Lab - Package Azure Virtual Desktop applications (AD DS)

Student lab manual

Lab dependencies

- An Azure subscription
- A Microsoft account or a Microsoft Entra account with the Global Administrator role in the Microsoft Entra tenant associated with the Azure subscription and with the Owner or Contributor role in the Azure subscription
- The completed lab **Prepare for deployment of Azure Virtual Desktop (AD DS)**
- The completed lab **Configure Conditional Access policies for AVD (AD DS)**
- The completed lab **Implement and Manage AVD Profiles (AD DS)**

Estimated Time

60 minutes

Lab scenario

You need to package and deploy Azure Virtual Desktop applications in an Active Directory Domain Services (AD DS) environment.

Objectives

After completing this lab, you will be able to:

- Prepare for and create MSIX app packages
- Implement an MSIX app attach image for Azure Virtual Desktop in an AD DS environment
- Implement the MSIX app attach on Azure Virtual Desktop in an AD DS environment

Lab files

- \\AZ-140\AllFiles\Labs\04\az140-42_azuredeploycl42.json
- \\AZ-140\AllFiles\Labs\04\az140-42_azuredeploycl42.parameters.json

Instructions

> **Important**: Microsoft renamed **Azure Active Directory** (**Azure AD**) to **Microsoft Entra ID**. For details regarding this change, refer to New name for Azure Active Directory. This is an ongoing effort, so you might still encounter instances where there is a mismatch between the lab instruction and the elements of the interface as you step through individual exercises. Take this into considerations (in particular, in this lab, the **Microsoft Entra Connect** designates the new name of **Azure Active Directory Connect**).

Exercise 1: Prepare for and create MSIX app packages

The main tasks for this exercise are as follows:

1. Prepare for configuration of Azure Virtual Desktop session hosts
2. Deploy an Azure VM running Windows 10 by using an Azure Resource Manager QuickStart template
3. Prepare the Azure VM running Windows 10 for MSIX packaging
4. Generate a signing certificate
5. Download software to package
6. Install the MSIX Packaging Tool
7. Create an MSIX package

Task 1: Prepare for configuration of Azure Virtual Desktop session hosts

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. On the lab computer and, in the web browser window displaying the Azure portal, open the **PowerShell** shell session within the **Cloud Shell** pane.

3. From the PowerShell session in the Cloud Shell pane, run the following to start the Azure Virtual Desktop session host Azure VMs you will be using in this lab:

```
Get-AzVM -ResourceGroup 'az140-21-RG' | Start-AzVM -NoWait
```

> **Note**: The command executes asynchronously (as determined by the -NoWait parameter), so while you will be able to run another PowerShell command immediately afterwards within the same PowerShell session, it will take a few minutes before the Azure VMs are actually started.

> **Note**: If you enabled PSRemoting on the session hosts in the az140-21-RG resource group in the first task of the previous lab (Implement and manage AVD profiles) then you may proceed directly to the next task without waiting for the Azure VMs to start. If you have not previously enabled PSRemoting on the session hosts in the az140-21-RG resource group, wait for the VMs to start and then run the following command.

4. From the PowerShell session of the **Cloud Shell**, run the following to enable PowerShell Remoting on the session hosts.

```
Get-AzVM -ResourceGroup 'az140-21-RG' | Enable-AzVMPSRemoting
```

Task 2: Deploy an Azure VM running Windows 10 by using an Azure Resource Manager QuickStart template

1. From your lab computer, in the web browser window displaying 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\04\az140-42_azuredeploycl42.json** and **\\AZ-140\AllFiles\Labs\04\az140-42_azuredeploycl42.parameters.json** into the Cloud Shell home directory.

