Contents

Note the changes!!!

The AZ-100 and AZ-101 certifications have been replaced by a new AZ-103 Microsoft Azure Administrator exam! You can read more about this announcement on Liberty Munson's blog at https://www.microsoft.com/en-us/learning/community-blog-post.aspx?BlogId=8&Id=375217

To support the new exam there is a new AZ-103 GitHub repository, available since May 3 2019. At that time, all the AZ-100 and AZ-101 labs in their respective repositories have been moved to the AZ-103 repository. Those labs are being reused in AZ-103 and we will be maintaining only one repository. The AZ-100 and AZ-101 lab numbering system has be retained, so if you are still teaching the AZ-100 or AZ-101 courses you will be able to easily identify the labs. You will also be able to get the latest version of the labs, and submit any issues you find.

What are we doing?

- We are publishing the lab instructions and lab files on GitHub to allow for interaction between the course authors and MCTs. We hope this will help keep the content current as the Azure platform changes.
- This is a GitHub repository for the AZ-101, Microsoft Azure Integration and Security course.
- You can access the repositories from https://github.com/orgs/MicrosoftLearning/dashboard
- Within each repository there are lab guides in the Markdown format in the Instructions folder. The lab guides in the PDF format are available from the MCT Download Center, however they are not being regularly updated. If appropriate, there are also additional files that are needed to complete the lab within the Allfiles\Labfiles folder. Not every course has corresponding lab files.
- For each delivery, trainers should download the latest files from GitHub. Trainers should also check the Issues tab to see if other MCTs have reported any errors.
- Lab timing estimates are provided but trainers should check to ensure this is accurate based on the audience.
- The lab content has been placed at the end of each course for consistency and convenience. However, as the instructor, you are the best judge to determine when the lab should be offered.
- To conduct you will need an internet connection and an Azure subscription. Please read the Instructor Prep Guide for more information on using the Cloud Shell.
- It is recommended that you provide these materials directly to your students rather than point them to the GitHub repository.

How are we doing?

- If as you are teaching these courses, you identify areas for improvement, please use the Issues tab to provide feedback. We will periodically create new files to incorporate the changes.

General comments regarding the AZ-101 course

- PowerShell scripts in all labs use the current version of Azure PowerShell Az module
- Although not required, it is a good idea to deprovision any existing resources when you have completed each lab. This will help mitigate the risk of exceeding the default vCPU quota limits and minimize usage charges.
- Availability of Azure regions and resources in these regions depends to some extent on the type of subscription you are using. To identify Azure regions available in your subscription, refer to https://azure.microsoft.com/en-us/regions/offers/ . To identify resources available in these regions, refer to https://azure.microsoft.com/en-us/global-infrastructure/services/ . These restrictions might result in failures during template validation or template deployment, in particular when provisioning Azure VMs. If this happens, review error messages and retry deployment with a different VM size or a different region.
- When launching Azure Cloud Shell for the first time, you will likely be prompted to create an Azure file share to persist Cloud Shell files. If so, you can typically accept the defaults, which will result in creation of a storage account in an automatically generated resource group. Note that this might happen again if you delete that storage account.
- Before you perform a template based deployments, you might need to register providers that handle provisioning of resource types referenced in the template. 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). You can perform registration from the subscription's Resource Providers blade in the Azure portal or by using Cloud Shell to run Register-AzResourceProvider PowerShell cmdlet or az provider Azure CLI command.

We hope using this GitHub repository brings a sense of collaboration to the labs and improves the overall quality of the lab experience.

Regards, Azure Administrator Courseware Team # AZ 101 Module 1 - Migrate Servers

1 Lab 2: Migrate on-premises Hyper-V VMs to Azure

Estimated Time: 120 minutes

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

> **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?view=azps-1.2.0

Lab files:

- **Allfiles/Labfiles/AZ-101.1/az-101-01b_azuredeploy.json**
- **Allfiles/Labfiles/AZ-101.1/az-101-01b_azuredeploy.parameters.json**
- **Allfiles/Labfiles/AZ-101.1/DSC/InstallHyperV.ps1**
- **Allfiles/Labfiles/AZ-101.1/DSC/InstallHyperV.zip**

1.0.1 Scenario

Adatum Corporation wants to implement Azure Site Recovery to facilitate migration and protection of on premises Hyper-V VMs

1.0.2 Objectives

After completing this lab, you will be able to:

- Implement Azure Site Recovery Vault
- Configure replication of Hyper-V VMs to Azure by using Azure Site Recovery

1.1 Exercise 1: Implement prerequisites for migration of Hyper-V VMs to Azure by using Azure Site Recovery

Estimated Time: 40 minutes

The main tasks for this exercise are as follows:

1. Provision an Azure VM with nested virtualization support by using an Azure Resource Manager template
2. Create an Azure Recovery Services vault

1.1.0.1 Task 1: Provision an Azure VM with nested virtualization support 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. From the Azure Portal, start a PowerShell session in the Cloud Shell.

   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, upload **az-101-01b_azuredeploy.ps1**, **az-101-01b_azuredeploy.json**, **az-101-01b_azuredeploy.parameters.json**, **InstallHyperV.ps1**, and **InstallHyperV.zip** files.
4. In the Cloud Shell pane, run the following in order to copy the **InstallHyperV.ps1** and **InstallHyperV.zip** files to a DSC subfolder in the home directory.

   ```
   Set-Location -Path $HOME

   New-Item -Type Directory -Path '.\DSC'

   Move-Item -Path '.\InstallHyperV.*' -Destination '.\DSC'
   ```
5. In the Cloud Shell pane, run the following in order to deploy a Standard_DS2_v3 Azure VM (substitute the `<location>` placeholder with the name of the Azure region where you want to perform deployment):

   ```
   ./az-101-01b_azuredeploy.ps1 -resourceGroupName 'az1010101b-RG' -resourceGroupLocation <location>
   ```

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

   Note: If the deployment fails due to the Standard_DS2_v3 size not being available, identify another Azure VM size that supports nested virtualization and specify this size explicitly during the deployment by using the following syntax (substitute the `<vm_Size>` placeholder with the intended Azure VM size)

   ```
   .\az-101-01b_azuredeploy.ps1 -resourceGroupName 'az1010101b-RG' -resourceGroupLocation <location>
   ```

   Note: Do not wait for the deployment to complete but proceed to the next task. You will use the virtual machine **az1010101b-vm1** in the next exercise of this lab.

1.1.0.2 Task 2: Implement an Azure Site Recovery vault

1. In the Azure portal, navigate to the **New** blade.
2. From the **New** blade, search Azure Marketplace for **Backup and Site Recovery (OMS)**.
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: **vaultaz1010102bb**
   - Subscription: the same Azure subscription you used in the previous task of this exercise
   - Resource group: the name of a new resource group **az1010102b-RG**
   - Location: the same Azure region that you selected in the previous task of this exercise.

Result: After you completed this exercise, you have initiated deployment of an Azure VM with nested virtualization support by using an Azure Resource Manager template and created an Azure Recovery Services vault

1.2 Exercise 2: Migrate a Hyper-V VM to Azure by using Azure Site Recovery

Estimated Time: 80 minutes

The main tasks for this exercise are as follows:

1. Provision a Hyper-V VM
2. Prepare infrastructure for Hyper-V VM replication
3. Configure Hyper-V VM replication
4. Review Hyper-V VM replication settings

1.2.0.1 Task 1: Provision a Hyper-V VM

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 **az1010101b-vm1** Azure VM.
2. From the **Overview** pane of the **az1010101b-vm1** blade, generate an RDP file and use it to connect to **az1010101b-vm1**.
3. When prompted, authenticate by specifying the following credentials:
   - User name: **Student**
   - Password: **Pa55w.rd1234**
4. Within the RDP session to **az1010101b-vm1**, start Hyper-V Manager.
5. In the Hyper-V Manager console, use Virtual Switch Manager to create an internal switch named **Internal**.
6. In the Hyper-V Manager console, use New Virtual Machine Wizard to create a Hyper-V VM with the following settings:
   - Name: **az1010101b-vm2**
   - Generation: **1**
   - Startup memory: **2048**
   - Connection: **Internal**
   - Create a virtual disk:
     - Name: **az1010101b-vm2.vhdx**
     - Location: **C:\Users\Public\Documents\Hyper-V\Virtual Hard Disks**
     - Size: **32 GB**
   - Installation Options: **Install an operating system later**

     Note: You will not be installing operating system on the Hyper-V VM. Its sole purpose is to illustrate configuration of the Site Recovery replication.

1.2.0.2 Task 2: Prepare infrastructure for Hyper-V VM replication

1. Within the RDP session, in Server Manager, navigate to the Local Server view and turn off temporarily **IE Enhanced Security Configuration**.
2. Within the RDP session, 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 **az1010102b-RG** resource group blade.
4. From the **az1010102b-RG** resource group blade, navigate to the **vaultaz1010102b** Recovery Services vault blade.

