Contents

#AZ-304: Lab File Names in GitHub

Source repo: https://github.com/MicrosoftLearning/AZ-304-Microsoft-Azure-Architect-Design/

Module 1 Lab

Lab title: Implementing Containers on Azure

- Current file name: **Module_1_Lab.md**
- Previous file name: LAB_01_Deploying Azure IaaS Solutions.md

Module 3 Lab

Lab title: Migrating Hyper-V VMs to Azure by using Azure Migrate

- Current file name: **Module_3_Lab.md**
- Previous file name: LAB_02_Data Storage.md

Module 4 Lab

Lab title: Managing Azure AD Authentication and Authorization

- Current file name: **Module_4_Lab.md**
- Previous file name: LAB_10_Security.md

Module 6 Lab

Lab title: Implementing Azure SQL Database-Based Applications

- Current file name: **Module_6_Lab.md**
- Previous file name: LAB_03_Data Storage.md

Module 13 Lab

Lab title: Implement Azure Logic Apps Integration with Azure Event Grid

- Current file name: **Module_13_Lab_a.md**
- Previous file name: LAB_04_Manage Compute_PaaS.md

1 AZ-304: Microsoft Azure Architect Design

- **Link to LABS in HTML format**
- **Download Latest Student Handbook and AllFiles Content**
- **Are you a MCT?** - Have a look at our GitHub User Guide for MCTs
- **Need to manually build the lab instructions?** - Instructions are available in the MicrosoftLearning/Docker-Build repository

1.1 What are we doing?

- To support this course, we will need to make frequent updates to the course content to keep it current with the Azure services used in the course. We are publishing the lab instructions and lab files on GitHub to allow for open contributions between the course authors and MCTs to keep the content current with changes in the Azure platform.
- We hope that this brings a sense of collaboration to the labs like we've never had before - when Azure changes and you find it first during a live delivery, go ahead and make an enhancement right in the lab source. Help your fellow MCTs.

1.2 How should I use these files relative to the released MOC files?

- The instructor handbook and PowerPoints are still going to be your primary source for teaching the course content.
- These files on GitHub are designed to be used in conjunction with the student handbook, but are in GitHub as a central repository so MCTs and course authors can have a shared source for the latest lab files.
- It will be recommended that for every delivery, trainers check GitHub for any changes that may have been made to support the latest Azure services, and get the latest files for their delivery.

1.3 What about changes to the student handbook?

- We will review the student handbook on a quarterly basis and update through the normal MOC release channels as needed.

1.4 How do I contribute?

- Any MCT can submit a pull request to the code or content in the GitHub repro, Microsoft and the course author will triage and include content and lab code changes as needed.
- You can submit bugs, changes, improvement and ideas. Find a new Azure feature before we have? Submit a new demo!

1.5 Notes

1.5.1 Classroom Materials

1.6 It is strongly recommended that MCTs and Partners access these materials and in turn, provide them separately to students. Pointing students directly to GitHub to access Lab steps as part of an ongoing class will require them to access yet another UI as part of the course, contributing to a confusing experience for the student. An explanation to the student regarding why they are receiving separate Lab instructions can highlight the nature of an always-changing cloud-based interface and platform. Microsoft Learning support for accessing files on GitHub and support for navigation of the GitHub site is limited to MCTs teaching this course only.

1.7 title: Online Hosted Instructions permalink: index.html layout: home

2 Content Directory

Files necessary to complete the labs can be DOWNLOADED HERE

Hyperlinks to each of the lab exercises are listed below.

2.1 Labs

{% assign labs = site.pages | where_exp:"page", "page.url contains '/Instructions/Labs'" %} | Module | Lab | | --- | --- | {% for activity in labs %}| {{ activity.lab.module }} | [{{ activity.lab.title }}{% if activity.lab.type %} - {{ activity.lab.type }}{% endif %}](/home/ll/Azure_clone/Azure_new/AZ-304-Microsoft-Azure-Architect-Design/{{ site.github.url }}{{ activity.url }}) | {% endfor %}

#AZ-304: Lab file names in GitHub

Source repo: https://github.com/MicrosoftLearning/AZ-304-Microsoft-Azure-Architect-Design/

Module 3 Lab

Lab title: Migrating Hyper-V VMs to Azure by using Azure Migrate

- Current file name: **Module_3_Lab.md**
- Previous file name: LAB_02_Data Storage.md

Module 4 Lab

Lab title: Managing Azure AD Authentication and Authorization

- Current file name: **Module_4_Lab.md**
- Previous file name: LAB_10_Security.md

Module 6 Lab

Lab title: Implementing Azure SQL Database-Based Applications

- Current file name: **Module_6_Lab.md**
- Previous file name: LAB_03_Data Storage.md

Module 13 Lab

Lab title: Implement Azure Logic Apps Integration with Azure Event Grid

- Current file name: **Module_13_Lab_a.md**
- Previous file name: LAB_04_Manage Compute_PaaS.md

2.2 lab: title: '13: Implement Azure Logic Apps integration with Azure Event Grid' module: 'Module 13: Design an Application Architecture'

3 Lab: Implement Azure Logic Apps integration with Azure Event Grid

4 Student lab manual

4.1 Lab scenario

Adatum Corporation has an extensive set of on-premises network monitoring framework that rely on the combination of agent-based and agentless solutions to provide visibility into any changes to its environment. The agentless solutions tend to be relatively inefficient since they rely on polling to determine state changes.

As Adatum is preparing to migrate some of its workloads to Azure, its Enterprise Architecture team wants to address these inefficiencies and evaluate the use of event driven architecture available in the cloud. The notion of using events in a solution or application is not new to the team. In fact, they have been promoting the idea of event-driven programming among its developers. One of the core tenets of an event-driven architecture is to reverse the dependencies that existing services may have with each other. Azure provides this functionality by relying on Event Grid, which is a fully managed service that supports the routing of events by utilizing a publisher-subscriber model. At its core, Event Grid is an event routing service that manages the routing and delivery of events from numerous sources and subscribers.

An event is created by a publisher such as a Blob Storage account, an Azure resource group, or even an Azure subscription. As events occur, they are published to an endpoint called a topic that the Event Grid service manages to digest all incoming messages. Event publishers are not limited to services on Azure. It is possible to use events that originate from custom applications or systems that can run from anywhere. This includes applications that are hosted on-premises, in a datacenter, or even on other clouds, if they can post an HTTP request to the Event Grid service.

Event handlers include several Azure services, including serverless technologies such as Functions, Logic Apps, or Azure Automation. Handlers are registered with Event Grid by creating an event subscription. If the event handler endpoint is publicly accessible and encrypted by Transport Layer Security, then messages can be pushed to it from Event Grid.

Unlike many other Azure services, there is no Event Grid namespace that needs to be provisioned or managed. Topics for native Azure resources are built in and completely transparent to users while custom topics are provisioned ad hoc and exist in a resource group. Event subscriptions are simply associated with a topic. This model simplifies management of topics as subscriptions and makes Event Grid highly multi-tenant, allowing for massive scale out.

Azure Event Grid is agnostic to any language or platform. While it integrates natively with Azure services, it can just as easily be leveraged by anything that supports the HTTP protocol, which makes it a very clever and innovative service.

To explore this functionality, the Adatum Architecture team wants to test integration of Azure Logic Apps with Event Grid to:

- detect when the state of a designated Azure VM is changed
- automatically generate an email notification in response to the event

4.2 Objectives

After completing this lab, you will be able to:

- Integrate Azure Logic Apps with Event Grid

- Trigger execution of Logic Apps in response to an event representing a change to a resource within a resource group

4.3 Lab Environment

Windows Server admin credentials

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

Estimated Time: 60 minutes

4.4 Lab Files

- \\AZ304\AllFiles\Labs\04\azuredeploy30304suba.json
- \\AZ304\AllFiles\Labs\04\azuredeploy30304rga.json
- \\AZ304\AllFiles\Labs\04\azuredeploy30304rga.parameters.json

4.5 Instructions

4.5.1 Exercise 0: Prepare the lab environment

The main tasks for this exercise are as follows:

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

4.5.1.1 Task 1: Deploy an Azure VM by using an Azure Resource Manager template

1. From your lab computer, start a web browser, navigate to the Azure portal, and sign in by providing credentials of a user account with the Owner role in the subscription you will be using in this lab.
2. In the Azure portal, open **Cloud Shell** pane by selecting on the toolbar icon directly to the right of the search textbox.
3. If prompted to select either **Bash** or **PowerShell**, select **PowerShell**.

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

4. From the Cloud Shell pane, run the following to register the **Microsoft.EventGrid** provider in your subscription:

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

5. In the toolbar of the Cloud Shell pane, select the **Upload/Download files** icon, in the drop-down menu select **Upload**, and upload the file **\\AZ304\AllFiles\Labs\04\azuredeploy30304suba.json** into the Cloud Shell home directory.
6. From the Cloud Shell pane, run the following to create a resource groups (replace the `<Azure region>` placeholder with the name of the Azure region that is available for deployment of Azure VMs in your subscription and which is closest to the location of your lab computer):

   ```
   $location = '<Azure region>'
   New-AzSubscriptionDeployment `
     -Location $location `
     -Name az30304subaDeployment `
     -TemplateFile $HOME/azuredeploy30304suba.json `
     -rgLocation $location `
     -rgName 'az30304a-labRG'
   ```

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

7. From the Cloud Shell pane, upload the Azure Resource Manager template **\\AZ304\AllFiles\Labs\04\azuredeploy**
8. From the Cloud Shell pane, upload the Azure Resource Manager parameter file **\\AZ304\AllFilesLabs\04\azuredep**

9. From the Cloud Shell pane, run the following to deploy a Azure VM running Windows Server 2019 that you will be using in this lab:

```
New-AzResourceGroupDeployment `
  -Name az30304rgaDeployment `
  -ResourceGroupName 'az30304a-labRG' `
  -TemplateFile $HOME/azuredeploy30304rga.json `
  -TemplateParameterFile $HOME/azuredeploy30304rga.parameters.json `
  -AsJob
```

   Note: Do not wait for the deployment to complete but instead proceed to the next exercise. The deployment should take less than 5 minutes.