2. From the PowerShell session in the Cloud Shell pane, run the following to deploy an Azure VM running Windows 10 that you will use for creating MSIX packages to and to join it to the Microsoft Entra DS domain:

```
$vNetResourceGroupName = 'az140-11-RG'
$location = (Get-AzResourceGroup -ResourceGroupName
$vNetResourceGroupName).Location
$resourceGroupName = 'az140-42-RG'
New-AzResourceGroup -ResourceGroupName $resourceGroupName -Location
$location
New-AzResourceGroupDeployment `
  -ResourceGroupName $resourceGroupName `
  -Location $location `
  -Name az140lab0402vmDeployment `
  -TemplateFile $HOME/az140-42_azuredeploycl42.json `
  -TemplateParameterFile $HOME/az140-42_azuredeploycl42.parameters.json
```

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

Task 3: Prepare the Azure VM running Windows 10 for MSIX packaging

1. From your lab computer, in the Azure portal, search for and select **Virtual machines** and, from the **Virtual machines** blade, in the list of virtual machines, select the **az140-cl-vm42** entry. This will open the **az140-cl-vm42** blade.

2. On the **az140-cl-vm42** blade, select **Connect**, in the drop-down menu, select **Connect via Bastion**.

3. When prompted, sign in with the **wvdadmin1@adatum.com** user name and the password you set when creating this user account.

4. Within the Bastion session to **az140-cl-vm42**, start **Windows PowerShell ISE** as administrator, from the **Administrator: Windows PowerShell ISE** console, run the following to prepare the operating system for MSIX packaging:

```
Schtasks /Change /Tn "\Microsoft\Windows\WindowsUpdate\Scheduled Start"
/Disable
reg add HKLM\Software\Policies\Microsoft\WindowsStore /v AutoDownload /t
REG_DWORD /d 0 /f
reg add
HKCU\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager /v
PreInstalledAppsEnabled /t REG_DWORD /d 0 /f
reg add
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager\Debug
/v ContentDeliveryAllowedOverride /t REG_DWORD /d 0x2 /f
reg add HKLM\Software\Microsoft\RDInfraAgent\MSIXAppAttach /v
PackageListCheckIntervalMinutes /t REG_DWORD /d 1 /f
reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System /v
EnableLUA /t REG_DWORD /d 0 /f
```

> **Note**: The last of these registry changes disables User Access Control. This is technically not required but simplifies the process illustrated in this lab.

Task 4: Generate a signing certificate

> **Note**: In this lab, you will use a self-signed certificate. In a production environment, you should be using a certificate issued by either a public Certification Authority or an internal one, depending on the intended use.

1. Within the Bastion session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** console, run the following to generate a self-signed certificate with the Common Name attribute set to **Adatum**, and store the certificate in the **Personal** folder of the **Local Machine** certificate store:

```
New-SelfSignedCertificate -Type Custom -Subject "CN=Adatum" -KeyUsage
DigitalSignature -KeyAlgorithm RSA -KeyLength 2048 -CertStoreLocation
"cert:\LocalMachine\My"
```

2. From the **Administrator: Windows PowerShell ISE** console, run the following to start the **Certificates** console targeting the Local Machine certificate store:

```
certlm.msc
```

3. In the **Certificates** console pane, expand the **Personal** folder, select the **Certificates** subfolder, right-click the **Adatum** certificate, in the right-click menu, select **All Tasks** followed by **Export**. This will launch the **Certificate Export Wizard**.

4. On the **Welcome to the Certificate Export Wizard** page of the **Certificate Export Wizard**, select **Next**.

5. On the **Export Private Key** page of the **Certificate Export Wizard**, select the option **Yes, export the private key** option and select **Next**.

6. On the **Export File Format** page of the **Certificate Export Wizard**, select the checkbox **Export all extended properties**, clear the checkbox **Enable certificate privacy**, and select **Next**.

7. On the **Security** page of the **Certificate Export Wizard**, select the **Password** checkbox, in the textboxes below, type **Pa55w.rd1234**, and select **Next**.

8. On the **File to Export** page of the **Certificate Export Wizard**, in the **File name** textbox, select **Browse**, in the **Save As** dialog box, navigate to the **C:\Allfiles\Labs\04** folder (create the folder first), in the **File name** textbox, type **adatum.pfx**, and select **Save**.

9. Back on the **File to Export** page of the **Certificate Export Wizard**, ensure that the textbox contains the entry **C:\Allfiles\Labs\04\adatum.pfx**, and select **Next**.