5. From the **vaultaz1010102b** blade, navigate to the **Site Recovery infrastructure** blade.
6. From the **Site Recovery infrastructure** blade, navigate to the **Site Recovery infrastructure - Hyper-V Sites** blade.
7. From the **Site Recovery infrastructure - Hyper-V Sites** blade, navigate to the **Create Hyper-V site** blade.
8. On the **Create Hyper-V site** blade, create a new site named **Adatum Hyper-V Site**.
9. From the **Site Recovery infrastructure** blade, navigate to the **Site Recovery infrastructure - Hyper-V Hosts** blade.
10. From the **Site Recovery infrastructure - Hyper-V Hosts** blade, navigate to the **Add Server** blade.
11. From the **Add Server** blade, use the **Download the installer for the Microsoft Azure Site Recovery Provider** link to download and initiate the installation of the Azure Site Recovery Provider.
12. Run Azure Site Recovery Provider Setup. On the initial page of the setup wizard, turn of the automatic check for updates.
13. After the installation completes, use the **Register** command button to continue to register the local Hyper-V host with the Azure Site Recovery vault by launching **Microsoft Azure Site Recovery Registration Wizard**.
14. Switch to the Azure portal and, from the **Add Server** blade, download the vault registration key for the **Adatum Hyper-V Site** to the **Downloads** folder on the local Hyper-V host.
15. Switch back to the **Microsoft Azure Site Recovery Registration Wizard** and, on the **Vault Settings** page, select the newly downloaded key file.
16. Verify that the subscription, vault name, and Hyper-V site name are correct and complete the registration.
17. Switch back to the Azure portal, navigate to the **Site Recovery infrastructure - Hyper-V Hosts** blade, and verify that the **az1010102b-vm1** appears on the list of servers with the **Connected** status.
18. From the **Site Recovery infrastructure - Hyper-V Hosts** blade, navigate to the **vaultaz1010102b - Site Recovery** blade.
19. From the **vaultaz1010102b - Site Recovery** blade, navigate to the **Prepare Infrastructure** blade and specify the following settings:
    - Protection goal:
      - Where are your machines located?: **On-premises**
      - Where do you want to replicate your machines to?: **To Azure**
      - Are your machines virtualized?: **Yes, with Hyper-V**
      - Are you using System Center VMM to manage your Hyper-V hosts?: **No**
    - Deployment planning:
      - Have you completed deployment planning?: **Yes, I have done it**
    - Prepare source:
      - Step 1: Select Hyper-V Site: **Adatum Hyper-V Site**
      - Step 2: Ensure Hyper-V servers are added: **az1010102b-vm1**
    - Target:
      - Step 1: Select Azure subscription:
        * Subscription: the same subscription you selected earlier in this lab
        * Select the deployment model used after failover: **Resource Manager**
      - Step 2: Ensure that at least one compatible Azure storage account exists:
        * Use the **+ Storage account** option to create a **Storage (general purpose v1) Standard** storage account with **Locally-redundant storage (LRS)** replication settings.

Note: The new storage account will be automatically created in the same resource group as the Azure Site Recovery vault.

- Step 3: Ensure that at least one compatible Azure virtual network exists:
  - Use the **+ Network** option to create a virtual network named **az-1010102b-vnet2** with the address space of **10.201.16.0/20**, a subnet named **subnet0**, and the subnet range of **10.201.16.0/24**.

Note: The new virtual network will be automatically created in the same resource group as the Azure Site Recovery vault.

- Replication policy:
  - Use the **+Create and Associate option** to navigate to the **Create and associate policy** and configure the policy with the following settings:
    - Name: **Adatum Hyper-V VM replication policy**
    - Source type: **Hyper-V**
    - Target type: **Azure**
    - Copy frequency: **5 Minutes**
    - Recovery point retention in hours: **2**
    - App-consistent snapshot frequency in hours: **1**
    - Initial replication start time: **Immediately**
    - Associated Hyper-V site: **Adatum Hyper-V Site**

Note: From the **Replication policy** blade, you can use the **View job in progress** links to navigate to the **Associate replication policy** blade and monitor progress of applying the replication policy.

20. Navigate back to the **Prepare infrastructure** blade and finalize the configuration.

1.2.0.3 Task 3: Enable Hyper-V VM replication

1. Within the RDP session, in the Azure portal, navigate to the **vaultaz1010102b** blade.
2. From the **vaultaz1010102b** blade, navigate to the **vaultaz1010102b - Replicated items** blade.
3. From the **vaultaz1010102b - Replicated items** blade, navigate to the **Enable replication** blade and enable Hyper-V VM replication with the following settings:
   - Source:
     - Source: **On-premises**
     - Source location: **Adatum Hyper-V Site**
   - Target:
     - Target: **Azure**
     - Subscription: the same subscription you selected earlier in this lab
     - Post-failover resource group: **az1010102b-RG**
     - Post-failover deployment model: **Resource Manager**
     - Storage account: the storage account you created in the previous task
     - Azure network: **Configure now for selected virtual machines**
     - Post-failover virtual network: **az-1010102b-vnet2**
     - Subnet: **subnet0 (10.201.16.0/24)**
   - Select virtual machines:
     - **az1010101b-vm2**
   - Configure properties:

- Defaults:
  - OS TYPE: **Windows**
  - OS DISK: **Need to select per VM.**
  - DISK TO REPLICATE: **Need to select per VM.**
- az1010101b-vm2:
  - OS TYPE: **Windows**
  - OS DISK: **az1010101b-vm2**
  - DISK TO REPLICATE: **All Disks [1]**

- Configure replication settings:
  - Replication policy: **Adatum Hyper-V VM replication policy**

4. Back on the **Enable replication blade**, enable replication.

1.2.0.4 Task 4: Review Hyper-V VM replication settings

1. Within the RDP session, in the Azure portal, on the **vaultaz1010102b - Replicated items** blade, ensure that there is an entry representing the **az1010101b-vm2** Azure VM and verify that its **REPLICATION HEALTH** is **Healthy**.

   Note: You might need to refresh the view of the page in order to view the replicated VM.

2. Monitor the **STATUS** column and wait until it changes to **Protected**.
3. From the **vaultaz1010102b - Replicated items** blade, navigate to the replicated item blade of the **vaultaz1010102b** Hyper-V VM.
4. On the **az1010101b-vm** replicated item blade, review the **Health and status**, **Failover readiness**, **Latest recovery points**, and **Infrastructure view** sections. Note the **Planned Failover**, **Failover** and **Test Failover** toolbar icons.

   Result: After you completed this exercise, you have provisioned a Hyper-V VM, prepared infrastructure for Hyper-V VM replication, configure Hyper-V VM replication, and reviewed Hyper-V VM replication settings. # AZ 101 Module 1 - Migrate Servers

2 Lab: Implement Azure Site Recovery between Azure regions

Estimated Time: 30 minutes

All tasks in this lab are performed from the Azure portal

Lab files:

- **Labfiles\AZ101\Mod01\az-101-01_azuredeploy.json**
- **Labfiles\AZ101\Mod01\az-101-01_azuredeploy.parameters.json**

2.0.1 Scenario

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

2.0.2 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

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

Estimated Time: 10 minutes

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

2.1.0.1 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, select the **Build your own template in the editor**.
6. From the **Edit template** blade, load the template file **Labfiles\AZ101\Mod01\az-101-01_azuredeploy.json**.

   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\AZ101\Mod01\az-101-01_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 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: To identify Azure regions where you can provision Azure VMs, refer to **https://azure.microsoft.com/en-us/regions/offers/**

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

2.1.0.2 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 (OMS)**.
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

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.

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

Estimated Time: 20 minutes

The main tasks for this exercise are as follows:

1. Configure Azure VM replication
2. Review Azure VM replication settings

2.2.0.1 Task 1: Configure Azure VM replication

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

1. In the 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.

2.2.0.2 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 replication**.
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.

   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.

   Result: After you completed this exercise, you have configured replication of an Azure VM and reviewed Azure VM replication settings. # AZ 101 Module 2 - Implement and Manage Application Services

3 Lab 2: Monitor changes to Azure resources by using Azure Event Grid and Azure Logic Apps

Estimated Time: 90 minutes

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

> **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?view=azps-1.2.0

Lab files: none

3.0.1 Scenario

Adatum Corporation wants to monitor changes to Azure resources by using Azure serverless services

3.0.2 Objectives

After completing this lab, you will be able to:

- Implement serverless services in Azure
- Configure Azure Logic App and Event Grid to facilitate event monitoring

3.1 Exercise 1: Implement prerequisites of the monitoring solution

Estimated Time: 40 minutes

The main tasks for this exercise are as follows:

1. Create an Azure Storage account

2. Create an Azure Logic App
3. Create an Azure AD service principal
4. Configure RBAC-based permissions
5. Register the Event Grid resource provider

3.1.0.1 Task 1: Create an Azure Storage account

1. From the lab virtual machine, start Microsoft Edge and browse to the Azure portal at **http://portal.azure.com** and sign in by using the Microsoft account that has the Owner role in the target Azure subscription.
2. In the Azure portal, navigate to the **New** blade.
3. From the **New** blade, search Azure Marketplace for **Storage account**.
4. Use the search results to navigate to the **Create storage account** blade.
5. On the **Basic** tab of the **Create storage account** blade, configure 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 **az1010201b-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 is closest to the lab location and where you can provision Azure resources
   - Performance: **Standard**
   - Account kind: **Storage (general purpose v1)**
   - Replication: **Locally-redundant storage (LRS)**
6. From the **Basic** tab of the **Create storage account** blade, switch to the **Advanced** tab and configure the following settings:
   - Secure transfer required: **Enabled**
   - VIRTUAL NETWORKS:
     - Allow access from: **All networks**
   - DATA LAKE STORAGE GEN2:
     - Hierarchical namespace: **Disabled**
7. From the **Advanced** tab of the **Create storage account** blade, switch to the **Review + create** tab and initiate creation of the storage account.