10. In the Azure portal, close the **Cloud Shell** pane.

4.5.2 Exercise 1: Configure authentication and authorization for an Azure logic app

1. Create an Azure Active Directory service principal
2. Assign the Reader role to the Azure AD service principal

4.5.2.1 Task 1: Create an Azure Active Directory service principal

1. In the Azure portal, start a **PowerShell** session within the **Cloud Shell**.
2. From the Cloud Shell pane, run the following 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.@z304'
$securePassword = ConvertTo-SecureString -Force -AsPlainText -String $password
$az30304aadapp = New-AzADApplication -DisplayName 'az30304aadsp' -HomePage 'http://az30304aadsp' -
```

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

```
New-AzADServicePrincipal -ApplicationId $az30304aadapp.ApplicationId.Guid -SkipAssignment
```

4. In the output of the **New-AzADServicePrincipal** command, note the value of the **ApplicationId** property. You will need it later in this exercise.
5. From the Cloud Shell pane, run the following 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 later in this exercise):

```
Get-AzSubscription
```

6. Close the Cloud Shell pane.

4.5.2.2 Task 2: Authorizing access to the Azure AD service principal

1. In the Azure portal, search for and select **Resource groups** and, on the **Resource groups** blade, select **az30304a-labRG**.
2. On the **az30304a-labRG** blade, select **Access control (IAM)**.
3. On the **az30304a-labRG | Access control (IAM)** blade, select **+ Add**, and select **Add role assignment**.
4. On the **Add role assignment** blade, specify the following settings and select **Save**:

Setting	Value
Role	**Reader**
Assign access to	**User, group, or service principal**
Select	**az30304aadsp**

4.5.3 Exercise 2: Implement an Azure logic app

The main tasks for this exercise are as follows:

1. Create an Azure logic app
2. Add a trigger to the Azure logic app
3. Add a condition to the Azure logic app
4. Add an action to the Azure logic app

4.5.3.1 Task 1: Create an Azure logic app

1. In the Azure portal, search for and select **Logic App** and, on the **Logic Apps** blade, select **+ Add** the select **Consumption**.
2. On the **Basics** tab of the **Logic App** blade, specify the following settings (leave others with their default values):

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	the name of a new resource group **az30304b-labRG**
Logic App name	**az30304b-logicapp1**
Select the location	**Region**
Location	the name of an Azure region that you chose in the previous exercise
Log Analytics	**Off**

3. Select **Review + create** and then select **Create**.

   Note: Wait for the logic app to be created. Provisioning should take about 2 minutes.

4.5.3.2 Task 2: Add a trigger to the Azure logic app

1. In the Azure portal, search for and select **Logic App** and, on the **Logic Apps** blade, select **az30304b-logicapp1**.
2. On the **Logic App Designer** blade, select **Blank Logic App**. This will display a blank designer workspace.
3. Use the **Search connectors and triggers** text box, to search for **Event Grid**, in the list of results, in the **Triggers** column, select **When a resource event occurs** Azure Event Grid trigger to add it to the designer workspace.
4. In the **Azure Event Grid** tile, select the **Connect with Service Principal** link, specify the following settings, and select **Create**:

Setting	Value
Connection Name	**az30304egconnection**
Client ID	the value of the **ApplicationId** property you identified earlier in this exercise
Client Secret	**Pa55w.rd1234.@z304**
Tenant	the value of the **TenantId** property you identified earlier in this exercise

5. In the **When a resource event occurs** tile, specify the following settings:

Setting	Value
Subscription	Choose your subscription from the drop-down list
Resource Type	**Microsoft.Resources.resourceGroups**
Resource Name	**az30304a-labRG**
Event Type Item - 1	**Microsoft.Resources.ResourceWriteSuccess**
Event Type Item - 2	**Microsoft.Resources.ResourceDeleteSuccess**

6. n the **When a resource event occurs** tile, select **Add new parameter** and select **Subscription Name**

7. In the **Subscription Name** text box, type **event-subscription-az30304b** and select **Save**.

4.5.3.3 Task 3: Add a condition to the Azure logic app

1. In the the Azure portal, on the Logic App Designer blade of the newly provisioned Azure logic app, select **+ New step**.
2. In the choose an operation tile, use the **Search connectors and actions** text box, to search for **Condition**, in the list of results, in the **Actions** column, select **Condition** to add it to the designer workspace.
3. Select the ellipsis symbol in the upper right corner of the **Condition** tile, in the pop-up menu, select **Rename**, and replace **Condition** with the text **If a virtual machine in the resource group has changed**.
4. Select the **Choose a value** text box on the left hand side of the condition, in the pop up window, in the **Expression** tab, enter this expression and select **OK**:

   ```
   triggerBody()?['data']['operationName']
   ```

5. Ensure that **is equal to** appears in the middle element of the condition and, in the **Choose a value** text box on the right hand side, type the value representing the opearation you intend to monitor:

   ```
   Microsoft.Compute/virtualMachines/write
   ```

6. On the **Logic Apps Designer** blade, select **Save**.

4.5.3.4 Task 4: 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, in the **True** tile, select **Add an action**.
2. In the **Choose an action** pane, in the **Search connectors and actions** text box, type **Outlook**.
3. In the list of results, select **Outlook.com**.
4. In the list of actions for **Outlook.com**, select **Send an email (V2)**.
5. In the **Outlook.com** pane, select **Sign in**.
6. When prompted, authenticate by using the Microsoft Account you are using in this lab.
7. When prompted for the consent to grant Azure Logic App permissions to access Outlook resources, select **Yes**.
8. In the **Outlook.com** pane, select the ellipsis symbol in the upper right corner of the **Send an email (V2)** tile, in the pop-up menu, select **Rename**, and replace **Send an email (v2)** with the text **Send an email**.
9. In the **Send an email** pane, specify the following settings and select **Save**:

Setting	Value
To	the primary e-mail address of your Microsoft Account
Subject	type **Resource updated:** and, in the **Dynamic Content** column to the right of the **Send an email** pane, sele
Body	type **Resource group:**, in the search text box under the **Dynamic Content** column to the right of the **Send a**

   You may need to confirm the account by going into the mailbox and entering your phone number.

10. On the **Logic Apps Designer** blade, select **Save**.

4.5.4 Exercise 3: Implement an event subscription

The main tasks for this exercise are as follows:

1. Configure event subscription
2. Review the functionality of the Azure logic app
3. Remove Azure resources deployed in the lab

4.5.4.1 Task 1: Configure event subscription

1. In the Azure portal, navigate to the **az30304b-logicapp1** blade, in the **Summary** section, select **Trigger history**.
2. On the **When_a_resource_event_occurs** blade, copy the value of the **Callback url [POST]** text box.
3. In the Azure portal, navigate to the **az30304a-LabRG** resource group and, in the vertical menu, select **Events**.
4. On the **az30304a-LabRG | Events** blade, select **+ Event subscription**.
5. On the **Create Event Subscription** blade, specify the following settings and select **Create**:

Setting	Value
Name	**event-subscription-az30304a-LabRG**
Event Schema	**Event Grid Schema**
System Topic name	**az30304b-eventgridtopic**
Filter to Event Types	**Resource Write Success**, **Resource Delete Success**, **Resource Action Success**
Endpoint Type	**Web Hook**
Endpoint	the URL string you copied at the beginning of this task

6. Select **Create**.

4.5.4.2 Task 2: Review the functionality of the Azure logic app

1. In the Azure portal, navigate to the **az30304a-labRG** resource group and, in the list of resources, select the entry representing the **az30304a-vm0** Azure VM.
2. On the **az30304a-vm0** blade, in the **Settings** section, select **Size**.
3. On the **az30304a-vm0 | Size** blade, select a size different from the one currently set and select **Resize** and verify that the resize operation completed successfully.
4. Navigate back to the **az30304b-logicapp1** blade, select **Refresh**, and note that the **Runs history** includes entries corresponding to changes of the state of the Azure VM.
5. In the **Runs history** listing, select an entry with the longest duration, representing the successful resizing on the Azure VM.
6. On the **Logic app run** blade, review the diagram representing the workflow of the logic app run.
7. On the **Logic app run** blade, select the **When a resource event occurs** rectangle to expand it and, in the **OUTPUTS** section, select **Show raw outputs**.
8. On the **Outputs** blade, review the details of the event and note that includes such details as the identity of your user account and the IP address from which the request to resize the Azure VM originated.
9. Navigate to the inbox of the email account you specified in the previous exercise and verify that includes an email generated by the logic app.

4.5.4.3 Task 3: Remove Azure resources deployed in the lab

1. From the Cloud Shell pane, run the following to list the resource group you created in this exercise:

   ```
   Get-AzResourceGroup -Name 'az30304*'
   ```

   Note: Verify that the output contains only the resource group you created in this lab. This group will be deleted in this task.

2. From the Cloud Shell pane, run the following to delete the resource group you created in this lab

   ```
   Get-AzResourceGroup -Name 'az30304*' | Remove-AzResourceGroup -Force -AsJob
   ```

3. Close the Cloud Shell pane.

4.6 lab: title: '3: Migrating Hyper-V VMs to Azure by using Azure Migrate' module: 'Module 3: Design for Migration'

5 Lab: Migrating Hyper-V VMs to Azure by using Azure Migrate

6 Student lab manual

6.1 Lab scenario

Despite its ambitions to modernize its workloads as part of migration to Azure, the Adatum Enterprise Architecture team realizes that, due to aggressive timelines, in many cases, it will be necessary to follow the lift-and-shift approach. To simplify this task, the Adatum Enterprise Architecture team started exploring the capabilities of Azure Migrate. Azure Migrate serves as a centralized hub to assess and migrate to Azure on-premises servers, infrastructure, applications, and data.

Azure Migrate provides the following features:

- Unified migration platform: A single portal to start, run, and track your migration to Azure.
- Range of tools: A range of tools for assessment and migration. Tools include Azure Migrate: Server Assessment and Azure Migrate: Server Migration. Azure Migrate integrates with other Azure services and with other tools and independent software vendor (ISV) offerings.
- Assessment and migration: In the Azure Migrate hub, you can assess and migrate:
- Servers: Assess on-premises servers and migrate them to Azure virtual machines.
- Databases: Assess on-premises databases and migrate them to Azure SQL Database or to SQL Managed Instance.
- Web applications: Assess on-premises web applications and migrate them to Azure App Service by using the Azure App Service Migration Assistant.
- Virtual desktops: Assess your on-premises virtual desktop infrastructure (VDI) and migrate it to Windows Virtual Desktop in Azure.
- Data: Migrate large amounts of data to Azure quickly and cost-effectively using Azure Data Box products.

While databases, web apps, and virtual desktops are in scope of the next stage of the migration initiative, Adatum Enterprise Architecture team wants to start by evaluating the use of Azure Migrate for migrating their on-premises Hyper-V virtual machines to Azure VM.

6.2 Objectives

After completing this lab, you will be able to:

- Prepare Hyper-V for assessment and migration by using Azure Migrate
- Assess Hyper-V for migration by using Azure Migrate
- Migrate Hyper-V VMs by using Azure Migrate

6.3 Lab Environment

Windows Server admin credentials

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

Estimated Time: 120 minutes

6.4 Lab Files

- \\AZ304\AllFiles\Labs\08\azuredeploy30308suba.json

6.4.1 Exercise 0: Prepare the lab environment

The main tasks for this exercise are as follows:

1. Deploy an Azure VM by using an Azure Resource Manager QuickStart template
2. Configure nested virtualization in the Azure VM

6.4.1.1 Task 1: Deploy an Azure VM by using an Azure Resource Manager QuickStart template

1. From your lab computer, start a web browser, navigate to the Azure portal - `portal.azure.com`, and sign in by providing credentials of a user account with the Owner role in the subscription you will be using in this lab.
2. In the Azure portal, open **Cloud Shell** pane by selecting on the toolbar icon directly to the right of the search textbox.
3. If prompted to select either **Bash** or **PowerShell**, select **PowerShell**.

   Note: If this is the first time you are starting **Cloud Shell** and you are presented with the **You have no storage mounted** message, select the subscription you are using in this lab, and select **Create storage**.
4. In the toolbar of the Cloud Shell pane, select the **Upload/Download files** icon, in the drop-down menu select **Upload**, and upload the file **\\AZ303\AllFiles\Labs\08\azuredeploy30308suba.json** into the Cloud Shell home directory.
5. From the Cloud Shell, run the following command to set a variable named location with an Azure Region near you (replace the ” placeholder with the name of the Azure region that is available for deployment of Azure VMs in your subscription and which is closest to the location of your lab computer, for example ’eastus’):

   ```
   $location = '<Azure region>'
   ```

   Note: To identify Azure regions where you can provision Azure VMs, refer to **https://azure.microsoft.com/en-us/regions/offers/**
6. From the Cloud Shell pane, run the following to create a resource group:

   ```
   New-AzSubscriptionDeployment `
     -Location $location `
     -Name az30308subaDeployment `
     -TemplateFile $HOME/azuredeploy30308suba.json `
     -rgLocation $location `
     -rgName 'az30308a-labRG'
   ```
7. In the Azure portal, close the **Cloud Shell** pane.
8. From your lab computer, open another browser tab, navigate to the 301-nested-vms-in-virtual-network Azure QuickStart template and select **Deploy to Azure**. This will automatically redirect the browser to the **Hyper-V Host Virtual Machine with nested VMs** blade in the Azure portal.

   `https://github.com/Azure/azure-quickstart-templates/tree/master/demos/nested-vms-in-virtual-networ`
9. On the **Hyper-V Host Virtual Machine with nested VMs** blade in the Azure portal, specify the following settings (leave others with their default values):

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	**az30308a-labRG**
Host Public IP Address Name	**az30308a-hv-vm-pip**
Virtual Network Name	**az30308a-hv-vnet**
Host Network Interface1Name	**az30308a-hv-vm-nic1**
Host Network Interface2Name	**az30308a-hv-vm-nic2**
Host Virtual Machine Name	**az30308a-hv-vm**
Host Admin Username	**Student**
Host Admin Password	**Pa55w.rd1234**

10. On the **Hyper-V Host Virtual Machine with nested VMs** blade, select **Review + create** and then select **Create**.

    Note: Wait for the deployment to complete. The deployment might take about 10 minutes.

6.4.1.2 Task 2: Configure nested virtualization in the Azure VM

1. In the Azure portal, search for and select **Virtual machines** and, on the **Virtual machines** blade, select **az30308a-hv-vm**.
2. On the **az30308a-hv-vm** blade, select **Networking**.
3. On the **az30308a-hv-vm | Networking** blade, ensure that the **az30308a-hv-vm-nic1** tab is selected and then select **Add inbound port rule**.

   Note: Make sure that you modify the settings of **az30308a-hv-vm-nic1**, which has the public IP address assigned to it.
4. On the **Add inbound security rule** blade, specify the following settings (leave others with their default values) and select **Add**:

Setting	Value
Destination port range	**3389**
Protocol	**Any**
Name	**AllowRDPInBound**

5. On the **az30308a-hv-vm** blade, select **Overview**.
6. On the **az30308a-hv-vm** blade, select **Connect**, in the drop-down menu, select **RDP**, and then click **Download RDP File**.
7. When prompted, click **Connect** and sign in with the following credentials:

Setting	Value
User Name	**Student**
Password	**Pa55w.rd1234**

8. Within the Remote Desktop session to **az30308a-hv-vm**, in the Server Manager window, click **Local Server**, click the **On** link next to the **IE Enhanced Security Configuration** label, and, in the **IE Enhanced Security Configuration** dialog box, select both **Off** options and then click **OK**.
9. From the Remote Desktop session, open File Explorer and navigate to the **F:**. Create a folder named **VHDs**.
10. Within the Remote Desktop session to **az30308a-hv-vm**, start Internet Explorer, navigate to the download page of Microsoft Edge, download Microsoft Edge installer and perform the installation.
11. Within the Remote Desktop session to **az30308a-hv-vm**, in Microsoft Edge, browse to Windows Server Evaluations, and download the Windows Server 2019 **VHD** file to the **F:\VHDs** folder.

    `https://www.microsoft.com/en-us/evalcenter/evaluate-windows-server-2019`

    Note: The evaluations page will ask for personal information to complete the download. Choose 'United Kingdom' or another country to be able to opt-out of notifications.
12. Within the Remote Desktop session to **az30308a-hv-vm**, click Start, and then click **Windows Administrative Tools**, then launch **Hyper-V Manager**.
13. In the **Hyper-V Manager** console, select the **az30308a-hv-vm** node.
14. Click **New** and, in the cascading menu, select **Virtual Machine**. This will start the **New Virtual Machine Wizard**.
15. On the **Before You Begin** page of the **New Virtual Machine Wizard**, select **Next >**.
16. On the **Specify Name and Location** page of the **New Virtual Machine Wizard**, specify the following settings and select **Next >**:

Setting	Value
Name	**az30308a-vm1**
Store the virtual machine in a different location	selected
Location	**F:\VMs**

Note: Make sure to create the **F:\VMs** folder.

17. On the **Specify Generation** page of the **New Virtual Machine Wizard**, ensure that the **Generation 1** option is selected and select **Next >**:
18. On the **Assign Memory** page of the **New Virtual Machine Wizard**, set **Startup memory** to **2048** and select **Next >**.
19. On the **Configure Networking** page of the **New Virtual Machine Wizard**, in the **Connection** drop-down list select **NestedSwitch** and select **Next >**.
20. On the **Connect Virtual Hard Disk** page of the **New Virtual Machine Wizard**, select the option **Use an existing virtual hard disk**, set location to the VHD file you downloaded to the **F:\VHDs** folder, and select **Next >**.
21. On the **Summary** page of the **New Virtual Machine Wizard**, select **Finish**.
22. In the **Hyper-V Manager** console, select the newly created virtual machine and select **Start**.
23. In the **Hyper-V Manager** console, verify that the virtual machine is running and select **Connect**.
24. In the Virtual Machine Connection window to **az30308a-vm1**, on the **Hi there** page, select **Next**.
25. In the Virtual Machine Connection window to **az30308a-vm1**, on the **License terms** page, select **Accept**.
26. In the Virtual Machine Connection window to **az30308a-vm1**, on the **Customize settings** page, set the password of the built-in Administrator account to **Pa55w.rd1234** and select **Finish**.
27. In the Virtual Machine Connection window to **az30308a-vm1**, sign in by using the newly set password.
28. In the Virtual Machine Connection window to **az30308a-vm1**, start Windows PowerShell and, in the **Administrator: Windows PowerShell** window run the following to set the computer name.