10. On the **Completing Certificate Export Wizard** page of the **Certificate Export Wizard**, select **Finish**, and select **OK** to acknowledge successful export.

    > **Note**: Since you are using a self-signed certificate, you need to install it in the **Trusted People** certificate store on the target session hosts.

11. From the **Administrator: Windows PowerShell ISE** console, run the following to install the newly generated certificate in the **Trusted People** certificate store on the target session hosts:

```
$wvdhosts = 'az140-21-p1-0','az140-21-p1-1','az140-21-p1-2'
$cleartextPassword = 'Pa55w.rd1234'
$securePassword = ConvertTo-SecureString $cleartextPassword -AsPlainText -
Force
$localPath = 'C:\Allfiles\Labs\04'
ForEach ($wvdhost in $wvdhosts){
   $remotePath = "\\$wvdhost\C$\Allfiles\Labs\04\"
   New-Item -ItemType Directory -Path $remotePath -Force
   Copy-Item -Path "$localPath\adatum.pfx" -Destination $remotePath -Force
   Invoke-Command -ComputerName $wvdhost -ScriptBlock {
      Import-PFXCertificate -CertStoreLocation
Cert:\LocalMachine\TrustedPeople -FilePath 'C:\Allfiles\Labs\04\adatum.pfx'
-Password $using:securePassword
   }
}
```

Task 5: Download software to package

1. Within the Bastion session to **az140-cl-vm42**, start **Microsoft Edge** and browse to **https://github.com/microsoft/XmlNotepad**.
2. On the **microsoft/XmlNotepad readme.md** page, select the download link for Standalone downloadable installer and download the compressed installation files.
3. Within the Bastion session to **az140-cl-vm42**, start File Explorer, navigate to the **Downloads** folder, open the compressed file, copy the content from within the folder in the compressed file, and paste it to the **C:\AllFiles\Labs\04** directory.

Task 6: Install the MSIX Packaging Tool

1. Within the Bastion session to **az140-cl-vm42**, start the **Microsoft Store** app.
2. In the **Microsoft Store** app, search for and select **MSIX Packaging Tool**, on the **MSIX Packaging Tool** page, select **Get**.
3. When prompted, skip signing in, wait for the installation to complete, select **Launch** and, in the **Send diagnostic data** dialog box, select **Decline**,

Task 7: Create an MSIX package

1. Within the Bastion session to **az140-cl-vm42**, switch to the **Administrator: Windows PowerShell ISE** window and, from the **Administrator: Windows PowerShell ISE** script pane, run the following to disable the Windows Search service:

```
$serviceName = 'wsearch'
Set-Service -Name $serviceName -StartupType Disabled
Stop-Service -Name $serviceName
```

2. Within the Bastion session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** console, run the following to create the folder that will host the MSIX package:

```
New-Item -ItemType Directory -Path 'C:\AllFiles\Labs\04\XmlNotepad' -Force
```

3. Within the Bastion session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** console, run the following to remove the Zone.Identifier alternate data stream from the extracted installer files, which has a value of "3" to indicate that they were downloaded from the Internet:

```
Get-ChildItem -Path 'C:\AllFiles\Labs\04' -Recurse -File | Unblock-File
```

4. Within the Bastion session to **az140-cl-vm42**, switch to the **MSIX Packaging Tool** interface, on the **Select task** page, select **Application package - Create your app package** entry. This will start the **Create new package** wizard.

5. On the **Select environment** page of the **Create new package** wizard, ensure that the **Create package on this computer** option is selected, select **Next**, and wait for the installation of the **MSIX Packaging Tool Driver**.

> **Note**: The installation of the MSIX Packaging Tool Driver will take 5-10. The status column will initially say **Checking** and, once installed, will say **Installed**.

6. On the **Prepare computer** page of the **Create new package** wizard, review the recommendations. If there is a pending reboot, restart the operating system, sign back in by using the **wvdadmin1@adatum.com** account, and restart the **MSIX Packaging Tool** before you proceed.

> **Note**: MSIX Packaging Tool disables temporarily Windows Update and Windows Search. In this case, the Windows Search service is already disabled.

