AZ-140: Configuring and Operating Windows Virtual Desktop

- **Download Latest Student Handbook and AllFiles Content /home/ll/Azure_clone/Azure_new/AZ-140-Configuring-and-Operating-Windows-Virtual-Desktop/../../releases/latest**
- **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

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.

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.

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.

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!

Notes

Classroom Materials

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.

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

---

Content Directory

Hyperlinks to each of the lab exercises and demos are listed below.

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 %} | {% endfor %}

Demos

{% assign demos = site.pages | where_exp:"page", "page.url contains '/Instructions/Demos'" %} | Module | Demo | | --- | --- | {% for activity in demos %}| {{ activity.demo.module }} | {{ activity.demo.title }} | {% endfor %}

---

demo: title: 'Demo: Deploying an ARM Template' module: 'Module 1: Exploring Azure Resource Manager'

Demo: Deploying an ARM Template

Instructions

1. Quisque dictum convallis metus, vitae vestibulum turpis dapibus non.

    1. Suspendisse commodo tempor convallis.

    2. Nunc eget quam facilisis, imperdiet felis ut, blandit nibh.

    3. Phasellus pulvinar ornare sem, ut imperdiet justo volutpat et.

2. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

3. Vestibulum hendrerit orci urna, non aliquet eros eleifend vitae.

4. Curabitur nibh dui, vestibulum cursus neque commodo, aliquet accumsan risus.

    ```
    Sed at malesuada orci, eu volutpat ex
    ```

5. In ac odio vulputate, faucibus lorem at, sagittis felis.

6. Fusce tincidunt sapien nec dolor congue facilisis lacinia quis urna.

    Note: Ut feugiat est id ultrices gravida.

7. Phasellus urna lacus, luctus at suscipit vitae, maximus ac nisl.

    - Morbi in tortor finibus, tempus dolor a, cursus lorem.
    - Maecenas id risus pharetra, viverra elit quis, lacinia odio.
    - Etiam rutrum pretium enim.

1. Curabitur in pretium urna, nec ullamcorper diam.

lab: title: 'Lab: Prepare for deployment of Azure Windows Virtual Desktop (Azure AD DS)' module: 'Module 1: Plan a WVD Architecture'

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

Student lab manual

Lab dependencies

- An Azure subscription
- A Microsoft account or an Azure AD account with the Global Administrator role in the Azure AD tenant associated with the Azure subscription and with the Owner or Contributor role in the Azure subscription

Note: At the time of authoring this course, the MSIX app attach functionality for Windows Virtual Desktop is in public preview. If you intend to run the lab that involves the use of MSIX app attach included in this course, you need to submit a request via on online form to enable MSIX app attach in your subscription. The approval and processing of requests can take up to 24 hours during business days. You'll receive an email confirmation once your request has been accepted and completed.

Estimated Time

150 minutes

> **Note**: Provisioning of an Azure AD DS takes involves about 90-minute wait time.

Lab scenario

You need to prepare for deployment of Azure Windows Virtual Desktop in an Azure Active Directory Domain Services (Azure AD DS) environment

Objectives

After completing this lab, you will be able to:

- Implement an Azure AD DS domain
- Configure the Azure AD DS domain environment

Lab files

- \\AZ-140\AllFiles\Labs\01\az140-11_azuredeploycl11a.json
- \\AZ-140\AllFiles\Labs\01\az140-11_azuredeploycl11a.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**.

1. 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 (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`):

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

1. Review the output of the command executed in the previous step and ensure that you have at least **20** available vCPUs in both the

Standard DSv3 Family and **StandardBSFamily** 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**.
3. On the subscription's **Usage + quotas** blade, select **Request Increase**.
4. On the **Basics** tab of the **New support request** blade, specify the following and select **Next: Solutions >**:

Setting	Value
Issue type	**Service and subscription limits (quotas)**
Subscription	the name of the Azure subscription you will be using in this lab
Quota type	**Compute-VM (cores-vCPUs) subscription limit increases**
Support plan	the name of the support plan associated with the target subscription

1. On the **Details** tab of the **New support request** blade, select the **Provide details** link.
2. On the **Quota details** tab of the **New support request** blade, specify the following and select **Save and continue**:

Setting	Value
Deployment model	**Resource Manager**
Location	the name of the Azure region you intend to use in this lab
Types	**Standard**
Standard	**BS Series**
New vCPU Limit	the new limit

Setting	Value
Standard	**DSv3 Series**
New vCPU Limit	the new limit

Note: The use of **BS Series** Azure VMs is in this case intended to minimize the cost of running the lab environment. It is not meant to represent the intended usage of the **BS Series** Azure VMs in the Windows Virtual Desktop scenarios.

1. Back on the **Details** tab of the **New support request** blade, specify the following and select **Next: Review + create >**:

Setting	Value
Severity	**C - Minimal impact**
Preferred contact method	choose your preferred option and provide your contact details

1. On the **Review + create** tab of the **New support request** blade, select **Create**.

Note: Quota increase requests within this range of vCPUs are typically completed within a few hours.

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

The main tasks for this exercise are as follows:

1. Create and configure an Azure AD user account for administration of Azure AD DS domain
2. Deploy an Azure AD DS instance by using the Azure portal
3. Configure the network and identity settings of the Azure AD DS deployment

Task 1: Create and configure an Azure AD user account for administration of Azure AD DS domain

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 and

the Global Administrator role in the Azure AD tenant associated with the Azure subscription.
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**.

1. From the Cloud Shell pane, run the following to sign in to your Azure AD tenant:

```
powershell Connect-AzureAD
```

1. From the Cloud Shell pane, run the following to retrieve the primary DNS domain name of the Azure AD tenant associated with your Azure subscription:

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

1. From the Cloud Shell pane, run the following to create a new Azure AD user:

```
powershell $passwordProfile = New-Object -TypeName
Microsoft.Open.AzureAD.Model.PasswordProfile
$passwordProfile.Password = 'Pa55w.rd1234'
$passwordProfile.ForceChangePasswordNextLogin = $false New-
AzureADUser -AccountEnabled $true -DisplayName 'aadadmin1' -
PasswordProfile $passwordProfile -MailNickName 'aadadmin1' -
UserPrincipalName "aadadmin1@$aadDomainName"
```

1. From the Cloud Shell pane, run the following to assign the Global Administrator role to the newly created Azure AD user:

```
powershell $aadUser = Get-AzureADUser -ObjectId
"aadadmin1@$aadDomainName" $aadRole = Get-
AzureADDirectoryRole | Where-Object {$_.displayName -eq
'Global administrator'} Add-AzureADDirectoryRoleMember -
ObjectId $aadRole.ObjectId -RefObjectId $aadUser.ObjectId
```

Note: Azure AD PowerShell module refers to the Global Administrator role as Company Administrator.

1. From the Cloud Shell pane, run the following to identify the user principal name of the newly created Azure AD user:

```
powershell (Get-AzureADUser -Filter "MailNickName eq
'aadadmin1'").UserPrincipalName
```

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

1. Close the Cloud Shell pane.
2. Within the Azure portal, search for and select **Subscriptions** and, from the **Subscriptions** blade, select the Azure subscription you are using in this lab.
3. On the blade displaying properties of your Azure subscription, select **Access control (IAM)**, select **+ Add**, and, in the drop-down list, select **Add role assignment**.
4. On the **Add role assignment** blade, specify the following settings and select **Save**:

Setting	Value
Role	**Owner**
Assign access to	**User, group, or service principal**
Select	**aadadmin1**

Note: You will use the **aadadmin1** account to manage your Azure subscription and the corresponding Azure AD tenant from an Azure AD DS joined Windows 10 Azure VM later in the lab.

Task 2: Deploy an Azure AD DS instance by using the Azure portal

1. From your lab computer, in the Azure portal, search for and select **Azure AD Domain Services** and, from the **Azure AD Domain Services** blade, select **+ Add**. This will open the **Create Azure AD Domain Services** blade.
2. On the **Basics** tab of the **Create Azure AD Domain Services** blade, specify the following settings and select **Next** (leave others with their existing 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 **az140-11a-RG**
Domain name	**adatum.com**
Region	the name of the region where you want to host your WVD deployment
SKU	**Standard**
Forest type	**User**

Note: While this is technically not required, in general, you should assign an Azure AD DS domain name different from any existing Azure or on-premises DNS name space.

1. On the **Networking** tab of the **Create Azure AD Domain Services** blade, next to the **Virtual network** drop-down list, select **Create new**.
2. On the **Create virtual network** blade, specify the following settings and select **OK** (leave others with their existing values):

Setting	Value
Name	**az140-aadds-vnet11a**
Address range	**10.10.0.0/16**
Subnet name	**aadds-Subnet**
Subnet name	**10.10.0.0/24**

1. Back on the **Networking** tab of the **Create virtual network** blade, select **Next** (leave others with their existing values).
2. On the **Administration** tab of the **Create Azure AD Domain Services** blade, accept the default settings and select **Next**.
3. On the **Synchronization** tab of the **Create Azure AD Domain Services** blade, ensure that **All** is selected and then select **Next**.
4. On the **Review + create** tab of the **Create Azure AD Domain Services** blade, select **Create**.
5. Review the notification regarding settings that you will not be able to change following creation of the Azure AD DS domain and select **OK**.

Note: The settings that you will not be able to change following provisioning of an Azure AD DS domain include its DNS name, its Azure subscription, its resource group, the virtual network and subnet hosting its domain controllers, and the forest type.

Note: Wait for the deployment to complete before you proceed to the next exercise. This might take about 90 minutes.

Task 3: Configure the network and identity settings of the Azure AD DS deployment

1. From your lab computer, in the Azure portal, search for and select **Azure AD Domain Services** and, from the **Azure AD Domain Services** blade, select the **adatum.com** entry to navigate to the newly provisioned Azure AD DS instance.
2. On the **adatum.com** blade of the Azure AD DS instance, click the warning stating **Network configuration issues detected. See detailed diagnosis**.
3. On the **adatum.com | Configuration diagnostics (preview)** blade, click **Run**.
4. In the **Validation** section, expand the **DNS records** pane and click **Fix**.
5. On the **DNS records** blade, click **Fix** again.
6. Navigate back to the **adatum.com** blade of the Azure AD DS instance and, in the **Required configuration steps** section, review the information regarding the Azure AD DS password hash synchronization.

Note: Any existing cloud-only users that need to be able to access Azure AD DS domain computers and their resources must either change their passwords or have them reset. This applies to the **aadadmin1** account you created earlier in this lab.

1. From your lab computer, in the Azure portal, open a **PowerShell** session in the **Cloud Shell** pane.
2. From the PowerShell session in the Cloud Shell pane, run the following to identify the objectID attribute of the Azure AD **aadadmin1** user account:

```
powershell Connect-AzureAD $objectId = (Get-AzureADUser -
Filter "MailNickName eq 'aadadmin1'").ObjectId
```

1. From the PowerShell session in the Cloud Shell pane, run the following to reset the password of the **aadadmin1** user account, which objectId you identified in the previous step:

```
powershell $password = ConvertTo-SecureString 'Pa55w.rd1234'
-AsPlainText -Force Set-AzureADUserPassword -ObjectId
$objectId -Password $password -ForceChangePasswordNextLogin
$false
```

Note: In real-world scenarios you would typically set the value of the **-ForceChangePasswordNextLogin** to $true. We chose **$false** in this case to simplify the lab steps.

Exercise 2: Configure the Azure AD DS domain environment

The main tasks for this exercise are as follows:

1. Deploy an Azure VM running Windows 10 by using an Azure Resource Manager QuickStart template
2. Review the default configuration of the Azure AD DS domain
3. Create AD DS users and groups that will be synchronized to Azure AD DS

Task 1: Deploy an Azure VM running Windows 10 by using an Azure Resource Manager QuickStart template

1. From your lab computer, in the Azure portal, from the PowerShell session in the Cloud Shell pane, run the following to add a subnet named **cl-Subnet** to the virtual network named **az140-aadds-vnet11a** you created in the previous task:

```
powershell $resourceGroupName = 'az140-11a-RG' $vnet = Get-
AzVirtualNetwork -ResourceGroupName $resourceGroupName -Name
'az140-aadds-vnet11a' $subnetConfig = Add-
AzVirtualNetworkSubnetConfig ` -Name 'cl-Subnet' ` -
AddressPrefix 10.10.255.0/24 ` -VirtualNetwork $vnet $vnet |
Set-AzVirtualNetwork
```

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 files **\\AZ-140\AllFiles\Labs\01\az140-11_azuredeploycl11a.json** and **\\AZ-140\AllFiles\Labs\01\az140-**

11_azuredeploycl11a.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 will serve as a Windows Virtual Desktop client and join it to the Azure AD DS domain:

```
powershell $resourceGroupName = 'az140-11a-RG' $location =
(Get-AzResourceGroup -ResourceGroupName
$resourceGroupName).Location New-AzResourceGroupDeployment `
-ResourceGroupName $resourceGroupName ` -Location $location `
-Name az140lab0101vmDeployment ` -TemplateFile $HOME/az140-
11_azuredeploycl11a.json ` -TemplateParameterFile
$HOME/az140-11_azuredeploycl11a.parameters.json
```

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

Task 2: Review the default configuration of the Azure AD DS domain

Note: Before you can sign in to the newly Azure AD DS joined computer, you need to add the user account you intend to sign in with to the **AAD DC Administrators** Azure AD group. This Azure AD group is created automatically in the Azure AD tenant associated with the Azure subscription where you provisioned the Azure AD DS instance.

Note: You have the option of populating this group with existing Azure AD user accounts when you provision an Azure AD DS instance.

1. From your lab computer, in the Azure portal, from the Cloud Shell pane, run the following to add the **aadadmin1** Azure AD user account to the **AAD DC Administrators** Azure AD group:

```
powershell Connect-AzureAD $groupObjectId = (Get-AzureADGroup
-Filter "DisplayName eq 'AAD DC Administrators'").ObjectId
$userObjectId = (Get-AzureADUser -Filter "MailNickName eq
'aadadmin1'").ObjectId Add-AzureADGroupMember -ObjectId
$groupObjectId -RefObjectId $userObjectId
```

1 Close the Cloud Shell pane. 1. From your lab computer, in the Azure portal, search for and select **Virtual machines** and, from the **Virtual**

machines blade, select the **az140-cl-vm11a** entry. This will open the **az140-cl-vm11a** blade. 1. On the **az140-cl-vm11a** blade, select **Connect**, in the drop-down menu, select **RDP**, on the **RDP** tab of the **az140-cl-vm11a | Connect** blade, in the **IP address** drop-down list, select the **Public IP address** entry, and then select **Download RDP File**. 1. When prompted, sign in with the following credentials:

Setting	Value
User Name	**ADATUM\aadadmin1**
Password	**Pa55w.rd1234**

1. Within the Remote Desktop to the **az140-cl-vm11a** Azure VM, start **Windows PowerShell ISE** as Administrator and, from the **Administrator: Windows PowerShell ISE** script pane, run the following to install the Active Directory and DNS-related Remote Server Administration Tools:

```
powershell Add-WindowsCapability -Name
Rsat.ActiveDirectory.DS-LDS.Tools~~~~0.0.1.0 -Online Add-
WindowsCapability -Name Rsat.Dns.Tools~~~~0.0.1.0 -Online
Add-WindowsCapability -Name
Rsat.GroupPolicy.Management.Tools~~~~0.0.1.0 -Online Add-
WindowsCapability -Name Rsat.ServerManager.Tools~~~~0.0.1.0 -
Online
```

Note: Wait for the installation to complete before you proceed to the next step. This might take about 2 minutes.

1. Within the Remote Desktop to the **az140-cl-vm11a** Azure VM, in the **Start** menu, navigate to the **Windows Administrative Tools** folder, expand it, and, from the list of tools, start **Active Directory Users and Computers**.
2. In the **Active Directory Users and Computers** console, review the default hierarchy, including the **AADDC Computers** and **AADDC Users** organizational units. Note that the former includes the **az140-cl-vm11a** computer account and the latter includes the user accounts synchronized from the Azure AD tenant associated with the Azure subscription hosting the deployment of Azure AD DS instance. The **AADDC Users** organizational unit also includes the **AAD DC Administrators** group synchronized from the same Azure AD tenant, along with its group membership. This membership cannot be modified directly within the Azure AD DS domain, but instead, you have to manage it within the Azure AD DS

tenant. Any changes are automatically synchronized with the replica of the group hosted in the Azure AD DS domain.

Note: Currently, the group includes only the **aadadmin1** user account.

1. In the **Active Directory Users and Computers** console, in the **AADDC Users** OU, select the **aadadmin1** user account, display its **Properties** dialog box, switch to the **Accounts** tab, and note that the user principal name suffix matches the primary Azure AD DNS domain name and is not modifiable.
2. In the **Active Directory Users and Computers** console, review the content of the **Domain Controllers** organizational unit and note that it includes computer accounts of two domain controllers with randomly generated names.

Task 3: Create AD DS users and groups that will be synchronized to Azure AD DS

1. Within the Remote Desktop to the **az140-cl-vm11a** Azure VM, start Microsoft Edge, navigate to the Azure portal, and sign in by providing user principal name of the **aadadmin1** user account with **Pa55w.rd1234** as its password.
2. In the Azure portal, open a PowerShell session in the **Cloud Shell**.
3. When prompted to select either **Bash** or **PowerShell**, select **PowerShell**.

Note: Since this is the first time you are starting **Cloud Shell** by using the **aadadmin1** user account, you will need to configure its Cloud Shell home directory. When presented with the **You have no storage mounted** message, select the subscription you are using in this lab, and select **Create storage**.

1. From the PowerShell session in the Cloud Shell pane, run the following to sign in to authenticate to your Azure AD tenant:

```
powershell Connect-AzureAD
```

1. From the PowerShell session in the Cloud Shell pane, run the following to retrieve the primary DNS domain name of the Azure AD tenant associated with your Azure subscription:

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

1. From the PowerShell session in the Cloud Shell pane, run the following to create the Azure AD user accounts you will use in the upcoming labs:

```
powershell $passwordProfile = New-Object -TypeName
Microsoft.Open.AzureAD.Model.PasswordProfile
$passwordProfile.Password = 'Pa55w.rd1234'
$passwordProfile.ForceChangePasswordNextLogin = $false
$aadUserNamePrefix = 'aaduser' $userCount = 1..9 foreach
($counter in $userCount) { New-AzureADUser -AccountEnabled
$true -DisplayName "$aadUserNamePrefix$counter" -
PasswordProfile $passwordProfile -MailNickName
"$aadUserNamePrefix$counter" -UserPrincipalName
"$aadUserNamePrefix$counter@$aadDomainName" }
```

1. From the PowerShell session in the Cloud Shell pane, run the following to create an Azure AD group named **az140-wvd-aadmins** and add to it the **aadadmin1** user account:

```
powershell $az140wvdaadmins = New-AzureADGroup -Description
'az140-wvd-aadmins' -DisplayName 'az140-wvd-aadmins' -
MailEnabled $false -SecurityEnabled $true -MailNickName
'az140-wvd-aadmins' $userObjectId = (Get-AzureADUser -Filter
"MailNickName eq 'aadadmin1'").ObjectId Add-
AzureADGroupMember -ObjectId $az140wvdaadmins.ObjectId -
RefObjectId $userObjectId
```

1. From the Cloud Shell pane, repeat the previous step to create Azure AD groups for Windows Virtual Desktop users that you will use in the upcoming labs and add to them previously created Azure AD user accounts:

```powershell $az140wvdausers = New-AzureADGroup -Description 'az140-wvd-ausers' -DisplayName 'az140-wvd-ausers' -MailEnabled $false -SecurityEnabled $true -MailNickName 'az140-wvd-ausers' $userObjectId = (Get-AzureADUser -Filter "MailNickName eq 'aaduser1'").ObjectId Add-AzureADGroupMember -ObjectId $az140wvdausers.ObjectId -RefObjectId $userObjectId $userObjectId = (Get-AzureADUser -Filter "MailNickName eq 'aaduser2'").ObjectId Add-AzureADGroupMember -ObjectId $az140wvdausers.ObjectId -RefObjectId $userObjectId $userObjectId = (Get-AzureADUser -Filter "MailNickName eq 'aaduser3'").ObjectId Add-AzureADGroupMember -ObjectId $az140wvdausers.ObjectId -RefObjectId $userObjectId
```

```
$userObjectId = (Get-AzureADUser -Filter "MailNickName eq
'aaduser4'").ObjectId Add-AzureADGroupMember -ObjectId
$az140wvdausers.ObjectId -RefObjectId $userObjectId $userObjectId =
(Get-AzureADUser -Filter "MailNickName eq 'aaduser5'").ObjectId
Add-AzureADGroupMember -ObjectId $az140wvdausers.ObjectId -
RefObjectId $userObjectId $userObjectId = (Get-AzureADUser -Filter
"MailNickName eq 'aaduser6'").ObjectId Add-AzureADGroupMember -
ObjectId $az140wvdausers.ObjectId -RefObjectId $userObjectId
$userObjectId = (Get-AzureADUser -Filter "MailNickName eq
'aaduser7'").ObjectId Add-AzureADGroupMember -ObjectId
$az140wvdausers.ObjectId -RefObjectId $userObjectId $userObjectId =
(Get-AzureADUser -Filter "MailNickName eq 'aaduser8'").ObjectId
Add-AzureADGroupMember -ObjectId $az140wvdausers.ObjectId -
RefObjectId $userObjectId $userObjectId = (Get-AzureADUser -Filter
"MailNickName eq 'aaduser9'").ObjectId Add-AzureADGroupMember -
ObjectId $az140wvdausers.ObjectId -RefObjectId $userObjectId

$az140wvdaremoteapp = New-AzureADGroup -Description "az140-
wvd-aremote-app" -DisplayName "az140-wvd-aremote-app" -
MailEnabled $false -SecurityEnabled $true -MailNickName "az140-
wvd-aremote-app" $userObjectId = (Get-AzureADUser -Filter
"MailNickName eq 'aaduser1'").ObjectId Add-AzureADGroupMember -
ObjectId $az140wvdaremoteapp.ObjectId -RefObjectId $userObjectId
$userObjectId = (Get-AzureADUser -Filter "MailNickName eq
'aaduser5'").ObjectId Add-AzureADGroupMember -ObjectId
$az140wvdaremoteapp.ObjectId -RefObjectId $userObjectId
$userObjectId = (Get-AzureADUser -Filter "MailNickName eq
'aaduser6'").ObjectId Add-AzureADGroupMember -ObjectId
$az140wvdaremoteapp.ObjectId -RefObjectId $userObjectId

$az140wvdapooled = New-AzureADGroup -Description "az140-wvd-
apooled" -DisplayName "az140-wvd-apooled" -MailEnabled $false -
SecurityEnabled $true -MailNickName "az140-wvd-apooled"
$userObjectId = (Get-AzureADUser -Filter "MailNickName eq
'aaduser1'").ObjectId Add-AzureADGroupMember -ObjectId
$az140wvdapooled.ObjectId -RefObjectId $userObjectId $userObjectId
= (Get-AzureADUser -Filter "MailNickName eq 'aaduser2'").ObjectId
Add-AzureADGroupMember -ObjectId $az140wvdapooled.ObjectId -
RefObjectId $userObjectId $userObjectId = (Get-AzureADUser -Filter
"MailNickName eq 'aaduser3'").ObjectId Add-AzureADGroupMember -
ObjectId $az140wvdapooled.ObjectId -RefObjectId $userObjectId
$userObjectId = (Get-AzureADUser -Filter "MailNickName eq
```

'aaduser4'").ObjectId Add-AzureADGroupMember -ObjectId $az140wvdapooled.ObjectId -RefObjectId $userObjectId

$az140wvdapersonal = New-AzureADGroup -Description "az140-wvd-apersonal" -DisplayName "az140-wvd-apersonal" -MailEnabled $false -SecurityEnabled $true -MailNickName "az140-wvd-apersonal"
$userObjectId = (Get-AzureADUser -Filter "MailNickName eq 'aaduser7'").ObjectId Add-AzureADGroupMember -ObjectId $az140wvdapersonal.ObjectId -RefObjectId $userObjectId
$userObjectId = (Get-AzureADUser -Filter "MailNickName eq 'aaduser8'").ObjectId Add-AzureADGroupMember -ObjectId $az140wvdapersonal.ObjectId -RefObjectId $userObjectId
$userObjectId = (Get-AzureADUser -Filter "MailNickName eq 'aaduser9'").ObjectId Add-AzureADGroupMember -ObjectId $az140wvdapersonal.ObjectId -RefObjectId $userObjectId ```

1. Close the Cloud Shell pane.
2. Within the Remote Desktop to the **az140-cl-vm11a** Azure VM, in the Microsoft Edge window displaying the Azure portal, search for and select **Azure Active Directory** blade, on your Azure AD tenant blade, in the vertical menu bar on the left side, in the **Manage* section, select** Users **and, on the** Users | All users** blade, verify that new user accounts have been created.
3. Navigate back to the Azure AD tenant blade, in the vertical menu bar on the left side, in the **Manage** section, select **Groups** and, on the **Groups | All groups** blade, verify that new group accounts have been created.
4. Within the Remote Desktop to the **az140-cl-vm11a** Azure VM, switch to the **Active Directory Users and Computers** console, in the **Active Directory Users and Computers** console, navigate to the **AADDC Users** OU, and verify that it contains the same user and group accounts.

>Note: You might have to refresh the view of the console.

lab: title: 'Lab: Prepare for deployment of Azure Windows Virtual Desktop (AD DS)' module: 'Module 1: Plan a WVD Architecture'

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

Student lab manual

Lab dependencies

- An Azure subscription you will be using in this lab.
- A Microsoft account or an Azure AD 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 Azure AD tenant associated with that Azure subscription.

Note: At the time of authoring this course, the MSIX app attach functionality for Windows Virtual Desktop is in public preview. If you intend to run the lab that involves the use of MSIX app attach included in this course, you need to submit a request via on online form to enable MSIX app attach in your subscription. The approval and processing of requests can take up to 24 hours during business days. You'll receive an email confirmation once your request has been accepted and completed.

Estimated Time

60 minutes

> **Note**: Provisioning of an Azure AD DS takes involves about 90-minute wait time.

Lab scenario

You need to prepare for deployment of Azure Windows Virtual Desktop in 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 an Azure Active Directory (Azure AD) 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**.