```
Rename-Computer -NewName 'az30308a-vm1' -Restart
```

6.4.2 Exercise 1: Prepare for assessment and migration by using Azure Migrate

The main tasks for this exercise are as follows:

1. Configure Hyper-V environment
2. Create an Azure Migrate project
3. Implement the target Azure environment

6.4.2.1 Task 1: Configure Hyper-V environment

1. Within the Remote Desktop session to **az30308a-hv-vm**, start Microsoft Edge, navigate to the Microsoft Download Center, and download the configuration PowerShell script to the **F:**.

   `https://aka.ms/migrate/script/hyperv`

   Note: The script performs the following tasks:

   - Checks that you're running the script on a supported PowerShell version.
   - Verifies that you have administrative privileges on the Hyper-V host.
   - Allows you to create a local user account that the Azure Migrate service uses to communicate with the Hyper-V host. This user account is added to Remote Management Users, Hyper-V Administrators and Performance Monitor Users groups on the Hyper-V host.
   - Checks that the host is running a supported version of Hyper-V, and the Hyper-V role.
   - Enables the WinRM service, and opens ports 5985 (HTTP) and 5986 (HTTPS) on the host. This is required for metadata collection.
   - Enables PowerShell remoting on the host.
   - Checks that the Hyper-V Integration Services is enabled on all VMs managed by the host.
   - Enables CredSSP on the host if needed.
2. Within the Remote Desktop session to **az30308a-hv-vm**, start **Windows PowerShell ISE**.

3. In the **Administrator: Windows PowerShell ISE** window, in the console pane, run the following to remove the Zone.Identifier alternate data stream, which, in this case, indicates that the file was downloaded from the Internet.