7. On the **Prepare computer** page of the **Create new package** wizard, click **Next**.

8. On the **Select installer** page of the **Create new package** wizard, next to the **Choose the installer you want to package** text box, select **Browse**, in the **Open** dialog box, browse to the **C:\AllFiles\Labs\04** folder, select **XmlNotepadSetup.msi**, and click **Open**,

9. On the **Select installer** page of the **Create new package** wizard, in the **Signing preference** drop-down list, select the **Sign with a certificate (.pfx)** entry, next to the **Browse for certificate** textbox, select **Browse**, in the **Open** dialog box, navigate to the **C:\AllFiles\Labs\04** folder, select the **adatum.pfx** file, click **Open**, in the **Password** text box, type **Pa55w.rd1234**, and select **Next**.

10. On the **Package information** page of the **Create new package** wizard, review the package information, validate that the publisher name is set to **CN=Adatum**, and select **Next**.
11. On the **Choose the Accelerator for applying to the package** page, select **Next**. This will trigger installation of the downloaded software.
12. In the **XMLNotepad Setup** window, accept the terms in the License Agreement and select **Install** and, once the installation completes, select **Finish**.
13. In the **Installation** page of the **Create new package** wizard, select **Next**.
14. On the **Manage first launch tasks** page of the **Create new package** wizard, review the provided information and select **Next**.
15. When prompted **Are you done?**, select **Yes, move on**.
16. On the **Services report** page of the **Create new package** wizard, verify that no services are listed and select **Next**.
17. On the **Create package** page of the **Create new package** wizard, in the **Save location** textbox, type **C:\Allfiles\Labs\04\XmlNotepad\XmlNotepad.msix** and click **Create**.
18. In the **Package successfully created** dialog box, note the location of the saved package and select **Close**.
19. Switch to the File Explorer window, navigate to the **C:\Allfiles\Labs\04\XmlNotepad** folder and verify that it contains the *.msix and *.xml files.
20. Copy the **XmlNotepad.msix** file to the **C:\Allfiles\Labs\04** folder.

Exercise 2: Implement an MSIX app attach image for Azure Virtual Desktop in an AD DS environment

The main tasks for this exercise are as follows:

1. Enable Hyper-V on the Azure VMs running Window 11 Enterprise multi-session
2. Create an MSIX app attach image

Task 1: Enable Hyper-V on the Azure VMs running Window 11 Enterprise multi-session

1. Within the Bastion session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** console, run the following to prepare the target Azure Virtual Desktop hosts for MSIX app attach:

```
$wvdhosts = 'az140-21-p1-0','az140-21-p1-1','az140-21-p1-2'
ForEach ($wvdhost in $wvdhosts){
   Invoke-Command -ComputerName $wvdhost -ScriptBlock {
      Schtasks /Change /Tn "\Microsoft\Windows\WindowsUpdate\Scheduled
Start" /Disable
      reg add HKLM\Software\Policies\Microsoft\WindowsStore /v AutoDownload
/t REG_DWORD /d 0 /f
      reg add
HKCU\Software\Microsoft\Windows\CurrentVersion\ContentDeliveryManager /v
```

```
PreInstalledAppsEnabled /t REG_DWORD /d 0 /f
      reg add
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliveryManager\Debug
/v ContentDeliveryAllowedOverride /t REG_DWORD /d 0x2 /f
      reg add HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
/v EnableLUA /t REG_DWORD /d 0 /f
      reg add HKLM\Software\Microsoft\RDInfraAgent\MSIXAppAttach /v
PackageListCheckIntervalMinutes /t REG_DWORD /d 1 /f
      Set-Service -Name wuauserv -StartupType Disabled
   }
}
```

2. Within the Bastion session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** console, run the following to install Hyper-V and its management tools, including the Hyper-V PowerShell module on the Azure Virtual Desktop hosts:

```
$wvdhosts = 'az140-21-p1-0','az140-21-p1-1','az140-21-p1-2'
ForEach ($wvdhost in $wvdhosts){
   Invoke-Command -ComputerName $wvdhost -ScriptBlock {
      Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V-
All
   }
}
```

3. When prompted to restrart the target operating system, select **Yes**.

4. Within the Bastion session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** console, run the following to install Hyper-V and its management tools, including the Hyper-V PowerShell module on the local computer:

```
Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Hyper-V-All
```

5. Once the installation of the Hyper-V components completes, select **Yes** to restart the operating system. Following the restart, sign back in with the **wvdadmin1@adatum.com** user name and the password you set when creating this user account.

