Lab - Prepare for deployment of Azure Virtual Desktop (AD DS)

Student lab manual

Lab dependencies

- An Azure subscription you will be using in this lab.
- A Microsoft account or an Microsoft Entra account with the Owner or Contributor role in the Azure subscription you will be using in this lab and with the Global Administrator role in the Microsoft Entra tenant associated with that Azure subscription.

Estimated Time

60 minutes

Lab scenario

You need to prepare for deployment of an Active Directory Domain Services (AD DS) environment

Objectives

After completing this lab, you will be able to:

- Deploy an Active Directory Domain Services (AD DS) single-domain forest by using Azure VMs
- Integrate an AD DS forest with a Microsoft Entra tenant

Lab files

- \\AZ-140\AllFiles\Labs\01\az140-11_azuredeploydc11.parameters.json
- \\AZ-140\AllFiles\Labs\01\az140-11_azuredeploycl11.json
- \\AZ-140\AllFiles\Labs\01\az140-11_azuredeploycl11.parameters.json

Instructions

Exercise 0: Increase the number of vCPU quotas

The main tasks for this exercise are as follows:

1. Identify current vCPU usage
2. Request vCPU quota increase

Task 1: Identify current vCPU usage

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 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 Azure portal, in the PowerShell session of the **Cloud Shell**, run the following to register the **Microsoft.Compute** and **Microsoft.Network** resource providers, in case they're not registered:

```
Register-AzResourceProvider -ProviderNamespace 'Microsoft.Compute'
Register-AzResourceProvider -ProviderNamespace 'Microsoft.Network'
```

5. In the Azure portal, in the PowerShell session of the **Cloud Shell**, run the following to verify the registration status of the **Microsoft.Compute** resource provider:

```
Get-AzResourceProvider -ListAvailable | Where-Object {$_.ProviderNamespace -
eq 'Microsoft.Compute'}
```

> **Note**: Verify that the status is listed as **Registered**. If not, wait a few minutes and repeat this step.

6. In the Azure portal, in the PowerShell session of the **Cloud Shell**, run the following to set the location for the next commands (replace the `<Azure_region>` placeholder with the name of the Azure region that you intend to use for this lab, such as, for example, `eastus`):

```
$location = '<Azure_region>'
```

7. In the Azure portal, in the PowerShell session of the **Cloud Shell**, run the following to identify the current usage of vCPUs and the corresponding limits for the **StandardDSv3Family** and **StandardBSFamily** Azure VMs:

```
Get-AzVMUsage -Location $location | Where-Object {$_.Name.Value -eq
'StandardDSv3Family'}
```

> **Note**: To identify the names of Azure regions, in the **Cloud Shell**, at the PowerShell prompt, run `(Get-AzLocation).Location`.

8. Review the output of the command executed in the previous step and ensure that you have at least **30** available vCPUs in the **Standard DSv3 Family vCPUs** of Azure VMs in the target Azure region. If that's

already the case, proceed directly to the next exercise. Otherwise, proceed to the next task of this exercise.

Task 2: Request vCPU quota increase

1. In the Azure portal, search for and select **Subscriptions** and, from the **Subscriptions** blade, select the entry representing the Azure subscription you intend to use for this lab.
2. In the Azure portal, on the subscription blade, in the vertical menu on the left side, in the **Settings** section, select **Usage + quotas**.

   Note: You might not need to raise a support ticket to increase quotas.

   Note: Requesting quota increase requires signing-in with multi-factor authentication (MFA). If you need to configure your account with MFA, refer to Plan an Azure Active Directory Multi-Factor Authentication deployment.
3. On the **Azure Pass – Sponsorship | Usage + quotas** blade, select **Region**, in the drop down list, select the checkbox next to the name of the Azure region you intend to use for this lab, select **Apply**, ensure that the **Compute** entry appears in the drop down list to the left of the **Region** entry, and, in the search box, type **Standard DSv3**.
4. In the list of results, select the checkbox next to the **Standard DSv3 Family vCPUs** item, select the **Request quota increase** entry in the toolbar, and, in the drop down list, select **Enter a new limit**.
5. In the **Request quota increase** pane, in the **New limit** column text box, type **30**, and then select **Submit**.
6. If prompted, on the **Request quota increase** pane, select **Authenticate with Multi-Factory Authentication** and follow the prompts to authenticate.
7. Allow the quota request to complete. After a few moments, the **Quota Details** blade will specify the request has been approved and Quota increased. Close the **Quota Details** blade.