```
Unblock-File -Path F:\MicrosoftAzureMigrate-Hyper-V.ps1
```

4. In the **Administrator: Windows PowerShell ISE** window, open the **MicrosoftAzureMigrate-Hyper-V.ps1** script residing in the **F:** folder and run it. When prompted for confirmation, press the **Y** key followed by the **Enter** key, with exception of the following prompts, in which case, type **N** followed by the **Enter** key:

- Do you use SMB share(s) to store the VHDs?
- Do you want to create non-administrator local user for Azure Migrate and Hyper-V Host communication?

6.4.2.2 Task 2: Create an Azure Migrate project

1. Within the Remote Desktop session to **az30308a-hv-vm**, start Microsoft Edge, navigate to the Azure portal, and sign in by providing credentials of a user account with the Owner role in the subscription you will be using in this lab.
2. In the Azure portal, search for and select **Azure Migrate**, on the **Azure Migrate** blade, in the **Migration goals** section, select **Windows, Linux and SQL Server** and then select **Create Project**.
3. On the **Azure Migrate** blade, specify the following settings (leave others with their default values) and select **Create**:

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	the name of a new resource group **az30308b-labRG**
Migrate project	**az30308b-migrate-project**
Geography	the name of your country or a geographical region

6.4.2.3 Task 3: Implement the target Azure environment

1. In the Azure portal, search for and select **Virtual networks** and, on the **Virtual networks** blade, select **+ Add** (or **+ Create**) on the command bar.
2. On the **Basics** tab of the **Create virtual network** blade, specify the following settings (leave others with their default values) and select **Next: IP Addresses**:

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	the name of a new resource group **az30308c-labRG**
Name	**az30308c-migration-vnet**
Region	the name of the Azure region into which you deployed the virtual machine earlier in this lab

3. On the **IP addresses** tab of the **Create virtual network** blade, in the **IPv4 address space** text box, type **10.8.0.0/16** and select **+ Add subnet**.
4. On the **Add subnet** blade, specify the following settings (leave others with their default values) and select **Add**:

Setting	Value
Subnet name	**subnet0**
Subnet address range	**10.8.0.0/24**

5. Back on the **IP addresses** tab of the **Create virtual network** blade, select **Review + create**.
6. On the **Review + create** tab of the **Create virtual network** blade, select **Create**.
7. In the Azure portal, search for and select **Virtual networks** and, on the **Virtual networks** blade, select

+ Add (or **+ Create**) on the command bar.

8. On the **Basics** tab of the **Create virtual network** blade, specify the following settings (leave others with their default values) and select **Next: IP Addresses**:

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	**az30308c-labRG**
Name	**az30308c-test-vnet**
Region	the name of the Azure region into which you deployed the virtual machine earlier in this lab

9. On the **IP addresses** tab of the **Create virtual network** blade, in the **IPv4 address space** text box, type **10.8.0.0/16** and select **+ Add subnet**.
10. On the **Add subnet** blade, specify the following settings (leave others with their default values) and select **Add**:

Setting	Value
Subnet name	**subnet0**
Subnet address range	**10.8.0.0/24**

11. Back on the **IP addresses** tab of the **Create virtual network** blade, select **Review + create**.
12. On the **Review + create** tab of the **Create virtual network** blade, select **Create**.
13. In the Azure portal, search for and select **Storage accounts** and, on the **Storage accounts** blade, select **+ Add** (or **+ Create**) on the command bar.
14. On the **Basics** tab of the **Create a storage account** blade, specify the following settings (leave others with their default values):

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	**az30308c-labRG**
Storage account name	any globally unique name between 3 and 24 in length consisting of letters and digits
Location	the name of the Azure region in which you created the virtual network earlier in this task
Performance	**Standard**
Account kind	**StorageV2 (general purpose v2)**
Redundancy	**Locally redundant storage (LRS)**

15. On the **Basics** tab of the **Create a storage account** blade, select **Review + create**.
16. On the **Review + create** tab of the **Create a storage account** blade, select **Create**.

6.4.3 Exercise 2: Assess Hyper-V for migration by using Azure Migrate

The main tasks for this exercise are as follows:

1. Deploy and configure the Azure Migrate appliance
2. Configure, run, and view an assessment

6.4.3.1 Task 1: Deploy and configure the Azure Migrate appliance

1. Within the Remote Desktop session to **az30308a-hv-vm**, in the Microsoft Edge window, in the Azure portal, search for and select **Azure Migrate**.
2. On the **Azure Migrate | Windows, Linux and SQL Server** blade, select **Discover** in the **Azure Migrate: Server Assessment** tile.
3. On the **Discover machines** blade, in the **Are your servers virtualized?** drop-down list, select **Yes, with Hyper-V**.

4. On the **Discover machines** blade, in the **Name your appliance** text box, type **az30308a-vma1** and select the **Generate key** button.

   Note: If you encounter a permission-related error while generating an Azure Migrate project key, in the Azure portal, navigate to the **Subscriptions** blade, select your subscription, on your subscription blade, select **Access Control (IAM)** and then assign the **Owner** role to your Azure AD user account.

5. Wait for the resource provisioning to complete, within the Remote Desktop session to **az30308a-hv-vm**, start Notepad, and copy the **Azure Migrate project key** into Notepad.

6. On the **Discover machines** blade, in the **Download Azure Migrate appliance** text box, select the **.VHD file** option, select **Download** and, when prompted, set the download location to the **F:\VMs** folder.

   Note: Wait for the download to complete. This might take about 5 minutes.

7. Once the download completes, extract the content of the downloaded .ZIP file into the **F:\VMs** folder.

8. Within the Remote Desktop session to **az30308a-hv-vm**, switch to the **Hyper-V Manager** console, select the **az30308a-hv-vm** node, select **Import Virtual Machine**. This will start the **Import Virtual Machine** wizard.

9. On the **Before You Begin** page of the **Import Virtual Machine** wizard, select **Next >**.

10. On the **Locate Folder** page of the **Import Virtual Machine** wizard, specify the location of the extracted **Virtual Machines** folder and select **Next >**:

11. On the **Select Virtual Machine** page of the **Import Virtual Machine** wizard, select **Next >**:

12. On the **Choose Import Type** page of the **Import Virtual Machine** wizard, select **Register the virtual machine in place (use the existing unique ID)** and select **Next >**.

13. On the **Configure Processor** page of the **Import Virtual Machine** wizard, set **Number of virtual processors** to **4**, and select **Next >**.

    Note: Ignore any error messages referring to the change of the number of virtual processors.

14. On the **Connect Network** page of the **Import Virtual Machine** wizard, in the **Connection** drop-down list select **NestedSwitch** and select **Next >**.

15. On the **Summary** page of the **Import Virtual Machine** wizard, select **Finish**.

    Note: Wait for the import to complete. This might take about 10 minutes.

16. In the **Hyper-V Manager** console, select the newly imported virtual machine, select **Rename** and set is name to **az30308a-vma1**.

17. In the **Hyper-V Manager** console, select the newly imported virtual machine and select **Start**.

18. In the **Hyper-V Manager** console, verify that the virtual machine is running and select **Connect**.

19. In the Virtual Machine Connection window to the virtual appliance, on the **License terms** page, select **Accept**.

20. In the Virtual Machine Connection window to the virtual appliance, on the **Customize settings** page, set the password of the built-in Administrator account to **Pa55w.rd1234** and select **Finish**.

21. In the Virtual Machine Connection window to the virtual appliance, sign in by using the newly set password.

22. Within the Virtual Machine Connection window to the virtual appliance, start Windows PowerShell and run the following to identify its IP address.

    ```
    (Get-NetIPAddress).IPAddress
    ```

23. Within the Remote Desktop session to **az30308a-hv-vm**, download Microsoft Edge and run the installation with the default settings.

24. Within the Remote Desktop session to **az30308a-hv-vm**, in the Microsoft Edge window, navigate to the https://IPaddress:44368, where the `IPaddress` placeholder represents the IP address you identified in the previous step.

    Note: Ignore the warning about the website's security certificate.

25. When prompted, sign in with the following credentials:

Setting	Value
User Name	**Administrator**
Password	**Pa55w.rd1234**

26. Within the Microsoft Edge window, on the **Appliance Configuration Manager** page, select the **I agree** button, wait for the prerequisites to be successfully verified.
27. Within the Microsoft Edge window, on the **Appliance Configuration Manager** page, in the **Register with Azure Migrate** section, in the **Provide Azure Migrate project key** text box, paste the key you copied into Notepad earlier in this exercise, select **Login**, accept the default code displayed and copy it to the clipboard, then select **Copy code and login** then in the **Enter code** pane in the browser page paste in the code you copied to the clipboard and select **Next**, sign in by providing credentials of a user account with the Owner role in the subscription you are using in this lab and close the browser page.
28. Within the Microsoft Edge window, on the **Appliance Configuration Manager** page, verify that registration was successful and select **Continue**.
29. Within the Microsoft Edge window, on the **Appliance Configuration Manager** page, in the **Manage credentials and discovery sources** section, select **Add credentials**, in the **Add credentials** pane, specify the following settings, select **Save**:

Setting	Value
Friendly Name	**az30308acreds**
User Name	**Student**
Password	**Pa55w.rd1234**

30. Within the Microsoft Edge window, on the **Appliance Configuration Manager** page, in the **Provide Hyper-V host/cluster details** section, select **Add discovery source**, in the **Add discovery source** pane, select the **Add single item** option, ensure that the **Discovery source** drop-down list is set to **Hyper-V Host/Cluster**, in the **Friendly name** drop-down list, select the **az30308acreds** entry, in the **IP address /FQDN** text box, type **10.0.2.1**, and select **Save**.
31. Within the Microsoft Edge window, on the **Appliance Configuration Manager** page, in the **Provide Hyper-V host/cluster details** section, select **Start discovery**.

    Note: In general, it might take about 15 minutes per host for metadata of discovered servers to appear in the Azure portal.

6.4.3.2 Task 2: Configure, run, and view an assessment

1. Within the Remote Desktop session to **az30308a-hv-vm**, in the Microsoft Edge window displaying the Azure portal, navigate back to the **Azure Migrate | Windows, Linux and SQL Server** blade, select **Refresh**, and, in the **Azure Migrate: Discovery and assessment** tile, open the dropdown **Assess** and select **Azure VM**.

    Note: You might need to refresh the page again.

2. On the **Assessment properties** blade select **Edit**, specify the following settings (leave others with their default values) and select **Save**:

Setting	Value
Target location	the name of the Azure region you are using in this lab
Storage type	**Automatic**
Reserved instances	**No reserved instances**
Sizing criterion	**As on premises**
VM series	**Dsv3_series**
Comfort factor	**1**
Offer	**Pay-As-You-Go**
Currency	US Dollar ($)

Setting	Value
Discount	**0**
VM uptime	**31** Day(s) per month and **24** Hour(s) per day

Note: Considering the limited time inherent to the lab environment, the only viable option in this case is **As on-premises** assessment.

3. Back on the **Assess servers** blade, select **Next** and then navigate to the **Select machines to assess** tab.
4. Set **Assessment name** to **az30308a-assessment**.
5. Ensure that the **Create new** option is selected, set the group name to **az30308a-assessment-group**, in the list of machines to be added to the group, select **az30308a-vm1**.
6. Click **Next**, and then click **Create assessment**.
7. Navigate back to the **Azure Migrate | Windows, Linux and SQL Server** blade, select **Refresh**, in the **Azure Migrate: Server Assessment** tile, verify that there the **Assessments** line contains **1** entry, and select it.
8. On the **Azure Migrate: Server Assessment | Assessments** blade, select the newly created assessment **az30308a-assessment**.
9. On the **az30308a-assessment** blade, review the information indicating Azure readiness and monthly cost estimate for both compute and storage.

Note: In real-world scenarios, you should consider installing dependency agent to provide more insights into server dependencies during the assessment stage.

6.4.4 Exercise 3: Migrate Hyper-V VMs by using Azure Migrate

The main tasks for this exercise are as follows:

1. Prepare for migration of Hyper-V VMs
2. Configure replication of Hyper-V VMs
3. Perform migration of Hyper-V VMs
4. Remove Azure resources deployed in the lab

6.4.4.1 Task 1: Prepare for migration of Hyper-V VMs

1. Within the Remote Desktop session to **az30308a-hv-vm**, in the Microsoft Edge window displaying the Azure portal, navigate back to the **Azure Migrate | Windows, Linux and SQL Server** blade.
2. On the **Azure Migrate | Windows, Linux and SQL Server** blade, in the **Azure Migrate: Server Migration** tile, select the **Discover** link.
3. On the **Discover machines** blade, specify the following settings (leave others with their default values) and select **Create resources**:

Setting	Value
Are your machines virtualized?	**Yes, with Hyper-V**
Target region	the name of the Azure region you are using in this lab
Confirm that the target region for migration	selected

Note: This step automatically triggers provisioning of an Azure Site Recovery vault.

4. On the **Discover machines** blade, in step **1. Prepare Hyper-V host servers**, select the first **Download** link (not the Download button), in order to download the Hyper-V replication provider software installer.
5. When prompted, launch **AzureSiteRecoveryProvider.exe**. This will start the **Azure Site Recovery Provider Setup (Hyper-V server)** wizard.
6. On the **Microsoft Update** page, select **Off** and select **Next**.

7. On the **Provider installation** page, select **Install**.
8. Switch to the Azure portal and, on the **Discover machines** blade, select the **Download** button in step 1 of the procedure for preparing on-premises Hyper-V hosts in order to download the vault registration key. When prompted, save the registration key in the **Downloads** folder.
9. Switch to the **Provider installation** page and select **Register**. This will start the **Microsoft Azure Site Recovery Registration Wizard**.
10. On the **Vault Settings** page of the **Microsoft Azure Site Recovery Registration Wizard**, select **Browse**, navigate to the **Downloads** folder, select the vault credentials file, and select **Open**.
11. Back on the **Vault Settings** page of the **Microsoft Azure Site Recovery Registration Wizard**, select **Next**.
12. On the **Proxy Settings** page of the **Microsoft Azure Site Recovery Registration Wizard**, accept the default settings and select **Next**.
13. On the **Registration** page of the **Microsoft Azure Site Recovery Registration Wizard**, select **Finish**.
14. Once the registration process completes, on the **Discover machines** blade, select **Finalize registration**.

    Note: You might have to refresh the browser page displaying the **Discover machines** blade and navigate back to it.

    Note: It might take up to 15 minutes for the discovery of virtual machines to complete.

6.4.4.2 Task 2: Configure replication of Hyper-V VMs

1. Once you receive the confirmation that the registration was finalized, navigate back to the **Azure Migrate | Windows, Linux and SQL Server** blade, in the **Azure Migrate: Server Migration** tile, select the **Replicate** link.

    Note: You might have to refresh the browser page displaying the **Azure Migrate | Windows, Linux and SQL Server** blade.
2. On the **Source settings** page of the **Replicate** blade, in the **Are your machines virtualized?** drop-down list, select **Yes, with Hyper-V** and select **Next: Virtual machines**.
3. On the **Virtual machines** page of the **Replicate** blade, specify the following settings (leave others with their default values) and select **Next: Next: Target settings**:

Setting	Value
Import migration settings from an Azure Migrate assessment	**Yes, apply migration settings from an Azure Migrat**
Select group	**az30308a-assessment-group**
Select assessment	**az30308a-assessment**
Virtual machines	**az30308a-vm1**

4. On the **Target settings** page of the **Replicate** blade, specify the following settings (leave others with their default values) and select **Next: Compute**:

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	**az30308c-labRG**
Replication Storage Account	the name of the storage account you created earlier in this lab
Virtual Network	**az30308c-migration-vnet**
Subnet	**subnet0**

5. On the **Compute** page of the **Replicate** blade, ensure that the **Standard_D2s_v3** is selected in the **Azure VM Size** drop-down list, in the **OS Type** drop-down list, select **Windows** and select **Next: Disks**.
6. On the **Disks** page of the **Replicate** blade, accept the default settings and select **Next: Review +**

Start replication.

7. On the **Review + Start replication** page of the **Replicate** blade, select **Replicate**.
8. To monitor the status of replication, navigate back to the **Azure Migrate | Windows, Linux and SQL Server** blade, in the **Azure Migrate: Server Migration** tile, select the **Replicating servers** entry and, on the **Azure Migrate: Server Migration | Replicating machines**, examine the **Status** column in the list of the replicating machines.
9. Wait until the status changes to **Protected**. This might take additional 15 minutes.

6.4.4.3 Task 3: Perform migration of Hyper-V VMs

1. In the Azure portal, on the **Azure Migrate: Server Migration | Replicating machines**, select the entry representing the **az30308a-vm1** virtual machine.
2. On the **az30308a-vm1** replicating machines blade, select **Test migration**.
3. On the **Test migration** blade, in the **Virtual network** drop-down list, select **az30308c-test-vnet** and select **Test migration**.

   Note: Wait for the test migration to complete. This might take about 5 minutes.
4. In the Azure portal, search for and select **Virtual machines** and, on the **Virtual machines** blade, note the entry representing the newly provisioned virtual machine **az30308a-vm1-test**.
5. In the Azure portal, navigate back to the **Azure Migrate: Server Migration | Replicating machines**, select **Refresh**, and verify that the **az30308a-vm1** virtual machine is listed with the **Cleanup test failover pending** status.
6. On the **Azure Migrate: Server Migration | Replicating machines** blade, select the entry representing the **az30308a-vm1** virtual machine.
7. On the **az30308a-vm1** replicating machines blade, select **Clean up test migration**.
8. On the **Test migrate cleanup** blade, select the checkbox **Testing is complete. Delete test virtual machine** and select **Cleanup Test**.
9. Once the test failover cleanup job completes, refresh the browser page displaying the **az30308a-vm1** replicating machines blade and note that the **Migrate** icon in the toolbar automatically became available.
10. On the **az30308a-vm1** replicating machines blade, select the **Migrate** link.
11. On the **Migrate** blade, in the **Shutdown machines before migration to minimize data loss?** drop-down list, select **Yes**, next select the checkbox next to the **az30308a-vm1** entry, and then select **Migrate**.
12. To monitor the status of migration, navigate back to the **Azure Migrate | Windows, Linux and SQL Server** blade, in the **Azure Migrate: Server Migration** tile, select the **Replicating servers** entry and, on the **Azure Migrate: Server Migration | Replicating machines**, examine the **Status** column in the list of the replicating machines. Verify that the status displayed the **Planned failover finished** status.

    Note: Migration is supposed to be a non-reversible action. If you want to see the completed information, navigate back to the Azure Migrate | Windows, Linux and SQL Server blade, refresh the page and verify that Migrated Servers in the Azure Migrate: Server Migration tile has a value of 1.

6.4.4.4 Task 4: Remove Azure resources deployed in the lab

1. Within the Remote Desktop session to **az30308a-hv-vm**, in the browser window displaying the Azure portal, start a PowerShell session within the Cloud Shell pane.
2. From the Cloud Shell pane, run the following to list the resource group you created in this exercise:

   ```
   Get-AzResourceGroup -Name 'az30308*'
   ```

   Note: Verify that the output contains only the resource group you created in this lab. This group will be deleted in this task.
3. From the Cloud Shell pane, run the following to delete the resource group you created in this lab