1. 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 (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`):

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

1. Review the output of the command executed in the previous step and ensure that you have at least **20** available vCPUs in both the

Standard DSv3 Family and **StandardBSFamily** 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**.
3. On the subscription's **Usage + quotas** blade, select **Request Increase**.
4. On the **Basics** tab of the **New support request** blade, specify the following and select **Next: Solutions >**:

Setting	Value
Issue type	**Service and subscription limits (quotas)**
Subscription	the name of the Azure subscription you will be using in this lab
Quota type	**Compute-VM (cores-vCPUs) subscription limit increases**
Support plan	the name of the support plan associated with the target subscription

1. On the **Details** tab of the **New support request** blade, select the **Provide details** link.
2. On the **Quota details** tab of the **New support request** blade, specify the following and select **Save and continue**:

Setting	Value
Deployment model	**Resource Manager**
Location	the name of the Azure region you intend to use in this lab
Types	**Standard**
Standard	**BS Series**
New vCPU Limit	the new limit

Setting	Value
Standard	**DSv3 Series**
New vCPU Limit	the new limit

Note: The use of **BS Series** Azure VMs is in this case intended to minimize the cost of running the lab environment. It is not meant to represent the intended usage of the **BS Series** Azure VMs in the Windows Virtual Desktop scenarios.

1. Back on the **Details** tab of the **New support request** blade, specify the following and select **Next: Review + create >**:

Setting	Value
Severity	**C - Minimal impact**
Preferred contact method	choose your preferred option and provide your contact details

1. On the **Review + create** tab of the **New support request** blade, select **Create**.

Note: Quota increase requests within this range of vCPUs are typically completed within a few hours.

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

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

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**.

1. 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 domain name prefix 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):

```
powershell $location = '<Azure_region>' Test-
AzDnsAvailability -Location $location -DomainNameLabel
<custom-name>
```

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

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

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:

```
powershell $resourceGroupName = 'az140-11-RG' New-
AzResourceGroup -Location $location -Name $resourceGroupName
```

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

2. From your lab computer, in the same web browser window, open another web browser tab and navigate to the QuickStart template named Create a new Windows VM and create a new AD Forest, Domain and DC.
3. 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.
4. On the **Create an Azure VM with a new AD Forest** blade, select **Edit parameters**.
5. 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**.
6. 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**
Dns Prefix	the DNS hostname you identified in the previous task

1. 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. This might take about 15 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, from the PowerShell session in the Cloud Shell pane, run the following to add a subnet named **cl-Subnet** to the virtual network named **az140-adds-vnet11** you created in the previous task:

```
powershell $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
```

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 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.
2. 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 Windows Virtual Desktop client into the newly created subnet:

```
powershell $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 exercise. The deployment might take about 10 minutes.

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

The main tasks for this exercise are as follows:

1. Create AD DS users and groups that will be synchronized to Azure AD
2. Configure AD DS UPN suffix
3. Create an Azure AD user that will be used to configure synchronization with Azure AD
4. Install Azure AD Connect
5. Configure hybrid Azure AD join

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

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 **RDP**, on the **RDP** tab of the **az140-dc-vm11 | Connect** blade, in the **IP address** drop-down list, select the **Load balancer DNS name** entry, and then select **Download RDP File**.
3. When prompted, sign in with the following credentials:

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

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

```
powershell $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
```

1. 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 Azure AD tenant used in this lab:

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

1. 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:

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

1. From the **Administrator: Windows PowerShell ISE** script pane, run the following to create AD DS user accounts that will be synchronized to the Azure AD tenant used in this lab:

```powershell $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 'Pa55w.rd1234' -AsPlainText -Force) -passThru }

$adUserNamePrefix = 'wvdadmin1' $adUPNSuffix = 'adatum.com' New-AdUser -Name $adUserNamePrefix -Path $ouPath -Enabled $True –ChangePasswordAtLogon $false -userPrincipalName $adUserNamePrefix@$adUPNSuffix -AccountPassword (ConvertTo-SecureString 'Pa55w.rd1234' -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**.

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

```
powershell 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
```

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

```
powershell 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','aduser8','aduser9' Get-ADGroup -Identity 'az140-
wvd-admins' | Add-AdGroupMember -Members 'wvdadmin1'
```

Task 2: Configure AD DS UPN suffix

1. Within the Remote Desktop 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):

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

1. 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):

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

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

```
powershell Connect-AzAccount
```

1. When prompted, provide the credentials of the user account with the Owner role in the subscription you are using in this lab.
2. From the **Administrator: Windows PowerShell ISE** console, run the following to retrieve the Id property of the Azure AD tenant associated with your Azure subscription:

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

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

```
powershell Install-Module -Name AzureAD -Force
```

1. From the **Administrator: Windows PowerShell ISE** console, run the following to authenticate to your Azure AD tenant:

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

1. When prompted, sign in with the same credentials you used earlier in this task.
2. From the **Administrator: Windows PowerShell ISE** console, run the following to retrieve the primary DNS domain name of the Azure AD tenant associated with your Azure subscription:

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

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

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

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

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

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

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

Task 3: Create an Azure AD user that will be used to configure directory synchronization

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to create a new Azure AD user:

```
powershell $userName = 'aadsyncuser' $passwordProfile = New-
Object -TypeName Microsoft.Open.AzureAD.Model.PasswordProfile
$passwordProfile.Password = 'Pa55w.rd1234'
$passwordProfile.ForceChangePasswordNextLogin = $false New-
AzureADUser -AccountEnabled $true -DisplayName $userName -
PasswordProfile $passwordProfile -MailNickName $userName -
UserPrincipalName "$userName@$aadDomainName"
```

1. From the **Administrator: Windows PowerShell ISE** script pane, run the following to assign the Global Administrator role to the newly created Azure AD user:

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

Note: Azure AD PowerShell module refers to the Global Administrator role as Company Administrator.

1. From the **Administrator: Windows PowerShell ISE** script pane, run the following to identify the user principal name of the newly created Azure AD user:

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

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

Task 4: Install Azure AD Connect

1. Within the Remote Desktop session to **az140-dc-vm11**, start Internet Explorer and navigate to the Microsoft Edge for Business download page.
2. 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.
3. 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 Azure AD credentials of the user account with the Owner role in the subscription you are using in this lab.

4. 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 **Azure Active Directory** blade and, on your Azure AD tenant blade, in the **Manage** section of the hub menu, select **Azure AD Connect**.
5. On the **Azure AD Connect** blade, select the **Download Azure AD Connect** link. This will automatically open a new browser tab displaying the **Microsoft Azure Active Directory Connect** download page.
6. On the **Microsoft Azure Active Directory Connect** download page, select **Download**.
7. When prompted whether to run or save the **AzureADConnect.msi** installer, select **Run** to start the **Microsoft Azure Active Directory Connect** wizard.
8. 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**.
9. On the **Express Settings** page of the **Microsoft Azure Active Directory Connect** wizard, select the **Customize** option.
10. On the **Install required components** page, leave all optional configuration options deselected and select **Install**.
11. On the **User sign-in** page, ensure that only the **Password Hash Synchronization** is enabled and select **Next**.
12. 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 **Pa55w.rd1234** as its password.

1. On the **Connect your directories** page, select the **Add Directory** button to the right of the **adatum.com** forest entry.
2. 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**

Setting	Value
Password	**Pa55w.rd1234**

1. Back on the **Connect your directories** page, ensure that the **adatum.com** entry appears as a configured directory and select **Next**
2. 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 Azure AD tenant does not have a verified custom DNS domain matching one of the UPN suffixes of the **adatum.com** AD DS.

1. 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**.
2. On the **Uniquely identifying your users** page, accept the default settings, and select **Next**.
3. On the **Filter users and devices** page, accept the default settings, and select **Next**.
4. On the **Optional features** page, accept the default settings, and select **Next**.
5. 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.

1. Review the information on the **Configuration complete** page and select **Exit** to close the **Microsoft Azure Active Directory Connect** window.
2. 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 Azure AD tenant.
3. 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 **Directory synced** column.

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

lab: title: 'Lab: Create and configure host pools and session hosts (Azure AD DS)' module: 'Module 2: Implement a WVD Infrastructure'

Lab - Create and configure host pools and session hosts (Azure AD DS)

Student lab manual

Lab dependencies

- An Azure subscription
- A Microsoft account or an Azure AD account with the Global Administrator role in the Azure AD 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 Windows Virtual Desktop (Azure AD DS)**

Estimated Time

60 minutes

Lab scenario

You need to create and configure host pools and session hosts in an Azure Active Directory Domain Services (Azure AD DS) environment.

Objectives

After completing this lab, you will be able to:

- Configure an Azure Windows Virtual Desktop environment in an Azure AD DS domain.
- Validate Azure Windows Virtual Desktop environment in an Azure AD DS domain.

Lab files

- None

Instructions

Exercise 1: Configure an Azure Windows Virtual Desktop environment

The main tasks for this exercise are as follows:

1. Prepare AD DS domain and the Azure subscription for deployment of an Azure Windows Virtual Desktop host pool
2. Deploy an Azure Windows Virtual Desktop host pool
3. Configure Windows Virtual Desktop application groups
4. Configure Windows Virtual Desktop workspaces

Task 1: Prepare AD DS domain and the Azure subscription for deployment of an Azure Windows Virtual Desktop host pool

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. From your lab computer, in the Azure portal, search for and select **Virtual machines** and, from the **Virtual machines** blade, select the **az140-cl-vm11a** entry. This will open the **az140-cl-vm11a** blade.
3. On the **az140-cl-vm11a** blade, in the toolbar, select **Connect**, in the drop-down menu, select **RDP**, on the **RDP** tab of the **az140-cl-vm11a | Connect** blade, in the **IP address** drop-down list, select the **Public IP address** entry, and then select **Download RDP File**.
4. When prompted, sign in with the following credentials:

Setting	Value
User Name	**ADATUM\aadadmin1**
Password	**Pa55w.rd1234**

1. Within the Remote Desktop to the **az140-cl-vm11a** Azure VM, start Microsoft Edge, navigate to the Azure portal, and sign in by providing user principal name of the **aadadmin1** user account with **Pa55w.rd1234** as its password.

Note: You can identify the user principal name (UPN) attribute of the **aadadmin1** account by reviewing its properties dialog box from

the Active Directory Users and Computers console or by switching back to your lab computer and reviewing its properties from the Azure AD tenant blade in the Azure portal.

1. Within the Remote Desktop session to **az140-cl-vm11a**, in the Microsoft Edge displaying the Azure portal, open a PowerShell session in the **Cloud Shell** and run the following register the **Microsoft.DesktopVirtualization** resource provider:

```
powershell Register-AzResourceProvider -ProviderNamespace
Microsoft.DesktopVirtualization
```

1. Within the Remote Desktop session to **az140-cl-vm11a**, in the Microsoft Edge displaying the Azure portal, search for and select **Virtual networks** and, from the **Virtual networks** blade, select the **az140-aadds-vnet11a** entry.
2. On the **az140-aadds-vnet11a** blade, select **Subnets**, on the **Subnets** blade, select **+ Subnet**, on the **Add subnet** blade, in the **Name** text box, type **hp1-Subnet**, leave all other settings with their default values, and select **Save**.

Task 2: Deploy an Azure Windows Virtual Desktop host pool

1. Within the Remote Desktop session to **az140-cl-vm11a**, in the Microsoft Edge window displaying the Azure portal, search for and select **Windows Virtual Desktop**, on the **Windows Virtual Desktop** blade, in the vertical menu on the left side, in the **Manage** section, select **Host pools** and, on the **Windows Virtual Desktop | Host pools** blade, select **+ Create**.
2. On the **Basics** tab of the **Create a host pool** blade, specify the following settings and select **Next: Virtual Machines >**:

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	the name of a new resource group **az140-21a-RG**
Host pool name	**az140-21a-hp1**
Location	the name of the Azure region into which you deployed the Azure AD DS instance earlier in this lab

Setting	Value
Validation environment	**No**
Host pool type	**Pooled**
Max session limit	**10**
Load balancing algorithm	**Breadth-first**

1. On the **Virtual machines** tab of the **Create a host pool** blade, specify the following settings and select **Next: Workspace >** (replace the placeholder with the name of the Azure AD tenant associated with the subscription into which you deployed the Azure AD DS instance):

Setting	Value
Add virtual machines	**Yes**
Resource group	**Defaulted to same as host pool**
Virtual machine location	the name of the Azure region into which you deployed resources in the first exercise of this lab
Virtual machine size	**Standard D2s v3**
Number of VMs	**2**
Name prefix	**az140-21-p1**
Image type	**Gallery**
Image	**Windows 10 Enterprise multi-session, Version 2004 + Microsoft 365 Apps**
OS disk type	**Standard SSD**
Virtual network	**az140-aadds-vnet11a**
Subnet	**hp1-Subnet (10.10.1.0/24)**
Public IP	**Yes**

Setting	Value
Configure SKU	**Basic**
Configure assignment	**Dynamic**
Network security group	**Basic**
Public inbound ports	**No**
Specify domain or unit	**Yes**
Domain to join	**adatum.com**
Organizational Unit path	**OU=AADDC Computers,DC=adatum,DC=com**
AD domain join UPN	**aadadmin1@**
Password	**Pa55w.rd1234**

1. On the **Workspace** tab of the **Create a host pool** blade, specify the following settings and select **Review + create**:

Setting	Value
Register desktop app group	**No**

1. On the **Review + create** tab of the **Create a host pool** blade, select **Create**.

Note: Wait for the deployment to complete. This should take about 15 minutes.

Task 3: Configure Windows Virtual Desktop application groups

1. Within the Remote Desktop session to **az140-cl-vm11a**, in the Azure portal, search for and select **Windows Virtual Desktop** and, on the **Windows Virtual Desktop** blade, select **Application groups**.
2. On the **Windows Virtual Desktop | Application groups** blade, select the auto-generated **az140-21a-hp1-DAG** desktop application

group.
3. On the **az140-21a-hp1-DAG** blade, in the vertical menu on the left side, in the **Manage** section, select **Assignments**.
4. On the **az140-21a-hp1-DAG | Assignments** blade, select **+ Add**.
5. On the **Select Azure AD users or user groups** blade, select **az140-wvd-apooled** and click **Select**.
6. Navigate back to the **Windows Virtual Desktop | Application groups** blade, and select **+ Create** again.
7. On the **Basics** tab of the **Create an application group** blade, specify the following settings and select **Next: Applications >**:

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	**az140-21a-RG**
Host pool	**az140-21a-hp1**
Application group type	**RemoteApp**
Application group name	**az140-21a-hp1-Office365-RAG**

1. On the **Applications** tab of the **Create an application group** blade, select **+ Add applications**.
2. On the **Add application** blade, specify the following settings and select **Save**:

Setting	Value
Application source	**Start menu**
Application	**Word**
Description	**Microsoft Word**
Require command line	**No**

1. Back on the **Applications** tab of the **Create an application group** blade, select **+ Add applications**.
2. On the **Add application** blade, specify the following settings and select **Save**:

Setting	Value
Application source	**Start menu**
Application	**Excel**

Setting	Value
Description	**Microsoft Excel**
Require command line	**No**

1. Back on the **Applications** tab of the **Create an application group** blade, select **+ Add applications**.
2. On the **Add application** blade, specify the following settings and select **Save**:

Setting	Value
Application source	**Start menu**
Application	**PowerPoint**
Description	**Microsoft PowerPoint**
Require command line	**No**

1. Back on the **Applications** tab of the **Create an application group** blade, select **Next: Assignments >**.
2. On the **Assignments** tab of the **Create an application group** blade, select **+ Add Azure AD users or user groups**.
3. On the **Select Azure AD users or user groups** blade, select **az140-wvd-aremote-app** and click **Select**.
4. Back on the **Assignments** tab of the **Create an application group** blade, select **Next: Workspace >**.
5. On the **Workspace** tab of the **Create a workspace** blade, specify the following setting and select **Review + create**:

Setting	Value
Register application group	**No**

1. On the **Review + create** tab of the **Create an application group** blade, select **Create**.

Note: Now you will create an application group based on file path as the application source

1. Within the Remote Desktop session to **az140-cl-vm11a**, search for and select **Windows Virtual Desktop** and, on the **Windows Virtual Desktop** blade, select **Application groups**.
2. On the **Windows Virtual Desktop | Application groups** blade, select **+ Create**.

3. On the **Basics** tab of the **Create an application group** blade, specify the following settings and select **Next: Applications >**:

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	**az140-21a-RG**
Host pool	**az140-21a-hp1**
Application group type	**RemoteApp**
Application group name	**az140-21a-hp1-Utilities-RAG**

1. On the **Applications** tab of the **Create an application group** blade, select **+ Add applications**.
2. On the **Add application** blade, specify the following settings and select **Save**:

Setting	Value
Application source	**File path**
Application path	**C:\Windows\system32\cmd.exe**
Application name	**Command Prompt**
Display name	**Command Prompt**
Icon path	**C:\Windows\system32\cmd.exe**
Icon index	**0**
Description	**Windows Command Prompt**
Require command line	**No**

1. Back on the **Applications** tab of the **Create an application group** blade, select **Next: Assignments >**.
2. On the **Assignments** tab of the **Create an application group** blade, select **+ Add Azure AD users or user groups**.
3. On the **Select Azure AD users or user groups** blade, select **az140-wvd-aremote-app** and **az140-wvd-aadmins** and click **Select**.
4. Back on the **Assignments** tab of the **Create an application group** blade, select **Next: Workspace >**.
5. On the **Workspace** tab of the **Create a workspace** blade, specify the following setting and select **Review + create**:

Setting	Value
Register application group	**No**

1. On the **Review + create** tab of the **Create an application group** blade, select **Create**.

Task 4: Configure Windows Virtual Desktop workspaces

1. Within the Remote Desktop session to **az140-cl-vm11a**, in the Microsoft Edge window displaying the Azure portal, search for and select **Windows Virtual Desktop** and, on the **Windows Virtual Desktop** blade, select **Workspaces**.
2. On the **Windows Virtual Desktop | Workspaces** blade, select **+ Create**.
3. On the **Basics** tab of the **Create a workspace** blade, specify the following settings and select **Next: Application groups >**:

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	**az140-21a-RG**
Workspace name	**az140-21a-ws1**
Friendly name	**az140-21a-ws1**

1. On the **Application groups** tab of the **Create a workspace** blade, specify the following settings:

Setting	Value
Register desktop app group	**Yes**

1. On the **Workspace** tab of the **Create a workspace** blade, select **+ Register application groups**.
2. On the **Add application groups** blade, select the plus sign next to the **az140-21a-hp1-DAG**, **az140-21a-hp1-Office365-RAG**, and **az140-21a-hp1-Utilities-RAG** entries and click **Select**.
3. Back on the **Application groups** tab of the **Create a workspace** blade, select **Review + create**.
4. On the **Review + create** tab of the **Create a workspace** blade, select **Create**.

Exercise 2: Validate Azure Windows Virtual Desktop environment

The main tasks for this exercise are as follows:

1. Install Microsoft Remote Desktop client (MSRDC) on a Windows 10 computer
2. Subscribe to a Windows Virtual Desktop workspace
3. Test Windows Virtual Desktop apps

Task 1: Install Microsoft Remote Desktop client (MSRDC) on a Windows 10 computer

1. Within the Remote Desktop session to **az140-cl-vm11a**, start Microsoft Edge and navigate to Windows Desktop client download page and, when prompted, run its installation by following prompts. Select the option **Install for all users on this machine**.
2. Once the installation completes, start the Remote Desktop client.

Task 2: Subscribe to a Windows Virtual Desktop workspace

1. In the **Remote Desktop** client window, select **Subscribe** and, when prompted, sign in with the **aaduser1** credentials (using its userPrincipalName attribute as the user name and **Pa55w.rd1234** as its password.

Note: Alternatively, in the **Remote Desktop** client window, select **Subscribe with URL**, in the **Subscribe to a Workspace** pane, in the **Email or Workspace URL**, type **https://rdweb.wvd.microsoft.com/api/arm/feeddiscovery**, select **Next**, and, once prompted, sign in with the **aaduser1** credentials (using its userPrincipalName attribute as the user name and **Pa55w.rd1234** as its password).

Note*: ***The user principal name of aaduser1 should be in the format ****aaduser1@****, where the** * placeholder matches the name of the Azure AD tenant associated with the subscription into which you deployed the Azure AD DS instance.

1. In the **Stay signed in to all your apps** window, clear the checkbox **Allow my organization to manage my device** checkbox and select

No, sign in to this app only.
2. Ensure that the **Remote Desktop** page displays the listing of applications that are included in the application groups associated with the user account **aaduser1** via its group membership.

Task 3: Test Windows Virtual Desktop apps

1. Within the Remote Desktop session to **az140-cl-vm11a**, in the **Remote Desktop** client window, in the list of applications, double-click **Command Prompt** and verify that it launches a **Command Prompt** window. When prompted to authenticate, type **Pa55w.rd1234** as the password of the **aaduser1** user account, select the checkbox **Remember me**, and select **OK**.

Note: Initially, it might take a few minutes for the application to start, but subsequently, the application startup should be much faster.

1. At the Command Prompt, type **hostname** and press the **Enter** key to display the name of the computer on which the Command Prompt is running.

Note: Verify that the displayed name is either **az140-21-p1-0** or **az140-21-p1-1**, not **az140-cl-vm11a**.

1. At the Command Prompt, type **logoff** and press the **Enter** key to log off from the current Remote App session.
2. Within the Remote Desktop session to **az140-cl-vm11a**, in the **Remote Desktop** client window, in the list of applications, double-click **SessionDesktop** and verify that it launches a Remote Desktop session.
3. Within the **Default Desktop** session, right-click **Start**, select **Run**, in the **Open** text box of the **Run** dialog box, type **cmd** and select **OK**.
4. Within the **Default Desktop** session, at the Command Prompt, type **hostname** and press the **Enter** key to display the name of the computer on which the Remote Desktop session is running.
5. Verify that the displayed name is either **az140-21-p1-0**, **az140-21-p1-1** or **az140-21-p1-2**.

---

lab: title: 'Lab: Deploy host pools and session hosts by using the Azure portal (AD DS)' module: 'Module 2: Implement a WVD Infrastructure'

Lab - Deploy host pools and session hosts by using the Azure portal (AD DS)

Student lab manual

Lab dependencies

- An Azure subscription you will be using in this lab.
- A Microsoft account or an Azure AD 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 Azure AD tenant associated with that Azure subscription.
- The completed lab **Prepare for deployment of Azure Windows Virtual Desktop (AD DS)**

Estimated Time

60 minutes

Lab scenario

You need to create and configure host pools and session hosts in an Active Directory Domain Services (AD DS) environment.

Objectives

After completing this lab, you will be able to:

- Implement an Azure Windows Virtual Desktop environment in an AD DS domain
- Validate Azure Windows Virtual Desktop environment in an AD DS domain

Lab files

- None

Instructions

Exercise 1: Implement an Azure Windows Virtual Desktop environment in an AD DS domain

The main tasks for this exercise are as follows:

1. Prepare AD DS domain and the Azure subscription for deployment of an Azure Windows Virtual Desktop host pool
2. Deploy an Azure Windows Virtual Desktop host pool
3. Manage the Azure Windows Virtual Desktop host pool session hosts
4. Configure Windows Virtual Desktop application groups
5. Configure Windows Virtual Desktop workspaces

Task 1: Prepare AD DS domain and the Azure subscription for deployment of an Azure Windows Virtual Desktop host pool

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 **Virtual machines** and, from the **Virtual machines** blade, select **az140-dc-vm11**.
3. On the **az140-dc-vm11** blade, select **Connect**, in the drop-down menu, select **RDP**, on the **RDP** tab of the **az140-dc-vm11 | Connect** blade, in the **IP address** drop-down list, select the **Load balancer DNS name** entry, and then select **Download RDP File**.
4. When prompted, sign in with the following credentials:

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

1. Within the Remote Desktop session to **az140-dc-vm11**, start **Windows PowerShell ISE** as administrator.
2. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to create an organizational unit that will host the computer objects of the Windows Virtual Desktop hosts:

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

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

```
powershell Connect-AzAccount
```

1. When prompted, provide the credentials of the user account with the Owner role in the subscription you are using in this lab.
2. From the **Administrator: Windows PowerShell ISE** console, run the following to identify the user principal name of the **aduser1** account:

```
powershell (Get-AzADUser -DisplayName
'aduser1').UserPrincipalName
```

Note: Record the user principal name you identified in this step. You will need it later in this lab.