Task 2: Create an MSIX app attach image

1. Within the Bastion session to **az140-cl-vm42**, start **Microsoft Edge**, browse to **https://aka.ms/msixmgr**. This will automatically download the **msixmgr.zip** file (the MSIX mgr tool archive) into the **Downloads** folder.

2. In File Explorer, navigate to the **Downloads** folder, open the compressed file and copy the content of the **x64** folder (including the folder) to the **C:\AllFiles\Labs\04** folder.

3. Within the Bastion session to **az140-cl-vm42**, start **Windows PowerShell ISE** as administrator and, from the **Administrator: Windows PowerShell ISE** script pane, run the following to create the folder

that will store the MSIX app attach image:

```
New-Item -ItemType Directory -Path 'C:\Allfiles\Labs\04\MSIXVhds' -Force
```

4. From the **Administrator: Windows PowerShell ISE** script pane, run the following to create the VHD that will host the MSIX files and unpack into it the MSIX package you created in the previous task:

```
$appName = 'XmlNotepad'
Set-Location -Path 'C:\AllFiles\Labs\04\x64'
.\msixmgr.exe -Unpack -packagePath ..\$appName.msix -destination
..\MSIXVhds\$appName.vhd -applyacls -create -filetype vhd -vhdSize 128 -
rootDirectory Apps
```

5. Within the Bastion session to **az140-cl-vm42**, in File Explorer, navigate to the **C:\AllFiles\Labs\04\MSIXVhds** folder and ensure you have a virtual disk called XmlNotepad.vhd.

Exercise 3: Implement MSIX app attach on Azure Virtual Desktop session hosts

The main tasks for this exercise are as follows:

1. Configure Active Directory groups containing Azure Virtual Desktop hosts
2. Set up the Azure Files share for MSIX app attach
3. Mount and register the MSIX App attach image on Azure Virtual Desktop session hosts
4. Publish MSIX apps to an application group
5. Validate the functionality of MSIX App attach

Task 1: Configure Active Directory groups containing Azure Virtual Desktop hosts

1. Switch to 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, provde 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 create an AD DS group object that will be synchronized to the Microsoft Entra tenant used in this lab:

```
$ouPath = "OU=WVDInfra,DC=adatum,DC=com"
New-ADGroup -Name 'az140-hosts-42-p1' -GroupScope 'Global' -GroupCategory
Security -Path $ouPath
```

> **Note**: You will use this group to grant Azure Virtual Desktop hosts permissions to the **az140-42-msixvhds** file share.

6. 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-hosts-42-p1' | Add-AdGroupMember -Members
'az140-21-p1-0$','az140-21-p1-1$','az140-21-p1-2$'
```

7. From the **Administrator: Windows PowerShell ISE** script pane, run the following to restart the servers which are members of the 'az140-hosts-42-p1' group:

```
$hosts = (Get-ADGroup -Identity 'az140-hosts-42-p1' | Get-ADGroupMember |
Select-Object Name).Name
$hosts | ForEach-Object {Restart-Computer -ComputerName $_ -Force}
```

> **Note**: This step ensures that the group membership change takes effect.

8. Within the Bastion session to **az140-dc-vm11**, in the **Start** menu, expand the **Microsoft Azure AD Connect** folder and select **Microsoft Azure AD Connect**.

9. On the **Welcome to Azure AD Connect** page of the **Microsoft Azure Active Directory Connect** window, select **Configure**.

10. On the **Additional tasks** page in the **Microsoft Entra Connect** window, select **Customize synchronization options** and select **Next**.

11. On the **Connect to Microsoft Entra** page in the **Microsoft Azure Active Directory Connect** window, authenticate by using the user principal name of the **aadsyncuser** user account you identified earlier in this task with the password you set when creating this user account.

12. On the **Connect your directories** page in the **Microsoft Azure Active Directory Connect** window, select **Next**.

13. On the **Domain and OU filtering** page in the **Microsoft Azure Active Directory Connect** window, ensure that the option **Sync selected domains and OUs** is selected, expand the **adatum.com** node, select the checkbox next to the **WVDInfra** OU (leave any other selected checkboxes unchanged), and select **Next**.

14. On the **Optional features** page in the **Microsoft Azure Active Directory Connect** window, accept the default settings, and select **Next**.

15. On the **Ready to configure** page in the **Microsoft Azure Active Directory Connect** window, ensure that the checkbox **Start the synchronization process when configuration completes** is selected and select **Configure**.

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

17. Within the Bastion session to **az140-dc-vm11**, start Microsoft Edge and navigate to the Azure portal. When prompted, sign in by using the Microsoft Entra credentials of the user account with the Global Administrator role in the Microsoft Entra tenant associated with the Azure subscription you are using in this lab.

18. Within the Bastion session to **az140-dc-vm11**, in the Microsoft Edge window displaying the Azure portal, search for and select **Microsoft Entra ID** to navigate to the Microsoft Entra tenant associated with the Azure subscription you are using for this lab.

19. On the Microsoft Entra ID blade, in the vertical menu bar on the left side, in the **Manage** section, click **Groups**.

20. On the **Groups | All groups** blade, in the list of groups, select the **az140-hosts-42-p1** entry.

    > **Note**: You might need to refresh the page for the group to be displayed.

21. On the **az140-hosts-42-p1** blade, in the vertical menu bar on the left side, in the **Manage** section, click **Members**.

22. On the **az140-hosts-42-p1 | Members** blade, verify that the list of **Direct members** include the three hosts of the Azure Virtual Desktop pool you added to the group earlier in this task.

Task 2: Set up the Azure Files share for MSIX app attach

1. On the lab computer, switch back to the Bastion session to **az140-cl-vm42**.

2. Within the Bastion session to **az140-cl-vm42**, start Microsoft Edge in the InPrivate mode, 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.

    > **Note**: Ensure to use the Microsoft Edge InPrivate mode.

3. Within the Bastion session to **az140-cl-vm42**, in the Microsoft Edge window displaying the Azure portal, search for and select **Storage accounts** and, on the **Storage accounts** blade, select the storage account you configured to host user profiles.

    > **Note**: This part of the lab is contingent on completing the lab **Implement and Manage Storage for AVD (AD DS)** or **Implement and Manage Storage for AVD (Microsoft Entra DS)**

    > **Note**: In production scenarios, you should consider using a separate storage account. This would require configuring that storage account for Microsoft Entra DS authentication, which you already implemented for the storage account hosting user profiles. You are using the same storage account to minimize duplicate steps across individual labs.

4. On the storage account blade, in the vertical menu on the left side, select **Access Control (IAM)**.

5. On the **Access Control (IAM)** blade of the storage account, select **+ Add** and, in the drop-down menu, select **Add role assignment**,

6. On the **Add role assignment** blade, on the **Role** tab, specify the following settings and select **Next**:

Setting	Value
Job function role	**Storage File Data SMB Share Contributor**

7. On the **Add role assignment** blade, on the **Members** tab, click **+ Select members**, specify the following settings and click **Select**.

Setting	Value
Select	**az140-wvd-users**

8. On the **Add role assignment** blade, select **Review + assign**, and then select **Review + assign**.

9. On the **Access Control (IAM)** blade of the storage account, select **+ Add** and, in the drop-down menu, select **Add role assignment**,

10. On the **Add role assignment** blade, on the **Role** tab, specify the following settings and select **Next**:

Setting	Value
Job function role	**Storage File Data SMB Share Elevated Contributor**

11. On the **Add role assignment** blade, on the **Members** tab, click **+ Select members**, specify the following settings and click **Select**.

Setting	Value
Select	**az140-wvd-admins**

12. On the **Add role assignment** blade, select **Review + assign**, and then select **Review + assign**.

13. On the **Access Control (IAM)** blade of the storage account, select **+ Add** and, in the drop-down menu, select **Add role assignment**,

14. On the **Add role assignment** blade, on the **Role** tab, specify the following settings and select **Next**:

Setting	Value
Job function role	**Storage File Data SMB Share Elevated Contributor**

15. On the **Add role assignment** blade, on the **Members** tab, click **+ Select members**, specify the following settings and click **Select**.

Setting	Value
Select	**az140-hosts-42-p1**

16. On the **Add role assignment** blade, select **Review + assign**, and then select **Review + assign**.

17. On the storage account blade, in the vertical menu on the left side, in the **Data storage** section, select **File shares** and then select **+ File share**.

18. On the **New file share** blade, specify the following settings and select **Next : Backup >** (leave other settings with their default values):

Setting	Value
Name	**az140-42-msixvhds**
Access tier	**Transaction optimized**

19. On the **Backup** blade, deselect the **Enable backup** checkbox, select **Review + Create**, wait for the validation process to complete, and then select **Create**.

20. In the Microsoft Edge displaying the Azure portal, in the list of file shares, select the newly created file share.

21. Within the Bastion session to **az140-cl-vm42**, start **Command Prompt** and, from the **Command Prompt** window, run the following to map a drive to the **az140-42-msixvhds** share (replace the `<storage-account-name>` placeholder with the name of the storage account) and verify that the command completes successfully:

```
net use Z: \\<storage-account-name>.file.core.windows.net\az140-42-msixvhds
```

22. Within the Bastion session to **az140-cl-vm42**, from the **Command Prompt** window, run the following to grant the required NTFS permissions to the computer accounts of session hosts:

```
icacls Z:\ /grant ADATUM\az140-hosts-42-p1:(OI)(CI)(RX) /T
icacls Z:\ /grant ADATUM\az140-wvd-users:(OI)(CI)(RX) /T
icacls Z:\ /grant ADATUM\az140-wvd-admins:(OI)(CI)(F) /T
```

> **Note**: You could also set these permissions by using File Explorer while signed in as **wvdadmin1@adatum.com**.

> **Note**: Next you will validate the functionality of MSIX App attach

23. Within the Bastion session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** window, run the following to copy the VHD file you created in the previous exercise to the Azure Files share you created earlier in this exercise:

```
New-Item -ItemType Directory -Path 'Z:\packages'
Copy-Item -Path 'C:\Allfiles\Labs\04\MSIXVhds\XmlNotepad.vhd' -Destination
'Z:\packages\' -Force
```

Task 3: Mount and register the MSIX App attach image on Azure Virtual Desktop session hosts

1. Within the Bastion session to **az140-cl-vm42**, in the Microsoft Edge window displaying the Azure portal, search for and select **Azure Virtual Desktop** and, on the **Azure Virtual Desktop** blade, in the vertical menu on the left side, in the **Manage** section, select **Host pools**.
2. On the **Azure Virtual Desktop | Host pools** blade, in the list of host pools, select the **az140-21-hp1** entry.
3. On the **az140-21-hp1 | Properties** blade, in the vertical menu on the left side, in the **Manage** section, select **MSIX packages**.
4. On the **az140-21-hp1 | MSIX packages** blade, click **+ Add**.
5. On the **Add MSIX package** blade, in the **MSIX image path** textbox, enter the path to the **XmlNotepad.vhd** file in the format `\\<storage-account-name>.file.core.windows.net\az140-42-msixvhds\packages\XmlNotepad.vhd` (replace the `<storage-account-name>` placeholder with the name of the storage account hosting the **az140-42-msixvhds** file share) and click **Add**.
6. On the **Add MSIX package** blade, specify the following settings and click **Add**:

Setting	Value
MSIX image path	**\\<storage-account-name>.file.core.windows.net\az140-42-msixvhds\XmlNotepad.vhd**, where the placeholder `<storage-account-name>` designates the name of the storage account hosting the **az140-42-msixvhds** file share
MSIX package	the name generated during package creation
Display name	**XML Notepad**
Registration type	**On-demand**
State	**Active**

Task 4: Publish MSIX apps to an application group

> **Note**: You will publish the MSIX app to both the remote app and the desktop app group.

1. Within the Bastion session to **az140-cl-vm42**, in the Microsoft Edge window displaying the Azure portal, search for and select **Azure Virtual Desktop** and, on the **Azure Virtual Desktop** blade, in the vertical menu on the left side, in the **Manage** section, select **Application groups**.
2. On the **Azure Virtual Desktop | Application groups** blade, select the **az140-21-hp1-Utilities-RAG** application group entry.
3. On the **az140-21-hp1-Utilities-RAG** blade, in the vertical menu on the left side, in the **Manage** section, select **Applications**.
4. On the **az140-21-hp1-Utilities-RAG | Applications** blade, click **+ Add**.

5. On the **Add application** blade, on the **Basics** and **Icon** tabs, specify the following settings and select **Review + add**:

Setting	Value