```
Get-AzResourceGroup -Name 'az30308*' | Remove-AzResourceGroup -Force -AsJob
```

4. Close the Cloud Shell pane.

6.5 lab: title: '4: Managing Azure AD Authentication and Authorization' module: 'Module 4: Design Authentication and Authorization'

7 Lab: Managing Azure AD Authentication and Authorization

8 Student lab manual

8.1 Lab scenario

As part of its migration to Azure, Adatum Corporation needs to define its identity strategy. Adatum has a single domain Active Directory forest named adatum.com and owns the corresponding, publicly registered DNS domain. As the Adatum Enterprise Architecture team is exploring the option of transitioning some of the on-premises workloads to Azure, it intends to evaluate integration between its Active Directory Domain Services (AD DS) environment and the Azure Active Directory (Azure AD) tenant associated with the target Azure subscription as the core component of its longer-term authentication and authorization model.

The new model should facilitate single sign-on, along with per-application step-up authentication that leverages multi-factor authentication capabilities of Azure AD. To implement single sign-on, the Architecture team plans to deploy Azure AD Connect and configure it for password hash synchronization, resulting in matching user objects in both identity stores. Choosing the optimal authentication method is the first concern for organizations wanting to move to the cloud. Azure AD password hash synchronization is the simplest way to implement single sign-on authentication for on-premises users when accessing Azure AD-integrated resources. This method is also required by some premium Azure AD features, such as Identity Protection.

To implement step-up authentication, the Adatum Enterprise Architecture team intends to take advantage of Azure AD Conditional Access policies. Conditional Access policies support enforcement of multi-factor authentication depending on the type of application or resource being accessed. Conditional Access policies are enforced after the first-factor authentication has been completed. Conditional Access can be based on a wide range of factors, including:

- User or group membership. Policies can be targeted to specific users and groups giving administrators fine-grained control over access.
- IP Location information. Organizations can create trusted IP address ranges that can be used when making policy decisions. Administrators can specify entire countries/regions IP ranges to block or allow traffic from.
- Device. Users with devices of specific platforms or marked with a specific state can be used when enforcing Conditional Access policies.
- Application. Users attempting to access specific applications can trigger different Conditional Access policies.
- Real-time and calculated risk detection. Signals integration with Azure AD Identity Protection allows Conditional Access policies to identify risky sign-in behavior. Policies can then force users to perform password changes or multi-factor authentication to reduce their risk level or be blocked from access until an administrator takes manual action.
- Microsoft Cloud App Security (MCAS). Enables user application access and sessions to be monitored and controlled in real time, increasing visibility and control over access to and activities performed within your cloud environment.

To accomplish these objectives the Adatum Enterprise Architecture team intends to test integration of its Active Directory Domain Services (AD DS) forest with its Azure Active Directory (Azure AD) tenant and evaluate the conditional access functionality for its pilot users.

8.2 Objectives

After completing this lab, you will be able to:

- Deploy an Azure VM hosting an AD DS domain controller
- Create and configure an Azure AD tenant

- Integrate an AD DS forest with an Azure AD tenant

8.3 Lab Environment

Windows Server admin credentials

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

Estimated Time: 120 minutes

8.4 Lab Files

- \\AZ304\AllFiles\Labs\10\azuredeploy30410suba.json

8.5 Instructions

8.5.1 Exercise 0: Prepare the lab environment

The main tasks for this exercise are as follows:

1. Identify an available DNS name for an Azure VM deployment
2. Deploy an Azure VM running an AD DS domain controller by using an Azure Resource Manager QuickStart template

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

1. From your lab computer, start a web browser, navigate to the Azure portal, and sign in by providing credentials of a user account with the Owner role in the subscription you will be using in this lab.
2. In the Azure portal, open **Cloud Shell** pane by selecting on the toolbar icon directly to the right of the search textbox.
3. If prompted to select either **Bash** or **PowerShell**, select **PowerShell**.

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

4. In the Cloud Shell pane, run the following to identify an available DNS name you will need to provide in the next task (substitute the placeholder `<custom-label>` with any valid DNS hostname which is likely to be globally unique and the placeholder `<Azure region>` with the name of the Azure region into which you want to deploy the Azure VM that will host an Active Directory domain controller):

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

   > **Note**: To identify Azure regions where you can provision Azure VMs, refer to https://azure.microsoft.com/en-us/regions/offers/, you can also get the list of the regions using **Powershell cmdlet**

   ```
   Get-AzLocation | FT
   ```

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

8.5.1.2 Task 2: Deploy an Azure VM running an AD DS domain controller by using an Azure Resource Manager QuickStart template

1. In the Azure portal, in the toolbar of the Cloud Shell pane, select the **Upload/Download files** icon, in the drop-down menu select **Upload**, and upload the file **\\AZ304\AllFiles\Labs\10\azuredeploy30410suba.json** into the Cloud Shell home directory.
2. From the Cloud Shell pane, run the following to create a resource groups (replace the `<Azure region>` placeholder with the name of the Azure region that you specified in the previous task):

   ```
   $location = '<Azure region>'
   ```