1. From the **Administrator: Windows PowerShell ISE** console, run the following to register the **Microsoft.DesktopVirtualization** resource provider:

```
powershell Register-AzResourceProvider -ProviderNamespace
Microsoft.DesktopVirtualization
```

1. Within the Remote Desktop session to **az140-dc-vm11**, start Microsoft Edge and navigate to the Azure portal. If prompted, sign in by using the Azure AD credentials of the user account with the Owner role in the subscription you are using in this lab.
2. Within the Remote Desktop session to **az140-dc-vm11**, 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 **Virtual networks** and, on the **Virtual networks** blade, select **az140-adds-vnet11**.
3. On the **az140-adds-vnet11** blade, select **Subnets**, on the **Subnets** blade, select **+ Subnet**, on the **Add subnet** blade, specify the following settings (leave all other settings with their default values) and click **Save**:

Setting	Value
Name	**hp1-Subnet**

Setting	Value
Subnet address range	**10.0.1.0/24**

Task 2: Deploy an Azure Windows Virtual Desktop host pool

1. Within the Remote Desktop session to **az140-dc-vm11**, in the Microsoft Edge window displaying the Azure portal, search for and select **Windows Virtual Desktop**, on the **Windows Virtual Desktop** blade, select **Host pools** and, on the **Windows Virtual Desktop | Host pools** blade, select **+ Add**.
2. On the **Basics** tab of the **Create a host pool** blade, specify the following settings and select **Next: Virtual Machines >**:

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	the name of a new resource group **az140-21-RG**
Host pool name	**az140-21-hp1**
Location	the name of the Azure region into which you deployed resources in the first exercise of this lab
Validation environment	**No**
Host pool type	**Pooled**
Max session limit	**50**
Load balancing algorithm	**Breadth-first**

1. On the **Virtual machines** tab of the **Create a host pool** blade, specify the following settings and select **Next: Workspace >**:

Setting	Value
Add virtual machines	**Yes**
Resource group	**Defaulted to same as host pool**

Setting	Value
Name prefix	**az140-21-p1**
Virtual machine location	the name of the Azure region into which you deployed resources in the first exercise of this lab
Availability options	No infrastructure redundancy required
Image type	**Gallery**
Image	**Windows 10 Enterprise multi-session, Version 20H2 + Microsoft 365 Apps**
Virtual machine size	**Standard D2s v3**
Number of VMs	**2**
OS disk type	**Standard SSD**
Virtual network	**az140-adds-vnet11**
Subnet	**hp1-Subnet (10.0.1.0/24)**
Network security group	**Basic**
Public inbound ports	**Yes**
Inbound ports to allow	**RDP**
Specify domain or unit	**Yes**
Domain to join	**adatum.com**
Organizational Unit path	**OU=WVDInfra,DC=adatum,DC=com**
AD domain join UPN	**student@adatum.com**
Password	**Pa55w.rd1234**
User name	Student
Password	Pa55w.rd1234
Confirm password	Pa55w.rd1234

1. On the **Workspace** tab of the **Create a host pool** blade, specify the following settings and select **Review + create**:

Setting	Value
Register desktop app group	**No**

1. On the **Review + create** tab of the **Create a host pool** blade, select **Create**.

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

Task 3: Manage the Azure Windows Virtual Desktop host pool session hosts

1. Within the Remote Desktop session to **az140-dc-vm11**, in the web browser window displaying the Azure portal, search for and select **Windows Virtual Desktop** and, on the **Windows Virtual Desktop** blade, in the vertical menu bar, in the **Manage section**, select **Host pools**.
2. On the **Windows Virtual Desktop | Host pools** blade, in the list of host pools, select **az140-21-hp1**.
3. On the **az140-21-hp1** blade, in the in the vertical menu bar, in the **Manage section**, select **Session hosts** and verify that the pool consists of two hosts.
4. On the **az140-21-hp1 | Session hosts** blade, select **+ Add**.
5. On the **Basics** tab of the **Add virtual machines to a host pool** blade, review the preconfigured settings and select **Next: Virtual Machines**.
6. On the **Virtual Machines** tab of the **Add virtual machines to a host pool** blade, specify the following settings and select **Review + create**:

Setting	Value
Resource group	**az140-21-RG**
Virtual machine location	the name of the Azure region into which you deployed the first two session host VMs
Number of VMs	**1**

Setting	Value
Image type	**Gallery**
Image	**Windows 10 Enterprise multi-session, Version 2004 + Microsoft 365 Apps**
Virtual network	**az140-adds-vnet11**
Subnet	**hp1-Subnet (10.0.1.0/24)**
Public IP	**Yes**
Configure SKU	**Basic**
Configure assignment	**Dynamic**
Network security group	**Basic**
Public inbound ports	**Yes**
Specify domain or unit	**Yes**
Domain to join	**adatum.com**
Organizational Unit path	**OU=WVDInfra,DC=adatum,DC=com**
AD domain join UPN	**student@adatum.com**
Password	**Pa55w.rd1234**

Note: As you likely noticed, it's possible to change the image and prefix of the VMs as you add session hosts to the existing pool. In general, this is not recommended unless you plan to replace all VMs in the pool.

1. On the **Review + create** tab of the **Add virtual machines to a host pool** blade, select **Create**

Note: Do not wait for the deployment to complete but instead proceed to the next task. The deployment might take about 5 minutes.

Task 4: Configure Windows Virtual Desktop application groups

1. Within the Remote Desktop session to **az140-dc-vm11**, in the web browser window displaying the Azure portal, search for and select

Windows Virtual Desktop and, on the **Windows Virtual Desktop** blade, select **Application groups**.
2. On the **Windows Virtual Desktop | Application groups** blade, note the existing, auto-generated **az140-21-hp1-DAG** desktop application group, and select it.
3. On the **az140-21-hp1-DAG** blade, select **Assignments**.
4. On the **az140-21-hp1-DAG | Assignments** blade, select **+ Add**.
5. On the **Select Azure AD users or user groups** blade, select **az140-wvd-pooled** and click **Select**.
6. Navigate back to the **Windows Virtual Desktop | Application groups** blade, select **+ Add**.
7. On the **Basics** tab of the **Create an application group** blade, specify the following settings and select **Next: Applications >**:

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	**az140-21-RG**
Host pool	**az140-21-hp1**
Application group type	**RemoteApp**
Application group name	**az140-21-hp1-Office365-RAG**

1. On the **Applications** tab of the **Create an application group** blade, select **+ Add applications**.
2. On the **Add application** blade, specify the following settings and select **Save**:

Setting	Value
Application source	**Start menu**
Application	**Word**
Description	**Microsoft Word**
Require command line	**No**

1. Back on the **Applications** tab of the **Create an application group** blade, select **+ Add applications**.
2. On the **Add application** blade, specify the following settings and select **Save**:

Setting	Value
Application source	**Start menu**
Application	**Excel**
Description	**Microsoft Excel**
Require command line	**No**

1. Back on the **Applications** tab of the **Create an application group** blade, select **+ Add applications**.
2. On the **Add application** blade, specify the following settings and select **Save**:

Setting	Value
Application source	**Start menu**
Application	**PowerPoint**
Description	**Microsoft PowerPoint**
Require command line	**No**

1. Back on the **Applications** tab of the **Create an application group** blade, select **Next: Assignments >**.
2. On the **Assignments** tab of the **Create an application group** blade, select **+ Add Azure AD users or user groups**.
3. On the **Select Azure AD users or user groups** blade, select **az140-wvd-remote-app** and click **Select**.
4. Back on the **Assignments** tab of the **Create an application group** blade, select **Next: Workspace >**.
5. On the **Workspace** tab of the **Create a workspace** blade, specify the following setting and select **Review + create**:

Setting	Value
Register application group	**No**

1. On the **Review + create** tab of the **Create an application group** blade, select **Create**.

Note: Wait for the Application Group to be created. This should take less than 1 minute.

Note: Next you will create an application group based on file path as the application source.

1. Within the Remote Desktop session to **az140-dc-vm11**, search for and select **Windows Virtual Desktop** and, on the **Windows Virtual Desktop** blade, select **Application groups**.
2. On the **Windows Virtual Desktop | Application groups** blade, select **+ Add**.
3. On the **Basics** tab of the **Create an application group** blade, specify the following settings and select **Next: Applications >**:

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	**az140-21-RG**
Host pool	**az140-21-hp1**
Application group type	**RemoteApp**
Application group name	**az140-21-hp1-Utilities-RAG**

1. On the **Applications** tab of the **Create an application group** blade, select **+ Add applications**.
2. On the **Add application** blade, specify the following settings and select **Save**:

Setting	Value
Application source	**File path**
Application path	**C:\Windows\system32\cmd.exe**
Application name	**Command Prompt**
Display name	**Command Prompt**
Icon path	**C:\Windows\system32\cmd.exe**
Icon index	**0**
Description	**Windows Command Prompt**
Require command line	**No**

1. Back on the **Applications** tab of the **Create an application group** blade, select **Next: Assignments >**.
2. On the **Assignments** tab of the **Create an application group** blade, select **+ Add Azure AD users or user groups**.
3. On the **Select Azure AD users or user groups** blade, select **az140-wvd-remote-app** and **az140-wvd-admins** and click **Select**.

4. Back on the **Assignments** tab of the **Create an application group** blade, select **Next: Workspace >**.
5. On the **Workspace** tab of the **Create a workspace** blade, specify the following setting and select **Review + create**:

Setting	Value
Register application group	**No**

1. On the **Review + create** tab of the **Create an application group** blade, select **Create**.

Task 5: Configure Windows Virtual Desktop workspaces

1. Within the Remote Desktop session to **az140-dc-vm11**, in the Microsoft Edge window displaying the Azure portal, search for and select **Windows Virtual Desktop** and, on the **Windows Virtual Desktop** blade, select **Workspaces**.
2. On the **Windows Virtual Desktop | Workspaces** blade, select **+ Add**.
3. On the **Basics** tab of the **Create a workspace** blade, specify the following settings and select **Next: Application groups >**:

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	**az140-21-RG**
Workspace name	**az140-21-ws1**
Friendly name	**az140-21-ws1**
Location	the name of the Azure region into which you deployed resources in the first exercise of this lab

1. On the **Application groups** tab of the **Create a workspace** blade, specify the following settings:

Setting	Value
Register desktop app group	**Yes**

1. On the **Workspace** tab of the **Create a workspace** blade, select **+ Register application groups**.
2. On the **Add application groups** blade, select the plus sign next to the **az140-21-hp1-DAG**, **az140-21-hp1-Office365-RAG**, and **az140-21-hp1-Utilities-RAG** entries and click **Select**.
3. Back on the **Application groups** tab of the **Create a workspace** blade, select **Review + create**.
4. On the **Review + create** tab of the **Create a workspace** blade, select **Create**.

Exercise 2: Validate Azure Windows Virtual Desktop environment

The main tasks for this exercise are as follows:

1. Install Microsoft Remote Desktop client (MSRDC) on a Windows 10 computer
2. Subscribe to a Windows Virtual Desktop workspace
3. Test Windows Virtual Desktop apps

Task 1: Install Microsoft Remote Desktop client (MSRDC) on a Windows 10 computer

1. Within the Remote Desktop session to **az140-dc-vm11**, in the browser window displaying the Azure portal, search for and select **Virtual machines** and, on the **Virtual machines** blade, select the **az140-cl-vm11** entry.
2. On the **az140-cl-vm11** blade, scroll down to the **Operations** section and select **Run Command**.
3. On the **az140-cl-vm11 | Run command** blade, select **EnableRemotePS** and select **Run**.

Note: Wait for the command to complete before you proceed to the next step. This might take about 1 minute.

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to add all members of the **ADATUM\az140-wvd-users** to the local **Remote Desktop Users** group on the Azure VM **az140-cl-vm11** running Windows 10 which you deployed in the lab

Prepare for deployment of Azure Windows Virtual Desktop (AD DS).

```
powershell $computerName = 'az140-cl-vm11' Invoke-Command -
ComputerName $computerName -ScriptBlock {Add-LocalGroupMember
-Group 'Remote Desktop Users' -Member 'ADATUM\az140-wvd-
users'}
```

1. Switch to your lab computer, from the lab computer, in the browser window displaying the Azure portal, search for and select **Virtual machines** and, on the **Virtual machines** blade, select the **az140-cl-vm11** entry.
2. On the **az140-cl-vm11** blade, select **Connect**, in the drop-down menu, select **RDP**, and then select **Download RDP File**.
3. When prompted, sign in as the **ADATUM\aduser1** user with **Pa55w.rd1234** as its password.
4. Within the Remote Desktop session to **az140-cl-vm11**, start Microsoft Edge and navigate to Windows Desktop client download page and, when prompted, select **Run** 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**. When prompted by User Account Control for administrative credentials, authenticate by using the **ADATUM\Student** username with **Pa55w.rd1234** as its password.
5. 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.

Task 2: Subscribe to a Windows Virtual Desktop workspace

1. In the **Remote Desktop** client window, select **Subscribe** and, when prompted, sign in with the **aduser1** credentials, by providing its userPrincipalName you identified earlier in this lab and **Pa55w.rd1234** as its password.

Note: Alternatively, in the **Remote Desktop** client window, select **Subscribe with URL**, in the **Subscribe to a Workspace** pane, in the **Email or Workspace URL**, type **https://rdweb.wvd.microsoft.com/api/arm/feeddiscovery**, select **Next**, and, once prompted, sign in with the **aduser1** credentials (using its userPrincipalName attribute as the user name and **Pa55w.rd1234** as its password).

1. In the **Stay signed in to all your apps** window, clear the checkbox **Allow my organization to manage my device** checkbox and select **No, sign in to this app only**.
2. Ensure that the **Remote Desktop** page displays the listing of applications that are included in the application groups published to the workspace and associated with the user account **aduser1** via its group membership.

Task 3: Test Windows Virtual Desktop apps

1. Within the Remote Desktop session to **az140-cl-vm11**, in the **Remote Desktop** client window, in the list of applications, double-click **Command Prompt** and verify that it launches a **Command Prompt** window. When prompted to authenticate, type **Pa55w.rd1234** as the password of the **aduser1** user account, select the checkbox **Remember me**, and select **OK**.

Note: Initially, it might take a few minutes for the application to start, but subsequently, the application startup should be much faster.

1. At the Command Prompt, type **hostname** and press the **Enter** key to display the name of the computer on which the Command Prompt is running.

Note: Verify that the displayed name is **az140-21-p1-0**, **az140-21-p1-1** or **az140-21-p1-2**, rather than **az140-cl-vm11**.

1. At the Command Prompt, type **logoff** and press the **Enter** key to log off from the current Remote App session.
2. Within the Remote Desktop session to **az140-cl-vm11**, in the **Remote Desktop** client window, in the list of applications, double-click **SessionDesktop** and verify that it launches a Remote Desktop session.
3. Within the **Default Desktop** session, right-click **Start**, select **Run**, in the **Open** text box of the **Run** dialog box, type **cmd** and select **OK**.
4. Within the **Default Desktop** session, at the Command Prompt, type **hostname** and press the **Enter** key to display the name of the computer on which the Remote Desktop session is running.
5. Verify that the displayed name is either **az140-21-p1-0**, **az140-21-p1-1** or **az140-21-p1-2**.

Exercise 3: Stop and deallocate Azure VMs provisioned in the lab

The main tasks for this exercise are as follows:

1. Stop and deallocate Azure VMs provisioned in the lab

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

Task 1: Deallocate Azure VMs provisioned 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 in this lab:

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

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

```
powershell Get-AzVM -ResourceGroup 'az140-21-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.

lab: title: 'Lab: Implement and manage storage for WVD (Azure AD DS)' module: 'Module 2: Implement a WVD Infrastructure'

Lab - Implement and manage storage for WVD (Azure AD DS)

Student lab manual

Lab dependencies

- An Azure subscription
- A Microsoft account or an Azure AD account with the Global Administrator role in the Azure AD 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 Windows Virtual Desktop (Azure AD DS)**

Estimated Time

30 minutes

Lab scenario

You need to implement and manage storage for a Windows Virtual Desktop deployment in an Azure Active Directory Domain Services (Azure AD DS) environment.

Objectives

After completing this lab, you will be able to:

- Configure Azure Files to store profile containers for Windows Virtual Desktop in Azure AD DS environment

Lab files

- None

Instructions

Exercise 1: Configure Azure Files to store profile containers for Windows Virtual Desktop

The main tasks for this exercise are as follows:

1. Create an Azure Storage account
2. Create an Azure Files share
3. Enable Azure AD DS authentication for the Azure Storage account
4. Configure the Azure Files share permissions
5. Configure the Azure Files directory and file level permissions

Task 1: Create an Azure Storage account

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. From your lab computer, in the Azure portal, search for and select **Virtual machines** and, from the **Virtual machines** blade, select the **az140-cl-vm11a** entry. This will open the **az140-cl-vm11a** blade.
3. On the **az140-cl-vm11a** blade, in the toolbar, select **Connect**, in the drop-down menu, select **RDP**, on the **RDP** tab of the **az140-cl-vm11a | Connect** blade, in the **IP address** drop-down list, select the **Public IP address** entry, and then select **Download RDP File**.
4. When prompted, sign in with the following credentials:

Setting	Value
User Name	**ADATUM\aadadmin1**
Password	**Pa55w.rd1234**

1. Within the Remote Desktop to the **az140-cl-vm11a** Azure VM, start Microsoft Edge, navigate to the Azure portal, and sign in by providing user principal name of the **aadadmin1** user account with **Pa55w.rd1234** as its password.

Note: You can identify the user principal name (UPN) attribute of the **aadadmin1** account by reviewing its properties dialog box from the Active Directory Users and Computers console or by switching

back to your lab computer and reviewing its properties from the Azure AD tenant blade in the Azure portal.

1. Within the Remote Desktop session to **az140-cl-vm11a**, in the Microsoft Edge window displaying the Azure portal, search for and select **Storage accounts** and, on the **Storage accounts** blade, select **+ Create**.
2. On the **Basics** tab of the **Create 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	the name of a new resource group **az140-22a-RG**
Storage account name	any globally unique name between 3 and 15 in length consisting of lower case letters and digits, starting with a letter
Location	the name of an Azure region hosting the Windows Virtual Desktop lab environment
Performance	**Standard**
Account kind	**StorageV2 (general purpose v2)**
Replication	**Locally redundant storage (LRS)**

Note: Make sure that the length of the storage account name does not exceed 15 characters. The name will be used to create a computer account in the Active Directory Domain Services (AD DS) domain that is integrated with the Azure AD tenant associated with the Azure subscription containing the storage account. This will allow for AD DS-based authentication when accessing file shares hosted in this storage account.

1. On the **Basics** tab of the **Create storage account** blade, select **Review + Create**, wait for the validation process to complete, and then select **Create**.

Note: Wait for the Storage account to be created. This should take about two minutes.

Task 2: Create an Azure Files share

1. Within the Remote Desktop session to **az140-cl-vm11a**, in the Microsoft Edge window displaying the Azure portal, navigate back to the **Storage accounts** blade and select the entry representing the newly created storage account.
2. On the storage account blade, in the vertical menu on the left side, in the **File services** section, select **File shares** and then select **+ File share**.
3. On the **New file share** blade, specify the following settings and select **Create** (leave other settings with their default values):

Setting	Value
Name	**az140-22a-profiles**

Task 3: Enable Azure AD DS authentication for the Azure Storage account

1. Within the Remote Desktop session to **az140-cl-vm11a**, in the Microsoft Edge window, in the Azure portal, on the blade displaying the properties of the storage account you created in the previous task, in the vertical menu on the left side, in the **Settings** section, select **Configuration**.
2. In the **Azure Active Directory Domain Services (Azure AD DS)** section, select the **Enabled** option, and select **Save**.

Task 4: Configure the Azure Files RBAC-based permissions

1. Within the Remote Desktop session to **az140-cl-vm11a**, in the Microsoft Edge window displaying the Azure portal, on the blade displaying properties of the storage account you created earlier in this exercise, in the vertical menu on the left side, in the **File services** section, select **File shares**, and in the list of shares, select the **az140-22a-profiles** entry.
2. On the **az140-22a-profiles** blade, in the vertical menu on the left side, select **Access Control (IAM)**.
3. On the **az140-22a-profiles | Access Control (IAM)** blade, select **+ Add** and, in the drop-down menu, select **Add role assignment**.
4. On the **Add role assignment** blade, specify the following settings and select **Save**:

Setting	Value
Role	**Storage File Data SMB Share Contributor**
Assign access to	**User, group, or service principal**
Select	**az140-wvd-ausers**

1. Back on the **Add role assignment** blade, specify the following settings and select **Save**:

Setting	Value
Role	**Storage File Data SMB Share Elevated Contributor**
Assign access to	**User, group, or service principal**
Select	**az140-wvd-aadmins**

Note: You will use the **aadadmin1** user account, which is a member of the **az140-wvd-aadmins** group to configure file share permissions.

Task 5: Configure the Azure Files directory and file level permissions

1. Within the Remote Desktop session to **az140-cl-vm11a**, start **Command Prompt** and, from the **Command Prompt** window, run the following to map a drive to the target share (replace the `<storage-account-name>` placeholder with the name of the storage account):

```
cmd net use Z: \\<storage-account-
name>.file.core.windows.net\az140-22a-profiles
```

1. Within the Remote Desktop session to **az140-cl-vm11a**, open File Explorer, navigate to the newly mapped Z: drive, display its **Properties** dialog box, select the **Security** tab, select **Edit**, select **Add**, in the **Select Users, Computers, Service Accounts, and Groups** dialog box, ensure that the **From this location** textbox contains the **adatum.com** entry, in the **Enter the object name to select** textbox, type **az140-wvd-ausers** and click **OK**.
2. Back on the **Security** tab of the dialog box displaying permissions of the mapped drive, ensure that the **az140-wvd-ausers** entry is selected, select the **Modify** checkbox in the **Allow** column, click

OK, review the message displayed in the **Windows Security** text box, and click **Yes**.

3. Back on the **Security** tab of the dialog box displaying permissions of the mapped drive, select **Edit**, select **Add**, in the **Select Users, Computers, Service Accounts, and Groups** dialog box, ensure that the **From this location** textbox contains the **adatum.com** entry, in the **Enter the object name to select** textbox, type **az140-wvd-aadmins** and click **OK**.
4. Back on the **Security** tab of the dialog box displaying permissions of the mapped drive, ensure that the **az140-wvd-aadmins** entry is selected, select the **Full control** checkbox in the **Allow** column, and click **OK**.
5. On the **Security** tab of the dialog box displaying permissions of the mapped drive, select **Edit**, in the list of groups and user names, select the **Authenticated users** entry, and select **Remove**.
6. On the **Security** tab of the dialog box displaying permissions of the mapped drive, select **Edit**, in the list of groups and user names, select the **Users** entry, select **Remove**, click **OK**, and then click **OK** twice to complete the process.

>Note: Alternatively, you could set permissions by using the icacls command-line utility.

lab: title: 'Lab: Implement and manage storage for WVD (AD DS)' module: 'Module 2: Implement a WVD Infrastructure'

Lab - Implement and manage storage for WVD (AD DS)

Student lab manual

Lab dependencies

- An Azure subscription you will be using in this lab.
- A Microsoft account or an Azure AD 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 Azure AD tenant associated with that Azure subscription.
- The completed lab **Prepare for deployment of Azure Windows Virtual Desktop (AD DS)**

Estimated Time

30 minutes

Lab scenario

You need to implement and manage storage for a Windows Virtual Desktop deployment in an Azure Active Directory Domain Services (Azure AD DS) environment.

Objectives

After completing this lab, you will be able to:

- Configure Azure Files to store profile containers for Windows Virtual Desktop

Lab files

- None

Instructions

Exercise 1: Configure Azure Files to store profile containers for Windows Virtual Desktop

The main tasks for this exercise are as follows:

1. Create an Azure Storage account
2. Create an Azure Files share
3. Enable AD DS authentication for the Azure Storage account
4. Configure the Azure Files RBAC-based permissions
5. Configure the Azure Files file system permissions

Task 1: Create an Azure Storage account

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 **Virtual machines** and, from the **Virtual machines** blade, select **az140-dc-vm11**.
3. On the **az140-dc-vm11** blade, select **Connect**, in the drop-down menu, select **RDP**, on the **RDP** tab of the **az140-dc-vm11 | Connect** blade, in the **IP address** drop-down list, select the **Load balancer DNS name** entry, and then select **Download RDP File**.
4. When prompted, sign in with the following credentials:

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

1. Within the Remote Desktop session to **az140-dc-vm11**, start Microsoft Edge and navigate to the Azure portal. If prompted, sign in by using the Azure AD credentials of the user account with the Owner role in the subscription you are using in this lab.
2. Within the Remote Desktop session to **az140-dc-vm11**, in the Microsoft Edge window displaying the Azure portal, search for and select **Storage accounts** and, on the **Storage accounts** blade, select **+ Add**.

3. On the **Basics** tab of the **Create 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	the name of a new resource group **az140-22-RG**
Storage account name	any globally unique name between 3 and 15 in length consisting of lower case letters and digits, starting with a letter
Location	the name of an Azure region hosting the Windows Virtual Desktop lab environment
Performance	**Standard**
Account kind	**StorageV2 (general purpose v2)**
Replication	**Read-access geo-redundant storage (RA-GRS)**

Note: Make sure that the length of the storage account name does not exceed 15 characters. The name will be used to create a computer account in the Active Directory Domain Services (AD DS) domain that is integrated with the Azure AD tenant associated with the Azure subscription containing the storage account. This will allow for AD DS-based authentication when accessing file shares hosted in this storage account.

1. On the **Basics** tab of the **Create storage account** blade, select **Review + Create**, wait for the validation process to complete, and then select **Create**.

Note: Wait for the Storage account to be created. This should take about 2 minutes.

Task 2: Create an Azure Files share

1. Within the Remote Desktop session to **az140-dc-vm11**, in the Microsoft Edge window displaying the Azure portal, navigate back to the **Storage accounts** blade and select the entry representing the newly created storage account.

2. On the storage account blade, in the **File services** section, select **File shares** and then select **+ File share**.
3. On the **New file share** blade, specify the following settings and select **Create** (leave other settings with their default values):

Setting	Value
Name	**az140-22-profiles**
Tiers	**Transaction optimized**

Task 3: Enable AD DS authentication for the Azure Storage account

1. Within the Remote Desktop session to **az140-dc-vm11**, open another tab in the Microsoft Edge window, navigate to the Azure Files samples GitHub repository, download [the most recent version of the compressed **AzFilesHybid.zip** PowerShell module, and extract its content into **C:\Allfiles\Labs\02** folder (create the folder if needed).
2. Within the Remote Desktop session to **az140-dc-vm11**, start **Windows PowerShell ISE** as administrator and, from the **Administrator: Windows PowerShell ISE** script pane, run the following to remove the **Zone.Identifier** alternate data stream, which has a value of **3**, indicating that it was downloaded from the Internet:

```
powershell Get-ChildItem -Path C:\Allfiles\Labs\02 -File -
Recurse | Unblock-File
```

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

```
powershell Connect-AzAccount
```

1. When prompted, sign in with the Azure AD credentials of the user account with the Owner role in the subscription you are using in this lab.
2. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to set the variables necessary to run the subsequent script:

```
powershell $subscriptionId = (Get-AzContext).Subscription.Id
$resourceGroupName = 'az140-22-RG' $storageAccountName =
(Get-AzStorageAccount -ResourceGroupName $resourceGroupName)
[0].StorageAccountName
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to create an AD DS computer object that represents the Azure Storage account you created earlier in this task and is used to implement its AD DS authentication:

```
powershell Set-Location -Path 'C:\Allfiles\Labs\02'
.\CopyToPSPath.ps1 Import-Module -Name AzFilesHybrid Join-
AzStorageAccountForAuth ` -ResourceGroupName
$ResourceGroupName ` -StorageAccountName $StorageAccountName
` -DomainAccountType 'ComputerAccount' ` -
OrganizationalUnitDistinguishedName
'OU=WVDInfra,DC=adatum,DC=com'
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to verify that the AD DS authentication is enabled on the Azure Storage account:

```
powershell $storageaccount = Get-AzStorageAccount -
ResourceGroupName $resourceGroupName -Name
$storageAccountName
$storageAccount.AzureFilesIdentityBasedAuth.ActiveDirectoryPr
operties
$storageAccount.AzureFilesIdentityBasedAuth.DirectoryServiceO
ptions
```

1. Verify that that the output of the command `$storageAccount.AzureFilesIdentityBasedAuth.ActiveDirectoryProperties` returns `AD`, representing the directory service of the storage account, and that the output of the `$storageAccount.AzureFilesIdentityBasedAuth.DirectoryServiceOptions` command, representing the directory domain information, resembles the following format (the values of `DomainGuid`, `DomainSid`, and `AzureStorageSid` will differ):

