales**
    - Properties: **Default**
    - Groups: **0 groups selected**
    - Directory role: **User**
    - Password: select the checkbox **Show Password** and note the string appearing in the **Password** text box. You will need it later in this lab.
3. From the **Users - All users** blade, create a new user with the following settings:
    - Name: **aaduser2**
    - User name: **aaduser2@<DNS-domain-name>.onmicrosoft.com** where <DNS-domain-name> represents the initial domain name you specified in the first task of this exercise.

    > ⓘNote
    >
    > **Note**: Take a note of this user name. You will need it later in this lab.

    - Profile:
        - Department: **Finance**
    - Properties: **Default**

- Directory role: **User**
- Password: select the checkbox **Show Password** and note the string appearing in the **Password** text box. You will need it later in this lab.

Task 4: Assign Azure AD Premium v2 licenses to Azure AD users

> ⓘNote
>
> **Note**: In order to assign Azure AD Premium v2 licenses to Azure AD users, you first have to set their location attribute.

1. From the **Users - All users** blade, navigate to the **aaduser1 - Profile** blade and set the **Usage location** to **United States**.
2. From the **aaduser1 - Profile** blade, navigate to the **aaduser1 - Licenses** blade and assign to the user an Azure Active Directory Premium P2 license with all licensing options enabled.
3. Return to the **Users - All users** blade, navigate to the **aaduser2 - Profile** blade, and set the **Usage location** to **United States**.
4. From the **aaduser2 - Profile** blade, navigate to the **aaduser2 - Licenses** blade and assign to the user an Azure Active Directory Premium P2 license with all licensing options enabled.
5. Return to the **Users - All users** blade, navigate to the Profile entry of your user account and set the **Usage location** to **United States**.
6. Navigate to **Licenses** blade of your user account and assign to it an Azure Active Directory Premium P2 license with all licensing options enabled.
7. Sign out from the portal and sign back in using the same account you are using for this lab.

   > ⓘNote
   >
   > **Note**: This step is necessary in order for the license assignment to take effect.

Task 5: Manage Azure AD group membership

1. In the Azure portal, navigate to the **Groups - All groups** blade.
2. From the **Groups - All groups** blade, navigate to the **Group** blade and create a new group with the following settings:
   - Group type: **Security**
   - Group name: **Sales**
   - Group description: **All users in the Sales department**
   - Membership type: **Dynamic User**
   - Dynamic user members:
     - Simple rule
     - Add users where: **department Equals Sales**
3. From the **Groups - All groups** blade, navigate to the **Group** blade and create a new group with the following settings:
   - Group type: **Security**
   - Group name: **Sales and Finance**

   - Membership type: **Dynamic User**
   - Dynamic user members:
      - Advanced rule: **(user.department -eq “Sales”) -or (user.department -eq “Finance”)**

4. From the **Groups - All groups** blade, navigate to the blades of **Sales** and **Sales and Finance** groups, and note that the group membership evaluation is in progress. Wait until the evalution completes, then navigate to the **Members** blade, and verify that the group membership is correct.

Task 6: Configure self-service password reset functionality

1. In the Azure portal, navigate to the **AdatumLab100-5b - Overview** blade.
2. From the **AdatumLab100-5b - Overview** blade, navigate to the **Password reset - Properties** blade.
3. On the **Password reset - Properties** blade, configure the following settings:
   - Self service password reset enabled: **Selected**
   - Selected group: **Sales**
4. From the **Password reset - Properties** blade, navigate to the **Password reset - Auhentication methods** blade and configure the following settings:
   - Number of methods required to reset: **1**
   - Methods available to users:
      - **Email**
      - **Mobile phone**
      - **Office phone**
      - **Security questions**
      - Number of security questions required to register: **3**
      - Number of security questions required to reset: **3**
      - Select security questions: select **Predefined** and add any combination of 5 predefined security questions
5. From the **Password reset - Authentication methods** blade, navigate to the **Password reset - Registration** blade, and ensure that the following settings are configured:
   - Require users to register when signing in?: **Yes**
   - Number of days before users are asked to re-confirm their authentication information: **180**

Task 7: Validate self-service password reset functionality

1. Open an InPrivate Microsoft Edge window.
2. In the new browser window, navigate to the Azure portal and sign in using the **aaduser1** user account. When prompted, change the password to a new value.

   > ⓘNote
   >
   > **Note**: You will need to provide a fully qualified name of the **aaduser1** user account, including the Azure AD tenant DNS domain name, as noted earlier in this lab.

3. When prompted with the **More information required** message, continue to the **don’t lose access to your account** page.

   - **Office phone**
   - **Authentication Phone**
   - **Authentication Email**
   - **Security Questions**

5. From the **don't lose access to your account** page, configure answers to 5 security questions you selected in the previous task
6. Verify that you successfully signed in to the Azure portal.
7. Sign out as **aaduser1** and close the InPrivate browser window.
8. Open an InPrivate Microsoft Edge window.
9. In the new browser window, navigate to the Azure portal and, on the **Pick an account** page, type in the **aaduser1** user account name.
10. On the **Enter password** page, click the **Forgot my password** link.
11. On the **Get back into your account** page, verify the **User ID**, enter the characters in the picture or the words in the audio, and proceed to the next page.
12. On the next page, provide answers to thre security questions using answers you specified in the previous task.
13. On the next page, enter twice a new password and complete the password reset process.
14. Verify that you can sign in to the Azure portal by using the newly reset password.