   Note: Do not wait for the storage acocunt to be created but instead proceed to the next task.

3.1.0.2 Task 2: Create an Azure Logic App

1. In the Azure portal, navigate to the **New** blade.
2. From the **New** blade, search Azure Marketplace for **Logic App**.
3. Use the search results to navigate to the **Logic App Create** blade and create an instance of Logic App with the following settings:
   - Name: **logicappaz1010201b**
   - Subscription: the name of the Azure subscription you are using in this lab
   - Resource group: the name of a new resource group **az1010202b-RG**
   - Location: the same Azure region into which you deployed the storage account in the previous task
   - Log Analytics: **Off**
4. Wait until the logic app is provisioned. This will take about a minute.

3.1.0.3 Task 3: Create an Azure AD service principal

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

   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 to create a new Azure AD application that you will associate with the service principal you create in the subsequent steps of this task:

   ```
   $password = 'Pa55w.rd1234'

   $securePassword = ConvertTo-SecureString -Force -AsPlainText -String $password

   $aadApp1010201b = New-AzADApplication -DisplayName 'aadApp1010201b' -HomePage 'http://aadApp101020
   ```

3. From the Cloud Shell pane, run the following command to create a new Azure AD service principal associated with the application you created in the previous step:

   ```
   New-AzADServicePrincipal -ApplicationId $aadApp1010201b.ApplicationId.Guid
   ```

4. In the output of the **New-AzureRmADServicePrincipal** cmdlet, note the value of the **ApplicationId** property. You will need this in the next exercise of this lab.

5. From the Cloud Shell pane, run the following cmdlet to identify the value of the **Id** property of the current Azure subscription and the value of the **TenantId** property of the Azure AD tenant associated with that subscription (you will also need them in the next exercise of this lab):

   ```
   Get-AzSubscription
   ```

6. Close the Cloud Shell pane.

3.1.0.4 Task 4: Configure RBAC-based permissions

1. In the Azure portal, navigate to the blade displaying properties of your Azure subscription.
2. From the Azure subscription blade, navigate to its **Access control (IAM)** blade.
3. From the **Access control (IAM)** blade, navigate to the **Add role assignment** blade.
4. From the **Add role assignment** blade, assign the **Reader** role within the scope of the Azure subscription to the **aadApp1010201b** service principal.

3.1.0.5 Task 5: Register the Event Grid resource provider

1. In the Azure portal, in the Microsoft Edge window, reopen the **PowerShell** session within the **Cloud Shell**.
2. From the Cloud Shell pane, run the following command to register the Microsoft.EventGrid resource provider:

   ```
   Register-AzResourceProvider -ProviderNamespace Microsoft.EventGrid
   ```

3. Close the Cloud Shell pane.

   Result: After you completed this exercise, you have created a storage account, a logic app that you will configure in the next exercise of this lab, an Azure AD service principal that you will reference during that configuration, assigned to that service principal the Reader role within the Azure subscription, and registered the Event Grid resource provider.

3.2 Exercise 2: Configure Azure Logic App and Event Grid

Estimated Time: 50 minutes

The main tasks for this exercise are as follows:

1. Add an Event Grid-based trigger to the Azure Logic App
2. Add an action to the Azure Logic App

3. Configure event subscription
4. Validate the functionality of the Azure Logic App

3.2.0.1 Task 1: Add an Event Grid-based trigger to the Azure Logic App

1. In the the Azure portal, navigate to the blade of the newly provisioned Azure logic app.
2. From the blade of the newly provisioned Azure logic app, navigate to its **Logic App Designer** blade.
3. From the **Logic App Designer** blade, use the **Blank Logic App** option to navigate to the blank designer workspace.
4. On the **Logic App Designer** workspace blade, display the list of connectors and triggers that can be added to the workspace.
5. Use the **Search connectors and triggers** text box to locate the **Event Grid** triggers and, from the list of results, add the **When a resource event occurs (preview) Azure Event Grid** trigger to the designer workspace.
6. On the **Azure Event Grid** tile, use the **Connect with Service Principal** link and specify the following values:
   - Connection Name: **egcaz1010201b**
   - Client ID: the **ApplicationId** property you identified in the previous exercise
   - Client Secret: **Pa55w.rd1234**
   - Tenant: the **TenantId** property you identified in the previous exercise
7. On the **When a resource event occurs** tile, specify the following values to configure the new connection:
   - Subscription: use the **Enter custom value** option to enter the subscription **Id** property you identified in the previous exercise
   - Resource Type: **Microsoft.Resources.resourceGroups**
   - Resource Name: use the **Enter custom value** option to enter **/subscriptions/*subscriptionId*/resourceGroup RG**, where ***subscriptionId*** is the subscription **Id** property you identified in the previous exercise
   - Event Type Item - 1: **Microsoft.Resources.ResourceWriteSuccess**
   - Event Type Item - 2: **Microsoft.Resources.ResourceDeleteSuccess**

3.2.0.2 Task 2: Add an action to the Azure Logic App

1. In the the Azure portal, on the **Logic App Designer** blade of the newly provisioned Azure logic app, use the **+ New step** option to display the **Choose an action** pane.
2. In the **Choose an action** pane, use the **Search connectors and actions** text box to select the **Outlook.com** and its **Send an email** action.
3. In the **Outlook.com** pane, select the **Sign in** button and, when prompted, provide the credentials of the Microsoft Account you are using in this lab.
4. When prompted for the consent, grant Azure Logic App permissions to access Outlook resources.
5. In the **Send an email** pane, specify the following settings:
   - To: the name of your Microsoft Account
   - Subject: **Resource updated:** followed by the **Subject Dynamic Content** entry.
   - Body: the **Topic Dynamic Content** entry, followed by **Event type Dynamic Content** entry, followed by **ID Dynamic Content** entry, followed by the **Event Time Dynamic Content** entry.
6. Save the changes you made on the **Logic App Designer** blade.

3.2.0.3 Task 3: Configure event subscription

Note: In order to configure event subscription, you need to first identify the callback URL of the Azure logic app

1. In the Azure portal, navigate back to the **logicappaz1010201b** blade and, in the **Summary** section, click **See trigger history**.
2. On the **When_a_resource_event_occurs** blade, copy the value of the **Callback url [POST]** text box into Clipboard.

   Note: Once you identified the callback URL, you can proceed to configure event subscription.
3. In the Azure portal, navigate to the **az1010201b-RG** resource group and display its **Events** blade.
4. On the **az1010201b-RG - Events** blade, use the **Web Hook** option to navigate to the **Create Event Subscription** blade.
5. On the **Create Event Subscription** blade, clear the **Subscribe to all event types** checkbox and, in the **Defined Event Types** drop down list, ensure that only the checkboxes next to the **Resource Write Success** and **Resource Delete Success** are selected.
6. In the **ENDPOINT DETAILS** section, in the **Endpoint type** drop-down list, select the **Web Hook** option and use the **Select an endpoint** link to display the **Select Web Hook** blade.
7. On the **Select Web Hook** blade, in the **Subscriber Endpoint**, paste the value of the **Callback url [POST]** of the Azure logic app you copied in the previous task and confirm the selection.
8. Set the **Name** in the **EVENT SUBSCRIPTION DETAILS** section to **event-subscription-az1010201b**.
9. Use the **Create** button to finalize your settings.

3.2.0.4 Task 4: Validate the functionality of the Azure Logic App

1. In the Azure portal, navigate to the **az1010201b-RG** resource group and, in the list of its resources, locate the Azure storage account you created in the first exercise.
2. Navigate to the storage account blade, from its **Configuration** blade, set the **Secure transfer required** setting to **Disabled**, and save the change.
3. Navigate to the **logicappaz1010201b** blade and note that the **Runs history** includes the entry corresponding to configuration change of the Azure storage account.
4. Navigate to the inbox of the email account of the Microsoft account you used in this exercise and verify that includes an email generated by the logic app.

   Result: After you completed this exercise, you have configured an Azure logic app to monitor changes to a resource group by adding an Event Grid-based trigger to the Azure Logic App, adding an action to the Azure Logic App, and configuring an event subscription. You also validated the functionality of the newly configured Azure Logic App.