```
DomainName : adatum.com NetBiosDomainName : adatum.com
ForestName : adatum.com DomainGuid : 47c93969-9b12-4e01-ab81-
1508cae3ddc8 DomainSid : S-1-5-21-1102940778-2483248400-
1820931179 AzureStorageSid : S-1-5-21-1102940778-2483248400-
1820931179-2109
```

1. Within the Remote Desktop session to **az140-dc-vm11**, switch to the Microsoft Edge window displaying the Azure portal, on the blade displaying the storage account properties, in the **Settings** section of the vertical menu, select **Configuration**.

2. On the configuration blade of the storage account, ensure that **Active Directory Domain Services (AD DS)** option is set to **Enabled** and that **Joined domain** entry is set to **adatum.com**.

Note: You might have to refresh the browser page for the change to be reflected within the Azure portal.

Task 4: Configure the Azure Files RBAC-based permissions

1. Within the Remote Desktop session to **az140-dc-vm11**, in the Microsoft Edge window displaying the Azure portal, on the blade displaying properties of the storage account you created earlier in this exercise, in the vertical menu on the left side, in the **File Services** section, select **File shares**.
2. On the **File shares** blade, in the list of shares, select the **az140-22-profiles** entry.
3. On the **az140-22-profiles** blade, in the vertical menu on the left side, select **Access Control (IAM)**.
4. On the **Access Control (IAM)** blade of the storage account, select **+ Add** and, in the drop-down menu, select **Add role assignment**,
5. On the **Add role assignment** blade, specify the following settings and select **Save**:

Setting	Value
Role	**Storage File Data SMB Share Contributor**
Assign access to	**User, group, or service principal**
Select	**az140-wvd-users**

1. On the **Access Control (IAM)** blade of the storage account, select **+ Add** and, in the drop-down menu, select **Add role assignment**,
2. On the **Add role assignment** blade, specify the following settings and select **Save**:

Setting	Value
Role	**Storage File Data SMB Share Elevated Contributor**
Assign access to	**User, group, or service principal**
Select	**az140-wvd-admins**

Task 5: Configure the Azure Files file system permissions

1. Within the Remote Desktop session to **az140-dc-vm11**, switch to the **Administrator: Windows PowerShell ISE** window and, from the **Administrator: Windows PowerShell ISE** script pane, run the following to create a variable referencing the name and key of the storage account you created earlier in this exercise:

```
powershell $resourceGroupName = 'az140-22-RG' $storageAccount
= (Get-AzStorageAccount -ResourceGroupName
$resourceGroupName)[0] $storageAccountName =
$storageAccount.StorageAccountName $storageAccountKey = (Get-
AzStorageAccountKey -ResourceGroupName $resourceGroupName -
Name $storageAccountName).Value[0]
```

1. From the **Administrator: Windows PowerShell ISE** script pane, run the following to create a drive mapping to the file share you created earlier in this exercise:

```
powershell $fileShareName = 'az140-22-profiles' net use Z:
"\\$storageAccountName.file.core.windows.net\$fileShareName"
/u:AZURE\$storageAccountName $storageAccountKey
```

1. From the **Administrator: Windows PowerShell ISE** console, run the following to view the current file system permissions:

```
powershell icacls Z:
```

Note: By default, both **NT Authority\Authenticated Users** and **BUILTIN\Users** have permissions that would allow users read other users' profile containers. You will remove them and add minimum required permissions instead.

1. From the **Administrator: Windows PowerShell ISE** script pane, run the following to adjust the file system permissions to comply with the principle of least privilege:

```
powershell $permissions = 'ADATUM\az140-wvd-admins'+':(F)'
cmd /c icacls Z: /grant $permissions $permissions =
'ADATUM\az140-wvd-users'+':(M)' cmd /c icacls Z: /grant
$permissions $permissions = 'Creator Owner'+':(OI)(CI)(IO)
(M)' cmd /c icacls Z: /grant $permissions icacls Z: /remove
'Authenticated Users' icacls Z: /remove 'Builtin\Users'
```

>Note: Alternatively, you could set permissions by using File Explorer.

lab: title: 'Lab: Deploy host pools and hosts by using Azure Resource Manager templates' module: 'Module 2: Implement a WVD Infrastructure'

Lab - Deploy host pools and hosts by using Azure Resource Manager templates

Student lab manual

Lab dependencies

- An Azure subscription you will be using in this lab.
- A Microsoft account or an Azure AD 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 Azure AD tenant associated with that Azure subscription.
- The completed lab **Prepare for deployment of Azure Windows Virtual Desktop (AD DS)** or **Prepare for deployment of Azure Windows Virtual Desktop (Azure AD DS)**
- The completed lab **Deploy host pools and session hosts by using the Azure portal (AD DS)** or **Deploy host pools and session hosts by using the Azure portal (Azure AD DS)**

Estimated Time

45 minutes

Lab scenario

You need to automate deployment of Windows Virtual Desktop host pools and hosts by using Azure Resource Manager templates.

Objectives

After completing this lab, you will be able to:

- Deploy Azure Windows Virtual Desktop host pools and hosts by using Azure Resource Manager templates

Lab files

- \\AZ-140\AllFiles\Labs\02\az140-23_azuredeployhp23.parameters.json
- \\AZ-140\AllFiles\Labs\02\az140-23_azuremodifyhp23.parameters.json

Instructions

Exercise 1: Deploy Azure Windows Virtual Desktop host pools and hosts by using Azure Resource Manager templates

The main tasks for this exercise are as follows:

1. Prepare for deployment of an Azure Windows Virtual Desktop host pool by using an Azure Resource Manager template
2. Deploy an Azure Windows Virtual Desktop host pool and hosts by using an Azure Resource Manager template
3. Verify deployment of the Azure Windows Virtual Desktop host pool and hosts
4. Prepare for adding of hosts to the existing Azure Windows Virtual Desktop host pool by using an Azure Resource Manager template
5. Add hosts to the existing Azure Windows Virtual Desktop host pool by using an Azure Resource Manager template
6. Verify changes to the Azure Windows Virtual Desktop host pool
7. Manage personal desktop assignments in the Azure Windows Virtual Desktop host pool

Task 1: Prepare for deployment of an Azure Windows Virtual Desktop host pool 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, search for and select **Virtual machines** and, from the **Virtual machines** blade, select **az140-dc-vm11**.
3. On the **az140-dc-vm11** blade, select **Connect**, in the drop-down menu, select **RDP**, on the **RDP** tab of the **az140-dc-vm11 | Connect** blade, in the **IP address** drop-down list, select the **Load balancer DNS name** entry, and then select **Download RDP File**.
4. When prompted, sign in with the following credentials:

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

1. Within the Remote Desktop session to **az140-dc-vm11**, start **Windows PowerShell ISE** as administrator.
2. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to identify the distinguished name of the organizational unit named **WVDInfra** that will host the computer objects of the Windows Virtual Desktop pool hosts:

```
powershell (Get-ADOrganizationalUnit -Filter "Name -eq
'WVDInfra'").distinguishedName
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to identify the user principal name attribute of the **ADATUM\Student** account that you will use to join the Windows Virtual Desktop hosts to the AD DS domain (**student@adatum.com**):

```
powershell (Get-ADUser -Filter "sAMAccountName -eq
'student'").userPrincipalName
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to identify the user principal name of the **ADATUM\aduser7** and **ADATUM\aduser8** accounts that you will use to test personal desktop assignments later in this lab:

```
powershell (Get-ADUser -Filter "sAMAccountName -eq
'aduser7'").userPrincipalName (Get-ADUser -Filter
"sAMAccountName -eq 'aduser8'").userPrincipalName
```

Note: Record all user principal name values you identified. You will need them later in this lab.

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to calculate the token expiration time necessary to perform a template-based deployment:

```
powershell $((get-
date).ToUniversalTime().AddDays(1).ToString('yyyy-MM-
ddTHH:mm:ss.fffffffZ'))
```

Note: The value should resemble the format `2020-12-27T00:51:28.3008055Z`. Record it since you will need it in the next task.

Note: A registration token is required to authorize a host to join the pool. The value of token's expiration date must be between one hour and one month from the current date and time.

1. Within the Remote Desktop session to **az140-dc-vm11**, start Microsoft Edge and navigate to the Azure portal. If prompted, sign in by using the Azure AD credentials of the user account with the Owner role in the subscription you are using in this lab.
2. Within the Remote Desktop session to **az140-dc-vm11**, 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 **Virtual networks** and, on the **Virtual networks** blade, select **az140-adds-vnet11**.
3. On the **az140-adds-vnet11** blade, select **Subnets**, on the **Subnets** blade, select **+ Subnet**, on the **Add subnet** blade, specify the following settings (leave all other settings with their default values) and click **Save**:

Setting	Value
Name	**hp2-Subnet**
Subnet address range	**10.0.2.0/24**

1. Within the Remote Desktop session to **az140-dc-vm11**, 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 **Network security groups** and, on the **Network security groups** blade, select the network security group in the **az140-11-RG** resource group.
2. On the network security group blade, in the vertical menu on the left, in the **Settings** section, click **Properties**.
3. On the **Properties** blade, click the **Copy to clipboard** icon on the right side of the **Resource ID** textbox.

Note: The value should resemble the format `/subscriptions/de8279a3-0675-40e6-91e2-5c3728792cb5/resourceGroups/az140-11-RG/providers/Microsoft.Network/networkSecurityGroups/az14`

`0-cl-vm11-nsg`, although the subscription ID will differ. Record it since you will need it in the next task.

Task 2: Deploy an Azure Windows Virtual Desktop host pool and hosts 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. From your lab computer, in the same web browser window, open another web browser tab and navigate to the GitHub Azure RDS templates repository page ARM Template to Create and provision new Windows Virtual Desktop hostpool.
3. On the **ARM Template to Create and provision new Windows Virtual Desktop hostpool** page, select **Deploy to Azure**. This will automatically redirect the browser to the **Custom deployment** blade in the Azure portal.
4. On the **Custom deployment** blade, select **Edit parameters**.
5. On the **Edit parameters** blade, select **Load file**, in the **Open** dialog box, select **\\AZ-140\AllFiles\Labs\02\az140-23_azuredeployhp23.parameters.json**, select **Open**, and then select **Save**.
6. Back on the **Custom deployment** 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	the name of a new resource group **az140-23-RG**
Location	the name of the Azure region into which you deployed Azure VMs hosting AD DS domain controllers in the lab **Prepare for deployment of Azure Windows Virtual Desktop (AD DS)**
Workspace location	the name of the same Azure region as the one set as the value of the **Location** parameters
Workspace Resource Group	none, since, if null, its value will be automatically set to match the deployment target resource group

Setting	Value
All Application Group Reference	none, since there are no existing application groups in the target workspace (there is no workspace)
Vm location	the name of the same Azure region as the one set as the value of the **Location** parameters
Create Network Security Group	**false**
Network Security Group Id	the value of the resourceID parameter of the existing network security group you identified in the previous task
Token Expiration Time	the value of the token expiration time you calculated in the previous task

Note: The deployment provisions a pool with personal desktop assignment type.

1. On the **Custom deployment** blade, select **Review + create** and select **Create**.

Note: Wait for the deployment to complete before you proceed to the next task. This might take about 15 minutes.

Task 3: Verify deployment of the Azure Windows Virtual Desktop host pool and hosts

1. From your lab computer, in the web browser displaying the Azure portal, search for and select **Windows Virtual Desktop**, on the **Windows Virtual Desktop** blade, select **Host pools** and, on the **Windows Virtual Desktop | Host pools** blade, select the entry **az140-23-hp2** representing the newly deployed pool.
2. On the **az140-23-hp2** blade, in the vertical menu on the left side, in the **Manage** section, click **Session hosts**.
3. On the **az140-23-hp2 | Session hosts** blade, verify that the deployment consists of two hosts.

4. On the **az140-23-hp2 | Session hosts** blade, in the vertical menu on the left side, in the **Manage** section, click **Application groups**.
5. On the **az140-23-hp2 | Application groups** blade, verify that the deployment includes the **Default Desktop** application group named **az140-23-hp2-DAG**.

Task 4: Prepare for adding of hosts to the existing Azure Windows Virtual Desktop host pool by using an Azure Resource Manager template

1. From your lab computer, switch to the Remote Desktop session to **az140-dc-vm11**.
2. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to generate the token necessary to join new hosts to the pool you provisioned earlier in this exercise:

```
powershell $registrationInfo = New-AzWvdRegistrationInfo -
ResourceGroupName 'az140-23-RG' -HostPoolName 'az140-23-hp2'
-ExpirationTime $((get-
date).ToUniversalTime().AddDays(1).ToString('yyyy-MM-
ddTHH:mm:ss.fffffffZ'))
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to retrieve the value of the token and paste it into Clipboard:

```
powershell $registrationInfo.Token | clip
```

Note: Record the value copied into Clipboard since you will need it in the next task. Make sure to that the value you are using includes a single line of text, without any line breaks.

Note: A registration token is required to authorize a host to join the pool. The value of token's expiration date must be between one hour and one month from the current date and time.

Task 5: Add hosts to the existing Azure Windows Virtual Desktop host pool by using an Azure Resource Manager template

1. From your lab computer, in the same web browser window, open another web browser tab and navigate to the GitHub Azure RDS templates repository page ARM Template to Add sessionhosts to an existing Windows Virtual Desktop hostpool.
2. On the **ARM Template to Add sessionhosts to an existing Windows Virtual Desktop hostpool** page, select **Deploy to Azure**. This will automatically redirect the browser to the **Custom deployment** blade in the Azure portal.
3. On the **Custom deployment** blade, select **Edit parameters**.
4. On the **Edit parameters** blade, select **Load file**, in the **Open** dialog box, select **\\AZ-140\AllFiles\Labs\02\az140-23_azuremodifyhp23.parameters.json**, select **Open**, and then select **Save**.
5. Back on the **Custom deployment** 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	**az140-23-RG**
Hostpool Token	the value of the token you generated in the previous task
Hostpool Location	the name of the Azure region into which you deployed the hostpool earlier in this lab
Vm Administrator Account Username	**student**
Vm Administrator Account Password	**Pa55w.rd1234**
Vm location	the name of the same Azure region as the one set as the value of the **Hostpool Location** parameters
Create Network Security Group	**false**
Network Security Group Id	the value of the resourceID parameter of the existing network security group you identified in the previous task

1. On the **Custom deployment** blade, select **Review + create** and select **Create**.

Note: Wait for the deployment to complete before you proceed to the next task. This might take about 5 minutes.

Task 6: Verify changes to the Azure Windows Virtual Desktop host pool

1. From your lab computer, in the web browser displaying the Azure portal, search for and select **Virtual machines** and, on the **Virtual machines** blade, note that the list includes an additional virtual machine named **az140-23-p2-2**.
2. From your lab computer, switch to the Remote Desktop session to **az140-dc-vm11**.
3. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to verify that the third host was successfully joined to the **adatum.com** AD DS domain:

```
powershell Get-ADComputer -Filter "sAMAccountName -eq 'az140-23-p2-2$'"
```

1. Switch back to your lab computer, in the web browser displaying the Azure portal, search for and select **Windows Virtual Desktop**, on the **Windows Virtual Desktop** blade, select **Host pools** and, on the **Windows Virtual Desktop | Host pools** blade, select the entry **az140-23-hp2** representing the newly modified pool. 1. On the **az140-23-hp2** blade, review the **Essentials** section and verify that the **Host pool type** is set to **Personal** with the **Assignment type** set to **Automatic**. 1. On the **az140-23-hp2** blade, in the vertical menu on the left side, in the **Manage** section, click **Session hosts**. 1. On the **az140-23-hp2 | Session hosts** blade, verify that the deployment consists of three hosts.

Task 7: Manage personal desktop assignments in the Azure Windows Virtual Desktop host pool

1. On your lab computer, in the web browser displaying the Azure portal, on the **az140-23-hp2 | Session hosts** blade, in the vertical menu on the left side, in the **Manage** section, select **Application groups**.

2. On the **az140-23-hp2 | Application groups** blade, in the list of application groups select **az140-23-hp2-DAG**.
3. On the **az140-23-hp2-DAG** blade, in the vertical menu on the left, select **Assignments**.
4. On the **az140-23-hp2-DAG | Assignments** blade, select **+ Add**.
5. On the **Select Azure AD users or user groups** blade, select **az140-wvd-personal** and click **Select**.

Note: Now let's review the experience of a user connecting to the Windows Virtual Desktop host pool.

1. From your lab computer, in the browser window displaying the Azure portal, search for and select **Virtual machines** and, on the **Virtual machines** blade, select the **az140-cl-vm11** entry.
2. On the **az140-cl-vm11** blade, select **Connect**, in the drop-down menu, select **RDP**, select the Public IP address, and then select **Download RDP File**.
3. When prompted, sign in as the **ADATUM\aduser7** user with **Pa55w.rd1234** as its password.
4. In the **Stay signed in to all your apps** window, clear the checkbox **Allow my organization to manage my device** checkbox and select **No, sign in to this app only**.
5. Within the Remote Desktop session to **az140-cl-vm11**, click **Start** and, in the **Start** menu, select the **Remote Desktop** client app.
6. In the **Remote Desktop** client window, select **Subscribe** and, when prompted, sign in with the **aduser7** credentials, by providing its userPrincipalName and **Pa55w.rd1234** as its password.

Note: Alternatively, in the **Remote Desktop** client window, select **Subscribe with URL**, in the **Subscribe to a Workspace** pane, in the **Email or Workspace URL**, type **https://rdweb.wvd.microsoft.com/api/arm/feeddiscovery**, select **Next**, and, once prompted, sign in with the **aduser7** credentials (using its userPrincipalName attribute as the user name and **Pa55w.rd1234** as its password).

1. On the **Remote Desktop** page, double-click the **SessionDesktop** icon, when prompted for credentials, type **Pa55w.rd1234**, select the **Remember me** checkbox, and click **OK**.
2. Verify that **aduser7** successfully signed in via Remote Desktop to a host.

3. Within the Remote Desktop session to one of the hosts as **aduser7**, right-click **Start**, in the right-click menu, select **Shut down or sign out** and, in the cascading menu, click **Sign out**.

Note: Now let's switch the personal desktop assignment from the direct mode to automatic.

1. Switch to your lab computer, to the web browser displaying the Azure portal, on the **az140-23-hp2-DAG | Assignments** blade, in the informational bar directly above the list of assignments, click the **Assign VM** link. This will redirect you to the **az140-23-hp2 | Session hosts** blade.
2. On the **az140-23-hp2 | Session hosts** blade, verify that one of the hosts has **aduser7** listed in the **Assigned User** column.

Note: This is expected since the host pool is configured for automatic assignment.

1. On your lab computer, 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 switch to the direct assignment mode:

```
powershell Update-AzWvdHostPool -ResourceGroupName
'az140-23-RG' -Name 'az140-23-hp2' -
PersonalDesktopAssignmentType Direct
```

3. On your lab computer, in the web browser window displaying the Azure portal, navigate to the **az140-23-hp2** host pool blade, review the **Essentials** section and verify that the **Host pool type** is set to **Personal** with the **Assignment type** set to **Direct**.

4. Switch back to the Remote Desktop session to **az140-cl-vm11**, in the **Remote Desktop** window, click the ellipsis icon in the upper right corner next to the **Windows Virtual Desktop** label, in the dropdown menu, click **Unsubscribe**, and, when prompted for confirmation, click **Continue**.
5. Within the Remote Desktop session to **az140-cl-vm11**, in the **Remote Desktop** window, on the **Let's get started** page, click **Subscribe**.

6. When prompted to sign in, on the **Pick an account** pane, click **Use another account**, and, when prompted, sign in by using the user principal name of the **aduser8** user account with **Pa55w.rd1234** as the password.
7. In the **Stay signed in to all your apps** window, clear the checkbox **Allow my organization to manage my device** checkbox and select **No, sign in to this app only**.
8. On the **Remote Desktop** page, double-click the **SessionDesktop** icon, verify that you receive an error message stating **We couldn't connect because there are currently no available resources. Try again later or contact tech support for help if this keeps happening**, and click **OK**.

Note: This is expected since the host pool is configured for direct assignment and **aduser8** has not been assigned a host.

1. Switch to your lab computer, to the web browser displaying the Azure portal and, on the **az140-23-hp2 | Session hosts** blade, select the **(Assign)** link in the **Assigned User** column next to one of the two remaining unassigned hosts.
2. On the **Assign a user**, select **aduser8**, click **Select** and, when prompted for confirmation, click **OK**.
3. Switch back to the Remote Desktop session to **az140-cl-vm11**, in the **Remote Desktop** window, double-click the **SessionDesktop** icon, when prompted for the password, type **Pa55w.rd1234**, click **OK**, and verify that you can successfully sign in to the assigned host.

Exercise 2: Stop and deallocate Azure VMs provisioned in the lab

The main tasks for this exercise are as follows:

1. Stop and deallocate Azure VMs provisioned in the lab

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

Task 1: Deallocate Azure VMs provisioned 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 in this lab:

```
powershell Get-AzVM -ResourceGroup 'az140-23-RG'
```

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

```
powershell Get-AzVM -ResourceGroup 'az140-23-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.

lab: title: 'Lab: Deploy and manage host pools and hosts by using PowerShell' module: 'Module 2: Implement a WVD Infrastructure'

Lab - Deploy and manage host pools and hosts by using PowerShell

Student lab manual

Lab dependencies

- An Azure subscription you will be using in this lab.
- A Microsoft account or an Azure AD 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 Azure AD tenant associated with that Azure subscription.
- The completed lab **Prepare for deployment of Azure Windows Virtual Desktop (AD DS)** or **Prepare for deployment of Azure Windows Virtual Desktop (Azure AD DS)**

Estimated Time

60 minutes

Lab scenario

You need to automate deployment of Windows Virtual Desktop host pools and hosts by using PowerShell in an Active Directory Domain Services (AD DS) environment.

Objectives

After completing this lab, you will be able to:

- Deploy Azure Windows Virtual Desktop host pools and hosts by using PowerShell
- Add hosts to the Windows Virtual Desktop host pool by using PowerShell

Lab files

- \\AZ-140\AllFiles\Labs\02\az140-24_azuredeployhp3.json
- \\AZ-140\AllFiles\Labs\02\az140-24_azuredeployhp3.parameters.json

Instructions

Exercise 1: Implement Azure Windows Virtual Desktop host pools and session hosts by using PowerShell

The main tasks for this exercise are as follows:

1. Prepare for deployment of Windows Virtual Desktop host pool by using PowerShell
2. Create a Windows Virtual Desktop host pool by using PowerShell
3. Perform a template-based deployment of an Azure VM running Windows 10 Enterprise by using PowerShell
4. Add an Azure VM running Windows 10 Enterprise as a session host to the Windows Virtual Desktop host pool by using PowerShell
5. Verify the deployment of the Azure Windows Virtual Desktop session host

Task 1: Prepare for deployment of Windows Virtual Desktop host pool by using PowerShell

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 **Virtual machines** and, from the **Virtual machines** blade, select **az140-dc-vm11**.
3. On the **az140-dc-vm11** blade, select **Connect**, in the drop-down menu, select **RDP**, on the **RDP** tab of the **az140-dc-vm11 | Connect** blade, in the **IP address** drop-down list, select the **Load balancer DNS name** entry, and then select **Download RDP File**.
4. When prompted, sign in with the following credentials:

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

1. Within the Remote Desktop session to **az140-dc-vm11**, start **Windows PowerShell ISE** as administrator.
2. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the

following to identify the distinguished name of the organizational unit named **WVDInfra** that will host the computer objects of the Windows Virtual Desktop pool session hosts:

```
powershell (Get-ADOrganizationalUnit -Filter "Name -eq
'WVDInfra'").distinguishedName
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to identify the UPN suffix of the **ADATUM\Student** account that you will use to join the Windows Virtual Desktop hosts to the AD DS domain (**student@adatum.com**):

```
powershell (Get-ADUser -Filter {sAMAccountName -eq 'student'}
-Properties userPrincipalName).userPrincipalName
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to install the DesktopVirtualization PowerShell module (when prompted, click **Yes to All**):

```
powershell Install-Module -Name Az.DesktopVirtualization -
Force
```

Note: Ignore any warnings regarding existing PowerShell modules in use.

1. Within the Remote Desktop session to **az140-dc-vm11**, start Microsoft Edge and navigate to the Azure portal. If prompted, sign in by using the Azure AD credentials of the user account with the Owner role in the subscription you are using in this lab.
2. Within the Remote Desktop session to **az140-dc-vm11**, 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 **Virtual networks** and, on the **Virtual networks** blade, select **az140-adds-vnet11**.
3. On the **az140-adds-vnet11** blade, select **Subnets**, on the **Subnets** blade, select **+ Subnet**, on the **Add subnet** blade, specify the following settings (leave all other settings with their default values) and click **Save**:

Setting	Value
Name	**hp3-Subnet**

Setting	Value
Subnet address range	**10.0.3.0/24**

1. Within the Remote Desktop session to **az140-dc-vm11**, 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 **Network security groups** and, on the **Network security groups** blade, select the security group in the **az140-11-RG** resource group.
2. On the network security group blade, in the vertical menu on the left, in the **Settings** section, click **Properties**.
3. On the **Properties** blade, click the **Copy to clipboard** icon on the right side of the **Resource ID** textbox.

Note: The value should resemble the format `/subscriptions/de8279a3-0675-40e6-91e2-5c3728792cb5/resourceGroups/az140-11-RG/providers/Microsoft.Network/networkSecurityGroups/az140-cl-vm11-nsg`, although the subscription ID will differ. Record it since you will need it in the next task.

Task 2: Create a Windows Virtual Desktop host pool by using PowerShell

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to sign in to your Azure subscription:

```
powershell Connect-AzAccount
```

1. When prompted, provide the credentials of the user account with the Owner role in the subscription you are using in this lab.
2. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to identify the Azure region hosting the Azure virtual network **az140-adds-vnet11**:

```
powershell $location = (Get-AzVirtualNetwork -
ResourceGroupName 'az140-11-RG' -Name 'az140-adds-
vnet11').Location
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the

following to create a resource group that will host the host pool and its resources:

```
powershell $resourceGroupName = 'az140-24-RG' New-
AzResourceGroup -Location $location -Name $resourceGroupName
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to create an empty host pool:

```
powershell $hostPoolName = 'az140-24-hp3' $workspaceName =
'az140-24-ws1' $dagAppGroupName = "$hostPoolName-DAG" New-
AzWvdHostPool -ResourceGroupName $resourceGroupName -Name
$hostPoolName -WorkspaceName $workspaceName -HostPoolType
Pooled -LoadBalancerType BreadthFirst -Location $location -
DesktopAppGroupName $dagAppGroupName -PreferredAppGroupType
Desktop
```

Note: The **New-AzWvdHostPool** cmdlet allows you to create a host pool, workspace, and the desktop app group, as well as to register the desktop app group with the workspace. You have the option of creating a new workspace or using an existing one.

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to retrieve the objectID attribute of the Azure AD group named **az140-wvd-pooled**:

```
powershell $aadGroupObjectId = (Get-AzADGroup -DisplayName
'az140-wvd-pooled').Id
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to assign the Azure AD group named **az140-wvd-pooled** to the default desktop app group of the newly created host pool:

```
powershell $roleDefinitionName = 'Desktop Virtualization
User' New-AzRoleAssignment -ObjectId $aadGroupObjectId -
RoleDefinitionName $roleDefinitionName -ResourceName
$dagAppGroupName -ResourceGroupName $resourceGroupName -
ResourceType
'Microsoft.DesktopVirtualization/applicationGroups'
```

Task 3: Perform a template-based deployment of an Azure VM running Windows 10 Enterprise by using PowerShell

1. From your lab computer, use the Remote Desktop session to the **az140-dc-vm11** Azure VM to copy the lab files **\\AZ-140\AllFiles\Labs\02\az140-24_azuredeployhp3.json** and **\\AZ-140\AllFiles\Labs\02\az140-24_azuredeployhp3.parameters.json** to the **C:\AllFiles\Labs\02** folder (create it if needed).
2. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to deploy an Azure VM running Windows 10 Enterprise (multi-session) that will serve as a Windows Virtual Desktop session host in the host pool you created in the previous task:

```
powershell $resourceGroupName = 'az140-24-RG' $location =
(Get-AzResourceGroup -ResourceGroupName
$resourceGroupName).Location New-AzResourceGroupDeployment `
-ResourceGroupName $resourceGroupName ` -Location $location `
-Name az140lab24hp3Deployment ` -TemplateFile
C:\AllFiles\Labs\02\az140-24_azuredeployhp3.json ` -
TemplateParameterFile C:\AllFiles\Labs\02\az140-
24_azuredeployhp3.parameters.json
```

Note: Wait for the deployment to complete before you proceed to the next task. This might take about 5 minutes.