> ⓘNote
>
> **Result**: After you completed this exercise, you have created a new Azure AD tenant, activated Azure AD Premium v2 trial, created and configured Azure AD users, assigned Azure AD Premium v2 licenses to Azure AD users, managed Azure AD group membership, as well as configured and validated self-service password reset functionality

Exercise 2: Manage Azure AD-integrated SaaS applications

The main tasks for this exercise are as follows:

1. Add an application from the Azure AD gallery
2. Configure the application for a single sign-on
3. Assign users to the application
4. Validate single sign-on for the application

Task 1: Add an application from the Azure AD gallery

1. In the Azure portal, navigate to the **AdatumLab100-5b - Overview** blade.
2. From the **AdatumLab100-5b - Overview** blade, navigate to the **Enterprise applications - All applications** blade.
3. From the **Enterprise applications - All applications** blade, navigate to the **Add an application** blade.
4. On the **Add an application** blade, search the application gallery for the **Microsoft OneDrive**.
5. Use the list of search results to navigate to the **Microsoft OneDrive** add app blade and add the app.

Task 2: Configure the application for a single sign-on

2. On the **Microsoft OneDrive - Getting started** blade, use the **Configure single sign-on (required)** option to navigate to the **Microsoft OneDrive - Single sign-on** blade.

3. On the **Microsoft OneDrive - Single sign-on** blade, select the **Password-based** option and save the configuration.

Task 3: Assign users to the application

1. Navigate back to the **Microsoft OneDrive - Getting started** blade.

2. On the **Microsoft OneDrive - Getting started** blade, use the **Assign a user for testing (required)** option to navigate to the **Users and groups** blade for **Microsoft OneDrive**.

3. From the **Users and groups** blade for **Microsoft OneDrive**, navigate to the **Add Assignment** blade and add the following assignment:

   - Users and groups: **Sales and Finance**

   - Select role: **Default access**

   - Assign Credentials:

     - Assign credentials to be shared among all group members: **Yes**

     - Email Address: the name of the Microsoft Account you are using for this lab

     - Password: the password of the Microsoft Account you are using for this lab

4. Sign out from the Azure portal and close the Microsoft Edge window.

Task 4: Validate single sign-on for the application

1. Open a Microsoft Edge window.

2. In the Microsoft Edge window, navigate to the Application Access Panel at http://myapps.microsoft.com and sign in by using the **aaduser2** user account. When prompted, change the password to a new value.

   > ⓘNote
   >
   > **Note**: You will need to provide a fully qualified name of the **aaduser2** user account, including the Azure AD tenant DNS domain name, as noted earlier in this lab.

3. On the Access Panel Applications page, click the **Microsoft OneDrive** icon.

4. When prompted, add the My Apps Secure Sign-in Extension and enable it, including the **Allow for InPrivate browsing** option.

5. Navigate again to the Application Access Panel at http://myapps.microsoft.com and sign in by using the **aaduser2** user account.

6. On the Access Panel Applications page, click the **Microsoft OneDrive** icon.

7. Verify that you have successfully accessed the Microsoft OneDrive application without having to re-authenticate.

8. Sign out from the Application Access Panel and close the Microsoft Edge window.

   > ⓘNote
   >
   > **Note**: Make sure to launch Microsoft Edge again, browse to the Azure portal, sign in by using the Microsoft account that has the Owner role in the Azure subscription you were using in this lab, and use the **Directory + subscription** filter to switch to your default Azure AD tenant once you complete this lab.

MicrosoftLearning/AZ-103-MicrosoftAzureAdministrator

ⓘNote

Result: After you completed this exercise, you have added an application from the Azure AD gallery, configured the application for a single sign-on, assigned users to the application, and validated single sign-on for the application.

MicrosoftLearning/AZ-103-MicrosoftAzureAdministrator

Lab: Role-Based Access Control

All tasks in this lab are performed from the Azure portal (including a PowerShell Cloud Shell session)

> ⓘNote
>
> **Note**: When not using Cloud Shell, the lab virtual machine must have the Azure PowerShell 1.2.0 module (or newer) installed https://docs.microsoft.com/en-us/powershell/azure/install-az-ps

Lab files: none

Scenario

Adatum Corporation wants to use Azure Role Based Access Control and Azure Policy to control provisioning and management of their Azure resources. It also wants to be able to automate and track provisioning and management tasks.

Objectives

After completing this lab, you will be able to:

- Configure delegation of provisioning and management of Azure resources by using built-in Role-Based Access Control (RBAC) roles and built-in Azure policies
- Verify delegation by provisioning Azure resources as a delegated admin and auditing provisioning events

Exercise 1: Configure delegation of provisioning and management of Azure resources by using built-in Role-Based Access Control (RBAC) roles and built-in Azure policies

The main tasks for this exercise are as follows:

1. Create Azure Active Directory (AD) users and groups
2. Create Azure resource groups
3. Delegate management of an Azure resource group via a built-in RBAC role
4. Assign a built-in Azure policy to an Azure resource group

Task 1: Create Azure AD users and groups

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 Azure subscription you intend to use in this lab and is a Global Administrator of the Azure AD tenant associated with that subscription.
2. In the Azure portal, navigate to the **Azure Active Directory** blade
3. From the **Azure Active Directory** blade, navigate to the **Custom domain names** blade and identify the primary DNS domain name associated the Azure AD tenant. Note its value - you will need it later in this task.
4. From the Azure AD **Custom domain names** blade, navigate to the **Users - All users** blade.
5. From the **Users - All users** blade, create a new user with the following settings:
   - Name: **aaduser100011**
   - User name: **aaduser100011@<DNS-domain-name>** where <DNS-domain-name> represents the primary DNS domain name you identified earlier in this task.
   - Profile: **Not configured**

   - Groups: **0 groups selected**
   - Directory role: **User**
   - Password: select the checkbox **Show Password** and note the string appearing in the **Password** text box. You will need it later in this lab.