   > **Note**: Depending on the choice of the Azure region and the current demand, it might be necessary to raise a support request. For instructions regarding the process of creating support request, refer to Create an Azure support request.

Exercise 1: Deploy an Active Directory Domain Services (AD DS) domain

The main tasks for this exercise are as follows:

1. Prepare for an Azure VM deployment
2. Deploy an Azure VM running an AD DS domain controller by using an Azure Resource Manager QuickStart template
3. Deploy an Azure VM running Windows 10 by using an Azure Resource Manager QuickStart template
4. Deploy Azure Bastion

Task 1: Prepare 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, use the **Search resources, services, and docs** text box at the top of the Azure portal page to search for and navigate to the **Microsoft Entra ID** blade.
3. From the **Overview** blade of the Microsoft Entra tenant and, in the vertical menu on the left side, in the **Manage** section, click **Properties**.
4. On the **Properties** blade of your Microsoft Entra tenant, at the very bottom of the blade, select the **Manage Security defaults** link.
5. On the **Enable Security defaults** blade, if needed, select **Disabled (not recommended)**, select the **My organization is planning to use Conditional Access** option button, and select **Save**, and then select **Disable**.
6. In the Azure portal, open **Cloud Shell** pane by selecting on the toolbar icon directly to the right of the search textbox.
7. 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**.

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

1. On the lab computer, in the web browser displaying the Azure portal, from the PowerShell session in the Cloud Shell pane, run the following to create a resource group (replace the `<Azure_region>` placeholder with the name of the Azure region that you intend to use for this lab, such as, for example, `eastus`):

```
$location = '<Azure_region>'
$resourceGroupName = 'az140-11-RG'
New-AzResourceGroup -Location $location -Name $resourceGroupName
```

2. In the Azure portal, close the **Cloud Shell** pane.
3. From your lab computer, in the same web browser window, open another web browser tab and navigate a customized version of QuickStart template named Create a new Windows VM and create a new AD Forest, Domain and DC.
4. On the **Create a new Windows VM and create a new AD Forest, Domain and DC** page, scroll down the page and 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.
5. On the **Create an Azure VM with a new AD Forest** blade, select **Edit parameters**.

6. On the **Edit parameters** blade, select **Load file**, in the **Open** dialog box, select **\\AZ-140\AllFiles\Labs\01\az140-11_azuredeploydc11.parameters.json**, select **Open**, and then select **Save**.

7. 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	**az140-11-RG**
Domain name	**adatum.com**

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

> **Note**: Wait for the deployment to complete before you proceed to the next exercise. The deployment might take 20-25 minutes.

Task 3: Deploy an Azure VM running Windows 10 by using an Azure Resource Manager QuickStart template

1. On the lab computer, in the web browser displaying the Azure portal, open a PowerShell session in the Cloud Shell pane, and run the following to add a subnet named **cl-Subnet** to the virtual network named **az140-adds-vnet11** you created in the previous task:

```
$resourceGroupName = 'az140-11-RG'
$vnet = Get-AzVirtualNetwork -ResourceGroupName $resourceGroupName -Name
'az140-adds-vnet11'
$subnetConfig = Add-AzVirtualNetworkSubnetConfig `
  -Name 'cl-Subnet' `
  -AddressPrefix 10.0.255.0/24 `
  -VirtualNetwork $vnet
$vnet | Set-AzVirtualNetwork
```

2. 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 files **\\AZ-140\AllFiles\Labs\01\az140-11_azuredeploycl11.json** and **\\AZ-140\AllFiles\Labs\01\az140-11_azuredeploycl11.parameters.json** into the Cloud Shell home directory.

3. From the PowerShell session in the Cloud Shell pane, run the following to deploy an Azure VM running Windows 10 that will serve as a client into the newly created subnet:

```
$location = (Get-AzResourceGroup -ResourceGroupName
$resourceGroupName).Location
New-AzResourceGroupDeployment `
  -ResourceGroupName $resourceGroupName `
  -Location $location `
```

```
    -Name az140lab0101vmDeployment `
    -TemplateFile $HOME/az140-11_azuredeploycl11.json `
    -TemplateParameterFile $HOME/az140-11_azuredeploycl11.parameters.json
```

> **Note**: Do not wait for the deployment to complete but instead proceed to the next task. The deployment might take about 10 minutes.