Note: The deployment uses an Azure Resource Manager template to provision an Azure VM and applies a VM extension that automatically joins the operating system to the **adatum.com** AD DS domain.

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to verify that the third session host was successfully joined to the **adatum.com** AD DS domain:

```
powershell Get-ADComputer -Filter "sAMAccountName -eq 'az140-
24-p3-0$'"
```

Task 4: Add an Azure VM running Windows 10 Enterprise as a host to the Windows Virtual Desktop host pool by using PowerShell

1. Within the Remote Desktop session to **az140-dc-vm11**, in the browser window displaying the Azure portal, search for and select **Virtual machines** and, on the **Virtual machines** blade, in the list of virtual machines, select **az140-24-p3-0**.

2. On the **az140-24-p3-0** blade, select **Connect**, in the drop-down menu, select **RDP**, on the **RDP** tab of the **az140-24-p3-0 | Connect** blade, in the **IP address** drop-down list, select the **Private IP address (10.0.3.4)** entry, and then select **Download RDP File**.
3. When prompted, sign in with the following credentials:

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

1. Within the Remote Desktop session to **az140-24-p3-0**, start **Windows PowerShell ISE** as administrator.
2. Within the Remote Desktop session to **az140-24-p3-0**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to create a folder that will host files required to add the newly deployed Azure VM as a session host to the host pool you provisioned earlier in this lab:

```
powershell $labFilesFolder = 'C:\AllFiles\Labs\02' New-Item -
ItemType Directory -Path $labFilesFolder
```

1. Within the Remote Desktop session to **az140-24-p3-0**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to download the Windows Virtual Desktop Agent and Boot Loader installers, required to add the session host to the host pool:

```
powershell $webClient = New-Object System.Net.WebClient
$wvdAgentInstallerURL =
'https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RW
rmXv' $wvdAgentInstallerName = 'WVD-Agent.msi'
$webClient.DownloadFile($wvdAgentInstallerURL,"$labFilesFolde
r/$wvdAgentInstallerName") $wvdBootLoaderInstallerURL =
'https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RW
rxrH' $wvdBootLoaderInstallerName = 'WVD-BootLoader.msi'
$webClient.DownloadFile($wvdBootLoaderInstallerURL,"$labFiles
Folder/$wvdBootLoaderInstallerName")
```

1. Within the Remote Desktop session to **az140-24-p3-0**, 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):

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

1. From the **Administrator: Windows PowerShell ISE** console, run the following to install the latest version of the Az.DesktopVirtualization PowerShell module:

```
powershell Install-Module -Name Az.DesktopVirtualization -
AllowClobber -Force Install-Module -Name Az -AllowClobber -
Force
```

1. From the **Administrator: Windows PowerShell ISE** console, run the following to modify the PowerShell execution policy and sign in to your Azure subscription:

```
powershell Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope
CurrentUser -Force Connect-AzAccount
```

1. When prompted, provide the credentials of the user account with the Owner role in the subscription you are using in this lab.
2. Within the Remote Desktopliveid session to **az140-24-p3-0**, from the **Administrator: Windows PowerShell ISE** console, run the following to generate the token necessary to join new session hosts to the pool you provisioned earlier in this exercise:

```
powershell $resourceGroupName = 'az140-24-RG' $hostPoolName =
'az140-24-hp3' $registrationInfo = New-AzWvdRegistrationInfo
-ResourceGroupName $resourceGroupName -HostPoolName
$hostPoolName -ExpirationTime $((get-
date).ToUniversalTime().AddDays(1).ToString('yyyy-MM-
ddTHH:mm:ss.fffffffZ'))
```

Note: A registration token is required to authorize a session host to join the host pool. The value of token's expiration date must be between one hour and one month from the current date and time.

1. Within the Remote Desktop session to **az140-24-p3-0**, from the **Administrator: Windows PowerShell ISE** console, run the following to install the Windows Virtual Desktop Agent:

```
powershell Set-Location -Path $labFilesFolder Start-Process -
FilePath 'msiexec.exe' -ArgumentList "/i
$WVDAgentInstallerName", "/quiet", "/qn", "/norestart",
"/passive", "REGISTRATIONTOKEN=$($registrationInfo.Token)",
"/l* $labFilesFolder\AgentInstall.log" | Wait-Process
```

1. Within the Remote Desktop session to **az140-24-p3-0**, from the **Administrator: Windows PowerShell ISE** console, run the following to install the Windows Virtual Desktop Boot Loader:

```
powershell Start-Process -FilePath "msiexec.exe" -
ArgumentList "/i $wvdBootLoaderInstallerName", "/quiet",
"/qn", "/norestart", "/passive", "/l*
$labFilesFolder\BootLoaderInstall.log" | Wait-process
```

Task 5: Verify the deployment of the Azure Windows Virtual Desktop host

1. Switch to the lab computer, in the web browser displaying the Azure portal, search for and select **Windows Virtual Desktop**, on the **Windows Virtual Desktop** blade, select **Host pools** and, on the **Windows Virtual Desktop | Host pools** blade, select the entry **az140-24-hp3** representing the newly modified pool.
2. On the **az140-24-hp3** blade, in the vertical menu on the left side, in the **Manage** section, click **Session hosts**.
3. On the **az140-24-hp3 | Session hosts** blade, verify that the deployment includes a single host.

Task 6: Manage app groups using PowerShell

1. From the lab computer, switch to the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to create a Remote App group:

```
powershell $subscriptionId = (Get-AzContext).Subscription.Id
$appGroupName = 'az140-24-hp3-Office365-RAG'
$resourceGroupName = 'az140-24-RG' $hostPoolName = 'az140-24-
hp3' $location = (Get-AzVirtualNetwork -ResourceGroupName
'az140-11-RG' -Name 'az140-adds-vnet11').Location New-
AzWvdApplicationGroup -Name $appGroupName -ResourceGroupName
$resourceGroupName -ApplicationGroupType 'RemoteApp' -
HostPoolArmPath
"/subscriptions/$subscriptionId/resourcegroups/$resourceGroup
Name/providers/Microsoft.DesktopVirtualization/hostPools/$hos
tPoolName"-Location $location
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to list the **Start** menu apps on the pool's hosts and review the output:

```
powershell Get-AzWvdStartMenuItem -ApplicationGroupName
$appGroupName -ResourceGroupName $resourceGroupName | Format-
List | more
```

Note: For any application you want to publish, you should record the information included in the output, including such parameters as **FilePath**, **IconPath**, and **IconIndex**.

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to publish Microsoft Word:

```
powershell $name = 'Microsoft Word' $filePath = 'C:\Program
Files\Microsoft Office\root\Office16\WINWORD.EXE' $iconPath =
'C:\Program Files\Microsoft
Office\Root\VFS\Windows\Installer\{90160000-000F-0000-1000-
0000000FF1CE}\wordicon.exe' New-AzWvdApplication -GroupName
$appGroupName -Name $name -ResourceGroupName
$resourceGroupName -Filepath $filePath -IconPath $iconPath -
IconIndex 0 -CommandLineSetting 'DoNotAllow' -
ShowInPortal:$true
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to publish Microsoft Word:

```
powershell $aadGroupObjectId = (Get-AzADGroup -DisplayName
'az140-wvd-remote-app').Id New-AzRoleAssignment -ObjectId
$aadGroupObjectId -RoleDefinitionName 'Desktop Virtualization
User' -ResourceName $appGroupName -ResourceGroupName
$resourceGroupName -ResourceType
'Microsoft.DesktopVirtualization/applicationGroups'
```

1. Switch to the lab computer, in the web browser displaying the Azure portal, on the **az140-24-hp3 | Session hosts** blade, in the vertical menu on the left side, in the **Manage** section, select **Application groups**.
2. On the **az140-24-hp3 | Application groups** blade, in the list of application groups, select the **az140-24-hp3-Office365-RAG** entry.
3. On the **az140-24-hp3-Office365-RAG** blade, verify the configuration of the application group, including the applications and assignments.

Exercise 2: Stop and deallocate Azure VMs provisioned in the lab

The main tasks for this exercise are as follows:

1. Stop and deallocate Azure VMs provisioned in the lab

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

Task 1: Deallocate Azure VMs provisioned 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 in this lab:

```
powershell Get-AzVM -ResourceGroup 'az140-24-RG'
```

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

```
powershell Get-AzVM -ResourceGroup 'az140-24-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.

lab: title: 'Lab: Create and manage session host images (AD DS)' module: 'Module 2: Implement a WVD Infrastructure'

Lab - Create and manage session host images (AD DS)

Student lab manual

Lab dependencies

- An Azure subscription you will be using in this lab.
- A Microsoft account or an Azure AD 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 Azure AD tenant associated with that Azure subscription.
- The completed lab **Prepare for deployment of Azure Windows Virtual Desktop (AD DS)** or **Prepare for deployment of Azure Windows Virtual Desktop (Azure AD DS)**

Estimated Time

60 minutes

Lab scenario

You need to create and manage Windows Virtual Desktop host images in an Active Directory Domain Services (AD DS) environment.

Objectives

After completing this lab, you will be able to:

- Create and manage WVD session host images

Lab files

- \\AZ-140\AllFiles\Labs\02\az140-25_azuredeployvm25.json
- \\AZ-140\AllFiles\Labs\02\az140-25_azuredeployvm25.parameters.json

Instructions

Exercise 1: Create and manage session host images

The main tasks for this exercise are as follows:

1. Prepare for configuration of a Windows Virtual Desktop host image
2. Configure a Windows Virtual Desktop host image
3. Create a Windows Virtual Desktop host image
4. Provision a Windows Virtual Desktop host pool by using the custom image

Task 1: Prepare for configuration of a Windows Virtual Desktop host image

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**.
4. 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 that will contain the Windows Virtual Desktop host image:

```
powershell $vnetResourceGroupName = 'az140-11-RG' $location =
(Get-AzResourceGroup -ResourceGroupName
$vnetResourceGroupName).Location $imageResourceGroupName =
'az140-25-RG' New-AzResourceGroup -Location $location -Name
$imageResourceGroupName
```

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 files **\\AZ-140\AllFiles\Labs\02\az140-25_azuredeployvm25.json** and **\\AZ-140\AllFiles\Labs\02\az140-25_azuredeployvm25.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 will

serve as a Windows Virtual Desktop client into the newly created subnet:

```
powershell New-AzResourceGroupDeployment ` -ResourceGroupName
$imageResourceGroupName ` -Name az140lab0205vmDeployment ` -
TemplateFile $HOME/az140-25_azuredeployvm25.json ` -
TemplateParameterFile $HOME/az140-
25_azuredeployvm25.parameters.json
```

Note: Wait for the deployment to complete but instead proceed to the next exercise. The deployment might take about 10 minutes.

Task 2: Configure a Windows Virtual Desktop host image

1. In the Azure portal, search for and select **Virtual machines** and, on the **Virtual machines** blade, select **az140-25-vm0**.
2. On the **az140-25-vm0** blade, select **Connect**, in the drop-down menu, select **RDP**, on the **RDP** tab of the **az140-25-vm0 | Connect** blade, in the **IP address** drop-down list, select the **Public IP address** entry, and then select **Download RDP File**.
3. When prompted, sign in with the following credentials:

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

Note: You will start by installing FSLogix binaries.

1. Within the Remote Desktop session to **az140-25-vm0**, start **Windows PowerShell ISE** as administrator.
2. Within the Remote Desktop session to **az140-25-vm0**, from the **Administrator: Windows PowerShell ISE** console, run the following to create a folder you will use as a temporary location for configuration of the image:

```
powershell New-Item -Type Directory -Path
'C:\Allfiles\Labs\02' -Force
```

1. Within the Remote Desktop session to **az140-25-vm0**, start Microsoft Edge, browse to FSLogix download page, download FSLogix compressed installation binaries into the **C:\Allfiles\Labs\02** folder and, from File Explorer, extract the **x64** subfolder into the same folder.

2. Within the Remote Desktop session to **az140-25-vm0**, switch to the **Administrator: Windows PowerShell ISE** window and, from the **Administrator: Windows PowerShell ISE** console, run the following to perform per-machine installation of OneDrive:

```
powershell Start-Process -FilePath
'C:\Allfiles\Labs\02\x64\Release\FSLogixAppsSetup.exe' -
ArgumentList '/quiet' -Wait
```

Note: Wait for the installation to complete. This might take about 1 minute.

Note: Next, you will install and configure Microsoft Teams.

1. Within the Remote Desktop session to **az140-25-vm0**, right-click **Start**, in the right-click menu, select **Run**, in the **Run** dialog box, in the **Open** textbox, type **cmd** and press the **Enter** key to start **Command Prompt**.
2. In the **Administrator: C:\windows\system32\cmd.exe** window, from the command prompt, run the following to prepare for per-machine installation of Microsoft Teams:

```
cmd reg add "HKLM\Software\Microsoft\Teams" /v
IsWVDEnvironment /t REG_DWORD /d 1 /f
```

1. Within the Remote Desktop session to **az140-25-vm0**, in Microsoft Edge, browse to the download page of Microsoft Visual C++ Redistributable, save **VC_redist.x64** into the **C:\Allfiles\Labs\02** folder.
2. Within the Remote Desktop session to **az140-25-vm0**, switch to the **Administrator: C:\windows\system32\cmd.exe** window and, from the command prompt, run the following to perform installation of Microsoft Visual C++ Redistributable:

```
cmd C:\Allfiles\Labs\02\vc_redist.x64.exe /install /passive
/norestart /log C:\Allfiles\Labs\02\vc_redist.log
```

1. Within the Remote Desktop session to **az140-25-vm0**, switch to the Microsoft Edge window, browse to the download page of Remote Desktop WebRTC Redirector Service, and save the installer package into the **C:\Allfiles\Labs\02** folder.
2. Within the Remote Desktop session to **az140-25-vm0**, start File Explorer, navigate to the **C:\Allfiles\Labs\02** folder, double-click

the newly downloaded installer, and run the installation with the default settings.

3. Within the Remote Desktop session to **az140-25-vm0**, in Microsoft Edge, browse to the documentation page titled Deploy the Teams desktop app to the VM, click the **64-bit version** link, and, when prompted, save the **Teams_windows_x64.msi** file into the **C:\Allfiles\Labs\02** folder.
4. Within the Remote Desktop session to **az140-25-vm0**, switch to the **Administrator: C:\windows\system32\cmd.exe** window and, from the command prompt, run the following to perform per-machine installation of Microsoft Teams:

```
cmd msiexec /i C:\Allfiles\Labs\02\Teams_windows_x64.msi /l*v
C:\Allfiles\Labs\02\Teams.log ALLUSER=1
```

Note: The installer supports the ALLUSER=1 and ALLUSERS=1 parameters. The ALLUSER=1 parameter is intended for per-machine installation in VDI environments. The ALLUSERS=1 parameter can be used in non-VDI and VDI environments.

1. Within the Remote Desktop session to **az140-25-vm0**, switch to the **Administrator: Windows PowerShell ISE** window and, from the **Administrator: Windows PowerShell ISE** console, run the following to install Microsoft Edge Chromium:

```
powershell Start-BitsTransfer -Source "https://aka.ms/edge-
msi" -Destination
'C:\Allfiles\Labs\02\MicrosoftEdgeEnterpriseX64.msi' Start-
Process -Wait -Filepath msiexec.exe -Argumentlist "/i
C:\Allfiles\Labs\02\MicrosoftEdgeEnterpriseX64.msi /q"
```

Note: Wait for the installation to complete. This might take about 2 minutes.

Note: When operating in a multi-language environment, you might need to install language packs. For details regarding this procedure, refer to the Microsoft Docs article Add language packs to a Windows 10 multi-session image.

Note: Next, you will disable Windows Automatic Updates, disable Storage Sense, configure time zone redirection, and configure collection of telemetry. In general, you should first apply all current updates first. In this lab, you skip this step in order to minimize the duration of the lab.

1. Within the Remote Desktop session to **az140-25-vm0**, switch to the **Administrator: C:\windows\system32\cmd.exe** window and, from the command prompt, run the following to disable Automatic Updates:

```
cmd reg add
"HKLM\SOFTWARE\Policies\Microsoft\Windows\WindowsUpdate\AU"
/v NoAutoUpdate /t REG_DWORD /d 1 /f
```

1. In the **Administrator: C:\windows\system32\cmd.exe** window, from the command prompt, run the following to disable Storage Sense:

```
cmd reg add
"HKCU\Software\Microsoft\Windows\CurrentVersion\StorageSense\
Parameters\StoragePolicy" /v 01 /t REG_DWORD /d 0 /f
```

1. In the **Administrator: C:\windows\system32\cmd.exe** window, from the command prompt, run the following to configure time zone redirection:

```
cmd reg add "HKLM\SOFTWARE\Policies\Microsoft\Windows
NT\Terminal Services" /v fEnableTimeZoneRedirection /t
REG_DWORD /d 1 /f
```

1. In the **Administrator: C:\windows\system32\cmd.exe** window, from the command prompt, run the following to disable feedback hub collection of telemetry data:

```
cmd reg add
"HKLM\SOFTWARE\Policies\Microsoft\Windows\DataCollection" /v
AllowTelemetry /t REG_DWORD /d 0 /f
```

1. In the **Administrator: C:\windows\system32\cmd.exe** window, from the command prompt, run the following to delete the temporary folder you created earlier in this task:

```
cmd rmdir C:\Allfiles /s /q
```

1. In the **Administrator: C:\windows\system32\cmd.exe** window, from the command prompt, run the Disk Cleanup utility and click **OK** once completed:

```
cmd cleanmgr /d C: /verylowdisk
```

Task 3: Create a Windows Virtual Desktop host image

1. Within the Remote Desktop session to **az140-25-vm0**, in the **Administrator: C:\windows\system32\cmd.exe** window, from the command prompt, run the sysprep utility in order to prepare the operating system for generating an image and automatically shut it down:

```
cmd C:\Windows\System32\Sysprep\sysprep.exe /oobe /generalize
/shutdown
```

Note: Wait for the sysprep process to complete. This might take about 2 minutes. This will automatically shut down the operating system.

1. From your lab computer, in the web browser displaying the Azure portal, search for and select **Virtual machines** and, from the **Virtual machines** blade, select **az140-25-vm0**.
2. On the **az140-25-vm0** blade, in the toolbar above the **Essentials** section, click **Refresh**, verify that the **Status** of the Azure VM changed to **Stopped**, click **Stop**, and, when prompted for confirmation, click **OK** to transition the Azure VM into the **Stopped (deallocated)** state.
3. On the **az140-25-vm0** blade, verify that the **Status** of the Azure VM changed to the **Stopped (deallocated)** state and, in the toolbar, click **Capture**. This will automatically display the **Create an image** blade.
4. On the **Basics** tab of the **Create an image** blade, specify the following settings:

Setting	Value
Share image to Shared image gallery	**Yes, share it to a gallery as an image version**
Automatically delete this virtual machine after creating the image	checkbox cleared
Target image gallery	the name of a new image gallery **az10425imagegallery**
Operating system state	**Generalized**

1. On the **Basics** tab of the **Create an image** blade, below the **Target image definition** textbox, click **Create new**.

2. On the **Create an image definition**, specify the following settings and click **OK**:

Setting	Value
Image definition name	**az140-25-host-image**
Publisher	**MicrosoftWindowsDesktop**
Offer	**office-365**
SKU	**20h1-evd-o365pp**

1. Back on the **Basics** tab of the **Create an image** blade, specify the following settings and click **Review + create**:

Setting	Value
Version number	**1.0.0**
Exclude from latest	checkbox cleared
End of life date	one year ahead from the current date
Default replica count	**1**
Target region replica count	**1**
Storage account type	**Premium SSD**

1. On the **Review + create** tab of the **Create an image** blade, click **Create**.

Note: Wait for the deployment to complete. This might take about 20 minutes.

1. From your lab computer, in the web browser displaying the Azure portal, search for and select **Shared image galleries*** ***and, on the Shared image galleries blade, select the az10425imagegallery entry, and, on the*** *az10425imagegallery* ***blade, verify the presence of the*** ***az140-25-host-image** entry representing the newly created image.

Task 4: Provision a Windows Virtual Desktop host pool by using a custom image

1. From the lab computer, 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 **Virtual networks** and, on the **Virtual networks** blade, select **az140-adds-vnet11**.

2. On the **az140-adds-vnet11** blade, select **Subnets**, on the **Subnets** blade, select **+ Subnet**, on the **Add subnet** blade, specify the following settings (leave all other settings with their default values) and click **Save**:

Setting	Value
Name	**hp4-Subnet**
Subnet address range	**10.0.4.0/24**

1. From the lab computer, in the Azure portal, in the web browser window displaying the Azure portal, search for and select **Windows Virtual Desktop**, on the **Windows Virtual Desktop** blade, select **Host pools** and, on the **Windows Virtual Desktop | Host pools** blade, select **+ Add**.
2. On the **Basics** tab of the **Create a host pool** blade, specify the following settings and select **Next: Virtual Machines >**:

Setting	Value
Subscription	the name of the Azure subscription you are using in this lab
Resource group	**az140-25-RG**
Host pool name	**az140-25-hp4**
Location	the name of the Azure region into which you deployed resources in the first exercise of this lab
Validation environment	**No**
Host pool type	**Pooled**
Max session limit	**50**
Load balancing algorithm	**Breadth-first**

1. On the **Virtual machines** tab of the **Create a host pool** blade, specify the following settings:

Setting	Value

Setting	Value
Add virtual machines	**Yes**
Resource group	**Defaulted to same as host pool**
Name prefix	**az140-25-p4**
Virtual machine location	the name of the Azure region into which you deployed resources in the first exercise of this lab
Availability options	No infrastructure redundancy required
Virtual machine size	**Standard D2s v3**
Number of VMs	**1**
Image type	**Gallery**
User name	Student
Password	Pa55w.rd1234
Confirm password	Pa55w.rd1234

1. On the **Virtual machines** tab of the **Create a host pool** blade, directly below the **Image** dropdown list, click the **See all images** link.
2. On the **Select an image** blade, click the **My Items** tab, click **Shared Images**, and, in the list of shared images, select **az140-25-host-image**.
3. Back on the **Virtual machines** tab of the **Create a host pool** blade, specify the following settings and select **Next: Workspace >**

Setting	Value
OS disk type	**Standard SSD**
Virtual network	**az140-adds-vnet11**
Subnet	**hp4-Subnet (10.0.4.0/24)**
Network security group	**Basic**
Public inbound ports	**Yes**
Inbound ports to allow	**RDP**
Specify domain or unit	**Yes**

Setting	Value
Domain to join	**adatum.com**
Organizational Unit path	**OU=WVDInfra,DC=adatum,DC=com**
AD domain join UPN	**student@adatum.com**
Password	**Pa55w.rd1234**

1. On the **Workspace** tab of the **Create a host pool** blade, specify the following settings and select **Review + create**:

Setting	Value
Register desktop app group	**No**

1. On the **Review + create** tab of the **Create a host pool** blade, select **Create**.

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

Note: Following deployment of hosts based on custom images, you should consider running the Virtual Desktop Optimization Tool, available from its GitHub repository.

Exercise 2: Stop and deallocate Azure VMs provisioned in the lab

The main tasks for this exercise are as follows:

1. Stop and deallocate Azure VMs provisioned in the lab

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

Task 1: Deallocate Azure VMs provisioned 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 in this lab:

```
powershell Get-AzVM -ResourceGroup 'az140-25-RG'
```

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

```
powershell Get-AzVM -ResourceGroup 'az140-25-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.

lab: title: 'Lab: Configure Conditional Access policies for WVD (AD DS)' module: 'Module 3: Manage Access and Security'

Lab - Configure Conditional Access policies for WVD (AD DS)

Student lab manual

Lab dependencies

- An Azure subscription
- A Microsoft account or an Azure AD account with the Global Administrator role in the Azure AD 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 Windows Virtual Desktop (AD DS)** or **Prepare for deployment of Azure Windows Virtual Desktop (Azure AD DS)**
- The completed lab **Deploy host pools and session hosts by using the Azure portal (AD DS)** or **Deploy host pools and session hosts by using the Azure portal (Azure AD DS)**

Estimated Time

60 minutes

Lab scenario

You need to control access to a deployment of Windows Virtual Desktop in an Active Directory Domain Services (AD DS) environment by using Azure Active Directory (Azure AD) conditional access.