6. From the **Users - All users** blade, navigate to the **Groups - All groups** blade.

7. From the **Groups - All groups** blade, create a new group with the following settings:

   - Group type: **Security**
   - Group name: **az1001 Contributors**
   - Group description: **az1001 Contributors**
   - Membership type: **Assigned**
   - Members: **aaduser100011**

Task 2: Create Azure resource groups

1. In the Azure portal, navigate to the **Resource groups** blade.

2. From the **Resource groups** blade, create the first resource group with the following settings:

   - Resource group name: **az1000101-RG**
   - Subscription: the name of the subscription you are using in this lab
   - Resource group location: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs.

   > ⓘNote
   >
   > **Note**: To identify Azure regions available in your subscription, refer to https://azure.microsoft.com/en-us/regions/offers/

3. From the **Resource groups** blade, create the second resource group with the following settings:

   - Resource group name: **az1000102-RG**
   - Subscription: the name of the subscription you selected in the previous step
   - Resource group location: the name of the Azure region you selected in the previous step

Task 3: Delegate management of an Azure resource group via a built-in RBAC role

1. In the Azure portal, from the **Resource groups** blade, navigate to the **az1000101-RG** blade.

2. From the **az1000101-RG** blade, display its **Access control (IAM)** blade.

3. From the **az1000101-RG - Access control (IAM)** blade, display the **Role assignments** blade.

4. From the **Role assignments** blade, create the following **role assignment**:

   - Role: **Contributor**
   - Assign access to: **Azure AD user, group, or service principal**
   - Select: **az1001 Contributors**

Task 4: Assign a built-in Azure policy to an Azure resource group

1. From the **az1000101-RG** blade, display its **Policies** blade.

3. Assign the policy with the following settings:
    - Scope: **az1000101-RG**
    - Exclusions: leave the entry blank
    - Policy definition: **Allowed virtual machine SKUs**
    - Assignment name: **Allowed virtual machine SKUs**
    - Description: **Allowed selected virtual machine SKUs (Standard_DS1_v2)**
    - Assigned by: leave the entry set to its default value
    - Allowed SKUs: **Standard_DS1_v2**
    - Create a Managed Identity: leave the entry blank

> ⓘNote
>
> **Result**: After you completed this exercise, you have created an Azure AD user and an Azure AD group, created two Azure resource groups, delegated management of the first Azure resource group via the built-in Azure VM Contributor RBAC role, and assigned to the same resource group the built-in Azure policy restricting SKUs that can be used for Azure VMs.

Exercise 2: Verify delegation by provisioning Azure resources as a delegated admin and auditing provisioning events

The main tasks for this exercise are as follows:

1. Identify an available DNS name for an Azure VM deployment
2. Attempt an automated deployment of a policy non-compliant Azure VM as a delegated admin
3. Perform an automated deployment of a policy compliant Azure VM as a delegated admin
4. Review Azure Activity Log events corresponding to Azure VM deployments

Task 1: Identify an available DNS name for an Azure VM deployment

1. From the Azure Portal, start a PowerShell session in the Cloud Shell.

    > ⓘNote
    >
    > **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.

2. In the Cloud Shell pane, run the following command, substituting the placeholder <custom-label> with any string which is likely to be unique and the placeholder <location-of-az1000101-RG> with the name of the Azure region in which you created the **az1000101-RG** resource group.

    Code Copy

    ```
    Test-AzDnsAvailability -DomainNameLabel <custom-label> -Location '<location-of-az1000101-RG>'
    ```

3. Verify that the command returned **True**. If not, rerun the same command with a different value of the <custom-label> until the command returns **True**.

4. Note the value of the <custom-label> that resulted in the successful outcome. You will need it in the next task

5. Run these commands:

```
Register-AzResourceProvider –ProviderNamespace Microsoft.Network
```

Code Copy

```
Register-AzResourceProvider –ProviderNamespace Microsoft.Compute
```

Note: These cmdlets register the Azure Resource Manager Microsoft.Network and Microsoft.Compute resource providers. This is a one-time operation (per subscription) required when using Azure Resource Manager templates to deploy resources managed by these resource providers (if these resource providers have not been yet registered).

Also Note: If you encounter an error after running these commands that mentions a token expiry set to a time that is before the current time, click the power button icon on our Cloud Shell UI and reboot your Cloud Shell instance. Once restarted, retry these commands.

Task 2: Attempt an automated deployment of a policy non-compliant Azure VM as a delegated admin

1. Launch another browser window in the Private mode.
2. In the new browser window, navigate to the Azure portal and sign in using the user account you created in the previous exercise. When prompted, change the password to a new value.
3. In the Azure portal, navigate to the **Resource groups** blade and note that you can view only the resource group **az1000101-RG**.
4. In the Azure portal, navigate to the **Create a resource** blade.
5. From the **Create a resource** blade, search Azure Marketplace for **Template deployment**.
6. Use the list of search results to navigate to the **Deploy a custom template** blade.
7. On the **Custom deployment** blade, in the **Load a GitHub quickstart template** drop-down list, select the **101-vm-simple-linux** entry and navigate to the **Edit template** blade.
8. On the **Edit template** blade, navigate to the **Variables** section and locate the **vmSize** entry.
9. Note that the template is using hard-coded **Standard_A1** VM size.
10. Discard any changes you might have made to the template and navigate to the **Deploy a simple Ubuntu Linux VM** blade.
11. From the **Deploy a simple Ubuntu Linux VM** blade, initiate a template deployment with the following settings:
    - Subscription: the same subscription you selected in the previous exercise
    - Resource group: **az1000101-RG**
    - Location: the name of the Azure region which you selected in the previous exercise
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Dns Label Prefix: the <custom-label> you identified in the previous task
    - Ubuntu OS Version: accept the default value
    - Location: accept the default value
12. Note that the initiation of the deployment fails. Navigate to the **Errors** blade and note that the deployment of the resource is not allowed by the policy **Allowed virtual machine SKUs**.