4 AZ 101 Module 2 - Implement and Manage Application Services

5 Lab: Implement and Manage Azure Web Apps

Estimated Time: 45 minutes

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

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?view=azps-1.2.0

Lab files: none

5.0.1 Scenario

Adatum Corporation wants to implement Azure web apps and configure them for scalability and performance

5.0.2 Objectives

After completing this lab, you will be able to:

- Implement Azure web apps
- Manage scalability and performance of Azure web apps

5.1 Exercise 1: Implement Azure web apps

Estimated Time: 20 minutes

The main tasks for this exercise are as follows:

1. Create an Azure web app
2. Configure web app deployment settings.
3. Create a staging deployment slot
4. Deploy code to the staging deployment slot and perform slot swap

5.1.0.1 Task 1: Create an Azure web app

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 **Web app**.
4. Use the list of search results to navigate to the **Web App Create** blade.
5. From the **Web App Create** blade, create a new web app with the following settings:
   - App name: any unique, valid name
   - Subscription: the name of the Azure subscription you intend to use in this lab
   - Resource group: the name of a new resource group **az1010201-RG**
   - OS: **Windows**
   - Publish: **Code**
   - App Service plan/Location: a new App Service plan with the following settings:
     - Name: **az1010201-AppServicePlan1**
     - Location: the name of an Azure region where you can provision Azure web apps
     - Pricing tier: **S1 Standard**
   - Application Insights: **Off**

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

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

     Note: Wait until the web app is created before you proceed to the next task. This should take about a minute.

5.1.0.2 Task 2: Create a staging deployment slot

1. In the Azure portal, navigate to the blade of the newly provisioned web app.
2. On the web app blade, use the **Browse** toolbar icon to open a new browser tab displaying the default App Service home page.
3. Close the browser tab displaying the default App Service home page.
4. In the Azure portal, from the web app blade, display its **Deployment slots** blade.

5. From the **Deployment slots** blade, add a slot with the following settings:
    - Name: **staging**
    - Configuration Source: **Don't clone configuration from an existing slot**
6. From the **Deployment slots** blade, navigate to the **staging** blade displaying the properties of the newly created deployment slot.
7. On the **staging** blade, use the **Browse** toolbar icon to open a new browser tab displaying the default App Service home page.
8. Close the browser tab displaying the default App Service home page in the staging deployment slot.

5.1.0.3 Task 3: Configure web app deployment settings.

1. In the Azure portal, from the **staging blade**, display the **staging - Deployment Center** blade.
2. On the **staging - Deployment Center** blade, configure the following settings:
    - Source control: **Local Git**
    - Build provider: **App Service Kudu build server**
3. Note the resulting **Git Clone Url**. You will need in the next task of this exercise.
4. From the **Deployment Center** blade, use the **Deployment Credentials** toolbar icon to display **User Credentials** pane.
5. In the **User Credentials** pane, set the password to **Pa55w.rd1234** and save the newly set credentials.

5.1.0.4 Task 4: Deploy code to the staging deployment slot and perform a slot swap

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

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

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

    ```
    git clone https://github.com/Azure-Samples/php-docs-hello-world
    ```

    Note: This command clones a remote repository containing a sample web app code

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

    ```
    Set-Location -Path $HOME/php-docs-hello-world/
    ```

    Note: This command sets the current location to the newly created clone of the local repository containing the sample web app code

4. In the Cloud Shell pane, run the following command, replacing the `<git_clone_url>` placeholder with the value of the **Git Clone Url** you identified in the previous task:

    ```
    git remote add azure <git_clone_url>
    ```

    Note: This command connects the local repo to the Git repo in Azure repos associated with the Azure web app

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

    ```
    git push azure master
    ```

    Note: This command pushes the sample web app code from the local repository to the Azure web app staging deployment slot

6. When prompted, type the password **Pa55w.rd1234** you set in the previous task.
7. Close the Cloud Shell pane.
8. In the Azure portal, navigate to the **Overview** section of the **staging** blade.

9. On the **staging** blade, use the **Browse** toolbar icon to open a new browser tab. Note that the browser displays the **Hello World!** page you just deployed.
10. Close the browser tab displaying the **Hello World!** page.
11. On the **staging** blade, use the **Swap** toolbar icon to display the **Swap** blade.
12. On the **Swap** blade, initiate slot swap with the following settings:
    - Swap type: **swap**
    - Source: **staging**
    - Destination: **production**
    - Preview Changes: review the changes that will be applied to the destination slot
13. On the **staging** blade, use the **Browse** toolbar icon to open a new browser tab. Note that the browser displays now the default App Service home page.
14. Close the browser tab displaying the default App Service home page.
15. From the **staging** blade, display its **Deployment slots** blade.
16. From the **Deployment slots** blade, navigate back to the blade displaying the properties of the production slot of the Azure web app.
17. On the web app blade, use the **Browse** toolbar icon to open a new browser tab. Note that the browser displays the **Hello World!** page.
18. Close the browser tab displaying the **Hello World!** page.

    Result: After you completed this exercise, you have created an Azure web app, configured web app deployment settings, created a staging deployment slot, deployed code to the staging deployment slot, and performed a slot swap.

5.2 Exercise 2: Manage scalability and performance of Azure web apps

Estimated Time: 25 minutes

The main tasks for this exercise are as follows:

1. Configure and test autoscaling of the Azure web app
2. Configure Content Delivery Network (CDN) for the Azure web app

5.2.0.1 Task 1: Configure and test autoscaling of the Azure web app

1. In the Azure portal, from the web app blade, display the **Scale out (App Service plan)** blade.
2. On the web app **Scale out (App Service plan)** blade, enable autoscale with the following settings:
    - Autoscale setting name: **az1010201-AutoScaling**
    - Resource group: **az1010201-RG**
    - **Default Auto created scale condition**:
        - Scale mode: **Scale based on a metric**
        - Rules: **Scale out**
        - When:
            * Time aggregation: **Average**
            * Metric name: **Data In**
            * Time grain statistics: **Average**
            * Operator: **Greater than**
            * Threshold: **1048576** bytes
            * Duration: **5** minutes
            * Cool down (minutes): **5**

- Instance limits (Minimum): **1**
- Instance limits (Maximum): **2**
- Instance limits (Default): **1**

- **Auto created scale condition 1**:
  - Scale mode: **Scale based on a metric**
  - Rules: **Scale in**
  - When:
    - Time aggregation: **Average**
    - Metric name: **Data In**
    - Time grain statistics: **Average**
    - Operator: **Greater than**
    - Threshold: **1048576** bytes
    - Duration: **5** minutes
    - Cool down (minutes): **5**
  - Instance limits (Minimum): **1**
  - Instance limits (Maximum): **1**
  - Instance limits (Default): **1**
  - Schedule: **Repeat specific days**
  - Repeat every: **Monday**, **Tuesday**, **Wednesday**, **Thursday**, **Friday**, **Saturday**, **Sunday**
  - Timezone: **(UTC-5:00) Eastern Time (US & Canada)**
  - Start time: **00:00**
  - End time: **23:59**

3. From the Azure Portal, start a PowerShell session in the Cloud Shell.
4. In the Cloud Shell pane, run the following commands:

```
$resourceGroup = Get-AzResourceGroup -Name 'az1010201-RG'
$webapp = Get-AzWebApp -ResourceGroupName $resourceGroup.ResourceGroupName
while ($true) { Invoke-WebRequest -Uri $webapp.DefaultHostName }
```

   Note: These commands submit requests to the Azure web app in a loop in order to trigger autoscaling

5. Minimize the Cloud Shell pane and, from the web app blade, display the **Process explorer** blade. This will allow you to monitor the number of instances and their resource utilization.
6. Once you notice that the number of instances has increased to 2, reopen the Cloud Shell pane and terminate the PowerShell script by pressing **Ctrl+C**.

   Note: It might take about 5 minutes for the number of instances to increase to 2

5.2.0.2 Task 2: Configure Content Delivery Network (CDN) for the Azure web app

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **CDN**.
3. Use the list of search results to navigate to the **CDN profiile** blade.
4. From the **CDN profile** blade, create a CDN profile with the following settings:
   - App name: **az1010202cdn-profile**
   - Subscription: the name of the Azure subscription you are using in this lab
   - Resource group: the name of a new resource group **az1010202-RG**

- Location: the name of an Azure region where you can provision Azure CDN
- Pricing tier: **Standard Microsoft**
- Create a new CDN endpoint now: enabled
- CDN endpoint name: any unique, valid name consisting of letters and digits
- Origin type: **Web App**
- Origin hostname: the fully qualified name of the web app you created in the first exercise

  Note: The green check mark in the **CDN endpoint name** text box will indicate whether the name you typed in is valid and unique.

  Note: Wait until the CDN endpoint is created before you proceed to the next task. This should take less than a minute.

5. In the Azure portal, navigate to the **az1010202cdn-profile** blade.
6. From the **az1010202cdn-profile** blade, note the list of endpoints and use it to navigate to the newly created endpoint.
7. From the endpoint blade, note the value of the **Endpoint hostname**.
8. Open a new tab of Microsoft Edge and browse to the URL representing the **Endpoint hostname**. Note that the browser displays the **Hello World!** page you just deployed.