Objectives

After completing this lab, you will be able to:

- Prepare for Azure Active Directory (Azure AD)-based Conditional Access for Windows Virtual Desktop
- Implement Azure AD-based Conditional Access for Windows Virtual Desktop

Lab files

- None

Instructions

Exercise 1: Prepare for Azure AD-based Conditional Access for Windows Virtual Desktop

The main tasks for this exercise are as follows:

1. Configure Azure AD Premium P2 licensing
2. Configure Azure AD Multi-Factor Authentication (MFA)
3. Register a user for Azure AD MFA
4. Configure hybrid Azure AD join
5. Trigger Azure AD Connect delta synchronization

Task 1: Configure Azure AD Premium P2 licensing

Note: Premium P1 or P2 licensing of Azure AD is required in order to implement Azure AD Conditional Access. You will use a 30-day trial for this lab.

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 and the Global Administrator role in the Azure AD tenant associated with that subscription.
2. In the Azure portal, search for and select **Azure Active Directory** to navigate to the Azure AD tenant associated with the Azure subscription you are using for this lab.
3. On the Azure Active Directory blade, in the vertical menu bar on the left side, in the **Manage** section, click **Users**.
4. On the **Users | All users (Preview)** blade, select **aduser5**.
5. On the **aduser5 | Profile** blade, in the toolbar, click **Edit**, in the **Settings** section, in the **Usage location** dropdown list, select country where the lab environment is located and, in the toolbar, click **Save**.

6. On the **aduser5 | Profile** blade, in the **Identity** section, identify the user principal name of the **aduser5** account.

   Note: Record this value. You will need it later in this lab.

7. On the **Users | All users (Preview)** blade, select the user account you used to sign at the beginning of this task and repeat the previous step in case your account does not have the **Usage location** assigned.

   Note: The **Usage location** property must be set in order to assign an Azure AD Premium P2 licenses to user accounts.

8. On the **Users | All users (Preview)** blade, select the **aadsyncuser** user account and identify its user principal name.

   Note: Record this value. You will need it later in this lab.

9. In the Azure portal, navigate back to the **Overview** blade of the Azure AD tenant and, in the vertical menu bar on the left side, in the **Manage** section, click **Licenses**.

10. On the **Licenses | Overview** blade, in the vertical menu bar on the left side, in the **Manage** section, click **All products**.
11. On the **Licenses | All products** blade, in the toolbar, click **+ Try/Buy**.
12. On the **Activate** blade, click **Free trial** in the **ENTERPRISE MOBILITY + SECURITY E5** section and then click **Activate**.
13. While on the **Licenses | Overview** blade, refresh the browser window to verify that the activation was successful.
14. On the **Licenses - All products** blade, select the **Enterprise Mobility + Security E5** entry.
15. On the **Enterprise Mobility + Security E5** blade, in the toolbar, click **+ Assign**.
16. On the **Assign license** blade, click **Add users and groups**, on the **Add users and groups** blade, select **aduser5** and your user accounts, and click **Select**.
17. Back on the **Assign license** blade, click **Assignment options**, on the **Assignment options** blade, verify that all options are enabled, click **Review + assign**, click **Assign**.

Task 2: Configure Azure AD Multi-Factor Authentication (MFA)

1. On your lab computer, in the web browser displaying the Azure portal, navigate back to the **Overview** blade of the Azure AD tenant and, in the vertical menu on the left side, in the **Manage** section, click **Security**.

2. On the **Security | Getting started** blade, in the vertical menu on the left side, in the **Manage** section, click **Identity Protection**.
3. On the **Identity Protection | Overview** blade, in the vertical menu on the left side, in the **Protect** section, click **MFA registration policy**.
4. On the **Identity Protection | MFA registration policy** blade, in the **Assignments** section of the **Multi-factor authentication registration policy**, click **All users**, on the **Include** tab, click the **Select individuals and groups** option, on the **Select users**, click **aduser5**, click **Select**, at the bottom of the blade, set the **Enforce policy** switch to **On**, and click **Save**.

Task 3: Register a user for Azure AD MFA

1. On your lab computer, open an **InPrivate** web browser session, navigate to the Azure portal, and sign in by providing the **user5** user principal name you identified earlier in this exercise and the **Pa55w.rd1234** as the password.
2. When presented with the message **More information required**, click **Next**. This will automatically redirect your browser to the **Microsoft Authenticator** page.
3. On the **Microsoft Authenticator** page, click the link **I want to set up a different method**.
4. In the **Choose a different method** dialog box, in the **Which method would you like to use?** dropdown list, select **Phone** an click **Confirm**.
5. On the **Phone** page, provide your phone number, verify that the **Text me a code** option is selected, and click **Next**.
6. On the **Phone** page, type the code included in the text you received on your phone and click **Next**.
7. Ensure that the SMS was successfully verified and your phone registered successfully and click **Next**.
8. On the **Success!** page, click **Done**.
9. On the Azure portal page, in the upper right corner, click the icon representing the user avatar, click **Sign out**, and close the **In private** browser window.

Task 4: Configure hybrid Azure AD join

Note: This functionality can be leveraged to implement additional security when setting up Conditional Access for devices based on

their Azure AD join status.

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 **RDP**, on the **RDP** tab of the **az140-dc-vm11 | Connect** blade, in the **IP address** drop-down list, select the **Load balancer DNS name** entry, and then select **Download RDP File**.
3. When prompted, sign in with the following credentials:

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

1. Within the Remote Desktop session to **az140-dc-vm11**, in the **Start** menu, expand the **Azure AD Connect** folder and select **Azure AD Connect**.
2. On the **Welcome to Azure AD Connect** page of the **Microsoft Azure Active Directory Connect** window, select **Configure**.
3. On the **Additional tasks** page in the **Microsoft Azure Active Directory Connect** window, select **Configure device options** and select **Next**.
4. On the **Overview** page in the **Microsoft Azure Active Directory Connect** window, review the information regarding **Hybrid Azure AD join** and **Device writeback** and select **Next**.
5. On the **Connect to Azure AD** page in the **Microsoft Azure Active Directory Connect** window, 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 lab and specify **Pa55w.rd1234** as its password.

1. On the **Device options** page in the **Microsoft Azure Active Directory Connect** window, ensure that the **Configure Hybrid Azure AD join** option is selected and select **Next**.
2. On the **Device operating systems** page in the **Microsoft Azure Active Directory Connect** window, select the **Windows 10 or later domain-joined devices** checkbox and select **Next**.

3. On the **SCP configuration** page in the **Microsoft Azure Active Directory Connect** window, select the checkbox next to the **adatum.com** entry, in the **Authentication Service** drop-down list, select **Azure Active Directory** entry, and select **Add**.
4. When prompted, in the **Enterprise Admin Credentials** dialog box, specify the following credentials, and select **OK**:

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

1. Back on the **SCP configuration** page in the **Microsoft Azure Active Directory Connect** window, select **Next**.
2. On the **Ready to configure** page in the **Microsoft Azure Active Directory Connect** window, select **Configure** and, once the configuration completes, select **Exit**.
3. Within the Remote Desktop session to **az140-dc-vm11**, start **Windows PowerShell ISE** as administrator.
4. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to move the **az140-cl-vm11** computer account to the **WVDClients** organizational unit (OU):

```
powershell Move-ADObject -Identity "CN=az140-cl-
vm11,CN=Computers,DC=adatum,DC=com" -TargetPath
"OU=WVDClients,DC=adatum,DC=com"
```

1. Within the Remote Desktop session to **az140-dc-vm11**, in the **Start** menu, expand the **Azure AD Connect** folder and select **Azure AD Connect**.
2. On the **Welcome to Azure AD Connect** page of the **Microsoft Azure Active Directory Connect** window, select **Configure**.
3. On the **Additional tasks** page in the **Microsoft Azure Active Directory Connect** window, select **Customize synchronization options** and select **Next**.
4. On the **Connect to Azure AD** page in the **Microsoft Azure Active Directory Connect** window, 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 lab and specify **Pa55w.rd1234**

as its password.

1. On the **Connect your directories** page in the **Microsoft Azure Active Directory Connect** window, select **Next**.
2. 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, ensure that the checkbox next to the **ToSync** OU is selected, select the checkbox next to the **WVDClients** OU, and select **Next**.
3. On the **Optional features** page in the **Microsoft Azure Active Directory Connect** window, accept the default settings, and select **Next**.
4. 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**.
5. Review the information on the **Configuration complete** page and select **Exit** to close the **Microsoft Azure Active Directory Connect** window.

Task 5: Trigger Azure AD Connect delta synchronization

1. Within the Remote Desktop session to **az140-dc-vm11**, switch to the **Administrator: Windows PowerShell ISE** window.
2. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console pane, run the following to trigger Azure AD Connect delta synchronization:

```
powershell Start-ADSyncSyncCycle -PolicyType Delta
```

1. Within the Remote Desktop session to **az140-dc-vm11**, start Microsoft Edge and navigate to the Azure portal. When prompted, sign in by using the Azure AD credentials of the user account with the Global Administrator role in the Azure AD tenant associated with the Azure subscription you are using in this lab.
2. Within the Remote Desktop session to **az140-dc-vm11**, in the Microsoft Edge window displaying the Azure portal, search for and select **Azure Active Directory** to navigate to the Azure AD tenant associated with the Azure subscription you are using for this lab.
3. On the Azure Active Directory blade, in the vertical menu bar on the left side, in the **Manage** section, click **Devices**.

4. On the **Devices | All devices** blade, review the list of devices and verify that the **az140-cl-vm11** device is listed with the **Hybrid Azure AD joined** entry in the **Join Type** column.

Exercise 2: Implement Azure AD-based Conditional Access for Windows Virtual Desktop

The main tasks for this exercise are as follows:

1. Create an Azure AD-based Conditional Access policy for all Windows Virtual Desktop connections
2. Test the Azure AD-based Conditional Access policy for all Windows Virtual Desktop connections
3. Modify the Azure AD-based Conditional Access policy to exclude hybrid Azure AD joined computers from the MFA requirement
4. Test the modified Azure AD-based Conditional Access policy

Task 1: Create an Azure AD-based Conditional Access policy for all Windows Virtual Desktop connections

Note: In this task, you will configure an Azure AD-based Conditional Access policy that requires MFA to sign in to a Windows Virtual Desktop session. The policy will also enforce reauthentication after the first 4 hours following a successful authentication.

1. On your lab computer, in the web browser displaying the Azure portal, navigate back to the **Overview** blade of the Azure AD tenant and, in the vertical menu on the left side, in the **Manage** section, click **Security**.
2. On the **Security | Getting started** blade, in the vertical menu on the left side, in the **Protect** section, click **Conditional Access**.
3. On the **Conditional Access | Policies** blade, in the toolbar, click **+ New policy**.

4. On the **New** blade, configure the following settings:

5. In the **Name** text box, type **az140-31-wvdpolicy1**

6. In the **Assignments** section, click **Users and groups**, click the **Select users and groups** option, click the **Users and groups**

checkbox, on the **Select** blade, click **aduser5**, and click **Select**.

7. In the **Assignments** section, click **Cloud apps or actions**, ensure that in the **Select what this policy applies to switch, the** Cloud apps **option is selected, click the** Select apps **option, on the** Select **blade, select the checkbox next to the** Windows Virtual Desktop **entry, and click** Select**.
8. In the **Assignments** section, click **Conditions**, click **Client apps**, on the **Client apps** blade, set the **Configure** switch to **Yes**, ensure that both the **Browser** and **Mobile apps and desktop clients** checkboxes are selected, and click **Done**.
9. In the **Access controls** section, click **Grant**, on the **Grant** blade, ensure that the **Grant access** option is selected, select the **Require multi-factor authentication** checkbox and click **Select**.
10. In the **Access controls** section, click **Session**, on the **Session** blade, select the **Sign-in frequency** checkbox, in the first textbox, type **4**, in the **Select units** dropdown list, select **Hours**, leave the **Persistent browser session** checkbox cleared, and click **Select**.

11. Set the **Enable policy** switch to **On**.

12. On the **New** blade, click **Create**.

Task 2: Test the Azure AD-based Conditional Access policy for all Windows Virtual Desktop connections

1. On your lab computer, open an **InPrivate** web browser session, navigate to the Azure portal, and sign in by providing the **user5** user principal name you identified earlier in this exercise and the **Pa55w.rd1234** as the password.

Note: Verify that you are not prompted to authenticate via MFA.

1. In the **InPrivate** web browser session, navigate to the Windows Virtual Desktop HTML5 web client page at https://rdweb.wvd.microsoft.com/arm/webclient.

Note: Verify that this will automatically trigger authentication via MFA.

1. In the **Enter code** pane, type the code included in the text you received on your phone and click **Verify**.

2. On the **All Resources** page, click **Command Prompt**, on the **Access local resources** pane, clear the **Printer** checkbox, and click **Allow**.
3. When prompted, in the **Enter your credentials**, in the **User name** textbox type the user principal name of **aduser5** and, in the **Password** textbox, type **Pa55word1234** and click **Submit**.
4. Verify that the **Command Prompt** Remote App was launched successfully.
5. In the **Command Prompt** Remote App window, at the command prompt, type **logoff** and press the **Enter** key.
6. Back on the **All Resources** page, in the upper right corner, click **aduser5**, in the dropdown menu, click **Sign Out**, and close the **InPrivate** web browser window.

Task 3: Modify the Azure AD-based Conditional Access policy to exclude hybrid Azure AD joined computers from the MFA requirement

Note: In this task, you will modify the Azure AD-based Conditional Access policy that requires MFA to sign in to a Windows Virtual Desktop session such that connections originating from Azure AD joined computers will not require MFA.

1. On your lab computer, in the browser window displaying the Azure portal, on the **Conditional Access | Policies** blade, click the entry representing the **az140-31-wvdpolicy1** policy.
2. On the **az140-31-wvdpolicy1** blade, in the **Assignments** section, click **Conditions**, in the list of conditions, click **Device state**, on the **Device state** blade, set the **Configure** switch to **Yes**, click the **Exclude** tab, on the **Exclude** tab, select **Device Hybrid Azure AD joined**, and click **Done**.
3. On the **az140-31-wvdpolicy1** blade, in the **Access controls** section, click **Grant**, on the **Grant** blade, select the **Require Hybrid Azure AD joined device** checkbox, ensure that the **Require one of the selected controls** option is enabled, and click **Select**.
4. On the **az140-31-wvdpolicy1** blade, click **Save**.

Task 4: Test the modified Azure AD-based Conditional Access policy

1. On your lab computer, in the browser window displaying the Azure portal, search for and select **Virtual machines** and, on the **Virtual machines** blade, select the **az140-cl-vm11** entry.
2. On the **az140-cl-vm11** blade, select **Connect**, in the drop-down menu, select **RDP**, and then select **Download RDP File**.
3. When prompted, sign in as the **ADATUM\aduser5** user with **Pa55w.rd1234** as its password.
4. Within the Remote Desktop session to **az140-cl-vm11**, start Microsoft Edge and navigate to navigate to the Windows Virtual Desktop HTML5 web client page at https://rdweb.wvd.microsoft.com/arm/webclient.

Note: Verify that this time you will not be prompted to authenticate via MFA. This is because **az140-cl-vm11** is Hybrid Azure AD-joined.

1. On the **All Resources** page, click **Command Prompt**, on the **Access local resources** pane, clear the **Printer** checkbox, and click **Allow**.
2. When prompted, in the **Enter your credentials**, in the **User name** textbox type the user principal name of **aduser5** and, in the **Password** textbox, type **Pa55word1234** and click **Submit**.
3. Verify that the **Command Prompt** Remote App was launched successfully.
4. In the **Command Prompt** Remote App window, at the command prompt, type **logoff** and press the **Enter** key.
5. Back on the **All Resources** page, in the upper right corner, click **aduser5**, in the dropdown menu, click **Sign Out**.
6. Within the Remote Desktop session to **az140-cl-vm11**, click **Start**, in the vertical bar directly above the **Start** button, click the icon representing the signed in user account, and, in the pop-up menu, click **Sign out**.

---

lab: title: 'Lab: Implement and manage Windows Virtual Desktop profiles (Azure AD DS)' module: 'Module 4: Manage User Environments and Apps'

Lab - Implement and manage Windows Virtual Desktop profiles (Azure AD DS)

Student lab manual

Lab dependencies

- An Azure subscription
- A Microsoft account or an Azure AD account with the Global Administrator role in the Azure AD tenant associated with the Azure subscription and with the Owner or Contributor role in the Azure subscription
- A Windows Virtual Desktop environment provisioned in the lab **Introduction to Windows Virtual Desktop (Azure AD DS)**

Estimated Time

30 minutes

Lab scenario

You need to implement Windows Virtual Desktop profile management in an Azure Active Directory Domain Services (Azure AD DS) environment.

Objectives

After completing this lab, you will be able to:

- Configure Azure Files to store profile containers for Windows Virtual Desktop in Azure AD DS environment
- Implement FSLogix based profiles for Windows Virtual Desktop in Azure AD DS environment

Lab files

- None

Instructions

Exercise 1: Implement FSLogix based profiles for Windows Virtual Desktop

The main tasks for this exercise are as follows:

1. Configure local Administrators group on Windows Virtual Desktop session host VMs
2. Configure FSLogix-based profiles on Windows Virtual Desktop session host VMs
3. Test FSLogix-based profiles with Windows Virtual Desktop

Task 1: Configure local Administrators group on Windows Virtual Desktop session host VMs

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. From your lab computer, in the Azure portal, search for and select **Virtual machines** and, from the **Virtual machines** blade, select the **az140-cl-vm11a** entry. This will open the **az140-cl-vm11a** blade.
3. From the **az140-cl-vm11a** blade, connect to the newly deployed Azure VM via Remote Desktop. When prompted to authenticate, specify **ADATUM\aadadmin1** as the username and the **Pa55w.rd1234** as its password.
4. Within the Remote Desktop session to **az140-cl-vm11a**, in the Start menu, navigate to the **Windows Administration Tools** folder, expand it, and select **Active Directory Users and Computers**.
5. In the **Active Directory Users and Computers** console, right-click the domain node, select **New**, followed by **Organizational Unit**, in the **New Object - Organizational Unit** dialog box, in the **Name** textbox, type **ADDC Users**, and select **OK**.
6. In the **Active Directory Users and Computers** console, right-click the **ADDC Users**, select **New**, followed by **Group**, in the **New Object - Group** dialog box, specify the following settings and select **OK**:

Setting	Value

Setting	Value
Group name	**Local Admins**
Group name (pre-Windows 2000)	**Local Admins**
Group scope	**Global**
Group type	**Security**

1. In the **Active Directory Users and Computers** console, display the properties of the **Local Admins** group, switch to the **Members** tab, select **Add**, in the **Select Users, Contacts, Computers, Service Accounts, or Groups** dialog box, in the **Enter the object names to select**, type **aadadmin1** and select **OK**.
2. Within the Remote Desktop session to **az140-cl-vm11a**, in the Start menu, navigate to the **Windows Administration Tools** folder, expand it, and select **Group Policy Management**.
3. In the **Group Policy Management** console, navigate to the **AADDC Computers** OU, right-click the **AADDC Computers GPO** icon and select **Edit**.
4. In the **Group Policy Management Editor** console, expand **Computer Configuration**, **Policies**, **Windows Settings**, **Security Settings**, right-click **Restriced Groups**, and select **Add Group**.
5. In the **Add Group** dialog box, in the **Group** text box, select **Browse**, in the **Select Groups** dialog box, in the **Enter the object names to select**, type **Local Admins** and select **OK**.
6. Back in the **Add Group** dialog box, select **OK**.
7. In the **ADATUM\Local Admins Properties** dialog box, in the section labeled **This group is a member of**, select **Add**, in the **Group Membership** dialog box, type **Administrators**, select **OK**, and select **OK** again to finalize the change.

Note: Make sure to use the section labeled **This group is a member of**

1. Within the Remote Desktop to the az140-cl-vm11a Azure VM, start PowerShell ISE as Administrator and run the following to restart the two Windows Virtual Desktop hosts in order to trigger Group Policy processing:

```
powershell $servers = 'az140-21-p1-0','az140-21-p1-1'
Restart-Computer -ComputerName $servers -Force -Wait
```

1. Wait for the script to complete. This should take about 3 minutes.

Task 2: Configure FSLogix-based profiles on Windows Virtual Desktop session host VMs

1. Within the Remote Desktop session to **az140-cl-vm11a**, start a Remote Desktop session to **az140-21-p1-0** and, when prompted, sign in with the following credentials:

Setting	Value
User Name	**ADATUM\wvdaadmin1**
Password	**Pa55w.rd1234**

1. Within the Remote Desktop session to **az140-21-p1-0**, start Microsoft Edge, browse to FSLogix download page, download FSLogix compressed installation binaries, extract them into the **C:\Source** folder, navigate to the **x64\Release** subfolder and use **FSLogixAppsSetup.exe** to install Microsoft FSLogix Apps with the default settings.
2. Within the Remote Desktop session to **az140-21-p1-0**, start **Windows PowerShell ISE** as administrator and, 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):

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

1. 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):

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

1. From the **Administrator: Windows PowerShell ISE** console, run the following to modify the execution policy:

```
powershell Set-ExecutionPolicy RemoteSigned -Force
```

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

```
powershell Connect-AzAccount
```

1. When prompted, sign in with the Azure AD credentials of the user account with the Owner role in the subscription you are using in this lab.
2. From the **Administrator: Windows PowerShell ISE** script pane, run the following to retrieve the name of the Azure Storage account you configured earlier in this lab:

```
powershell $resourceGroupName = 'az140-22a-RG'
$storageAccountName = (Get-AzStorageAccount -
ResourceGroupName $resourceGroupName)[0].StorageAccountName
```

1. Within the Remote Desktop session to **az140-21-p1-0**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to configure profile registry settings:

```
powershell $profilesParentKey = 'HKLM:\SOFTWARE\FSLogix'
$profilesChildKey = 'Profiles' $fileShareName = 'az140-22a-
profiles' New-Item -Path $profilesParentKey -Name
$profilesChildKey -Force New-ItemProperty -Path
$profilesParentKey\$profilesChildKey -Name 'Enabled' -
PropertyType DWord -Value 1 New-ItemProperty -Path
$profilesParentKey\$profilesChildKey -Name 'VHDLocations' -
PropertyType MultiString -Value
"\\$storageAccountName.file.core.windows.net\$fileShareName"
```

1. Within the Remote Desktop session to **az140-21-p1-0**, right-click **Start**, in the right-click menu, select **Run**, in the **Run** dialog box, in the **Open** text box, type the following and select **OK** to launch the **Local Users and Groups** window:

```
cmd lusrmgr.msc
```

1. In the **Local Users and Groups** console, note the four groups which names start with the **FSLogix** string:

2. FSLogix ODFC Exclude List

3. FSLogix ODFC Include List
4. FSLogix Profile Exclude List

5. FSLogix Profile Include List

6. In the **Local Users and Groups** console, double-click the **FSLogix Profile Include List** group enty, note that it includes the **\Everyone** group, and select **OK** to close the group **Properties** window.

7. In the **Local Users and Groups** console, double-click the **FSLogix Profile Exclude List** group enty, note that it does not include any group members by default, and select **OK** to close the group **Properties** window.

Note: To provide consistent user experience, you need to install and configure FSLogix components on all Windows Virtual Desktop session hosts. You will perform this task in the unattended manner on the other session host in our lab environment.

1. Within the Remote Desktop session to **az140-21-p1-0**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to install FSLogix components on the **az140-21-p1-1** session host:

```
powershell $server = 'az140-21-p1-1' $localPath =
'C:\Source\x64' $remotePath =
"\\$server\C$\Source\x64\Release" Copy-Item -Path
$localPath\Release -Destination $remotePath -Filter '*.exe' -
Force -Recurse Invoke-Command -ComputerName $server -
ScriptBlock { Start-Process -FilePath
$using:localPath\Release\FSLogixAppsSetup.exe -ArgumentList
'/quiet' -Wait }
```

1. Within the Remote Desktop session to **az140-21-p1-0**, start **Windows PowerShell ISE** as administrator and, from the **Administrator: Windows PowerShell ISE** script pane, run the following to configure profile registry settings on the **az140-21-p1-1** session host:

```
powershell $profilesParentKey = 'HKLM:\SOFTWARE\FSLogix'
$profilesChildKey = 'Profiles' $fileShareName = 'az140-22a-
profiles' Invoke-Command -ComputerName $server -ScriptBlock {
New-Item -Path $using:profilesParentKey -Name
$using:profilesChildKey -Force New-ItemProperty -Path
$using:profilesParentKey\$using:profilesChildKey -Name
'Enabled' -PropertyType DWord -Value 1 New-ItemProperty -Path
$using:profilesParentKey\$using:profilesChildKey -Name
'VHDLocations' -PropertyType MultiString -Value
"\\$using:storageAccountName.file.core.windows.net\$using:fil
eShareName" }
```

Note: Before you test the FSLogix-based profile functionality, you need to remove the locally cached profile of the ADATUM\wvdaadmin1 account you will be using for testing from

the Windows Virtual Desktop session hosts you used in the previous lab.

1. Switch to the Remote Desktop session to **az140-cl-vm11a**, within the Remote Desktop session to **az140-cl-vm11a**, switch to the **Administrator: Windows PowerShell ISE** window and, from the **Administrator: Windows PowerShell ISE** script pane, run the following to remove the locally cached profile of the ADATUM\aaduser1 account:

```
powershell $userName = 'aaduser1' $servers = 'az140-21-p1-
0','az140-21-p1-1' Get-CimInstance -ComputerName $servers -
Class Win32_UserProfile | Where-Object {
$_.LocalPath.split('\')[-1] -eq $userName } | Remove-
CimInstance
```

Task 3: Test FSLogix-based profiles with Windows Virtual Desktop

1. Within the Remote Desktop session to **az140-cl-vm11a**, switch to the Remote Desktop client.
2. Within the Remote Desktop session to **az140-cl-vm11a**, in the **Remote Desktop** client window, in the list of applications, double-click **Command Prompt**, when prompted, provide the password, and verify that it launches a **Command Prompt** window.

Note: Initially, it might take a few minutes for the application to start, but subsequently, the application startup should be much faster.