Task 3: Perform an automated deployment of a policy compliant Azure VM as a delegated admin

2. On the **Edit template** blade, navigate back to the **Variables** section and locate the **vmSize** entry.
3. Replace the value **Standard_A1** with **Standard_DS1_v2** and save the change.
4. Initiate a deployment again. Note that this time validation is successful.
5. Do not wait for the deployment to complete but proceed to the next task.

Task 4: Review Azure Activity Log events corresponding to Azure VM deployments

1. Switch to the browser window that you used in the previous exercise.
2. In the Azure portal, navigate to the **az1000101-RG** resource group blade.
3. From the **az1000101-RG** resource group blade, display its **Activity log** blade.
4. In the list of operations, note the ones corresponding to the failed and successful validation events.
5. Refresh the view of the blade and observe events corresponding to the Azure VM provisioning, including the final one representing the successful deployment.

> ⓘNote
>
> **Result**: After you completed this exercise, you have identified an available DNS name for an Azure VM deployment, attempted an automated deployment of a policy non-compliant Azure VM as a delegated admin, performed an automated deployment of a policy compliant Azure VM as the same delegated admin, and reviewed Azure Activity Log entries corresponding to both Azure VM deployments.

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:

Shell Copy

```
az group list --query "[?starts_with(name,'az1000')].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

Shell Copy

```
az group list --query "[?starts_with(name,'az1000')].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.

ⓘNote

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

MicrosoftLearning/AZ-103-MicrosoftAzureAdministrator

Lab: Implementing governance and compliance with Azure initiatives and resource locks

All tasks in this lab are performed from the Azure portal (including a PowerShell Cloud Shell session)

> ⓘNote
>
> **Note**: When not using Cloud Shell, the lab virtual machine must have the Azure PowerShell 1.2.0 module (or newer) installed https://docs.microsoft.com/en-us/powershell/azure/install-az-ps

Lab files:

- **Labfiles\Module_11\Governance_and_Compliance\AZ-100.1/az-100-01b_azuredeploy.json**
- **Labfiles\Module_11\Governance_and_Compliance\az-100-01b_azuredeploy.parameters.json**

Scenario

Adatum Corporation wants to use Azure policies and initiatives in order to enforce resource tagging in its Azure subscription. Once the environment is compliant, Adatum wants to prevent unintended changes by implementing resource locks.

Objectives

After completing this lab, you will be able to:

- Implement Azure tags by using Azure policies and initiatives
- Implement Azure resource locks

Exercise 1: Implement Azure tags by using Azure policies and initiatives

The main tasks for this exercise are as follows:

1. Provision Azure resources by using an Azure Resource Manager template.
2. Implement an initiative and policy that evaluate resource tagging compliance.
3. Implement a policy that enforces resource tagging compliance.
4. Evaluate tagging enforcement and tagging compliance.
5. Implement remediation of resource tagging non-compliance.
6. Evaluate effects of the remediation task on compliance.

Task 1: Provision Azure resources 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 Azure subscription you intend to use in this lab.
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 **Custom deployment** blade.
5. On the **Custom deployment** blade, select the **Build your own template in the editor**.
6. From the **Edit template** blade, load the template file **az-100-01b_azuredeploy.json**.

> ⓘNote
>
> **Note**: Review the content of the template and note that it defines deployment of an Azure VM hosting Windows Server 2016 Datacenter, including tags on some of its resources.

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 **az-100-01b_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 are using in this lab
    - Resource group: the name of a new resource group **az1000101b-RG**
    - Location: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs
    - Vm Size: **Standard_DS1_v2**
    - Vm Name: **az1000101b-vm1**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1000101b-vnet1**
    - Environment Name: **lab**

> ⓘNote
>
> **Note**: To identify Azure regions where you can provision Azure VMs, refer to **https://azure.microsoft.com/en-us/regions/offers/**

> ⓘNote
>
> **Note**: Do not wait for the deployment to complete before you proceed to the next step.

12. In the Azure portal, navigate to the **Tags** blade.
13. From the **Tags** blade, display all resources with the **environment** tag set to the value **lab**. Note that only some of the resources deployed in the previous task have this tag assigned.

> ⓘNote
>
> **Note**: At this point, only some of the resources have been provisioned, however, you should see at least a few without tags assigned to them.

Task 2: Implement a policy and an initiative that evaluate resource tagging compliance.

1. In the Azure portal, navigate to the **Policy** blade.
2. From the **Policy** blade, navigate to the **Policy - Definitions** blade.
3. From the **Policy Definitions** blade, display the **Require tag and its value** policy definition.