Task 4: Deploy Azure Bastion

> **Note**: Azure Bastion allows for connection to the Azure VMs without public endpoints which you deployed in the previous task of this exercise, while providing protection against brute force exploits that target operating system level credentials.

> **Note**: Ensure that your browser has the pop-up functionality enabled.

1. In the browser window displaying the Azure portal, open another tab and, in the browser tab, navigate to the Azure portal.

2. In the Azure portal, open **Cloud Shell** pane by selecting on the toolbar icon directly to the right of the search textbox.

3. From the PowerShell session in the Cloud Shell pane, run the following to add a subnet named **AzureBastionSubnet** to the virtual network named **az140-adds-vnet11** you created earlier in this exercise:

```
$resourceGroupName = 'az140-11-RG'
$vnet = Get-AzVirtualNetwork -ResourceGroupName $resourceGroupName -Name
'az140-adds-vnet11'
$subnetConfig = Add-AzVirtualNetworkSubnetConfig `
  -Name 'AzureBastionSubnet' `
  -AddressPrefix 10.0.254.0/24 `
  -VirtualNetwork $vnet
$vnet | Set-AzVirtualNetwork
```

4. Close the Cloud Shell pane.

5. In the Azure portal, search for and select **Bastions** and, from the **Bastions** blade, select **+ Create**.

6. On the **Basic** tab of the **Create a Bastion** blade, specify the following settings and select **Review + create**:

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	**az140-11-RG**
Name	**az140-11-bastion**

Setting	Value
Region	the same Azure region to which you deployed the resources in the previous tasks of this exercise
Tier	**Basic**
Virtual network	**az140-adds-vnet11**
Subnet	**AzureBastionSubnet (10.0.254.0/24)**
Public IP address	**Create new**
Public IP name	**az140-adds-vnet11-ip**

7. On the **Review + create** tab of the **Create a Bastion** blade, select **Create**:

> **Note**: Wait for the deployment to complete before you proceed to the next exercise. The deployment might take about 10 minutes.

Exercise 2: Integrate an AD DS forest with a Microsoft Entra tenant

The main tasks for this exercise are as follows:

1. Create AD DS users and groups that will be synchronized to Microsoft Entra
2. Configure AD DS UPN suffix
3. Create a Microsoft Entra user that will be used to configure synchronization with Microsoft Entra
4. Install Microsoft Entra Connect

Task 1: Create AD DS users and groups that will be synchronized to Microsoft Entra

1. On the lab computer, in the web browser displaying the Azure portal, search for and select **Virtual machines** and, from the **Virtual machines** blade, select **az140-dc-vm11**.
2. On the **az140-dc-vm11** blade, select **Connect**, in the drop-down menu, select **Connect via Bastion**.
3. When prompted, provide the following credentials and select **Connect**:

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

4. Within the Bastion session to **az140-dc-vm11**, start **Windows PowerShell ISE** as administrator.
5. From the **Administrator: Windows PowerShell ISE** script pane, run the following to disable Internet Explorer Enhanced Security for Administrators:

```
$adminRegEntry = 'HKLM:\SOFTWARE\Microsoft\Active Setup\Installed
Components\{A509B1A7-37EF-4b3f-8CFC-4F3A74704073}'
Set-ItemProperty -Path $AdminRegEntry -Name 'IsInstalled' -Value 0
Stop-Process -Name Explorer
```

6. From the **Administrator: Windows PowerShell ISE** console, run the following to create an AD DS organizational unit that will contain objects included in the scope of synchronization to the Microsoft Entra tenant used in this lab:

```
New-ADOrganizationalUnit 'ToSync' -path 'DC=adatum,DC=com' -
ProtectedFromAccidentalDeletion $false
```

7. From the **Administrator: Windows PowerShell ISE** console, run the following to create an AD DS organizational unit that will contain computer objects of Windows 10 domain-joined client computers:

```
New-ADOrganizationalUnit 'WVDClients' -path 'DC=adatum,DC=com' -
ProtectedFromAccidentalDeletion $false
```

8. From the **Administrator: Windows PowerShell ISE** script pane, run the following to create AD DS user accounts that will be synchronized to the Microsoft Entra tenant used in this lab (replace both `<password>` placeholders with random, complex passwords):