1. In the upper left corner of the **Command Prompt** window, right-click the **Command Prompt** icon and, in the drop-down menu, select **Properties**.
2. In the **Command Prompt Properties** dialog box, select the **Font** tab, modify the size and font settings, and select **OK**.
3. From the **Command Prompt** window, type **logoff** and press the **Enter** key to sign out from the Remote Desktop session.
4. Within the Remote Desktop session to **az140-cl-vm11a**, in the **Remote Desktop** client window, in the list of applications, double-click **SessionDesktop** and verify that it launches a Remote Desktop session.
5. Within the **SessionDesktop** session, right-click **Start**, in the right-click menu, select **Run**, in the **Run** dialog box, in the **Open** text

box, type **cmd** and select **OK** to launch a **Command Prompt** window:

6. Verify that the **Command Prompt** window properties match those you set earlier in this task.
7. Within the **SessionDesktop** session, minimize all windows, right-click the desktop, in the right-click menu, select **New** and, in the cascading menu, select **Shortcut**.
8. On the **What item would you like to create a shortcut for?** page of the **Create Shortcut** wizard, in the **Type the location of the item** text box, type **Notepad** and select **Next**.
9. On the **What would you like to name the shortcut** page of the **Create Shortcut** wizard, in the **Type a name for this shortcut** text box, type **Notepad** and select **Finish**.
10. Within the **SessionDesktop** session, right-click **Start**, in the right-click menu, select **Shut down or sign out** and then, in the cascading menu, select **Sign out**.
11. Back in the Remote Desktop session to **az140-cl-vm11a**, in the **Remote Desktop** client window, in the list of applications, and double-click **SessionDesktop** to start a new Remote Desktop session.
12. Within the **SessionDesktop** session, verify that the **Notepad** shortcut appears on the desktop.
13. Within the **SessionDesktop** session, right-click **Start**, in the right-click menu, select **Shut down or sign out** and then, in the cascading menu, select **Sign out**.
14. Switch to the Remote Desktop session to **az140-cl-vm11a**, switch to the Microsoft Edge window displaying the Azure portal.
15. In the Microsoft Edge window displaying the Azure portal, navigate back to the **Storage accounts** blade and select the entry representing the storage account you created in the prevous exercise.
16. On the storage account blade, in the **File services** section, select **File shares** and then, in the list of file shares, select **az140-22a-profiles**.
17. On the **az140-22a-profiles** blade, verify that its content includes a folder which name consists of a combination of the Security Identifier (SID) of the **ADATUM\aaduser1** account followed by the **_aaduser1** suffix.
18. Select the folder you identified in the previous step and note that it contains a single file named **Profile_aaduser1.vhd**.

---

lab: title: 'Lab: Implement and manage Windows Virtual Desktop profiles (AD DS)' module: 'Module 4: Manage User Environments and Apps'

Lab - Implement and manage Windows Virtual Desktop profiles (AD DS)

Student lab manual

Lab dependencies

- An Azure subscription you will be using in this lab.
- A Microsoft account or an Azure AD 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 Azure AD tenant associated with that Azure subscription.
- The completed lab **Prepare for deployment of Azure Windows Virtual Desktop (AD DS)**
- The completed lab **Implement and manage storage for WVD (AD DS)**

Estimated Time

30 minutes

Lab scenario

You need to implement Windows Virtual Desktop profile management in an Active Directory Domain Services (AD DS) environment.

Objectives

After completing this lab, you will be able to:

- Implement FSLogix based profiles for Windows Virtual Desktop

Lab files

- None

Instructions

Exercise 1: Implement FSLogix based profiles for Windows Virtual Desktop

The main tasks for this exercise are as follows:

1. Configure FSLogix-based profiles on Windows Virtual Desktop session host VMs
2. Test FSLogix-based profiles with Windows Virtual Desktop
3. Remove Azure resources deployed in the lab

Task 1: Configure FSLogix-based profiles on Windows Virtual Desktop session host VMs

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 **Virtual machines** and, from the **Virtual machines** blade, select **az140-dc-vm11**.
3. On the **az140-dc-vm11** blade, select **Connect**, in the drop-down menu, select **RDP**, on the **RDP** tab of the **az140-dc-vm11 | Connect** blade, in the **IP address** drop-down list, select the **Load balancer DNS name** entry, and then select **Download RDP File**.
4. When prompted, sign in with the following credentials:

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

1. Within the Remote Desktop session to **az140-dc-vm11**, start **Windows PowerShell ISE** as administrator.

Note: The next step is necessary to prepare for the next lab. Restarting the hosts ensures that the user profiles are offloaded.

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to restart both session host VMs

```
powershell $servers = 'az140-21-p1-0','az140-21-p1-1'
Restart-Computer -ComputerName $servers -Force
```

1. Within the Remote Desktop session to **az140-dc-vm11**, start Microsoft Edge and navigate to the Azure portal. If prompted, sign in by using the Azure AD credentials of the user account with the Owner role in the subscription you are using in this lab.
2. Within the Remote Desktop session to **az140-dc-vm11**, in the Microsoft Edge window displaying the Azure portal, search for and select **Virtual machines** blade and, on the **Virtual machines** blade, select **az140-21-p1-0**.
3. On the **az140-21-p1-0** blade, select **Connect**, in the drop-down menu, select **RDP**, on the **az140-21-p1-0 | Connect** blade, in the **IP address** dropdown list, select the **Private IP address** entry, select **Download RDP File**, and then select **Open**.
4. When prompted, sign in with the following credentials:

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

1. Within the Remote Desktop session to **az140-21-p1-0**, start Microsoft Edge, browse to FSLogix download page, download FSLogix compressed installation binaries, extract them into the **C:\Allfiles\Labs\04** folder (create the folder if needed), navigate to the **x64\Release** subfolder, double-click the **FSLogixAppsSetup.exe** file to launch the **Microsoft FSLogix Apps Setup** wizard, and step through the installation of Microsoft FSLogix Apps with the default settings.
2. Within the Remote Desktop session to **az140-21-p1-0**, start **Windows PowerShell ISE** as administrator and, 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):

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

1. 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):

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

1. From the **Administrator: Windows PowerShell ISE** console, run the following to modify the execution policy:

```
powershell Set-ExecutionPolicy RemoteSigned -Force
```

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

```
powershell Connect-AzAccount
```

1. When prompted, sign in with the Azure AD credentials of the user account with the Owner role in the subscription you are using in this lab.
2. Within the Remote Desktop session to **az140-21-p1-0**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to retrieve the name of the Azure Storage account you configured earlier in this lab:

```
powershell $resourceGroupName = 'az140-22-RG'
$storageAccountName = (Get-AzStorageAccount -
ResourceGroupName $resourceGroupName)[0].StorageAccountName
```

1. Within the Remote Desktop session to **az140-21-p1-0**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to configure profile registry settings:

```
powershell $profilesParentKey = 'HKLM:\SOFTWARE\FSLogix'
$profilesChildKey = 'Profiles' $fileShareName = 'az140-22-
profiles' New-Item -Path $profilesParentKey -Name
$profilesChildKey -Force New-ItemProperty -Path
$profilesParentKey\$profilesChildKey -Name 'Enabled' -
PropertyType DWord -Value 1 New-ItemProperty -Path
$profilesParentKey\$profilesChildKey -Name 'VHDLocations' -
PropertyType MultiString -Value
"\\$storageAccountName.file.core.windows.net\$fileShareName"
```

1. Within the Remote Desktop session to **az140-21-p1-0**, right-click **Start**, in the right-click menu, select **Run**, in the **Run** dialog box, in the **Open** text box, type the following and select **OK** to launch the **Local Users and Groups** console:

```
cmd lusrmgr.msc
```

1. In the **Local Users and Groups** console, note the four groups which names start with the **FSLogix** string:

2. FSLogix ODFC Exclude List

3. FSLogix ODFC Include List
4. FSLogix Profile Exclude List

5. FSLogix Profile Include List

6. In the **Local Users and Groups** console, in the list of groups, double-click the **FSLogix Profile Include List** group, note that it includes the **\Everyone** group, and select **OK** to close the group **Properties** window.

7. In the **Local Users and Groups** console, in the list of groups, double-click the **FSLogix Profile Exclude List** group, note that it does not include any group members by default, and select **OK** to close the group **Properties** window.

Note: To provide consistent user experience, you need to install and configure FSLogix components on all Windows Virtual Desktop session hosts. You will perform this task in the unattended manner on the other session host in our lab environment.

1. Within the Remote Desktop session to **az140-21-p1-0**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to install FSLogix components on the **az140-21-p1-1** session host:

```
powershell $server = 'az140-21-p1-1' $localPath =
'C:\Allfiles\Labs\04\x64' $remotePath =
"\\$server\C$\Allfiles\Labs\04\x64\Release" Copy-Item -Path
$localPath\Release -Destination $remotePath -Filter '*.exe' -
Force -Recurse Invoke-Command -ComputerName $server -
ScriptBlock { Start-Process -FilePath
$using:localPath\Release\FSLogixAppsSetup.exe -ArgumentList
'/quiet' -Wait }
```

Note: Wait for the script execution to complete. This might take about 2 minutes.

1. Within the Remote Desktop session to **az140-21-p1-0**, from the **Administrator: Windows PowerShell ISE** script pane, run the

following to configure profile registry settings on the **az140-21-p1-1** session host:

```
powershell $profilesParentKey = 'HKLM:\SOFTWARE\FSLogix'
$profilesChildKey = 'Profiles' $fileShareName = 'az140-22-
profiles' Invoke-Command -ComputerName $server -ScriptBlock {
New-Item -Path $using:profilesParentKey -Name
$using:profilesChildKey -Force New-ItemProperty -Path
$using:profilesParentKey\$using:profilesChildKey -Name
'Enabled' -PropertyType DWord -Value 1 New-ItemProperty -Path
$using:profilesParentKey\$using:profilesChildKey -Name
'VHDLocations' -PropertyType MultiString -Value
"\\$using:storageAccountName.file.core.windows.net\$using:fil
eShareName" }
```

Note: Before you test the FSLogix-based profile functionality, you need to remove the locally cached profile of the **ADATUM\aduser1** account you will be using for testing from the Windows Virtual Desktop session hosts you used in the previous lab.

1. Within the Remote Desktop session to **az140-21-p1-0**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to remove the locally cached profile of the **ADATUM\aduser1** account on both Azure VMs serving as session hosts:

```
powershell $userName = 'aduser1' $servers = 'az140-21-p1-
0','az140-21-p1-1' Get-CimInstance -ComputerName $servers -
Class Win32_UserProfile | Where-Object {
$_.LocalPath.split('\')[-1] -eq $userName } | Remove-
CimInstance
```

Task 2: Test FSLogix-based profiles with Windows Virtual Desktop

1. Switch to your lab computer, from the lab computer, in the browser window displaying the Azure portal, search for and select **Virtual machines** and, on the **Virtual machines** blade, select the **az140-cl-vm11** entry.
2. On the **az140-cl-vm11** blade, select **Connect**, in the drop-down menu, select **RDP**, and then select **Download RDP File**.
3. When prompted, sign in with the **ADATUM\aduser1** credentials.
4. Within the Remote Desktop session to **az140-cl-vm11**, click **Start** and, in the **Start** menu, click **Remote Desktop** to start the Remote Desktop client.

5. Within the Remote Desktop session to **az140-cl-vm11**, in the **Remote Desktop** client window, in the list of applications, double-click **Command Prompt**, when prompted, provide the password of the **aduser1** account, and verify a **Command Prompt** window opens successfully.
6. In the upper left corner of the **Command Prompt** window, right-click the **Command Prompt** icon and, in the drop-down menu, select **Properties**.
7. In the **Command Prompt Properties** dialog box, select the **Font** tab, modify the size and font settings, and select **OK**.
8. From the **Command Prompt** window, type **logoff** and press the **Enter** key to sign out from the Remote Desktop session.
9. Within the Remote Desktop session to **az140-cl-vm11**, in the **Remote Desktop** client window, in the list of applications, double-click **SessionDesktop** under az-140-21-ws1 and verify that it launches a Remote Desktop session.
10. Within the **SessionDesktop** session, right-click **Start**, in the right-click menu, select **Run**, in the **Run** dialog box, in the **Open** text box, type **cmd** and select **OK** to launch a **Command Prompt** window:
11. Verify that the **Command Prompt** window settings match those you configured earlier in this task.
12. Within the **SessionDesktop** session, minimize all windows, right-click the desktop, in the right-click menu, select **New** and, in the cascading menu, select **Shortcut**.
13. On the **What item would you like to create a shortcut for?** page of the **Create Shortcut** wizard, in the **Type the location of the item** text box, type **Notepad** and select **Next**.
14. On the **What would you like to name the shortcut** page of the **Create Shortcut** wizard, in the **Type a name for this shortcut** text box, type **Notepad** and select **Finish**.
15. Within the **SessionDesktop** session, right-click **Start**, in the right-click menu, select **Shut down or sign out** and then, in the cascading menu, select **Sign out**.
16. Back in the Remote Desktop session to **az140-cl-vm11**, in the **Remote Desktop** client window, in the list of applications, and double-click **SessionDesktop** to start a new Remote Desktop session.
17. Within the **SessionDesktop** session, verify that the **Notepad** shortcut appears on the desktop.
18. Within the **SessionDesktop** session, right-click **Start**, in the right-click menu, select **Shut down or sign out** and then, in the

cascading menu, select **Sign out**.

19. Switch to your lab computer and, in the Microsoft Edge window displaying the Azure portal, navigate to the **Storage accounts** blade and select the entry representing the storage account you created in the previous exercise.
20. On the storage account blade, in the **File services** section, select **File shares** and then, in the list of file shares, select **az140-22-profiles**.
21. On the **az140-22-profiles** blade, verify that its content includes a folder which name consists of a combination of the Security Identifier (SID) of the **ADATUM\aduser1** account followed by the **_aduser1** suffix.
22. Select the folder you identified in the previous step and note that it contains a single file named **Profile_aduser1.vhd**.

---

lab: title: 'Lab: Package Windows Virtual Desktop applications (AD DS)' module: 'Module 4: Manage User Environments and Apps'

Lab - Package Windows Virtual Desktop applications (AD DS)

Student lab manual

Lab dependencies

- An Azure subscription
- A Microsoft account or an Azure AD account with the Global Administrator role in the Azure AD 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 Windows Virtual Desktop (AD DS)** or **Prepare for deployment of Azure Windows Virtual Desktop (Azure AD DS)**
- The completed lab **Windows Virtual Desktop profile management (AD DS)** or **Windows Virtual Desktop profile management (Azure AD DS)**

Note: At the time of authoring this lab, the MSIX app attach functionality for Windows Virtual Desktop is in public preview. In order to try it, you need to submit a request via on online form to enable MSIX app attach in your subscription. The approval and processing of requests can take up to 24 hours during business days. You'll receive an email confirmation once your request has been accepted and completed.

Estimated Time

90 minutes

Lab scenario

You need to package and deploy Windows 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 MSIX app attach container for Windows Virtual Desktop in AD DS environment
- Implement the MSIX app attach on Windows Virtual Desktop in AD DS environment

Lab files

- \\AZ-140\AllFiles\Labs\04\az140-42_azuredeployc142.json
- \\AZ-140\AllFiles\Labs\04\az140-42_azuredeployc142.parameters.json

Instructions

Exercise 1: Prepare for and create MSIX app packages

The main tasks for this exercise are as follows:

1. Prepare for configuration of Windows 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 Windows 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 Windows Virtual Desktop session host Azure VMs you will be using in this lab:

```
powershell 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**: Proceed directly to the next task without waiting for the Azure VMs to start.

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 Azure AD DS domain:

```
powershell $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 **RDP**, on the **RDP** tab of the **az140-cl-vm42 | Connect** blade, and then select **Download RDP File**.
3. When prompted, sign in with the following credentials:

Setting	Value
User Name	**ADATUM\wvdadmin1**

Setting	Value
Password	**Pa55w.rd1234**

1. Within the Remote Desktop 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:

```
powershell 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\ContentDeliver
yManager /v PreInstalledAppsEnabled /t REG_DWORD /d 0 /f reg
add
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliver
yManager\Debug /v ContentDeliveryAllowedOverride /t REG_DWORD
/d 0x2 /f reg add
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Syste
m /v EnableLUA /t REG_DWORD /d 0 /f reg add
HKLM\Software\Microsoft\RDInfraAgent\MSIXAppAttach /v
PackageListCheckIntervalMinutes /t REG_DWORD /d 1 /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 Remote Desktop session to **az140-cl-vm42**, start **Windows PowerShell ISE** as administrator, 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:

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

1. Within the Remote Desktop session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** console, run the following to start the **Certificates** console targeting the Local Machine certificate store:

```
powershell certlm.msc
```

1. 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**.
2. On the **Welcome to the Certificate Export Wizard** page of the **Certificate Export Wizard**, select **Next**.
3. On the **Export Private Key** page of the **Certificate Export Wizard**, select the option **Yes, export the private key** option and select **Next**.
4. 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**.
5. On the **Security** page of the **Certificate Export Wizard**, select the **Password** checkbox, in the textboxes below, type **Pa55w.rd1234**, and select **Next**.
6. 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 if needed), in the **File name** textbox, type **adatum.pfx**, and select **Save**.
7. 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**.
8. 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.

1. 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:

```
powershell $wvdhosts = 'az140-21-p1-0','az140-21-p1-
1','az140-21-p1-2' $cleartextPassword = 'Pa55w.rd1234'
$securePassword = ConvertTo-SecureString $cleartextPassword -
AsPlainText -Force ForEach ($wvdhost in $wvdhosts){
$localPath = 'C:\Allfiles\Labs\04' $remotePath =
"\\$wvdhost\C$\Allfiles\Labs\04\" Copy-Item -Path
"$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 Remote Desktop 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 Remote Desktop session to **az140-cl-vm42**, start File Explorer, navigate to the **Downloads** folder, open the compressed file, copy its content, create a folder **C:\AllFiles\Labs\04**, and paste the copied content into the newly created folder.

Task 6: Install the MSIX Packaging Tool

1. Within the Remote Desktop 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 Remote Desktop 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:

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

1. Within the Remote Desktop 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:

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

1. Within the Remote Desktop 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:

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

1. Within the Remote Desktop 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.
2. 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**.
3. 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 in back by using the **ADATUM\wvdadmin1** 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.

1. On the **Prepare computer** page of the **Create new package** wizard, click **Next**.
2. 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**,

3. 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**.
4. 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**. This will trigger installation of the downloaded software.
5. In the **XMLNotepad Setup** window, accept the terms in the License Agreement and select **Install** and, once the installation completes, select the **Launch XML Notepad** checkbox and select **Finish**.
6. Verify that XML Notepad is running, close it, switch back to the **Create new package** wizard in the **MSIX Packaging Tool** window, and select **Next**.

Note: In this case, restart is not required to complete the installation.

1. On the **First launch tasks** page of the **Create new package** wizard, review the provided information and select **Next**.
2. When prompted **Are you done?**, select **Yes, move on**.
3. On the **Services report** page of the **Create new package** wizard, verify that no services are listed and select **Next**.
4. On the **Create package** page of the **Create new package** wizard, in the **Save location** textbox, type **C:\Allfiles\Labs\04\XmlNotepad** and click **Create**.
5. In the **Package successfully created** dialog box, note the location of the saved package and select **Close**.
6. 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.

Exercise 2: Implement MSIX app attach container for Windows Virtual Desktop in Azure AD DS environment

The main tasks for this exercise are as follows:

1. Enable Hyper-V on the Azure VMs running Window 10 Enterprise Edition
2. Create an app attach container

Task 1: Enable Hyper-V on the Azure VMs running Window 10 Enterprise Edition

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

```
powershell $wvdhosts = 'az140-21-p1-0','az140-21-p1-
1','az140-21-p1-2' ForEach ($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\ContentDeliver
yManager /v PreInstalledAppsEnabled /t REG_DWORD /d 0 /f reg
add
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\ContentDeliver
yManager\Debug /v ContentDeliveryAllowedOverride /t REG_DWORD
/d 0x2 /f reg add
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\Syste
m /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 } }
```

1. Within the Remote Desktop 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 Windows Virtual Desktop hosts:

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

1. Following the installation of the Hyper-V components on each host, type **Y** and press the **Enter** key to restart the target operating

system.
2. Within the Remote Desktop 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:

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

1. Once the installation of the Hyper-V components completes, type **Y** and press the **Enter** key to restart the operating system. Following the restart, sign in back by using the **ADATUM\wvdadmin1** account with the **Pa55w.rd1234** password.

Task 2: Create an app attach container

1. Within the Remote Desktop session to **az140-cl-vm42**, start **Microsoft Edge**, browse to **https://aka.ms/msixmgr** and, when prompted whether to open or save **msixmgr.zip** file, click **Save**. This will download 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 **x64** folder to the **C:\AllFiles\Labs\04** folder.
3. Within the Remote Desktop 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 VHD file that will serve as the app attach container:

```
powershell New-Item -ItemType Directory -Path
'C:\Allfiles\Labs\04\MSIXVhds' -Force New-VHD -SizeBytes
128MB -Path 'C:\Allfiles\Labs\04\MSIXVhds\XmlNotepad.vhd' -
Dynamic -Confirm:$false
```

1. Within the Remote Desktop session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to mount the newly created VHD file:

```
powershell $vhdObject = Mount-VHD -Path
'C:\Allfiles\Labs\04\MSIXVhds\XmlNotepad.vhd' -Passthru
```

1. Within the Remote Desktop session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to initialize the disk, create a new partition, format it, and assign to it the first available drive letter:

```
powershell $disk = Initialize-Disk -Passthru -Number
$vhdObject.Number $partition = New-Partition -
AssignDriveLetter -UseMaximumSize -DiskNumber $disk.Number
Format-Volume -FileSystem NTFS -Confirm:$false -DriveLetter
$partition.DriveLetter -Force
```

1. Within the Remote Desktop session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to create a folder structure that will host the MSIX files and unpack into it the MSIX package you created in the previous task:

```
powershell $appName = 'XmlNotepad' $msixPackage = Get-
ChildItem -Path "C:\AllFiles\Labs\04\$appName" -Filter *.msix
-File C:\AllFiles\Labs\04\x64\msixmgr.exe -Unpack -
packagePath $msixPackage.FullName -destination
"$($partition.DriveLetter):\Apps\$appName" -applyacls
```

1. Within the Remote Desktop session to **az140-cl-vm42**, in File Explorer, navigate to the **F:\Apps\XmlNoteppad** folder and review its content.
2. Within the Remote Desktop session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to identify the GUID of volume hosting the unpacked MSIX package

```
powershell $uniqueId = (Get-Volume -DriveLetter
"$($partition.DriveLetter)").UniqueId $regex = [regex]"\
{(.*)\}" [regex]::match($uniqueId, $regex).Groups[1].value
```

Note: Record the GUID value you identified. You will need it in the next exercise of this lab.

1. Within the Remote Desktop session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** console, run the following to unmount the VHD file that will serve as the app attach container:

```
powershell Dismount-VHD -Path
"C:\Allfiles\Labs\04\MSIXVhds\$appName.vhd" -Confirm:$false
```

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

The main tasks for this exercise are as follows:

1. Configure Active Directory groups containing Windows Virtual Desktop hosts
2. Set up the Azure Files share for MSIX app attach
3. Mount and register the MSIX App attach container on Windows 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 Windows 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 **RDP**, on the **RDP** tab of the **az140-dc-vm11 | Connect** blade, in the **IP address** drop-down list, select the **Load balancer DNS name** entry, and then select **Download RDP File**.
3. When prompted, sign in with the following credentials:

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

1. Within the Remote Desktop session to **az140-dc-vm11**, start **Windows PowerShell ISE** as administrator.
2. From the **Administrator: Windows PowerShell ISE** script pane, run the following to create an AD DS group object that will be synchronized to the Azure AD tenant used in this lab:

```
powershell $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 Windows Virtual Desktop hosts permissions to the **az140-42-msixvhds** file share.

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

```
powershell Get-ADGroup -Identity 'az140-hosts-42-p1' | Add-
AdGroupMember -Members 'az140-21-p1-0$','az140-21-p1-
1$','az140-21-p1-2$'
```

1. 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:

```
powershell $hosts = (Get-ADGroup -Identity 'az140-hosts-42-
p1' | Get-ADGroupMember | Select-Object Name).Name $hosts |
Restart-Computer
```

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

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to identify the user principal name of the **aadsyncuser** Azure AD user account:

```
powershell (Get-AzADUser -DisplayName
'aadsyncuser').UserPrincipalName
```

Note: Record the user principal name you identified in this step. You will need it later in this task.

1. Within the Remote Desktop session to **az140-dc-vm11**, in the **Start** menu, expand the **Azure AD Connect** folder and select **Azure AD Connect**.
2. On the **Welcome to Azure AD Connect** page of the **Microsoft Azure Active Directory Connect** window, select **Configure**.
3. On the **Additional tasks** page in the **Microsoft Azure Active Directory Connect** window, select **Customize synchronization options** and select **Next**.
4. On the **Connect to Azure AD** 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 and the **Pa55w.rd1234** password.
5. On the **Connect your directories** page in the **Microsoft Azure Active Directory Connect** window, select **Next**.

6. 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**.
7. On the **Optional features** page in the **Microsoft Azure Active Directory Connect** window, accept the default settings, and select **Next**.
8. 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**.
9. Review the information on the **Configuration complete** page and select **Exit** to close the **Microsoft Azure Active Directory Connect** window.
10. Within the Remote Desktop session to **az140-dc-vm11**, start Internet Explorer and navigate to the Azure portal. When prompted, sign in by using the Azure AD credentials of the user account with the Global Administrator role in the Azure AD tenant associated with the Azure subscription you are using in this lab.
11. Within the Remote Desktop session to **az140-dc-vm11**, in the Internet Explorer window displaying the Azure portal, search for and select **Azure Active Directory** to navigate to the Azure AD tenant associated with the Azure subscription you are using for this lab.
12. On the Azure Active Directory blade, in the vertical menu bar on the left side, in the **Manage** section, click **Groups**.
13. On the **Groups | All groups** blade, in the list of groups, select the **az140-hosts-42-p1** entry.
14. On the **az140-hosts-42-p1** blade, in the vertical menu bar on the left side, in the **Manage** section, click **Members**.
15. On the **az140-hosts-42-p1 | Members** blade, verify that the list of **Direct members** include the three hosts of the Windows 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 Remote Desktop session to **az140-cl-vm42**.
2. Within the Remote Desktop 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.

1. Within the Remote Desktop 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 **Windows Virtual Desktop profile management (AD DS)** or **Windows Virtual Desktop profile management (Azure AD DS)**

Note: In production scenarios, you should consider using a separate storage account. This would require configuring that storage account for Azure AD 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.

1. On the storage account blade, in the vertical menu on the left side, in the **File services** section, select **File shares** and then select **+ File share**.
2. On the **New file share** blade, specify the following settings and select **Create** (leave other settings with their default values):

Setting	Value
Name	**az140-42-msixvhds**

1. In the Microsoft Edge displaying the Azure portal, in the list of file shares, select the newly created file share.
2. On the **az140-42a-msixvhds** blade, in the vertical menu on the left side, select **Access Control (IAM)**.
3. On the **az140-42a-msixvhds | Access Control (IAM)** blade of the storage account, select **+ Add** and, in the drop-down menu, select **Add role assignment**,
4. On the **Add role assignment** blade, specify the following settings and select **Save**:

Setting	Value
Role	**Storage File Data SMB Share Elevated Contributor**

Setting	Value
Assign access to	**User, group, or service principal**
Select	**az140-wvd-admins**

Note: The **az140-wvd-admins** group contains the **wvdadmin1** user account, which you'll use to configure share permissions.

1. Repeat the previous two steps to configure the following role assignments:

Setting	Value
Role	**Storage File Data SMB Share Elevated Contributor**
Assign access to	**User, group, or service principal**
Select	**az140-hosts-42-p1**

Setting	Value
Role	**Storage File Data SMB Share Reader**
Assign access to	**User, group, or service principal**
Select	**az140-wvd-users**

Note: Windows Virtual Desktop users and hosts need at least read access to the file share.

1. Within the Remote Desktop 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:

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

1. Within the Remote Desktop 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:

```
cmd 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 **ADATUM\wvdadmin1**.