- Definition location: the name of the subscription you are using in this lab
- Name: **az10001b - Audit tag and its value**
- Description: **Audits a required tag and its value. Does not apply to resource groups.**
- Category: the name of a new category **Lab**
- Policy rule: in the existing policy rule, change the **effect** from **deny** to **audit**, such that the policy definition has the following content:

Code Copy

```
{
  "mode": "indexed",
  "policyRule": {
    "if": {
      "not": {
        "field": "[concat('tags[', parameters('tagName'), ']')]",
        "equals": "[parameters('tagValue')]"
      }
    },
    "then": {
      "effect": "audit"
    }
  },
  "parameters": {
    "tagName": {
      "type": "String",
      "metadata": {
        "displayName": "Tag Name",
        "description": "Name of the tag, such as 'environment'"
      }
    },
    "tagValue": {
      "type": "String",
      "metadata": {
        "displayName": "Tag Value",
        "description": "Value of the tag, such as 'production'"
      }
    }
  }
}
```

5. From the **Policy - Definitions** blade, navigate to the **New Initiative definition** blade.

6. From the **New Initiative definition** blade, create a new initiative definition with the following settings:

   - Definition location: the name of the subscription you are using in this lab
   - Name: **az10001b - Tagging initiative**
   - Description: **Collection of tag policies.**
   - Category: **Lab**
   - AVAILABLE DEFINITIONS: search for and select **az10001b - Audit tag and its value**
     - Tag Name: **environment**
     - Tag Value: **lab**

8. From the **Policy - Assignments** blade, navigate to the **Assign initiative** blade and create a new initative assignment with the following settings:

    - Scope: the name of the subscription you are using in this lab
    - Exclusions: none
    - Initiative definition: **az10001b - Tagging initiative**
    - Assignment name: **az10001b - Tagging initiative assignment**
    - Description: **Assignment of az10001b - Tagging initiative**
    - Assigned by: the default value
    - Create a Managed Identity: **unchecked**

9. Navigate to the **Policy - Compliance** blade. Note that **COMPLIANCE STATE** is set to either **Not registered** or **Not started**.

    ⓘNote

    Note: On average, it takes about 10 minutes for a compliance scan to start. Rather than waiting for the compliance scan, proceed to the next task. You will review the compliance status later in this exercise.

Task 3: Implement a policy that enforces resource tagging compliance.

1. Navigate to the **Policy - Definitions** blade.
2. From the **Policy - Definitions** blade, navigate to the **az10001b - Tagging initiative** blade.
3. From the **az10001b - Tagging initiative** blade, navigate to its **Edit initiative definition** blade.
4. Add the built-in policy definition named **Require tag and its value** to the initiative and set its parameters to the following values:

    - Tag Name: **environment**
    - Tag Value: **lab**

    ⓘNote

    Note: At this point, your initiative contains two policies. The first of them evaluates the compliance status and the second one enforces tagging during deployment.

Task 4: Evaluate tagging enforcement and tagging compliance.

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **New** blade, search Azure Marketplace for **Template deployment**.
3. Use the list of search results to navigate to the **Custom deployment** blade.
4. On the **Custom deployment** blade, select the **Build your own template in the editor**.
5. From the **Edit template** blade, load the template file **az-100-01b_azuredeploy.json**.

    ⓘNote

    Note: This is the same template that you used for deployment in the first task of this exercise.

6. Save the template and return to the **Custom deployment** blade.

8. From the **Edit parameters** blade, load the parameters file **az-100-01b_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 **az1000102b-RG**
    - Location: the name of the Azure region which you chose in the first task of this exercise
    - Vm Size: **Standard_DS1_v2**
    - Vm Name: **az1000102b-vm1**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1000102b-vnet1**
    - Environment Name: **lab**

    > ⓘNote
    >
    > **Note**: The deployment will fail. This is expected.

11. You will be presented with the message indicating validation erors. Review the error details, indicating that deployment of resource **az1000102b-vnet1** was disallowed by the policy **Require tag and its value** which is included in the **az10001b - Tagging initiative assignment**.

12. Navigate to the **Policy - Compliance** blade. Identify the entry in the **COMPLIANCE STATE** column.

13. Navigate to the **az10001b - Tagging initiative assignment** blade and review the summary of the compliance status.

14. Display the listing of resource compliance and note which resources have been identified as non-compliant.

    > ⓘNote
    >
    > **Note**: You might need to click **Refresh** button on the **Policy - Compliance** blade in order to see the update to the compliance status.

Task 5: Implement remediation of resource tagging non-compliance.

1. In the Azure portal, navigate to the **az10001b - Tagging initiative** blade.

2. From the **az10001b - Tagging initiative** blade, navigate to its **Edit initiative definition** blade.

3. Add the built-in policy definition named **Append tag and its default value** to the initiative and set its parameters to the following values:

    - Tag Name: **environment**
    - Tag Value: **lab**

4. Delete the custom policy definition named **az10001b - Audit tag and its value** from the initiative.

5. Delete the built-in policy definition named **Require tag and its value** from the initiative and save the changes.

MicrosoftLearning/AZ-103-MicrosoftAzureAdministrator

> ⓘNote
>
> **Note**: At this point, your initiative contains a single policy that automatically remediates tagging non-compliance during deployment of new resources and provides evaluation of compliance status.

6. From the Azure Portal, start a PowerShell session in the Cloud Shell.

> ⓘNote
>
> **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.

7. In the Cloud Shell pane, run the following commands.

Code Copy

```
Get-AzResource -ResourceGroupName 'az1000101b-RG' | ForEach-Object {Set-AzResource -ResourceId $_.ResourceId -Tag @{environment="lab"} -Force }
```

> ⓘNote
>
> **Note**: These commands assign the **environment** tag with the value **lab** to each resource in the resource group **az1000101b-RG**, overwriting any already assigned tags.

> ⓘNote
>
> **Note**: Wait until the commands successfully complete.