   Result: After you completed this exercise, you have configured and tested autoscaling of the Azure web app as well as configured Content Delivery Network (CDN) for the Azure web app. # AZ 101 Module 3 - Implement Advanced Virtual Networking

6 Lab 2: Use Azure Network Watcher for monitoring and troubleshooting network connectivity

Estimated Time: 90 minutes

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

> **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?view=azps-1.2.0

Lab files:

- **Allfiles/Labfiles/AZ-101.3/az-101-03b_01_azuredeploy.json**
- **Allfiles/Labfiles/AZ-101.3/az-101-03b_02_azuredeploy.json**
- **Allfiles/Labfiles/AZ-101.3/az-101-03b_01_azuredeploy.parameters.json**
- **Allfiles/Labfiles/AZ-101.3/az-101-03b_02_azuredeploy.parameters.json**

6.0.1 Scenario

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

6.0.2 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

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

Estimated Time: 45 minutes

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

6.1.0.1 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 **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-101-03b_01_azuredeploy.json**.

   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_DS1_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: 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):

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

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):

```
Get-AzSqlCapability -LocationName <location>
```

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

12. In the Azure portal, navigate to the **New** blade.
13. From the **New** blade, search Azure Marketplace for **Template deployment**.
14. Use the list of search results to navigate to the **Custom deployment** blade.
15. On the **Custom deployment** blade, select the **Build your own template in the editor**.
16. From the **Edit template** blade, load the template file **az-101-03b_02_azuredeploy.json**.

    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_DS1_v2**
    - Vm Name: **az1010302b-vm2**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1010302b-vnet2**

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

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

6.1.0.2 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.

6.1.0.3 Task 3: Establish peering between Azure virtual networks

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**

- Allow virtual network access: **Enabled**
- Allow forwarded traffic: disabled
- Allow gateway transit: disabled
- Use remote gateways: disabled

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

6.1.0.4 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 **az1010301b-vnet1 - Service endpoints** blade.
3. From the **az1010301b-vnet1 - Service endpoints** blade, add service endpoints with the following settings:
   - Service: **Microsoft.Storage**
   - Subnets: **subnet0**
   - Service: **Microsoft.Sql**
   - Subnets: **subnet0**
4. In the Azure portal, navigate to the **az1000301b-RG** resource group blade.
5. From the **az1010301b-RG** resource group blade, navigate to the blade of the storage account included in the resource group.
6. From the storage account blade, navigate to its **Firewalls and virtual networks** blade.
7. From the **Firewalls and virtual networks** blade of the storage account, configure the following settings:
   - Allow access from: **Selected networks**
   - Virtual networks:
     - VIRTUAL NETWORK: **az1010301b-vnet1**
       - 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**

8. In the Azure portal, navigate to the **az1010301b-RG** resource group blade.
9. From the **az1010301b-RG** resource group blade, navigate to the **az1010301b-db1** Azure SQL Database blade.
10. From the **az1010301b-db1** Azure SQL Database blade, navigate to its server's **Firewall settings** blade.
11. From the **Firewall settings** 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**

    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.

6.2 Exercise 2: Use Azure Network Watcher to monitor network connectivity

Estimated Time: 45 minutes

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

6.2.0.1 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 **Network Watcher - 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**
      - Source type: **Virtual machine**
      - Virtual machine: **az1010301b-vm1**
    - Destination: **Specify manually**
      - URI, FQDN or IPv4: **10.203.16.4**

        Note: **10.203.16.4** is the private IP address of the second Azure VM az1010301b-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: If this is the first time you are using Network Watcher, the check can take up to 5 minutes.

6.2.0.2 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: 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:

   ```
   [System.Net.Dns]::GetHostAddresses($(Get-AzStorageAccount -ResourceGroupName 'az1010301b-RG')[0].S
   ```

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 blob service endpoint of the storage account you identified in the previous step of this task
   - 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: 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**.

   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**

6.2.0.3 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].FullyQ
   ```
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 low latency.

   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: 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 Azure SQL Database server you identified earlier in this task

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

    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. # AZ 101 Module 3 - Implement Advanced Virtual Networking

7 Lab: Implement Advanced Virtual Networking

Estimated Time: 60 minutes

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

Lab files:

- **Labfiles\AZ101\Mod03\az-101-03_01_azuredeploy.json**
- **Labfiles\AZ101\Mod03\az-101-03_01_1_azuredeploy.parameters.json**
- **Labfiles\AZ101\Mod03\az-101-03_01_2_azuredeploy.parameters.json**

7.0.1 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

7.0.2 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

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

Estimated Time: 20 minutes

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

7.1.0.1 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, select the **Build your own template in the editor**.
6. From the **Edit template** blade, load the template file **Labfiles\AZ101\Mod03\az-101-03_01_azuredeploy.json**.

   Note: Review the content of the template and note that it defines deployment of two Azure VMs hosting Windows Server 2016 Datacenter Core into an availability set.
7. Save the template and return to the **Custom deployment** blade.
8. From the **Custom deployment** blade, navigate to the **Edit parameters** blade.
9. From the **Edit parameters** blade, load the parameters file **Labfiles\AZ101\Mod03\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_DS1_v2**
    - Virtual Network Name: **az1010301-vnet**
    - Address Prefix: **10.101.31.0/24**
    - Virtual Network Resource Group: **az1010301-RG**
    - Subnet0Name: **subnet0**
    - Subnet0Prefix: **10.101.31.0/26**
    - Availability Set Name: **az1010301w-avset**
    - Network Security Group Name: **az1010301w-vm-nsg**
    - Modules Url: **https://github.com/Azure/azure-quickstart-templates/raw/master/dsc-extension-iis-server-windows-vm/ContosoWebsite.ps1.zip**
    - Configuration Function: **ContosoWebsite.ps1\\ContosoWebsite**

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

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

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

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Template deployment**.
3. Use the list of search results to navigate to the **Deploy a custom template** blade.
4. On the **Custom deployment** blade, select the **Build your own template in the editor**.

5. From the **Edit template** blade, load the template file **Labfiles\AZ101\Mod03\az-101-03_01_azuredeploy.json**.

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

6. Save the template and return to the **Custom deployment** blade.
7. From the **Custom deployment** blade, navigate to the **Edit parameters** blade.
8. From the **Edit parameters** blade, load the parameters file **Labfiles\AZ101\Mod03\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_DS1_v2**
    - Virtual Network Name: **az1010302-vnet**
    - Address Prefix: **10.101.32.0/24**
    - Virtual Network Resource Group: **az1010302-RG**
    - Subnet0Name: **subnet0**
    - Subnet0Prefix: **10.101.32.0/26**
    - Availability Set Name: **az1010302w-avset**
    - Network Security Group Name: **az1010302w-vm-nsg**
    - Modules Url: **https://github.com/Azure/azure-quickstart-templates/raw/master/dsc-extension-iis-server-windows-vm/ContosoWebsite.ps1.zip**
    - Configuration Function: **ContosoWebsite.ps1\\ContosoWebsite**

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

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

7.2 Exercise 1: Implement Azure Load Balancing

Estimated Time: 30 minutes

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

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

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

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Load Balancer**.
3. Use the list of search results to navigate to the **Create load balancer** blade.
4. From the **Create load balancer** blade, create a new Azure Load Balancer with the following settings:
   - Name: **az1010301w-lb**
   - Type: **Public**
   - SKU: **Basic**
   - Public IP address: a new public IP address named **az1010301w-lb-pip**
   - Assignment: **Dynamic**
   - Subsciption: 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 setttings:
   - Name: **az1010301w-bepool**
   - 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: It is possible that the IP addresses of the Azure VMs are asssigned in the reverse order.

     Note: Wait for the operation to complete. This should take less than a minute.
8. From the **az1010301w-lb - Backend pools** blade, display the **az1010301w-lb - Health probes** blade.
9. From the **az1010301w-lb - Health probes** blade, add a health probe with the following settings:
   - Name: **az1010301w-healthprobe**
   - Protocol: **TCP**
   - Port: **80**
   - Interval: **5** seconds
   - Unhealthy threshold: **2** consecutive failures

     Note: Wait for the operation to complete. This should take less than a minute.
10. From the **az1010301w-lb - Health probes** blade, display the **az1010301w-lb - Load balancing rules** blade.

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

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

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

1. In the Azure portal, navigate to the **Create a resource** blade.
2. From the **Create a resource** blade, search Azure Marketplace for **Load Balancer**.
3. Use the list of search results to navigate to the **Create load balancer** blade.
4. From the **Create load balancer** blade, create a new Azure Load Balancer with the following settings:
    - Name: **az1010302w-lb**
    - Type: **Public**
    - SKU: **Basic**
    - Public IP address: a new public IP address named **az1010302w-lb-pip**
    - Assignment: **Dynamic**
    - Subsciption: the name of the subscription you are using in this lab
    - Resource group: **az1010302-RG**
    - Location: the name of the Azure region in which you deployed Azure VMs in the second task of the previous exercise
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 setttings:
    - 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: It is possible that the IP addresses of the Azure VMs are asssigned in the reverse order.

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

8. From the **az1010302w-lb - Backend pools** blade, display the **az1010302w-lb - Health probes** blade.
9. From the **az1010302w-lb - Health probes** blade, add a health probe with the following settings:
   - Name: **az1010302w-healthprobe**
   - Protocol: **TCP**
   - Port: **80**
   - Interval: **5** seconds
   - Unhealthy threshold: **2** consecutive failures

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

7.2.0.3 Task 3: Implement Azure NAT rules in the first region

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

   Note: The NAT functionality does not rely on health probes.
3. From the **az1010301w-lb - Inbound NAT rules** blade, add the first inbound NAT rule with the following settings:
   - Name: **az1010301w-vm0-RDP**
   - Frontend IP address: **LoadBalancedFrontEnd**
   - IP Version: **IPv4**
   - Service: **Custom**
   - Protocol: **TCP**
   - Port: **33890**
   - Target virtual machine: **az1010301w-vm0**
   - Network IP configuration: **ipconfig1 (10.101.31.4)** or **ipconfig1 (10.101.31.5)**
   - Port mapping: **Custom**
   - Floating IP (direct server return): **Disabled**
   - Target port: **3389**

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

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

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

7.2.0.4 Task 4: Implement Azure NAT rules in the second region

1. In the Azure portal, navigate to the blade of the Azure load balancer **az1010302w-lb**.
2. From the **az1010302w-lb** blade, display the **az1010302w-lb - Inbound NAT rules** blade.
3. From the **az1010302w-lb - Inbound NAT rules** blade, add the first inbound NAT rule with the following settings:
   - Name: **az1010302w-vm0-RDP**
   - Frontend IP address: **LoadBalancedFrontEnd**
   - IP Version: **IPv4**
   - Service: **Custom**
   - Protocol: **TCP**
   - Port: **33890**
   - Target virtual machine: **az1010302w-vm0**
   - Network IP configuration: **ipconfig1 (10.101.32.4)** or **ipconfig1 (10.101.32.5)**
   - Port mapping: **Custom**
   - Floating IP (direct server return): **Disabled**
   - Target port: **3389**

     Note: Wait for the operation to complete. This should take less than a minute.
4. From the **az1010302w-lb - Inbound NAT rules** blade, add the second inbound NAT rule with the following settings:
   - Name: **az1010302w-vm1-RDP**
   - Frontend IP address: **LoadBalancedFrontEnd**
   - IP Version: **IPv4**
   - Service: **Custom**
   - Protocol: **TCP**
   - Port: **33891**
   - Target virtual machine: **az1010302w-vm1**

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

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

7.2.0.5 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:

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

   Note: This command initiates a Remote Desktop session to the **az1010301w-vm0** Azure VM by using the **az1010301w-vm0-RDP** NAT rule you created in the previous task.
9. When prompted to sign in, provide the following credentials:
   - Admin Username: **Student**
   - Admin Password: **Pa55w.rd1234**
10. Once you sign in, from the command prompt, run the following command:

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

    Note: Repeat the same tests for the second region.

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

7.3 Exercise 2: Implement Azure Traffic Manager load balancing

Estimated Time: 10 minutes

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

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

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

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

3. From the **az1010301w-lb-pip - Configuration** blade set the **DNS name label** of the public IP address to a unique value.

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

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

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

7.3.0.2 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 **az1010303-tm**.
6. From the **az1010303-tm** blade, display the **az1010303-tm - Configuration** blade and review the configuration settings.

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

7. From the **az1010303-tm** blade, display the **az1010303-tm - Endpoints** blade.
8. From the **az1010303-tm - 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 **az1010303-tm - Endpoints** blade, add the second endpoint with the following settings:
   - Type: **Azure endpoint**
   - Name: **az1010302w-lb-pip**
   - Target resource type: **Public IP address**
   - Target resource: **az1010302w-lb-pip**
   - Weight: **100**
   - Custom Header settings: leave blank

- Add as disabled: leave blank

10. On the **az1010303-tm - 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.

7.3.0.3 Task 3: Verify Azure Traffic Manager load balancing

1. From the **az1010303-tm - Endpoints** blade, switch to the **az1010303-tm** blade to display the **Overview** section.
2. Note the DNS name assigned to the Traffic Manager profile (the string following the **http://** prefix).
3. From the Azure Portal, start a PowerShell session in the Cloud Shell pane.

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

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

   ```
   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:

   ```
   nslookup <TM_DNS_name>
   ```

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

   Result: After you completed this exercise, you have implemented and verified Azure Traffic Manager load balancing # AZ 101 Module 4 - Secure Identities

8 Lab 2: Implement and validate Azure AD Identity Protection

Estimated Time: 90 minutes

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: none

8.0.1 Scenario

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

8.0.2 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

8.0.3 Exercise 0: Prepare the lab environment

Estimated Time: 15 minutes

The main tasks for this exercise are as follows:

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

8.0.3.1 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: 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
    - Resource group: the name of a new resource group **az1010401b-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: **az1010401b-vm1**
    - Admin Username: **Student**
    - Admin Password: **Pa55w.rd1234**
    - Virtual Network Name: **az1010401b-vnet1**

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

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

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.

8.1 Exercise 1: Implement Azure MFA

Estimated Time: 45 minutes

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

8.1.0.1 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: Take a note of the initial domain name. You will need it later in this lab.

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

   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 **Products** blade.
5. From the **Products** blade, navigate to the **Activate** blade and activate **Azure AD Premium P2** free trial.

8.1.0.3 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@**.onmicrosoft.com** where represents the initial domain name you specified in the first task of this exercise.

     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.
3. From the **Users - All users** blade, create a new user with the following settings:
   - Name: **aaduser2**
   - User name: **aaduser2@**.onmicrosoft.com** where represents the initial domain name you specified in the first task of this exercise.

     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.

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

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: This step is necessary in order for the license assignment to take effect.

8.1.0.5 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 ensure that all **verification options**, including **Call to phone**, **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 **Enforced**.

   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**

8.1.0.6 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: 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 at least one of the following options:
   - **Authentication phone**
   - **Office phone**
   - **Mobile app**
5. Select the **Authentication phone** or **Office phone** option and select the **Call me** method of contact.
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.
10. Open an InPrivate Microsoft Edge window again, navigate to the Azure portal and, when prompted, sign in by using the **aaduser1** user account. This will automatically trigger the call to the phone number you provided.
11. Answer the call, press **0#**, and listen to the remainder of the message.

    Note: At this point, your account has been automatically blocked.
12. To unblock the **aaduser1** account, sign in to the Azure portal by using the Microsoft account you used to create the **AdatumLab101-4b** Azure AD tenant, navigate to the **Multi-Factor Authentication - Block/unblock users** blade, and, use the **Unblock** link next to the **aaduser1** entry to unblock this user account.

    Note: To unblock a user, you need to provide **Reason for unblocking** on the **Unblock a user** blade.

8.2 Exercise 2: Implement Azure AD Identity Protection:

Estimated Time: 35 minutes

The main tasks for this exercise are as follows:

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

8.2.0.1 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: 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.

8.2.0.2 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**
     - Conditions:
       * User risk: **Medium and above**
   - Controls:
     - Access: **Allow access**
     - **Require password change**
   - Enforce Policy: **On**

8.2.0.3 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:
       * User risk: **Medium and above**
   - Controls:
     - Access: **Allow access**
     - **Require multi-factor authentication**
   - Enforce Policy: **On**

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

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 **az1010401b-vm1** blade.
2. 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**
3. Within the Remote Desktop session, open an InPrivate Internet Explorer window.

4. In the new browser window, navigate to the ToR Browser Project at **https://www.torproject.org/projects/torbrow** download the ToR Browser, and install it with the default options.
5. 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**
6. When prompted, sign in with the **aaduser2** account you created in the previous exercise.
7. 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.
8. 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.
9. 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.
10. Use the **Verify** option and specify whether you want to verify your identity via text or a call.
11. Complete the verification and ensure that you successfully signed in to the Application Access Panel.
12. Sign out as **aaduser1** and close the ToR Browser window.
13. 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.
14. 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**.
15. 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**.

    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 # AZ 101 Module 4 - Secure Identities

9 Lab: Secure Identities

Estimated Time: 60 minutes

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

> **Note**: When not using Cloud Shell, the lab virtual machine must have Azure AD PowerShell module installed **https://docs.microsoft.com/en-us/powershell/azure/active-directory/install-adv2?view=azureadps-2.0**
>
> **Note**: Exercise 3 and Exercise 4 of this lab require the use of a mobile phone

Lab files:

- **Labfiles\AZ101\Mod04\az-101-04_01_azuredeploy.json**
- **Labfiles\AZ101\Mod04\az-101-04_01_azuredeploy.parameters.json**
- **Labfiles\AZ101\Mod04\az-101-04_01_customRoleDefinition.json**

9.0.1 Scenario

Adatum Corporation wants to leverage Azure Active Directory (AD) capabilities to delegate management of its resources and identities. It also wants to take advantage of more advanced capabilities available in Azure AD Premium P2.

9.0.2 Objectives

After completing this lab, you will be able to:

- Deploy an Azure VM by using an Azure Resource Manager template

- Create Azure AD users and groups
- Delegate management of Azure resources by using custom Role-Based Access Control (RBAC) roles
- Delegate management of Azure AD by using Privileged Identity Management directory roles
- Delegate management of Azure resources by using Privileged Identity Management resource roles

9.1 Exercise 0: Deploy an Azure VM by using an Azure Resource Manager template

Estimated Time: 5 minutes

The main tasks for this exercise are as follows:

1. Deploy an Azure VM running Windows Server 2016 Datacenter by using an Azure Resource Manager template

9.1.0.1 Task 1: Deploy an Azure VM running Windows Server 2016 Datacenter 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 the 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 **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, select the **Build your own template in the editor**.
6. From the **Edit template** blade, load the template file **Labfiles\AZ101\Mod04\az-101-04_01_azuredeploy.json**.