```
New-AzSubscriptionDeployment `
  -Location $location `
  -Name az30410subaDeployment `
  -TemplateFile $HOME/azuredeploy30410suba.json `
  -rgLocation $location `
  -rgName 'az30410a-labRG'
```

3. In the Azure portal, close the **Cloud Shell** pane.
4. From your lab computer, open another browser tab and navigate to the https://github.com/Azure/azure-quickstart-templates/tree/master/application-workloads/active-directory/active-directory-new-domain.
5. On the **Create a new Windows VM and create a new AD Forest, Domain and DC** page, select **Deploy to Azure**. This will automatically redirect the browser to the **Create an Azure VM with a new AD Forest** blade in the Azure portal.
6. On the **Create an Azure VM with a new AD Forest** blade, select **Edit parameters**.
7. On the **Edit parameters** blade, select **Load file**, in the **Open** dialog box, select **\\AZ304\AllFiles\Labs\10\azure** select **Open**, and then select **Save**.
8. On the **Create an Azure VM with a new AD Forest** blade, specify the following settings (leave others with their existing values):

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	**az30410a-labRG**
Dns Prefix	the DNS hostname you identified in the previous task

9. On the **Create an Azure VM with a new AD Forest** blade, select **Review + create** and select **Create**.

   Note: Do not wait for the deployment to complete but instead proceed to the next exercise. The deployment might take about 15 minutes. You will use the virtual machine deployed in this task in the third exercise of this lab.

8.5.2 Exercise 1: Create and configure an Azure AD tenant

The main tasks for this exercise are as follows:

1. Create an Azure AD tenant
2. Create and configure Azure AD users
3. Activate and assign Azure AD Premium P2 licensing

8.5.2.1 Task 1: Create an Azure AD tenant

1. In the Azure portal, search for and select **Azure Active Directory** and, on the Azure Active Directory blade, select **+ Create a tenant**.
2. On the **Basics** tab of the **Create a directory** blade, select the **Azure Active Directory** option and select **Next: Configuration >**.
3. On the **Configuration** tab of the **Create a directory** blade, specify the following settings (leave others with their existing values):

Setting	Value
Organization name	**Adatum Lab**
Initial domain name	any valid DNS name consisting of lower case letters and digits and starting with a letter
Country/Region	**United States**

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

4. Select **Next: Review + create** and then select **Create**.
5. Refresh the browser page displaying the Azure portal, search for and select **Azure Active Directory** and, on the Azure Active Directory blade, select **Switch tenant**.
6. In the **Directory + subscription** blade, on the **Adatum Lab** card, click **Switch**.

8.5.2.2 Task 2: Create and configure Azure AD users

1. On the **Adatum Lab | Overview** Azure Active Directory blade, in the **Manage** section, select **Users**, on the **Users | All users** blade, select your user account to display its **Profile** settings.
2. On the profile blade of your user account, select **Edit**, in the **Settings** section, set **Usage location** to **United States** and select **Save** to save the change.

   Note: This is necessary in order to assign an Azure AD Premium P2 license to your user account later in this lab.
3. Navigate back to the **Users - All users** blade, and then select + **New user**.
4. On the **New user** blade, specify the following settings (leave others with their defaults):

Setting	Value
User name	**az30410-aaduser1**
Name	**az30410-aaduser1**
Auto-generate password	enabled
Show password	enabled
Roles	**Global administrator**
Usage location	**United States**

   Note: Record the full user name (including the domain name) and the auto-generated password. You will need it later in this task.
5. On the **New user** blade, select **Create**
6. On the lab computer, open an **InPrivate** browser window and sign in to the Azure portal using the newly created **az30410-aaduser1** user account. When prompted to update the password, change the password to **Pa55w.rd1234**.
7. Sign out as the **az30410-aaduser1** user from the Azure portal and close the InPrivate browser window.

8.5.2.3 Task 3: Activate and assign Azure AD Premium P2 licensing

1. Back in the browser window displaying the Azure portal, navigate to the **Overview** blade of the **Adatum Lab** Azure AD tenant and, in the **Manage** section, select **Licenses**.
2. On the **Licenses | Overview** blade, select **All products**, select + **Try/Buy**.
3. On the **Activate** blade, in the **Azure AD Premium P2** section, select **Free trial** and then select **Activate**.
4. Refresh the browser window showing the **Licenses | All products** blade to verify that the activation was successful.
5. On the **Licenses - All products** blade, select the **Azure Active Directory Premium P2** entry.
6. On the **Azure Active Directory Premium P2 | Licensed users** blade, select + **Assign**.
7. On the **Assign license** blade, select **Users**, and on the **Users** blade, select both your account and the **az30410-aaduser1** user account and click **Select** for each.
8. Back on the **Assign license** blade, select **Assignment options**, review the options listed on the **License options** blade, and select **OK**.
9. On the **Assign license** blade, select **Assign**.

8.5.3 Exercise 2: Integrate an AD DS forest with an Azure AD tenant

The main tasks for this exercise are as follows:

1. Assign a custom domain name to the Azure AD tenant
2. Configure AD DS in the Azure VM
3. Install Azure AD Connect
4. Configure properties of synchronized user accounts

8.5.3.1 Task 1: Assign a custom domain name to the Azure AD tenant

1. In the Azure portal, navigate to the **Azure Active Directory Adatum Lab | Overview** blade.
2. On the **Adatum Lab | Overview** blade, select **Custom domain names**.
3. On the **Adatum Lab | Custom domain names** blade, identify the primary, default DNS domain name associated with the Azure AD tenant.

   Note: Record the value of the primary DNS name of the Azure AD tenant. You will need it in the next task.
4. On the **Adatum Lab | Custom domain names** blade, select **+ Add custom domain**.
5. On the **Custom domain name** blade, in the **Custom domain name** text box, type **adatum.com**, and select **Add domain**.
6. On the **adatum.com** blade, review the information necessary to perform verification of the Azure AD domain name and close the blade.

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

8.5.3.2 Task 2: Configure AD DS in the Azure VM

Note: Make sure that the deployment of the Azure VM you initiated at the beginning of the lab has completed before you start this exercise.

1. In the Azure portal, search for and select **Azure Active Directory** and, on the Azure Active Directory blade, select **Switch tenant**.
2. On the **Switch tenant** blade, click the **Switch** button in the tile representing the Azure AD tenant associated with the Azure subscription into which you deployed the **az30410a-vm1** Azure VM in the previous exercise of this lab.
3. In the Azure portal, search for and select **Virtual machines** and, on the **Virtual machines** blade, select **az30410a-vm1**.
4. On the **az30410a-vm1** blade, select **Connect**, in the drop-down menu, select **RDP**, on the **RDP** tab of the **az30410a-vm1 | Connect** blade, in the **IP address** drop-down list, select the **Load balancer public IP address** entry, select **Download RDP File** and open the downloaded RDP file.
5. When prompted, sign in with the following credentials:

Setting	Value
User Name	**Student**
Password	**Pa55w.rd1234**

6. Within the Remote Desktop session to **az30410a-vm1**, in the Server Manager window, select **Local Server**, select the **On** link next to the **IE Enhanced Security Configuration** label, and, in the **IE Enhanced Security Configuration** dialog box, select both **Off** options.
7. Within the Remote Desktop session to **az30410a-vm1**, in the Server Manager window, select **Tools** and, in the drop-down menu, select **Active Directory Administrative Center**

8. In **Active Directory Administrative Center**, select **adatum (local)**, in the **Tasks** pane, select **New**, and, in the cascading menu, select **Organizational Unit**.
9. In the **Create Organizational Unit** window, in the **Name** text box, type **ToSync** and select **OK**.
10. Double-click the newly crated **ToSync** organizational unit such that it its content appears in the details pane of the Active Directory Administrative Center console.
11. In the **Tasks** pane, within the **ToSync** section, select **New**, and, in the cascading menu, select **User**.
12. In the **Create User** window, create a new user account with the following settings (leave others with their existing values) and select **OK**:

Setting	Value
Full Name	**aduser1**
User UPN logon	**aduser1**
User SamAccountName logon	**aduser1**
Password	**Pa55w.rd1234**
Other password options	**Password never expires**

8.5.3.3 Task 3: Install Azure AD Connect

1. Within the Remote Desktop session to **az30410a-vm1**, start Internet Explorer, navigate to the download page of Microsoft Edge, download Microsoft Edge installer and perform the installation.
2. Within the Remote Desktop session to **az30410a-vm1**, in Microsoft Edge, navigate to the Azure portal, and sign in by using the **az30410-aaduser1** user account you created the previous exercise. When prompted, specify the full user name you recorded and the **Pa55w.rd1234** password.
3. In the Azure portal, search for and select **Azure Active Directory** and, on the **Adatum Lab | Overview** blade, select **Azure AD Connect**.
4. On the **Adatum Lab | Azure AD Connect** blade, select the **Download Azure AD Connect** link. You will be redirected to the **Microsoft Azure Active Directory Connect** download page.
5. On the **Microsoft Azure Active Directory Connect** download page, select **Download**.
6. When prompted, select **Run** to start the **Microsoft Azure Active Directory Connect** wizard.
7. On the **Welcome to Azure AD Connect** page of the **Microsoft Azure Active Directory Connect** wizard, select the checkbox **I agree to the license terms and privacy notice** and select **Continue**.
8. On the **Express Settings** page of the **Microsoft Azure Active Directory Connect** wizard, select the **Customize** option.
9. On the **Install required components** page, leave all optional configuration options deselected and select **Install**.
10. On the **User sign-in** page, ensure that only the **Password Hash Synchronization** is enabled and select **Next**.
11. On the **Connect to Azure AD** page, authenticate by using the credentials of the **az30410-aaduser1** user account you created in the previous exercise and select **Next**.
12. On the **Connect your directories** page, select the **Add Directory** button to the right of the **adatum.com** forest entry.
13. In the **AD forest account** window, ensure that the option to **Create new AD account** is selected, specify the following credentials, and select **OK**:

Setting	Value
User Name	**ADATUM\Student**
Password	**Pa55w.rd1234**

14. Back on the **Connect your directories** page, ensure that the **adatum.com** entry appears as a configured directory and select **Next**

15. On the **Azure AD sign-in configuration** page, note the warning stating **Users will not be able to sign-in to Azure AD with on-premises credentials if the UPN suffix does not match a verified domain name**, enable the checkbox **Continue without matching all UPN suffixes to verified domain**, and select **Next**.

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

16. On the **Domain and OU filtering** page, select the option **Sync selected domains and OUs**, clear all checkboxes, select only the checkbox next to the **ToSync** OU, and select **Next**.
17. On the **Uniquely identifying your users** page, accept the default settings, and select **Next**.
18. On the **Filter users and devices** page, accept the default settings, and select **Next**.
19. On the **Optional features** page, accept the default settings, and select **Next**.
20. On the **Ready to configure** page, ensure that the **Start the synchronization process when configuration completes** checkbox is selected and select **Install**.

    Note: Installation should take about 2 minutes.

21. Review the information on the **Configuration complete** page and select **Exit** to close the **Microsoft Azure Active Directory Connect** window.

8.5.3.4 Task 4: Configure properties of synchronized user accounts

1. Within the Remote Desktop session to **az30410a-vm1**, in the Microsoft Edge window displaying the Azure portal, navigate to the **Users - All users** blade of the Adatum Lab Azure AD tenant.
2. On the **Users | All users** blade, note that the list of user objects includes the **aduser1** account, with the **Yes** entry appearing in the **Directory synced** column.

    Note: You might have to wait a few minutes and select **Refresh** for the **aduser1** user account to appear.

3. On the **Users | All users** blade, select the **aduser1** entry.
4. On the **aduser1 | Profile** blade, note the full name of the user account.

    Note: Record the full user name. You will need it in the next exercise.

5. On the **aduser1 | Profile** blade, in the **Job info** section, note that the **Department** attribute is not set.
6. Within the Remote Desktop session to **az30410a-vm1**, switch to **Active Directory Administrative Center**, select the **aduser1** entry in the list of objects in the **ToSync** OU, and, in the **Tasks** pane, in the **ToSync** section, select **Properties**.
7. In the **aduser1** window, in the **Organization** section, in the **Department** text box, type **Sales**, and select **OK**.
8. Within the Remote Desktop session to **az30410a-vm1**, start **Windows PowerShell**.
9. From the **Administrator: Windows PowerShell** console, run the following to start Azure AD Connect delta synchronization:

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

Start-ADSyncSyncCycle -PolicyType Delta
```