8. In the Azure portal, navigate to the **Tags** blade.

9. From the **Tags** blade, display all resources with the **environment** tag set to the value **lab**. Verify that all resources in the resource group **az1000101b-RG** are listed.

Task 6: Evaluate effects of the remediation task on compliance.

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

2. From the **New** blade, search Azure Marketplace for **Template deployment**.

3. Use the list of search results to navigate to the **Custom deployment** blade.

4. On the **Custom deployment** blade, select the **Build your own template in the editor**.

5. From the **Edit template** blade, load the template file **az-100-01b_azuredeploy.json**.

> ⓘNote
>
> **Note**: This is the same template that you used for deployment in the first task of this exercise.

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 **az-100-01b_azuredeploy.parameters.json**.

9. Save the parameters and return to the **Custom deployment** blade.

- Subscription: the name of the subscription you are using in this lab
- Resource group: **az1000102b-RG**
- Location: the name of the Azure region which you chose in the first task of this exercise
- Vm Size: **Standard_DS1_v2**
- Vm Name: **az1000102b-vm1**
- Admin Username: **Student**
- Admin Password: **Pa55w.rd1234**
- Virtual Network Name: **az1000102b-vnet1**
- Environment Name: **lab**

> ⓘNote
>
> **Note**: The deployment will succeed this time. This is expected.

> ⓘNote
>
> **Note**: Do not wait for the deployment to complete before you proceed to the next step.

11. In the Azure portal, navigate to the **Tags** blade.
12. From the **Tags** blade, display all resources with the **environment** tag set to the value **lab**. Note that all the resources deployed to the resource group **az1000102b-RG** have this tag with the same value automatically assigned.

> ⓘNote
>
> **Note**: At this point, only some of the resources have been provisioned, however, you should see that all of them have tags assigned to them.

13. Navigate to the **Policy - Compliance** blade. Identify the entry in the **COMPLIANCE STATE** column.
14. Navigate to the **az10001b - Tagging initiative assignment** blade. Identify the entry in the **COMPLIANCE STATE** column. If the column contains the **Not started** entry, wait until it the compliance scan runs.

> ⓘNote
>
> **Note**: You might need to wait for up to 10 minutes and click **Refresh** button on the **Policy - Compliance** blade in order to see the update to the compliance status.

> ⓘNote
>
> **Note**: Do not wait until the status is listed as compliant but instead proceed to the next exercise.

> ⓘNote
>
> **Result**: After you completed this exercise, you have implemented an initiative and policies that evaluate, enforce, and remediate resource tagging compliance. You also evaluated the effects of policy assignment.

1. Create resource group-level locks to prevent accidental changes
2. Validate functionality of the resource group-level locks

Task 1: Create resource group-level locks to prevent accidental changes

1. In the Azure portal, navigate to the **az1000101b-RG** resource group blade.
2. From the **az1000101b-RG** resource group blade, display the **az1000101b-RG - Locks** blade.
3. From the **az1000101b-RG - Locks** blade, add a lock with the following settings:
   - Lock name: **az1000101b-roLock**
   - Lock type: **Read-only**

Task 2: Validate functionality of the resource group-level locks

1. In the Azure portal, navigate to the **az1000102b-vm1** virtual machine blade.
2. From the **az1000102b-vm1** virtual machine blade, navigate to the **az1000102b-vm1 - Tags** blade.
3. Try setting the value of the **environment** tag to **dev**. Note that the operation is successful.
4. In the Azure portal, navigate to the **az1000101b-vm1** virtual machine blade.
5. From the **az1000101b-vm1** virtual machine blade, navigate to the **az1000101b-vm1 - Tags** blade.
6. Try setting the value of the **environment** tag to **dev**. Note that this time the operation fails. The resulting error message indicates that the resource refused tag assignment, with resource lock being the likely reason.
7. Navigate to the blade of the storage account created in the **az1000101b-RG - Locks** resource group.
8. From the storage account blade, navigate to its **Access keys** blade. Note the resulting error message stating that you cannot access the data plane because a read lock on the resource or its parent.
9. In the Azure portal, navigate to the **az1000101b-RG** resource group blade.
10. From the **az1000101b-RG** resource group blade, navigate to its **Tags** blade.
11. From the **Tags** blade, attempt assigning the **environment** tag with the value **lab** to the resource group and note the error message.

> ⓘNote
>
> **Result**: After you completed this exercise, you have created a resource group-level lock to prevent accidental changes and validated its functionality.

MicrosoftLearning/AZ-103-MicrosoftAzureAdministrator

Lab: Implement Azure File Sync

All tasks in this lab are performed from the Azure portal, except for steps in Exercise 1 and Exercise 2 performed within a Remote Desktop session to an Azure VM.

Lab files:

- **Labfiles\Module_12\Implementing_File_Sync\az-100-02b_azuredeploy.json**
- **Labfiles\Module_12\Implementing_File_Sync\az-100-02b_azuredeploy.parameters.json**

Scenario

Adatum Corporation hosts its file shares in on-premises file servers. Considering its plans to migrate majority of its workloads to Azure, Adatum is looking for the most efficient method to replicate its data to file shares that will be available in Azure. To implement it, Adatum will use Azure File Sync.