> **Note**: Ensure that you record the passwords used. You will need them later in this and subsequent labs.

```
$ouName = 'ToSync'
$ouPath = "OU=$ouName,DC=adatum,DC=com"
$adUserNamePrefix = 'aduser'
$adUPNSuffix = 'adatum.com'
$userCount = 1..9
foreach ($counter in $userCount) {
  New-AdUser -Name $adUserNamePrefix$counter -Path $ouPath -Enabled $True `
    -ChangePasswordAtLogon $false -userPrincipalName
$adUserNamePrefix$counter@$adUPNSuffix `
    -AccountPassword (ConvertTo-SecureString '<password>' -AsPlainText -
Force) -passThru
}

$adUserNamePrefix = 'wvdadmin1'
$adUPNSuffix = 'adatum.com'
New-AdUser -Name $adUserNamePrefix -Path $ouPath -Enabled $True `
    -ChangePasswordAtLogon $false -userPrincipalName
$adUserNamePrefix@$adUPNSuffix `
    -AccountPassword (ConvertTo-SecureString '<password>' -AsPlainText -
Force) -passThru
```

```
Get-ADGroup -Identity 'Domain Admins' | Add-AdGroupMember -Members
'wvdadmin1'
```

> **Note**: The script creates nine non-privileged user accounts named **aduser1** - **aduser9** and one privileged account that is a member of the **ADATUM\Domain Admins** group named **wvdadmin1**.

9. From the **Administrator: Windows PowerShell ISE** script pane, run the following to create AD DS group objects that will be synchronized to the Microsoft Entra tenant used in this lab:

```
New-ADGroup -Name 'az140-wvd-pooled' -GroupScope 'Global' -GroupCategory
Security -Path $ouPath
New-ADGroup -Name 'az140-wvd-remote-app' -GroupScope 'Global' -GroupCategory
Security -Path $ouPath
New-ADGroup -Name 'az140-wvd-personal' -GroupScope 'Global' -GroupCategory
Security -Path $ouPath
New-ADGroup -Name 'az140-wvd-users' -GroupScope 'Global' -GroupCategory
Security -Path $ouPath
New-ADGroup -Name 'az140-wvd-admins' -GroupScope 'Global' -GroupCategory
Security -Path $ouPath
```

10. From the **Administrator: Windows PowerShell ISE** console, run the following to add members to the groups you created in the previous step:

```
Get-ADGroup -Identity 'az140-wvd-pooled' | Add-AdGroupMember -Members
'aduser1','aduser2','aduser3','aduser4'
Get-ADGroup -Identity 'az140-wvd-remote-app' | Add-AdGroupMember -Members
'aduser1','aduser5','aduser6'
Get-ADGroup -Identity 'az140-wvd-personal' | Add-AdGroupMember -Members
'aduser7','aduser8','aduser9'
Get-ADGroup -Identity 'az140-wvd-users' | Add-AdGroupMember -Members
'aduser1','aduser2','aduser3','aduser4','aduser5','aduser6','aduser7','aduse
r8','aduser9'
Get-ADGroup -Identity 'az140-wvd-admins' | Add-AdGroupMember -Members
'wvdadmin1'
```

Task 2: Configure AD DS UPN suffix

1. Within the Bastion session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to install the latest version of the PowerShellGet module (select **Yes** when prompted for confirmation):

```
[Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12
Install-Module -Name PowerShellGet -Force -SkipPublisherCheck
```

2. From the **Administrator: Windows PowerShell ISE** console, run the following to install the latest version of the Az PowerShell module (select **Yes to All** when prompted for confirmation):

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

> **Note**: You may need to wait 3-5 minutes before any output from the installation of the Az module appears. You may also need to wait a further 5 minutes **after** output has stopped. This is expected behavior.

3. From the **Administrator: Windows PowerShell ISE** console, run the following to disable Windows Account Manager:

```
Update-AzConfig -EnableLoginByWam $false
```

4. From the **Administrator: Windows PowerShell ISE** console, run the following to sign in to your Azure subscription:

```
Connect-AzAccount
```

5. When prompted, provide the credentials of an Entra ID user account with the Owner role in the subscription you are using in this lab.