Application source	**App Attach**
Package	the name representing the package included in the image
Application	**XMLNOTEPAD**
Application identifier	**XML Notepad**
Display name	**XML Notepad**
Description	**XML Notepad**

6. Review the configured settings, then select **Add**.
7. Navigate back to the **Azure Virtual Desktop | Application groups** blade and select the **az140-21-hp1-DAG** application group entry.
8. On the **az140-21-hp1-DAG** blade, in the vertical menu on the left side, in the **Manage** section, select **Applications**.
9. On the **az140-21-hp1-DAG | Applications** blade, click **+ Add**.
10. On the **Add application** blade, specify the following settings and select **Review + add**:

Setting	Value
Application source	**MSIX package**
MSIX package	the name representing the package included in the image
Application identifier	**XML Notepad**
Display name	**XML Notepad**
Description	**XML Notepad**

11. Review the configured settings, then select **Add**.

Task 5: Validate the functionality of MSIX App attach

1. Within the Bastion session to **az140-cl-vm42**, start Microsoft Edge, navigate to Windows Desktop client download page and, once download completes, select **Open file** to start its installation. On the **Installation Scope** page of the **Remote Desktop Setup** wizard, select the option **Install for all users of this machine** and click **Install**.
2. Once the installation completes, ensure that the **Launch Remote Desktop when setup exits** checkbox is selected and click **Finish** to start the Remote Desktop client.
3. In the **Remote Desktop** client window, select **Subscribe** and, when prompted, sign in with the **aduser1** user principal name and the password you set when creating this user account.

4. In the **Remote Desktop** client window, within the **az140-21-ws1** section, double-click the **XML Notepad** icon, when prompted, provide the password, and verify that the XML Notepad launches successfully.
5. Within the Bastion session to **az140-cl-vm42**, right-click **Start**, in the right-click menu, select **Shut down or sign out** and then, in the cascading menu, select **Sign out**.
6. In the **Disconnected** dialog, select **Close**.

Exercise 4: Stop and deallocate Azure VMs provisioned and used in the lab

The main tasks for this exercise are as follows:

1. Stop and deallocate Azure VMs provisioned and used in the lab

> **Note**: In this exercise, you will deallocate the Azure VMs provisioned and used in this lab to minimize the corresponding compute charges

Task 1: Deallocate Azure VMs provisioned and used in the lab

1. Switch to the lab computer and, in the web browser window displaying the Azure portal, open the **PowerShell** shell session within the **Cloud Shell** pane.

2. From the PowerShell session in the Cloud Shell pane, run the following to list all Azure VMs created and used in this lab:

```
Get-AzVM -ResourceGroup 'az140-21-RG'
Get-AzVM -ResourceGroup 'az140-42-RG'
```

3. From the PowerShell session in the Cloud Shell pane, run the following to stop and deallocate all Azure VMs you created and used in this lab:

```
Get-AzVM -ResourceGroup 'az140-21-RG' | Stop-AzVM -NoWait -Force
Get-AzVM -ResourceGroup 'az140-42-RG' | Stop-AzVM -NoWait -Force
```

> **Note**: The command executes asynchronously (as determined by the -NoWait parameter), so while you will be able to run another PowerShell command immediately afterwards within the same PowerShell session, it will take a few minutes before the Azure VMs are actually stopped and deallocated.