10. Switch to the Microsoft Edge window displaying the **aduser1 | Profile** blade, refresh the page and note that the **Department** property is set to **Sales**.

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

11. On the **aduser1 | Profile** blade, select **Edit**.
12. On the **aduser1 | Profile** blade, in the **Settings** section, in the **Usage location** drop-down list, select **United States** and then select **Save**.
13. On the **aduser1 | Profile** blade, select **Licenses**.

14. On the **aduser1 | Licenses** blade, select **+ Assignments**.
15. On the **Update license assignments** blade, select the **Azure Active Directory Premium P2** checkbox and select **Save**.

8.5.4 Exercise 3: Implement Azure AD conditional access

The main tasks for this exercise are as follows:

1. Disable Azure AD security defaults.
2. Create an Azure AD conditional access policy
3. Verify Azure AD conditional access
4. Remove Azure resources deployed in the lab

8.5.4.1 Task 1: Disable Azure AD security defaults.

1. Within the Remote Desktop session to **az30410a-vm1**, in the Microsoft Edge window displaying the Azure portal, navigate to the **Adatum Lab | Overview** blade of the Adatum Lab Azure AD tenant.
2. On the **Adatum Lab | Overview** blade, in the **Manage** section, select **Properties**.
3. On the **Adatum Lab | Properties** blade, select the **Manage Security defaults** link at the bottom of the page.
4. On the **Enable Security defaults** blade, set **Enable Security defaults** switch to **No**, select the checkbox **My organization is using Conditional Access**, and select **Save**.

8.5.4.2 Task 2: Create an Azure AD conditional access policy

1. On the **Adatum Lab | Properties** blade, in the **Manage** section, select the **Security**.
2. On the **Security | Getting started** blade, select **Conditional Access**.
3. On the **Conditional Access | Policies** blade, select **+ New policy**.
4. On the **New** blade, in the **Name** text box, type **Azure portal MFA enforcement**.
5. On the **New** blade, in the **Assignments** section, select **Users and groups**, on the **Include** tab, select **Select users and groups**, select the **Users and groups** checkbox, on the **Select** blade, select **aduser1**, and confirm your choice by clicking **Select**.
6. Back on the **New** blade, in the **Assignments** section, select **Cloud apps or actions**, on the **Include** tab, select **Select apps**, click **Select**, on the **Select** blade, select **Microsoft Azure Management** checkbox, and confirm your choice by clicking **Select**.
7. Back on the **New** blade, in the **Access controls** section, select **Grant**, on the **Grant** blade, ensure that the **Grant** option is selected, select **Require multi-factor authentication**, and confirm your choice by clicking **Select**.
8. Back on the **New** blade, set the **Enable policy** switch to **On** and select **Create**.

8.5.4.3 Task 3: Verify Azure AD conditional access

1. Within the Remote Desktop session to **az30410a-vm1**, in the **Microsoft Edge** window, select **Settings** menu header, in the **Settings** menu, select **Safety**, in the cascading menu, select **InPrivate Browsing**, and, in the InPrivate Microsoft Edge window, navigate to the Access Panel Applications portal https://myapplications.microsoft.com.
2. When prompted, sign in by using the synchronized Azure AD account of the **aduser1**, using the full user name you recorded in the previous exercise and the **Pa55w.rd1234** password.
3. Verify that you can successfully sign in to the Access Panel Applications portal.
4. In the same browser window, navigate to the Azure portal.
5. Note that, this time, you are presented with the message **More information required**. Within the page displaying the message, select **Next**.

6. At that point, you will be redirected to the **Additional security verification** page, which will step you through configuring multi-factor authentication.

   Note: Completing the multi-factor authentication configuration is optional. If you proceed, you will need to designate your mobile device as an authentication phone or to use it to run a mobile app.

8.5.4.4 Task 4: Remove Azure resources deployed in the lab

1. Within the Remote Desktop session to **az30410a-vm1**, start Microsoft Edge and browse to the Microsoft Online Services Sign-In Assistant for IT Professionals RTW at https://www.microsoft.com/en-us/Download/details.aspx?id=28177.
2. On the Microsoft Online Services Sign-In Assistant for IT Professionals RTW download page, select **Download**, on the **Choose the download you want** page, select **en\msoidcli_64.msi**, and select **Next**.
3. When prompted, run **Microsoft Online Services Sign-in Assistant Setup** with the default options.
4. Once the setup completes, within the Remote Desktop session to **az30410a-vm1**, start **Windows PowerShell** console.
5. In the **Administrator: Windows PowerShell** window, run the following to install the required PowerShell module:

   ```
   [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
   Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force
   Install-Module MSOnline -Force
   ```

6. In the **Administrator: Windows PowerShell** window, run the following to authenticate to the **Adatum Lab** Azure AD tenant:

   ```
   Connect-MsolService
   ```

7. When prompted to authenticate, provide the credentials of the **az30410-aaduser1** user account.
8. In the **Administrator: Windows PowerShell** window, run the following to disable Azure AD Connect synchronization:

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

   Note: If you receive an error message at this point, you might have to wait for up to 12 hours and try again.

9. From the lab computer, in the browser window displaying the Azure portal, switch to the **Adatum Lab** tenant, navigate to the **Azure Active Directory Premium P2 - Licensed users** blade, select the user accounts to which you assigned licenses in this lab, select **Remove license**, and, when prompted to confirm, select **OK**.
10. In the Azure portal, navigate to the **Users - All users** blade and ensure that all user accounts you created in this lab are no longer listed as **Directory synced**.
11. On the **Users - All users** blade, select each user accounts you created in this lab and select **Delete** in the toolbar.
12. Navigate to the **Adatum Lab - Overview** blade of the Adatum Lab Azure AD tenant, select **Delete tenant**, on the **Delete directory 'Adatum Lab'** blade, select the **Get permission to delete Azure resources** link, on the **Properties** blade of Azure Active Directory, set **Access management for Azure resources** to **Yes** and select **Save**.
13. Sign out from the Azure portal and sign in back.
14. Navigate back to the **Delete directory 'Adatum Lab'** blade and select **Delete**.
15. On the lab computer, in the browser window displaying the Azure portal, make sure you are connected to the original Azure Active Directory tenant, and start a PowerShell session within the Cloud Shell pane.
16. From the Cloud Shell pane, run the following to list the resource group you created in this exercise:

    ```
    Get-AzResourceGroup -Name 'az30410*'
    ```

Note: Verify that the output contains only the resource group you created in this lab. This group will be deleted in this task.