   Note: Review the content of the template and note that it defines deployment of an Azure VM hosting Linux Ubuntu.
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\AZ101\Mod04\az-101-04_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 **az1010401-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**
    - Ubuntu OS Version: **16.04.0-LTS**

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

    Note: Do not wait for the deployment to complete but proceed to the next task. You will use the Azure VM provisioned by this deployment in Exercise 2 Task 3.

9.2 Exercise 1: Create Azure AD users and groups

Estimated Time: 10 minutes

The main tasks for this exercise are as follows:

1. Create an Azure AD user
2. Create an Azure AD security group and add the Azure AD user to the group.

9.2.0.1 Task 1: Create an Azure AD user

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

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

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

   ```
   $domainName = ((Get-AzureAdTenantDetail).VerifiedDomains)[0].Name
   $domainName
   ```

   Note: These commands identify the verified DNS name of the Azure AD tenant associated with the Azure subscription you are using in this lab. Take a note of this value since you will need it later in this lab.

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

   ```
   $passwordProfile = New-Object -TypeName Microsoft.Open.AzureAD.Model.PasswordProfile
   $passwordProfile.Password = 'Pa55w.rd1234'
   $passwordProfile.ForceChangePasswordNextLogin = $false
   $aadUser = New-AzureADUser -AccountEnabled $true -DisplayName 'aaduser 101041' -PasswordProfile $p
   ```

   Note: These commands create an Azure AD user which you will use later in this lab.

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

   ```
   (Get-AzureADUser -Filter "MailNickName eq 'aaduser101041'").UserPrincipalName
   ```

   Note: This command identifies the user principal name of the newly created Azure AD user. Take a note of this name. You will use it later in this lab.

9.2.0.2 Task 2: Create an Azure AD security group and add the Azure AD user to the group.

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

   ```
   $aadGroup = New-AzureADGroup -Description "VM Operators" -DisplayName "VM Operators" -MailEnabled
   ```

   Note: This command creates an Azure AD security group

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

   ```
   Add-AzureADGroupMember -ObjectId $aadGroup.ObjectId -RefObjectId $aadUser.ObjectId
   ```

   Note: This command adds the user you created in the previous task to the newly created Azure AD group

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

   ```
   Get-AzureADGroupMember -ObjectId $aadGroup.ObjectId
   ```

   Note: This command returns the list of members of the newly created Azure AD group

4. Close the Cloud Shell pane.

9.3 Exercise 2: Delegate management of Azure resources by using custom Role-Based Access Control roles

Estimated Time: 15 minutes

The main tasks for this exercise are as follows:

1. Identify actions to delegate via RBAC
2. Create a custom RBAC role in the Azure AD tenant
3. Assign the custom RBAC role and test the role assignment

9.3.0.1 Task 1: Identify actions to delegate via RBAC

1. In the Azure portal, navigate to the blade of the resource group **az1010401-RG**.
2. From the **az1010401-RG** blade, display its **Access Control (IAM)** blade.
3. From the **az1010401-RG - Access Control (IAM)** blade, display the **Roles** blade.
4. From the **Roles** blade, display the **Owner** blade.
5. From the **Owner** blade, display the **Permissions (preview)** blade.
6. From the **Permissions (preview)** blade, display the **Microsoft Compute** blade.
7. From the **Microsoft Compute** blade, display the **Virtual Machines** blade.
8. On the **Virtual Machines** blade, review the list of management actions that can be delegated through RBAC. Note that they include the **Deallocate Virtual Machine** and **Start Virtual Machine** actions.

9.3.0.2 Task 2: Create a custom RBAC role in the Azure AD tenant

1. On the lab virtual machine, open the file **Labfiles\AZ101\Mod04\az-101-04_01_custom.role.definition.json** and review its content:

```
{
   "Name": "Virtual Machine Operator (Custom)",
   "Id": null,
   "IsCustom": true,
   "Description": "Allows to start and stop (deallocate) Azure VMs",
   "Actions": [
       "Microsoft.Compute/*/read",
       "Microsoft.Compute/virtualMachines/deallocate/action",
       "Microsoft.Compute/virtualMachines/start/action"
   ],
   "NotActions": [
   ],
   "AssignableScopes": [
       "/subscriptions/SUBSCRIPTION_ID"
   ]
}
```

2. From the Azure Portal, start a PowerShell session in the Cloud Shell.
3. From the Cloud Shell pane, upload the file **Labfiles\AZ101\Mod04\az-101-04_01_custom.role.definition.json** into the home directory.
4. In the Cloud Shell pane, run the following commands:

```
$subscription_id = (Get-AzureRmSubscription).Id
(Get-Content -Path $HOME/az-101-04_01_custom.role.definition.json) -Replace 'SUBSCRIPTION_ID', "$s
```

   Note: These commands replace the **SUBSCRIPTION_ID** placeholder with the actual Id of the Azure subscription you are using in this lab.

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

```
New-AzureRmRoleDefinition -InputFile $HOME/az-101-04_01_custom.role.definition.json
```

   Note: This command creates the custom role definition object in the Azure AD tenant by using the JSON file you uploaded to the Cloud Shell

6. From the Cloud Shell pane, run the following command:

```
Get-AzureRmRoleDefinition -Name 'Virtual Machine Operator (Custom)'
```

Note: This command verifies that the role has been successfully defined by retrieving the role definition object and displaying its properties

7. Close the Cloud Shell pane.

9.3.0.3 Task 3: Assign the custom RBAC role and test the role assignment

Note: Before you test the custom RBAC role assignment this task, ensure that the template deployment you started in Exercise 0 has completed.

1. In the Azure portal, navigate to the blade of the resource group **az1010401-RG**.
2. From the **az1010401-RG** blade, display its **Access Control (IAM)** blade.
3. From the **az1010401-RG - Access Control (IAM)** blade, display the **Add permissions** pane.
4. From the **Add permissions** pane, assign the custom RBAC role to the Azure AD security group **VM Operators** you created in the previous exercise by using the following settings:
   - Role: **Virtual Machine Operator (Custom)**
   - Assign access to: **Azure AD user, group, or application**
   - Select: **VM Operators**
5. Launch another Microsoft Edge window in the InPrivate mode, browse to the Azure portal at **http://portal.azure.com** and sign in by using the **aaduser101041** user account:
   - Username: the user principal name of the Azure AD user **aaduser101041** you identified in the first exercise of this lab.
   - Password: **Pa55w.rd1234**
6. In the Azure portal, navigate to the **Resource groups** blade. Note that you are not able to see any resource groups.
7. In the Azure portal, navigate to the **All resources** blade. Note that you are able to see only the **az1010401-vm** and its managed disk.
8. In the Azure portal, navigate to the **az1010401-vm** blade.
9. Try restarting the virtual machine. Review the error message in the notification area and note that this action failed because the current user is not authorized to carry it out.
10. Stop the virtual machine and verify that the action completed successfully.
11. Start the virtual machine and verify that the action completed successfully.
12. Sign out as the Azure AD user **aaduser101041** and close the Microosft Edge InPrivate mode window.

Result: After you completed this exercise, you have defined, assigned, and tested a custom RBAC role

9.4 Exercise 3: Delegate management of Azure AD by using Privileged Identity Management directory roles

Estimated Time: 15 minutes

The main tasks for this exercise are as follows:

1. Activate Azure AD Premium P2 trial
2. Assign Azure AD Premium P2 licenses
3. Activate Privileged Identity Management
4. Sign up PIM for Azure AD roles
5. Delegate management of Azure AD roles
6. Validate delegation of management of Azure AD roles

9.4.0.1 Task 1: Activate Azure AD Premium P2 trial.

Note: In order to use Azure AD Privileged Identity Management, you need to have Azure AD Premium 2 licenses

1. In the Azure portal, while signed in by using the Microsoft account that has the Owner role in the Azure subscription and is a Global Administrator of the Azure AD tenant associated with that subscription, navigate to the Azure AD tenant blade.
2. From the Azure AD tenant blade, navigate to the **Licenses** blade.
3. From the the **Licenses** blade, navigate to the **Licenses - All products** blade.
4. From the **Licenses - All products** blade, navigate to the **Activate** blade and activate the **Azure AD Premium P2** trial.

9.4.0.2 Task 2: Assign Azure AD Premium P2 licenses.

1. Navigate to the **Users - All users** blade of the Azure AD tenant associated with your Azure subscription.
2. From the **Users - All users** blade, display the **aaduser 101041 - Profile** blade.
3. From the **aaduser 101041 - Profile** blade, edit **Settings** of the **aaduser 101041** user account by selecting the **Usage location** matching the location of the Azure AD tenant.
4. Navigate back to the **Licenses - Overview** blade of the Azure AD tenant associated with your Azure subscription.
5. From the **Licenses - Overview** blade, navigate to the **Products** blade.
6. From the **Products** blade, navigate to the **Azure Active Directory Premium P2 - Licensed users** blade.
7. From the **Azure Active Directory Premium P2 - Licensed users**, navigate to the **Assign license** blade.
8. From the **Assign license** blade, assign an Azure AD Premium P2 license to the **aaduser 101041** user account.