6. From the **Administrator: Windows PowerShell ISE** console, run the following to retrieve the Id property of the Microsoft Entra tenant associated with your Azure subscription:

```
$tenantId = (Get-AzContext).Tenant.Id
```

7. From the **Administrator: Windows PowerShell ISE** console, run the following to install and import the latest version of the Azure AD PowerShell module:

```
Install-Module -Name AzureAD -Force
Import-Module -Name AzureAD
```

8. From the **Administrator: Windows PowerShell ISE** console, run the following to authenticate to your Microsoft Entra tenant:

```
Connect-AzureAD -TenantId $tenantId
```

9. When prompted, sign in with the same credentials you used earlier in this task (the user account with the Owner role in the subscription you are using in this lab).

10. From the **Administrator: Windows PowerShell ISE** console, run the following to retrieve the primary DNS domain name of the Microsoft Entra tenant associated with your Azure subscription:

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

11. From the **Administrator: Windows PowerShell ISE** console, run the following to add the primary DNS domain name of the Microsoft Entra tenant associated with your Azure subscription to the list of UPN suffixes of your AD DS forest:

```
Get-ADForest|Set-ADForest -UPNSuffixes @{add="$aadDomainName"}
```

12. From the **Administrator: Windows PowerShell ISE** script pane, run the following to assign the primary DNS domain name of the Microsoft Entra tenant associated with your Azure subscription as the UPN suffix of all users in the AD DS domain:

```
$domainUsers = Get-ADUser -Filter {UserPrincipalName -like '*adatum.com'} -
Properties userPrincipalName -ResultSetSize $null
$domainUsers | foreach {$newUpn =
$_.UserPrincipalName.Replace('adatum.com',$aadDomainName); $_ | Set-ADUser -
UserPrincipalName $newUpn}
```

13. From the **Administrator: Windows PowerShell ISE** console, run the following to assign the **adatum.com** UPN suffix back to the **Student** domain user:

```
$domainAdminUser = Get-ADUser -Filter {sAMAccountName -eq 'Student'} -
Properties userPrincipalName
$domainAdminUser | Set-ADUser -UserPrincipalName 'student@adatum.com'
```

Task 3: Create a Microsoft Entra user that will be used to configure directory synchronization

1. Within the Bastion session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to create a new Microsoft Entra user (replace the `<password>` placeholder with a random, complex password):

> **Note**: Ensure that you record the password you used. **You will need it later in this and subsequent labs**.

```
$userName = 'aadsyncuser'
$passwordProfile = New-Object -TypeName
```

```
Microsoft.Open.AzureAD.Model.PasswordProfile
$passwordProfile.Password = '<password>'
$passwordProfile.ForceChangePasswordNextLogin = $false
New-AzureADUser -AccountEnabled $true -DisplayName $userName -
PasswordProfile $passwordProfile -MailNickName $userName -UserPrincipalName
"$userName@$aadDomainName"
```

2. From the **Administrator: Windows PowerShell ISE** script pane, run the following to assign the Global Administrator role to the newly created Microsoft Entra user:

```
$aadUser = Get-AzureADUser -ObjectId "$userName@$aadDomainName"
$aadRole = Get-AzureADDirectoryRole | Where-Object {$_.displayName -eq
'Global administrator'}
Add-AzureADDirectoryRoleMember -ObjectId $aadRole.ObjectId -RefObjectId
$aadUser.ObjectId
```

3. From the **Administrator: Windows PowerShell ISE** script pane, run the following to identify the user principal name of the newly created Microsoft Entra user:

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

> **Note**: Record the user principal name **and** password. You will need it later in this exercise.