Objectives

After completing this lab, you will be able to:

- Deploy an Azure VM by using an Azure Resource Manager template
- Prepare Azure File Sync infrastructure
- Implement and validate Azure File Sync

Exercise 0: Prepare the lab environment

The main tasks for this exercise are as follows:

1. Deploy an Azure VM by using an Azure Resource Manager template

Task 1: Deploy an Azure VM 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 Azure subscription you intend to use in this lab.
2. In the Azure portal, navigate to the **New** blade.
3. From the **New** blade, search Azure Marketplace for **Template deployment**.
4. Use the list of search results to navigate to the **Custom deployment** blade.
5. On the **Custom deployment** blade, select the **Build your own template in the editor**.
6. From the **Edit template** blade, load the template file **az-100-02b_azuredeploy.json**.

   > ⓘNote
   >
   > **Note**: Review the content of the template and note that it defines deployment of an Azure VM hosting Windows Server 2016 Datacenter with a single data disk.

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 **az-100-02b_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:

- Resource group: the name of a new resource group **az1000201b-RG**
- Location: the name of the Azure region which is closest to the lab location and where you can provision Azure VMs
- Vm Size: **Standard_DS1_v2**
- Vm Name: **az1000201b-vm1**
- Admin Username: **Student**
- Admin Password: **Pa55w.rd1234**
- Virtual Network Name: **az1000201b-vnet1**

> ⓘNote
>
> **Note**: To identify Azure regions where you can provision Azure VMs, refer to **https://azure.microsoft.com/en-us/regions/offers/**

> ⓘNote
>
> **Note**: Do not wait for the deployment to complete but proceed to the next exercise. You will use the virtual machine included in this deployment in the next exercise of this lab.

> ⓘNote
>
> **Note**: Keep in mind that the purpose of Azure VM **az1000201b-vm1** is to emulate an on-premises file server in our scenario.

> ⓘNote
>
> **Result**: After you completed this exercise, you have initiated a template deployment of an Azure VM **az1000201b-vm1** that you will use in the next exercise of this lab.

Exercise 1: Prepare Azure File Sync infrastructure

The main tasks for this exercise are as follows:

1. Create an Azure Storage account and a file share
2. Prepare Windows Server 2016 for use with Azure File Sync
3. Run Azure File Sync evaluation tool

Task 1: Create an Azure Storage account and a file share

1. In the Azure portal, navigate to the **New** blade.
2. From the **New** blade, search Azure Marketplace for **Storage account**.
3. Use the list of search results to navigate to the **Create storage account** blade.
4. From the **Create storage account** blade, create a new storage account with the following settings:
    - Subscription: the same subscription you selected in the previous task
    - Resource group: the name of a new resource group **az1000202b-RG**
    - Storage account name: any valid, unique name between 3 and 24 characters consisting of lowercase

   - Location: the name of the Azure region which you selected in the previous task
   - Performance: **Standard**
   - Account kind: **Storage (general purpose v1)**
   - Replication: **Locally-redundant storage (LRS)**
   - Secure transfer required: **Disabled**
   - Allow access from: **All networks**
   - Hierarchical namespace: **Disabled**

   > ⓘNote
   >
   > **Note**: Wait for the storage account to be provisioned then proceed to the next step.

5. In the Azure portal, navigate to the blade representing the newly provisioned storage account.

6. From the storage account blade, display the properties of its File Service.

7. From the storage account **Files** blade, create a new file share with the following settings:

   - Name: **az10002bshare1**
   - Quota: none

Task 2: Prepare Windows Server 2016 for use with Azure File Sync

> ⓘNote
>
> **Note**: Before you start this task, ensure that the template deployment you started in Exercise 0 has completed.

1. In the Azure portal, navigate to the **az1000201b-vm1** blade.

2. From the **az1000201b-vm1** blade, connect to the Azure VM via the RDP protocol and, when prompted to sign in, provide the following credentials:

   - Admin Username: **Student**
   - Admin Password: **Pa55w.rd1234**

3. Within the RDP session to the Azure VM, in Server Manager, navigate to **File and Storage Services**, locate the data disk attached to the Azure VM, initialize it as a **GPT** disk, and use **New Volume Wizard** to create a single volume occupying entire disk with the following settings:

   - Drive letter: **S**
   - File system: **NTFS**
   - Allocation unit size: **Default**
   - Volume label: **Data**

4. Within the RDP session, start a Windows PowerShell session as administrator.

5. From the Windows PowerShell console, set up a file share by running the following:

   Code Copy

```
$directory = New-Item -Type Directory -Path 'S:\az10002bShare'

New-SmbShare -Name $directory.Name -Path $directory.FullName -FullAccess 'Administrators' -
ReadAccess Everyone

Copy-Item -Path 'C:\WindowsAzure\*' -Destination $directory.FullName -Recurse
```

ⓘNote

Note: To populate the file share with sample data, we use content of the *C:\WindowsAzure* folder, which should contain about 100 MB worth of files

6. From the Windows PowerShell console, install the latest Az PowerShell module by running the following:

Code Copy

```
Install-Module -Name Az -AllowClobber
```

ⓘNote

Note: When prompted, confirm that you want to proceed with the installation from PSGallery repository.

Task 3: Run Azure File Sync evaluation tool

1. Within the RDP session to the Azure VM, from the Windows PowerShell console, install the latest version of Package Management and PowerShellGet by running the following:

Code Copy

```
Install-Module -Name PackageManagement -Repository PSGallery -Force

Install-Module -Name PowerShellGet -Repository PSGallery -Force
```

ⓘNote

Note: When prompted, confirm that you want to proceed with the installation of the NuGet provider.