9.4.0.3 Task 3: Activate Privileged Identity Management

1. In the Azure portal, navigate to the **Privileged Identity Management** blade.
2. From the **Privileged Identity Management** blade, initiate the **Consent to PIM** action.
3. From the **Privileged Identity Management - Consent to PIM** blade, proceed to the **Verify my identity** task.
4. On the **Aditional security verification** page, specify the following settings:
   - **Step 1: How should we contact you?**
     - Authentication phone: select your country or region and specify a mobile phone number you intend to use in this lab
     - Method: **Send me a code by text message**
   - **Step 2: We've send a text message to your phone**
     - Use the code in the text message you received
5. Back on the **Privileged Identity Management** blade, grant consent and, when prompted, confirm your decision.

9.4.0.4 Task 4: Sign up PIM for Azure AD roles

1. In the Azure portal, from the **Privileged Identity Management - Quick start** blade, navigate to the **Azure AD roles - Quick start** blade.
2. From the **Azure AD roles - Quick start** blade, navigate to the **Azure AD roles - Sign up PIM for Azure AD Roles** blade.

3. From the **Azure AD roles - Sign up PIM for Azure AD Roles** blade, sign up for Azur eAD PIM for Azure AD roles.

9.4.0.5 Task 5: Delegate management of Azure AD roles

1. In the Azure portal, return to the **Privileged Identity Management - Quick start** blade and then navigate to the **Azure AD roles - Quick start** blade.

   Note: This step allows you to access the Azure AD roles management features in the portal.

2. From the **Azure AD roles - Quick start** blade, display the **Azure AD roles - Roles** blade.
3. From the **Azure AD roles - Roles** blade, display the **Add managed members** blade.
4. From the **Add managed members** blade, specify the following settings in order to designate the **aaduser101041** user account as an eligible member of the **User Administrator** role:
   - Select a role: **User Administrator**
   - Select members: **aaduser 101041**
5. From the **Azure AD roles - Roles** blade, display the **Azure AD roles - Members** blade and note that the **aaduser101041** user account is listed as an eligible member of the **User Administrator** role.

9.4.0.6 Task 6: Validate delegation of management of Azure AD roles

1. Launch another Microsoft Edge window in the InPrivate mode, browse to the Azure portal at **http://portal.azure.com** and sign in by using the **aaduser101041** user account:
   - Username: the user principal name of the Azure AD user **aaduser101041** you identified in the first exercise of this lab.
   - Password: **Pa55w.rd1234**
2. In the Azure portal, navigate to the **Privileged Identity Management - Quick start** blade.
3. From the **Privileged Identity Management - Quick start** blade, display the **My roles - Azure AD roles** blade.
4. On the **My roles - Azure AD roles** blade, on the **Eligible roles** tab, note that you are eligible to activate the assignment to the **User Administrator** role.
5. From the **My roles - Azure AD roles** blade, initiate the activation. This will display the **User Administrator** blade.
6. From the **User Administrator** blade, proceed to the **Verify my identity** task.
7. On the **Aditional security verification** page, specify the following settings:
   - **Step 1: How should we contact you?**
     - Authentication phone: select your country or region and specify a mobile phone number you intend to use in this lab
     - Method: **Send me a code by text message**
   - **Step 2: We've send a text message to your phone**
     - Use the code in the text message you received
8. Back on the **User Administrator** blade, initiate activation. This will display the **Activation** blade.
9. From the the **Activation** blade, perform activation using the following settings:
   - Activation duration (hours): **1**
   - Activation reason: **testing PIM-based Azure AD role delegation**
10. In the Azure portal, navigate back to the **My roles - Azure AD roles** blade.
11. On the **My roles - Azure AD roles** blade, on the **Active roles** tab, note that the role assignment has been activated.
12. In the Azure portal, navigate to the **Users - All users** blade of the Azure AD tenant associated with your Azure subscription.

13. From the **Users - All users** blade, create a new Azure AD user account with the following settings:
    - Name: **aaduser101042**
    - User name: **aaduser101042@** where represents the primary DNS domain name you identified in the first exercise of this lab.
    - Profile: **Not configured**
    - Properties: **Default**
    - Groups: **0 groups selected**
    - Directory role: **User**
    - Password: accept the default vaule.
14. Verify that the user was created successfully.
15. Sign out as the Azure AD user **aaduser101041** and close the Microosft Edge InPrivate mode window.

    Result: After you completed this exercise, you have activated Azure AD Premium P2 trial, assigned Azure AD Premium P2 licenses, activated Privileged Identity Management, signed up PIM for Azure AD roles, delegated management of Azure AD roles, and validated delegation of management of Azure AD roles.

9.5 Exercise 4: Delegate management of Azure resources by using Privileged Identity Management resource roles

Estimated Time: 15 minutes

The main tasks for this exercise are as follows:

1. Onboard the Azure subscription for PIM resource management
2. Delegate management of Azure AD resources
3. Validate delegation of management of Azure AD resources

9.5.0.1 Task 1: Onboard the Azure subscription for PIM resource management

1. In the Azure portal, while signed in by using the Microsoft account that has the Owner role in the Azure subscription and is a Global Administrator of the Azure AD tenant associated with that subscription, navigate to the **Privileged Identity Management - Quick start** blade.
2. From the **Privileged Identity Management - Quick start** blade, navigate to the **Privileged Identity Management - Azure resources** blade.
3. From the **Privileged Identity Management - Azure resources** blade, display the **Azure resources - Discovery** blade.
4. On the **Privileged Identity Management - Azure resources** blade, select the entry representing your Azure subscription and initiate the resource management action. When prompted, confirm your decision to onboard selected resource for management.

9.5.0.2 Task 2: Delegate management of Azure AD resources

1. Navigate back to the **Privileged Identity Management - Azure resources** blade and note that that your Azure subscription is listed as the discovered resource.
2. From the **Privileged Identity Management - Azure resources** blade, select the entry representing your Azure subscription. This will automatically display the blade representing Privileged Identity Management resource settings for this subscription.
3. From the blade representing Privileged Identity Management resource settings for your Azure subscription, display the list of resource roles.
4. From the blade listing resource roles, display the **Contributor** blade.
5. From the **Contributor** blade, display the **New assignment** blade.

6. From the **New assignment** blade, specify the following settings in order to designate the **aaduser101041** user account as an eligible member of the **Contributor** role:
   - Select a role: **Contributor**
   - Select a member or group: **aaduser 101041**
   - Set membership settings:
     - Assignment type: **Eligible**
     - Assignment starts: select current date and time
     - Assignment ends: select the date and time 24 hours ahead of the current date and time
7. From the blade representing Privileged Identity Management resource settings for your Azure subscription, display the **Members** blade and, on the **Eligible roles** tab, note that the **aaduser101041** user account is listed as member of the **Contributor** role.

9.5.0.3 Task 3: Validate delegation of management of Azure resources

1. Launch another Microsoft Edge window in the InPrivate mode, browse to the Azure portal at **http://portal.azure.com** and sign in by using the **aaduser101041** user account:
   - Username: the user principal name of the Azure user **aaduser101041** you identified in the first exercise of this lab
   - Password: **Pa55w.rd1234**
2. In the Azure portal, navigate to the **Privileged Identity Management - Azure resources** blade.
3. From the **Privileged Identity Management - Azure resources** blade, select the entry representing your Azure subscription. This will automatically display the blade representing Privileged Identity Management resource settings for this subscription.
4. On the blade representing Privileged Identity Management resource settings for this subscription, note the message indicating that you have eligible roles that can be activated.
5. From the blade representing Privileged Identity Management resource settings for this subscription, navigate to the **My roles** blade.
6. On the **My roles** blade, on the **Eligible resources** tab, note that you are eligible to activate the assignment to the **Contributor** role.
7. From the **My roles** blade, initiate the activation. This will display the **Activate** blade.
8. From the the **Activate** blade, perform activation using the following settings:
   - Scope: the name of your Azure subscription
   - Start time: current date and time
   - Duration (hours): **8**
   - Reason: **testing PIM-based Azure resource role delegation**
9. Back on the **My roles** blade, refresh the page and display the **Active roles** tab. Note that the role assignment has been activated.
10. Sign out as the Azure user **aaduser101041** and then sign back in.
11. In the Azure portal, navigate to the **All resources** blade and note that you are able to see all Azure resources in your Azure subscription.
12. In the Azure portal, navigate to the **Subscriptions** blade and select the entry representing your Azure subscription.
13. From the blade displaying properties of your Azure subscription, navigate to its **Access control (IAM)** blade.
14. On the **Access control (IAM)** blade, note that the **aaduser101041** user account is listed as a member of the **Contributor** role.
15. Sign out as the Azure user **aaduser101041** and close the Microosft Edge InPrivate mode window.

Result: After you completed this exercise, you have onboarded the Azure subscription for PIM resource management, delegated management of Azure AD resources, and validate delegation of management of Azure AD resources.