Task 4: Install Microsoft Entra Connect

1. Within the Bastion session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to eanble TLS 1.2:

```
New-Item 'HKLM:\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319' -
Force | Out-Null
New-ItemProperty -path
'HKLM:\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319' -name
'SystemDefaultTlsVersions' -value '1' -PropertyType 'DWord' -Force | Out-
Null
New-ItemProperty -path
'HKLM:\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319' -name
'SchUseStrongCrypto' -value '1' -PropertyType 'DWord' -Force | Out-Null
New-Item 'HKLM:\SOFTWARE\Microsoft\.NETFramework\v4.0.30319' -Force | Out-
Null
New-ItemProperty -path 'HKLM:\SOFTWARE\Microsoft\.NETFramework\v4.0.30319' -
name 'SystemDefaultTlsVersions' -value '1' -PropertyType 'DWord' -Force |
Out-Null
New-ItemProperty -path 'HKLM:\SOFTWARE\Microsoft\.NETFramework\v4.0.30319' -
name 'SchUseStrongCrypto' -value '1' -PropertyType 'DWord' -Force | Out-Null
New-Item
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols
```

```
\TLS 1.2\Server' -Force | Out-Null
New-ItemProperty -path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols
\TLS 1.2\Server' -name 'Enabled' -value '1' -PropertyType 'DWord' -Force |
Out-Null
New-ItemProperty -path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols
\TLS 1.2\Server' -name 'DisabledByDefault' -value 0 -PropertyType 'DWord' -
Force | Out-Null
New-Item
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols
\TLS 1.2\Client' -Force | Out-Null
New-ItemProperty -path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols
\TLS 1.2\Client' -name 'Enabled' -value '1' -PropertyType 'DWord' -Force |
Out-Null
New-ItemProperty -path
'HKLM:\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols
\TLS 1.2\Client' -name 'DisabledByDefault' -value 0 -PropertyType 'DWord' -
Force | Out-Null
Write-Host 'TLS 1.2 has been enabled.'
```

2. Within the Bastion session to **az140-dc-vm11**, start Internet Explorer and navigate to the Microsoft Edge for Business download page.

3. From the Microsoft Edge for Business download page download the latest stable version of Microsoft Edge, install it, launch it, and configure it with the default settings.

4. Within the Remote Desktop session to **az140-dc-vm11**, use Microsoft Edge to navigate to the Azure portal. If prompted, sign in by using the Microsoft Entra credentials of the user account with the Owner role in the subscription you are using in this lab.

5. In the Azure portal, use the **Search resources, services, and docs** text box at the top of the Azure portal page to search for and navigate to the **Microsoft Entra ID** blade and, on your Microsoft Entra tenant blade, in the **Manage** section of the hub menu, select **Microsoft Entra Connect**.

6. On the **Microsoft Entra Connect** blade, select the **Connect Sync** link from the service menu and then select the **Download Microsoft Entra Connect** link. This will automatically open a new browser tab displaying the **Microsoft Entra Connect** download page.

7. On the **Microsoft Entra Connect** download page, select **Download**.

8. If prompted whether to run or save the **AzureADConnect.msi** installer, select **Run**. Otherwise, open the file after it downloads to start the **Microsoft Azure Active Directory Connect** wizard.

9. 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**.

10. On the **Express Settings** page of the **Microsoft Azure Active Directory Connect** wizard, select the **Customize** option.

11. On the **Install required components** page, leave all optional configuration options deselected and select **Install**.

12. On the **User sign-in** page, ensure that only the **Password Hash Synchronization** is enabled and select **Next**.

13. On the **Connect to Azure AD** page, authenticate by using the credentials of the **aadsyncuser** user account you created in the previous exercise and select **Next**.

    > **Note**: Provide the userPrincipalName attribute of the **aadsyncuser** account you recorded earlier in this exercise and specify the password you set earlier in this lab as its password.

14. On the **Connect your directories** page, select the **Add Directory** button to the right of the **adatum.com** forest entry.

15. 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**

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

17. 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**: This is expected, since the Microsoft Entra tenant does not have a verified custom DNS domain matching one of the UPN suffixes of the **adatum.com** AD DS.

18. On the **Domain and OU filtering** page, select the option **Sync selected domains and OUs**, expand the adatum.com node, clear all checkboxes, select only the checkbox next to the **ToSync** OU, and select **Next**.

19. On the **Uniquely identifying your users** page, accept the default settings, and select **Next**.

20. On the **Filter users and devices** page, accept the default settings, and select **Next**.

21. On the **Optional features** page, accept the default settings, and select **Next**.

22. 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 5 minutes.

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

24. Within the Remote Desktop session to **az140-dc-vm11**, in the Microsoft Edge window displaying the Azure portal, navigate to the **Users - All users** blade of the Adatum Lab Microsoft Entra tenant.

25. On the **Users | All users** blade, note that the list of user objects includes the listing of AD DS user accounts you created earlier in this lab, with the **Yes** entry appearing in the **On-premises sync enabled** column.

    > **Note**: You might have to wait a few minutes and refresh the browser page for the AD DS user accounts to appear.