17. From the Cloud Shell pane, run the following to delete the resource group you created in this lab

```
Get-AzResourceGroup -Name 'az30410*' | Remove-AzResourceGroup -Force -AsJob
```

18. Close the Cloud Shell pane.

8.6 lab: title: '6: Implementing Azure SQL Database-Based Applications' module: 'Module 6: Design a Solution for Databases'

9 Lab: Implementing Azure SQL Database-Based Applications

10 Student lab manual

10.1 Lab scenario

Adatum Corporation has a number two tier applications with .NET Core-based front end and SQL Server-based backend. The Adatum Enterprise Architecture team is exploring the possibility of implementing these applications by leveraging Azure SQL Database as the data tier. Given intermittent, unpredictable usage of the existing SQL Server backend and relatively high tolerance for latency built into the front-end apps, Adatum is considering the serverless tier of Azure SQL Database.

Serverless is a compute tier for individual Azure SQL Databases instances that automatically scales compute based on workload demand and bills for compute used per second. The serverless compute tier is also capable of automatically pausing databases during inactive periods when only storage is billed and automatically resumes databases when activity returns.

The Adatum Enterprise Architecture team is also interested in evaluating network-level security provided by the Azure SQL Databases, in order to ensure that it is possible to restrct inbound connections to specific ranges of IP addresses, in scenarios where the apps must be able to connect from its on-premises locations without relying on hybrid connectivity via Site-to-Site VPN or ExpressRoute.

To accomplish these objectives, the Adatum Architecture team will test Azure SQL Database-based applications, including:

- Implementing serverless tier of Azure SQL Database
- Implementing .NET Core console apps that use Azure SQL Database as their data store

10.2 Objectives

After completing this lab, you will be able to:

- Implement serverless tier of Azure SQL Database
- Configure .NET Core-based console apps that use Azure SQL Database as their data store

10.3 Lab Environment

Windows Server admin credentials

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

Estimated Time: 60 minutes

10.4 Lab Files

- None

10.4.1 Exercise 1: Implement Azure SQL Database

The main tasks for this exercise are as follows:

1. Create Azure SQL Database
2. Connect to and query Azure SQL Database

10.4.1.1 Task 1: Create Azure SQL Database

1. From your lab computer, start a web browser, navigate to the Azure portal, and sign in by providing credentials of a user account with the Owner role in the subscription you will be using in this lab.
2. In the Azure portal, search for and select **SQL database** and, on the **SQL databases** blade, select **+ Add**.
3. On the **Basics** tab of the **Create SQL Database** blade, specify the following settings (leave others with their default values):

Setting	Value
Subscription	the name of the Azure subscription you will be using in this lab
Resource group	the name of a new resource group **az30303a-labRG**
Database name	**az30303a-db1**

4. Directly below the **Server** drop down list, select the **Create new** and, on the **New server** blade, specify the following settings and select **OK** (leave others with their default values):

Setting	Value
Server name	any valid, globally unique name
Server admin login	**sqladmin**
Password	**Pa55w.rd1234**
Location	the name of an Azure region where you can provision SQL databases

5. Next to the **Compute + storage** label, select the **Configure database** link.
6. On the **Configure** blade, select **Serverless**, review the corresponding hardware configuration and auto-pause delay settings, leave the **Enable auto-pause** checkbox enabled, and select **Apply**.
7. Back on the **Basics** tab of the **Create SQL Database** blade, select **Next: Networking >**.
8. On the **Networking** tab of the **Create SQL Database** blade, specify the following settings (leave others with their default values):

Setting	Value
Connectivity method	**Public endpoint**
Allow Azure services and resources to access this server	**No**
Add current client IP address	**Yes**

9. Select **Next: Security >**.
10. On the **Security** tab of the **Create SQL Database** blade, specify the following settings (leave others with their default values):

Setting	Value
Enable Azure Defender for SQL	**Not now**

11. Select **Next: Additional settings >**.
12. On the **Additional settings** tab of the **Create SQL Database** blade, specify the following settings (leave others with their default values):

Setting	Value
Use existing data	**Sample**

13. Select **Review + create** and then select **Create**.

    Note: Wait for the SQL database to be created. Provisioning should take about 2 minutes.

10.4.1.2 Task 2: Connect to and query Azure SQL Database

1. In the Azure portal, search for and select **SQL database** and, on the **SQL databases** blade, select the entry representing the newly created **az30303a-db1** Azure SQL database.
2. On the SQL database blade, select **Query editor (preview)**.
3. In the **SQL Server authentication** section, in the **Password** textbox, type **Pa55w.rd1234** and select **OK**.
4. In the **Query editor (preview)** pane, on the **Query 1** tab, enter the following query and select **Run**:

```
SELECT TOP 20 pc.Name as CategoryName, p.name as ProductName
FROM SalesLT.ProductCategory pc
JOIN SalesLT.Product p
ON pc.productcategoryid = p.productcategoryid;
```

5. Review the **Results** tab to verify that the query completed successfully.

10.4.2 Exercise 2: Implement a .NET Core console app that uses Azure SQL Database as their data store

The main tasks for this exercise are as follows:

1. Identify ADO.NET connection information of Azure SQL Database
2. Create and configure a .NET Core console app
3. Test the .NET Core console app
4. Configure Azure SQL database firewall
5. Verify the functionality of the .NET Core console app
6. Remove Azure resources deployed in the lab

10.4.2.1 Task 1: Identify ADO.NET connection information of Azure SQL Database

1. In the Azure portal, on the blade of the Azure SQL database you deployed in the previous exercise, in the **Settings** section, select **Connection strings**.
2. On the **ADO.NET** tab, note the ADO.NET connection string for SQL authentication.

10.4.2.2 Task 2: Create and configure a .NET Core console app

1. In the Azure portal, open the **Cloud Shell** pane by selecting on the toolbar icon directly to the right of the search textbox.
2. If prompted to select either **Bash** or **PowerShell**, select **Bash**.

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

3. From the Cloud Shell pane, run the following to create a new folder named **az30303a1** and set it as your current directory:

```
mkdir az30303a1
cd az30303a1/
```

4. From the Cloud Shell pane, run the following to create a new app project file for a .NET Core-based app based on the desktop template:

```
dotnet new console
```

5. In the Cloud Shell pane, use the built in editor to open and modify the **az30303a1.csproj** file by adding the following XML element between the `<Project>` tags:

```
<ItemGroup>
    <PackageReference Include="System.Data.SqlClient" Version="4.6.0" />
</ItemGroup>
```

6. Save and close the **az30303a1.csproj** file.

7. In the Cloud Shell pane, use the built in editor to open and modify the **Program.cs** file by replacing its content with the following code:

```
using System;
using System.Data.SqlClient;
using System.Text;

namespace sqltest
{
    class Program
    {
        static void Main(string[] args)
        {
            try
            {
                SqlConnectionStringBuilder builder = new SqlConnectionStringBuilder();
                builder.ConnectionString="<your_ado_net_connection_string>";

                using (SqlConnection connection = new SqlConnection(builder.ConnectionString))
                {
                    Console.WriteLine("\nQuery data example:");
                    Console.WriteLine("=========================================\n");

                    connection.Open();
                    StringBuilder sb = new StringBuilder();
                    sb.Append("SELECT TOP 20 pc.Name as CategoryName, p.name as ProductName ");
                    sb.Append("FROM [SalesLT].[ProductCategory] pc ");
                    sb.Append("JOIN [SalesLT].[Product] p ");
                    sb.Append("ON pc.productcategoryid = p.productcategoryid;");
                    String sql = sb.ToString();

                    using (SqlCommand command = new SqlCommand(sql, connection))
                    {
                        using (SqlDataReader reader = command.ExecuteReader())
                        {
                            while (reader.Read())
                            {
                                Console.WriteLine("{0} {1}", reader.GetString(0), reader.GetString
                            }
                        }
                    }
                }
                catch (SqlException e)
                {
                    Console.WriteLine(e.ToString());
                }
                Console.WriteLine("\nDone. Press enter.");
                Console.ReadLine();
            }
        }
    }
```

8. Leave the editor window open.
9. In the Azure portal, on the blade displaying the connection strings for the **az30303a-db1** database, copy the ADO.NET connection string.
10. Switch back to the editor window and replace the placeholder `<your_ado_net_connection_string>` with the value of the connection string you copied in the previous step.
11. In the connection string you copied into the editor window, replace the placeholder `{your_password}` with **Pa55w.rd1234**.
12. Save and close the **Program.cs** file.

10.4.2.3 Task 3: Test the .NET Core console app

1. From the Cloud Shell pane, run the following to compile the newly created .NET Core-based console app:

   ```
   dotnet restore
   ```

2. From the Cloud Shell pane, run the following to execute the newly created .NET Core-based console app:

   ```
   dotnet run
   ```

3. Note that the execution of the console app will trigger an error.

   > **Note**: This is expected, since the connection from IP address assigned to the virtual machine running the Cloud Shell session must be explicitly allowed.

10.4.2.4 Task 4: Configure Azure SQL database firewall

1. From the Cloud Shell pane, run the following to identify the public IP address of the virtual machine running the Cloud Shell session:

   ```
   curl -s checkip.dyndns.org | sed -e 's/.*Current IP Address: //' -e 's/<.*$//'
   ```

2. In the Azure portal, on the blade displaying the connection strings for the **az30303a-db1** database, select **Overview** and, in the toolbar, select **Set server firewall**.
3. On the **Firewall settings** blade, set the following entries and select **Save**:

Setting	Value
Rule name	**cloudshell**
Start IP	the IP address you identified earlier in this task
End IP	the IP address you identified earlier in this task

> **Note**: Obviously this is meant for the lab purposes only, since that IP address will change after you restart the Cloud Shell session.

10.4.2.5 Task 5: Verify the functionality of the .NET Core console app

1. From the Cloud Shell pane, run the following to execute the newly created .NET Core-based console app:

   ```
   dotnet run
   ```

2. Note that the execution of the console app will this time be successful and that it returns the same results as those displayed in the query editor within the Azure portal SQL database blade.

10.4.2.6 Task 6: Remove Azure resources deployed in the lab

1. From the Cloud Shell pane, run the following to list the resource group you created in this exercise:

   ```
   az group list --query "[?starts_with(name,'az30303')]".name --output tsv
   ```

   > **Note**: Verify that the output contains only the resource group you created in this lab. This group will be deleted in this task.

2. From the Cloud Shell pane, run the following to delete the resource group you created in this lab

   ```
   az group list --query "[?starts_with(name,'az30303')]".name --output tsv | xargs -L1 bash -c 'az g
   ```

3. From the Cloud Shell pane, run the following to remove the folder named **az30303a1**:

```
rm -r ~/az30303a1
```

4. Close the Cloud Shell pane.