2. Restart the PowerShell session.

3. From the Windows PowerShell console, install the Azure File Sync PowerShell module by running the following:

Code Copy

```
Install-Module -Name Az.StorageSync -AllowClobber -Force
```

4. From the Windows PowerShell console, install the Azure File Sync PowerShell module by running the following:

Code Copy

```
Invoke-AzStorageSyncCompatibilityCheck -Path 'S:\az10002bShare'
```

5. Review the results and verify that no compatibility issues have been found.

MicrosoftLearning/AZ-103-MicrosoftAzureAdministrator

> ⓘNote
>
> **Result**: After you completed this exercise, you have created an Azure Storage account and a file share, prepare Windows Server 2016 for use with Azure File Sync, and run Azure File Sync evaluation tool

Exercise 2: Prepare Azure File Sync infrastructure

The main tasks for this exercise are as follows:

1. Deploy the Storage Sync Service
2. Install the Azure File Sync Agent
3. Register the Windows Server with Storage Sync Service
4. Create sync groups and a cloud endpoint
5. Create a server endpoint
6. Validate Azure File Sync operations

Task 1: Deploy the Storage Sync Service

1. Within the RDP session to the Azure VM, in Server Manager, navigate to the Local Server view and turn off temporarily **IE Enhanced Security Configuration**.
2. Within the RDP session to the Azure VM, start Internet Explorer, browse to the Azure portal at **http://portal.azure.com** and sign in by using the same Microsoft account you used previously in this lab.
3. In the Azure portal, navigate to the **New** blade.
4. From the **New** blade, search Azure Marketplace for **Azure File Sync**.
5. Use the list of search results to navigate to the **Deploy Storage Sync** blade.
6. From the **Deploy Storage Sync** blade, create a Storage Sync Service with the following settings:
   - Name: **az1000202b-ss**
   - Subscription: the same subscription you selected in the previous task
   - Resource group: the name of a new resource group **az1000203b-RG**
   - Location: the name of the Azure region in which you created the storage account earlier in this exercise

Task 2: Install the Azure File Sync Agent.

1. Within the RDP session, start another instance of Internet Explorer, browse to Microsoft Download Center at **https://go.microsoft.com/fwlink/?linkid=858257** and download the Azure File Sync Agent Windows Installer file **StorageSyncAgent_V6_WS2016.msi**.
2. Once the download completes, run the Storage Sync Agent Setup wizard with the default settings to install Azure File Sync Agent.
3. After the Azure File Sync agent installation completes, the **Azure File Sync - Server Registration** wizard will automatically start.

Task 3: Register the Windows Server with Storage Sync Service

1. From the initial page of the **Azure File Sync - Server Registration** wizard, sign in by using the same Microsoft account you used previously in this lab.
2. On the **Choose a Storage Sync Service** page of the **Azure File Sync - Server Registration** wizard, specify

- Azure Subscription: the name of the subscription you are using in this lab
- Resource group: **az1000203b-RG**
- Storage Sync Service: **az1000202b-ss**

3. When prompted, sign in again by using the same Microsoft account you used previously in this lab.

Task 4: Create a sync group and a cloud endpoint

1. Within the RDP session to the Azure VM, in the Azure portal, navigate to the **az1000202b-ss** Storage Sync Service blade.
2. From the **az1000202b-ss** Storage Sync Service blade, navigate to the **Sync group** blade and create a new sync group with the following settings:
   - Sync group name: **az1000202b-syncgroup1**
   - Azure Subscription: the name of the subscription you are using in this lab
   - Storage account: the resource id of the storage account you created in the previous exercise
   - Azure File Share: **az10002bshare1**

Task 5: Create a server endpoint

1. Within the RDP session to the Azure VM, in the Azure portal, from the **az1000202b-ss** Storage Sync Service blade, navigate to the **az1000202b-syncgroup1** blade.
2. From the **az1000202b-syncgroup1** blade, navigate to the **Add server endpoint** blade and create a new server endpoint with the following settings:
   - Registered server: **az1000201b-vm1**
   - Path: **S:\az10002bShare**
   - Cloud Tiering: **Enabled**
     - Always preserve the specified percentage of free space on the volume: **15**
     - Only cache files that were accessed or modified within the specified number of days: **30**
   - Offline Data Transfer: **Disabled**

Task 6: Validate Azure File Sync operations

1. Within the RDP session to the Azure VM, in the Azure portal, monitor the health status of the server endpoint **az100021b-vm1** on the **az1000202b-syncgroup1** blade, as it changes from **Provisioning** to **Pending** and, eventually, to a green checkmark.

   > ⓘNote
   >
   > **Note**: You should be able to proceed to the next step after a few minutes.

2. In the Azure portal, navigate to the blade for the storage account you created earlier in the lab, switch to the **Files** tab and then click **az10002bshare1**.
3. On the **az10002bshare1** blade, click **Connect**.
4. From the **Connect** blade, copy into Clipboard the PowerShell commands that connect to the file share from a Windows computer.
5. Within the RDP session, start a Windows PowerShell ISE session.
6. From the Windows PowerShell ISE session, open the script pane and paste into it the content of your local

7. Add the ` -Persist switch to the end of the line containing the New-PSDrive` cmdlet.

8. Execute the script and verify that its output confirms successful mapping of the Z: drive to the Azure Storage File Service share.

9. Within the RDP session, start File Explorer, navigate to the Z: drive, and verify that it contains the same content as S:\az10002bShare

10. Display the Properties window of individual folders on the Z: drive, review the Security tab, and note that the entries represent NTFS permissions assigned to the corresponding folders on the S: drive.

> ⓘNote
>
> **Result**: After you completed this exercise, you have deployed the Storage Sync Service, installed the Azure File Sync Agent, registered the Windows Server with Storage Sync Service, created a sync group and a cloud endpoint, created a server endpoint, and validated Azure File Sync operations.