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

1. Within the Remote Desktop 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:

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

Task 3: Run the MSIX app attach staging script on Windows Virtual Desktop hosts

1. Within the Remote Desktop session to **az140-cl-vm42**, 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):

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

1. 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):

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

1. From the **Administrator: Windows PowerShell ISE** console, run the following to modify the execution policy:

```
powershell Set-ExecutionPolicy RemoteSigned -Force
```

1. From the **Administrator: Windows PowerShell ISE** console, run the following to authenticate to the Azure subscription using in this lab:

```
powershell Connect-AzAccount
```

1. When prompted, sign in with the Azure AD credentials of the user account with the Owner role in the subscription you are using in this lab.
2. Within the Remote Desktop session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** console, run the following to identify the name of the Azure storage account hosting the file share containing the MSIX package:

```
powershell $resourceGroupName = 'az140-22-RG'
$storageAccountName = (Get-AzStorageAccount -
ResourceGroupName $resourceGroupName)[0].StorageAccountName
```

1. Within the Remote Desktop session to **az140-cl-vm42**, start Command Prompt as Administrator, from the Command Prompt, run the following to enable WinRM (when prompted for confirmation, type **Y** and press the **Enter** key:

```
cmd winrm qc
```

1. Within the Remote Desktop session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** console, run the following to enable CredSSP authentication for PowerShell Remoting to the target Windows Remote Desktop hosts:

```
powershell Enable-WSManCredSSP -Role client -DelegateComputer
* -Force
```

1. Within the Remote Desktop session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** console, run the following to enable CredSSP authentication for PowerShell Remoting to the target Windows Remote Desktop hosts:

```
powershell $wvdhosts = 'az140-21-p1-0','az140-21-p1-
1','az140-21-p1-2' ForEach ($wvdhost in $wvdhosts){ Invoke-
Command -ComputerName $wvdhost -ScriptBlock { winrm qc
Enable-PSRemoting -Force Enable-WSManCredSSP -Role server -
Force } }
```

Note: This is done to allow running the staging script remotely. Alternatively, you could run the script locally on each target host.

1. Within the Remote Desktop session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** console, run the following to perform the MSIX app attach staging (replace the

<volume_guid> placeholder with the volume GUID you identified in the previous exercise):

```powershell $username = 'ADATUM\wvdadmin1' $cleartextPassword = 'Pa55w.rd1234' $securePassword = ConvertTo-SecureString -AsPlainText $cleartextPassword -Force $creds = New-Object System.Management.Automation.PSCredential -ArgumentList $username,$securePassword

$wvdhosts = 'az140-21-p1-0','az140-21-p1-1','az140-21-p1-2' ForEach ($wvdhost in $wvdhosts){ Invoke-Command -ComputerName $wvdhost -Authentication Credssp -Credential $creds -ScriptBlock {

    $vhdSrc = "\\$using:storageAccountName.file.core.windows.net\az140-42-msixvhds\packages\XmlNotepad.vhd"
    Mount-Diskimage -ImagePath $vhdSrc -NoDriveLetter -Access ReadOnly
    $volumeGuid = '<volume_guid>'
    $msixDest = "\\?\Volume{" + $volumeGuid + "}\"

    $parentFolder = '\Apps\XmlNotepad\'
    $msixJunction = 'C:\Allfiles\Labs\04\AppAttach\'
    $packageName = "XmlNotepad_2.8.0.0_x64__4vm7ty4fw38e8"

    If (!(Test-Path -Path $msixJunction)) {New-Item -ItemType Directory -Path $msixJunction}
    $msixJunction = $msixJunction + $packageName
    cmd.exe /c mklink /j $msixJunction $msixDest


[Windows.Management.Deployment.PackageManager,Windows.Management.Deployment,ContentType=WindowsRuntime] | Out-Null
    Add-Type -AssemblyName System.Runtime.WindowsRuntime
    $asTask = ([System.WindowsRuntimeSystemExtensions].GetMethods() | Where { $_.ToString() -eq 'System.Threading.Tasks.Task`1[TResult] AsTask[TResult,TProgress] (Windows.Foundation.IAsyncOperationWithProgress`2[TResult,TProgress])'})[0]
    $asTaskAsyncOperation = $asTask.MakeGenericMethod([Windows.Management.Deployment.DeploymentResult], [Windows.Management.Deployment.DeploymentProgress])
    $packageManager = [Windows.Management.Deployment.PackageManager]::new()
    $path = $msixJunction + $parentFolder + $packageName
    $path = ([System.Uri]$path).AbsoluteUri
    $asyncOperation =
```

```
$packageManager.StagePackageAsync($path, $null,
"StageInPlace")
     $task = $asTaskAsyncOperation.Invoke($null,
@($asyncOperation))
     $task
  }

}```
```

Task 4: Initiate a Remote Desktop session to one of the pool hosts

1. Within the Remote Desktop session to **az140-cl-vm42**, start Microsoft Edge and navigate to Windows Desktop client download page and, when prompted, select **Run** 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 **Pa55w.rd1234** as its password.
4. If prompted, in the **Stay signed in to all your apps** window, clear the **Allow my organization to manage my device** checkbox and click **No, sign in to this app only**.
5. In the **Remote Desktop** client window, within the **az140-21-ws1** section, double-click **SessionDesktop** icon to open a Remote Desktop session to the host pool that is part of the **az140-21-ws1** workspace. When prompted provide the password for the **aduser1** account.

Task 5: Run the MSIX app attach registration script on Windows Virtual Desktop hosts

1. Within the Remote Desktop session from **az140-cl-vm42** to a host pool in the **az140-21-ws1** workspace, while signed in as **aduser1**, start **Windows PowerShell ISE** and, from the **Windows PowerShell ISE** console, run the following to perform the MSIX app attach registration:

```
powershell $packageName =
'XmlNotepad_2.8.0.0_x64__4vm7ty4fw38e8' $path = 'C:\Program
```

```
Files\WindowsApps\' + $packageName + '\AppxManifest.xml' Add-
AppxPackage -Path $path -DisableDevelopmentMode -Register
```

1. Within the Remote Desktop session from **az140-cl-vm42** to a host pool in the **az140-21-ws1** workspace, while signed in as **ADATUM\aduser1**, click **Start**, in the **Start** menu, click **XML Notepad** and verify that it successfully launches the XML Notepad app.
2. Close XML Notepad and the **Windows PowerShell ISE** window, right-click **Start**, in the right-click menu, select **Shut down or sign out** and, in the cascading menu, select **Sign out**.

Task 6: Run the MSIX app attach deregistration script

1. Back within the Remote Desktop 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 perform the MSIX app deregistration on all hosts in the **az140-21-hp1** host pool:

```
powershell $wvdhosts = 'az140-21-p1-0','az140-21-p1-
1','az140-21-p1-2' ForEach ($wvdhost in $wvdhosts){ Invoke-
Command -ComputerName $wvdhost -Authentication Credssp -
Credential $creds -ScriptBlock { $packageName =
"XmlNotepad_2.8.0.0_x64__4vm7ty4fw38e8" Remove-AppxPackage -
AllUsers -Package $packageName } }
```

1. Verify that the script completed successfully.

Task 7: Run the MSIX app attach destaging script

1. Within the Remote Desktop session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** console, run the following to identify the name of the Azure storage account hosting the file share containing the MSIX package:

```
powershell $resourceGroupName = 'az140-22-RG'
$storageAccountName = (Get-AzStorageAccount -
ResourceGroupName $resourceGroupName)[0].StorageAccountName
```

1. Within the Remote Desktop session to **az140-cl-vm42**, from the **Administrator: Windows PowerShell ISE** script pane, run the

following to perform the MSIX app destaging on all hosts in the **az140-21-hp1** host pool:

```
powershell $wvdhosts = 'az140-21-p1-0','az140-21-p1-
1','az140-21-p1-2' ForEach ($wvdhost in $wvdhosts){ Invoke-
Command -ComputerName $wvdhost -Authentication Credssp -
Credential $creds -ScriptBlock { $vhdSrc =
"\\$using:storageAccountName.file.core.windows.net\az140-42-
msixvhds\packages\XmlNotepad.vhd" $packageName =
"XmlNotepad_2.8.0.0_x64__4vm7ty4fw38e8" $msixJunction =
'C:\Allfiles\Labs\04\AppAttach' Remove-Item
"$msixJunction\$packageName" -Recurse -Force -Verbose
Dismount-DiskImage -ImagePath $vhdSrc -Confirm:$false } }
```

1. Verify that the script completed successfully.

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:

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

1. 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:

```
powershell 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.

lab: title: 'Lab: Implement autoscaling in host pools (AD DS)' module: 'Module: Monitor and Maintain a WVD Infrastructure'

Lab - Implement autoscaling in host pools (AD DS)

Student lab manual

Lab dependencies

- An Azure subscription you will be using in this lab.
- A Microsoft account or an Azure AD 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 Azure AD tenant associated with that Azure subscription.
- The completed lab **Prepare for deployment of Azure Windows Virtual Desktop (AD DS)**
- The completed lab **Deploy host pools and session hosts by using the Azure portal (AD DS)**

Estimated Time

60 minutes

Lab scenario

You need to configure autoscaling of Windows Virtual Desktop session hosts in an Active Directory Domain Services (AD DS) environment.

Objectives

After completing this lab, you will be able to:

- Configure autoscaling of Windows Virtual Desktop session hosts
- Verify autoscaling of Windows Virtual Desktop session hosts

Lab files

- None

Instructions

Exercise 1: Configure autoscaling of Windows Virtual Desktop session hosts

The main tasks for this exercise are as follows:

1. Prepare for autoscaling of Windows Virtual Desktop session hosts
2. Create and configure an Azure Automation account
3. Create an Azure Logic app

Task 1: Prepare for autoscaling of Windows 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 Windows Virtual Desktop session host Azure VMs you will be using in this lab:

```
powershell 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.

Task 2: Create and configure an Azure Automation account

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 **Virtual machines** and, from the **Virtual machines** blade, select **az140-dc-vm11**.
3. On the **az140-dc-vm11** blade, select **Connect**, in the drop-down menu, select **RDP**, on the **RDP** tab of the **az140-dc-vm11 | Connect** blade, in the **IP address** drop-down list, select the **Load balancer DNS name** entry, and then select **Download RDP File**.
4. When prompted, sign in with the following credentials:

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

1. Within the Remote Desktop session to **az140-dc-vm11**, start **Windows PowerShell ISE** as administrator.
2. From the **Administrator: Windows PowerShell ISE** console, run the following to sign in to your Azure subscription:

```
powershell Connect-AzAccount
```

1. When prompted, sign in with the Azure AD credentials of the user account with the Owner role in the subscription you are using in this lab.
2. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to download the PowerShell script you will use to create the Azure Automation account that is part of the autoscaling solution:

```
powershell [Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12 $labFilesfolder =
'C:\Allfiles\Labs\05' New-Item -ItemType Directory -Path
$labFilesfolder -Force Set-Location -Path $labFilesfolder
$uri = 'https://raw.githubusercontent.com/Azure/RDS-
Templates/master/wvd-templates/wvd-scaling-
script/CreateOrUpdateAzAutoAccount.ps1' Invoke-WebRequest -
Uri $Uri -OutFile '.\CreateOrUpdateAzAutoAccount.ps1'
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to set the values of variables that you will assign to script parameters:

```
powershell $aadTenantId = (Get-AzContext).Tenant.Id
$subscriptionId = (Get-AzContext).Subscription.Id
```

```
$resourceGroupName = 'az140-51-RG' $location = (Get-
AzVirtualNetwork -ResourceGroupName 'az140-11-RG' -Name
'az140-adds-vnet11').Location $suffix = Get-Random
$automationAccountName = "az140-automation-51$suffix"
$workspaceName = "az140-workspace-51$suffix"
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to create the resource group you will use in this lab:

```
powershell New-AzResourceGroup -ResourceGroupName
$resourceGroupName -Location $location
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to create an Azure Log Analytics workspace you will use in this lab:

```
powershell New-AzOperationalInsightsWorkspace -Location
$location -Name $workspaceName -Sku Standard -
ResourceGroupName $resourceGroupName
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE**, open the **C:\Allfiles\Labs\05\CreateOrUpdateAzAutoAccount.ps1** script and enclose the code between lines **82** an **86** into the multiline comment, such that they look as follows:

```
powershell <# # Get the Role Assignment of the authenticated
user $RoleAssignments = Get-AzRoleAssignment -SignInName
$AzContext.Account -ExpandPrincipalGroups if (!
($RoleAssignments | Where-Object { $_.RoleDefinitionName -in
@('Owner', 'Contributor') })) { throw 'Authenticated user
should have the Owner/Contributor permissions to the
subscription' } #>
```

1. Within the Remote Desktop session to **az140-dc-vm11**, open a new tab in the **Administrator: Windows PowerShell ISE** script pane, paste the following script, and run it to create the Azure Automation account that is part of the autoscaling solution:

```powershell $Params = @{ "AADTenantId" = $aadTenantId
"SubscriptionId" = $subscriptionId "UseARMAPI" = $true
"ResourceGroupName" = $resourceGroupName
"AutomationAccountName" = $automationAccountName "Location" =
$location "WorkspaceName" = $workspaceName }
```

```
[Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12
.\CreateOrUpdateAzAutoAccount.ps1 @Params ```
```

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

1. Within the Remote Desktop session to **az140-dc-vm11**, in the **Administrator: Windows PowerShell ISE** script pane, review the output of the script.

Note: The output includes a webhook URI, the Log Analytics Workspace ID and the corresponding primary key values that you need to provide when provisioning the Azure Logic App that is part of the autoscaling solution.

1. To verify the configuration of the Azure Automation account, within the Remote Desktop session to **az140-dc-vm11**, start Microsoft Edge and navigate to the Azure portal. If prompted, sign in by using the Azure AD credentials of the user account with the Owner role in the subscription you are using in this lab.
2. Within the Remote Desktop session to **az140-dc-vm11**, in the Microsoft Edge window displaying the Azure portal, search for and select **Automation accounts** and, on the **Automation accounts** blade, select the entry representing the newly provisioned Azure Automation account (with the name starting with the **az140-automation-51** prefix).
3. On the Automation Account blade, in the vertical menu on the left side, in the **Process Automation** section, select **Runbooks** and, in the list of runbooks, verify the presence of the **WVDAutoScaleRunbookARMBased** runbook.
4. On the Automation Account blade, in the vertical menu on the left side, in the **Account Settings** section, select **Run as accounts** and, in the list of accounts on the right side, next to the **Azure Run As Account**, click **+ Create**.
5. On the **Add Azure Run As Account** blade, click **Create** and verify that the new account was successfully created.

Task 3: Create an Azure Logic app

1. Within the Remote Desktop session to **az140-dc-vm11**, switch to the **Administrator: Windows PowerShell ISE** window and, from

the **Administrator: Windows PowerShell ISE** script pane, run the following to run the following to download the PowerShell script you will use to create the Azure Logic app that is part of the autoscaling solution:

```
powershell $labFilesfolder = 'C:\Allfiles\Labs\05' Set-
Location -Path $labFilesfolder $uri =
"https://raw.githubusercontent.com/Azure/RDS-
Templates/master/wvd-templates/wvd-scaling-
script/CreateOrUpdateAzLogicApp.ps1" Invoke-WebRequest -Uri
$uri -OutFile ".\CreateOrUpdateAzLogicApp.ps1"
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE**, open the **C:\Allfiles\Labs\05\CreateOrUpdateAzLogicApp.ps1** script and enclose the code between lines **134** an **138** into the multiline comment, such that they look as follows:

```
powershell <# # Get the Role Assignment of the authenticated
user $RoleAssignments = Get-AzRoleAssignment -SignInName
$AzContext.Account -ExpandPrincipalGroups if (!
($RoleAssignments | Where-Object { $_.RoleDefinitionName -in
@('Owner', 'Contributor') })) { throw 'Authenticated user
should have the Owner/Contributor permissions to the
subscription' } #>
```

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to set the values of variables that you will assign to script parameters:

```powershell $AADTenantId = (Get-AzContext).Tenant.Id
$AzSubscription = (Get-AzContext).Subscription.Id $ResourceGroup =
Get-AzResourceGroup -Name 'az140-51-RG' $WVDHostPool = Get-
AzResource -ResourceType
"Microsoft.DesktopVirtualization/hostpools" -Name 'az140-21-hp1'
$LogAnalyticsWorkspace = (Get-AzOperationalInsightsWorkspace -
ResourceGroupName $ResourceGroup.ResourceGroupName)[0]
$LogAnalyticsWorkspaceId = $LogAnalyticsWorkspace.CustomerId
$LogAnalyticsWorkspaceKeys = (Get-
AzOperationalInsightsWorkspaceSharedKey -ResourceGroupName
$ResourceGroup.ResourceGroupName -Name
$LogAnalyticsWorkspace.Name) $LogAnalyticsPrimaryKey =
$LogAnalyticsWorkspaceKeys.PrimarySharedKey $RecurrenceInterval
= 2 $BeginPeakTime = '1:00' $EndPeakTime = '1:01' $TimeDifference =
```

'0:00' $SessionThresholdPerCPU = 1 $MinimumNumberOfRDSH = 1 $MaintenanceTagName = 'CustomMaintenance' $LimitSecondsToForceLogOffUser = 5 $LogOffMessageTitle = 'Autoscaling' $LogOffMessageBody = 'Forcing logoff due to autoscaling'

$AutoAccount = (Get-AzAutomationAccount -ResourceGroupName $ResourceGroup.ResourceGroupName)[0] $AutoAccountConnection = Get-AzAutomationConnection -ResourceGroupName $AutoAccount.ResourceGroupName -AutomationAccountName $AutoAccount.AutomationAccountName

$WebhookURIAutoVar = Get-AzAutomationVariable -Name 'WebhookURIARMBased' -ResourceGroupName $AutoAccount.ResourceGroupName -AutomationAccountName $AutoAccount.AutomationAccountName ```

Note: The values of parameters are geared towards accelerating the autoscaling behavior. In your production environment, you should adjust them to match your own specific requirements.

1. Within the Remote Desktop session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** script pane, run the following to create the Azure Logic app that is part of the autoscaling solution:

```powershell $Params = @{ "AADTenantId" = $AADTenantId # Optional. If not specified, it will use the current Azure context "SubscriptionID" = $AzSubscription.Id # Optional. If not specified, it will use the current Azure context "ResourceGroupName" = $ResourceGroup.ResourceGroupName # Optional. Default: "WVDAutoScaleResourceGroup" "Location" = $ResourceGroup.Location # Optional. Default: "West US2" "UseARMAPI" = $true "HostPoolName" = $WVDHostPool.Name "HostPoolResourceGroupName" = $WVDHostPool.ResourceGroupName # Optional. Default: same as ResourceGroupName param value "LogAnalyticsWorkspaceId" = $LogAnalyticsWorkspaceId # Optional. If not specified, script will not log to the Log Analytics "LogAnalyticsPrimaryKey" = $LogAnalyticsPrimaryKey # Optional. If not specified, script will not log to the Log Analytics "ConnectionAssetName" = $AutoAccountConnection.Name # Optional. Default: "AzureRunAsConnection" "RecurrenceInterval" = $RecurrenceInterval #
```

Optional. Default: 15 "BeginPeakTime" = $BeginPeakTime # Optional. Default: "09:00" "EndPeakTime" = $EndPeakTime # Optional. Default: "17:00" "TimeDifference" = $TimeDifference # Optional. Default: "-7:00" "SessionThresholdPerCPU" = $SessionThresholdPerCPU # Optional. Default: 1 "MinimumNumberOfRDSH" = $MinimumNumberOfRDSH # Optional. Default: 1 "MaintenanceTagName" = $MaintenanceTagName # Optional. "LimitSecondsToForceLogOffUser" = $LimitSecondsToForceLogOffUser # Optional. Default: 1 "LogOffMessageTitle" = $LogOffMessageTitle # Optional. Default: "Machine is about to shut down." "LogOffMessageBody" = $LogOffMessageBody # Optional. Default: "Your session will be logged off. Please save and close everything." "WebhookURI" = $WebhookURIAutoVar.Value }

.\CreateOrUpdateAzLogicApp.ps1 @Params ```

Note: Wait for the script to complete. This might take about 2 minutes.

1. To verify the configuration of the Azure Logic app, within the Remote Desktop session to **az140-dc-vm11**, switch to the Microsoft Edge window displaying the Azure portal, search for and select **Logic Apps** and, on the **Logic apps** blade, select the entry representing the newly provisioned Azure Logic app named **az140-21-hp1_Autoscale_Scheduler**.
2. On the **az140-21-hp1_Autoscale_Scheduler** blade, in the vertical menu on the left side, in the **Development Tools** section, select **Logic app designer**.
3. On the designer pane, click the rectangle labeled **Recurrence** and note that you can use it to control frequency in which the need for autoscaling is evaluated.

Exercise 2: Verify and review autoscaling of Windows Virtual Desktop session hosts

The main tasks for this exercise are as follows:

1. Verify autoscaling of Windows Virtual Desktop session hosts
2. Use Azure Log Analytics to track Windows Virtual Desktop events

Task 1: Verify autoscaling of Windows Virtual Desktop session hosts

1. To verify the autoscaling of the Windows Virtual Desktop session hosts, within the Remote Desktop session to **az140-dc-vm11**, in the Microsoft Edge window displaying the Azure portal, search for and select **Virtual machines** and, on the **Virtual machines** blade, review the status of the three Azure VMs in the **az140-21-RG** resource group.
2. Verify that two of the three Azure VMs are either in the process of being deallocated or are already **Stopped (deallocated)**.

Note: As soon as you verify that autoscaling is working, you should disable the Azure Logic app to minimize the corresponding charges.

1. To disable the Azure Logic app, within the Remote Desktop session to **az140-dc-vm11**, in the Microsoft Edge window displaying the Azure portal, search for and select **Logic Apps** and, on the **Logic apps** blade, select the entry representing the newly provisioned Azure Logic app named **az140-21-hp1_Autoscale_Scheduler**.
2. On the **az140-21-hp1_Autoscale_Scheduler** blade, in the toolbar, click **Disable**.
3. On the **az140-21-hp1_Autoscale_Scheduler** blade, in the **Essentials** section, review the information including the number of successful runs in the last 24 hours and the **Summary** section providing the frequency of recurrence.
4. Within the Remote Desktop session to **az140-dc-vm11**, in the Microsoft Edge window displaying the Azure portal, search for and select **Automation accounts** and, on the **Automation accounts** blade, select the entry representing the newly provisioned Azure Automation account (with the name starting with the **az140-automation-51** prefix).
5. On the **Automation Account** blade, in the vertical menu on the left side, in the **Process Automation** section, select **Jobs** and review the list of jobs corresponding to individual invocations of the **WVDAutoScaleRunbookARMBased** runbook.
6. Select the most recent job and, on its blade, click **All Logs** tab header. This will display detailed listing of job execution steps.

Task 2: Use Azure Log Analytics to track Windows Virtual Desktop events

Note: To analyze autoscaling and any other Windows Virtual Desktop events, you can use Log Analytics.

1. Within the Remote Desktop session to **az140-dc-vm11**, in the Microsoft Edge window displaying the Azure portal, search for and select **Log Analytics workspaces** and, on the **Log Analytics workspaces** blade, select the entry representing the Azure Log Analytics workspace used in this lab (which name starts with the **az140-workspace-51** prefix.
2. On the Log Analytics workspace blade, in the vertical menu on the left side, in the **General** section, click **Logs** and, on the **Welcome to Log Analytics** pane, click **Get Started**.
3. On the **Queries** pane, in the **All Queries** vertical menu on the left side, select **Windows Virtual Desktop** and review the predefined queries.
4. Close the **Queries** pane. This will automatically display the **New Query 1** tab.
5. In the query window, paste the following query, click **Run** to display all events for the host pool used in this lab:

```
kql WVDTenantScale_CL | where hostpoolName_s == "az140-21-
hp1" | project TimeStampUTC = TimeGenerated, TimeStampLocal =
TimeStamp_s, HostPool = hostpoolName_s, LineNumAndMessage =
logmessage_s, AADTenantId = TenantId
```

Note: If you don't see any results, wait a few minutes and try again.

1. In the query window, paste the following query, click **Run** to display the total number of currently running session hosts and active user sessions in the target host pool:

```
kql WVDTenantScale_CL | where logmessage_s contains "Number
of running session hosts:" or logmessage_s contains "Number
of user sessions:" or logmessage_s contains "Number of user
sessions per Core:" | where hostpoolName_s == "az140-21-hp1"
| project TimeStampUTC = TimeGenerated, TimeStampLocal =
TimeStamp_s, HostPool = hostpoolName_s, LineNumAndMessage =
logmessage_s, AADTenantId = TenantId
```

1. In the query window, paste the following query, click **Run** to display the status of all session host VMs in a host pool:

```
kql WVDTenantScale_CL | where logmessage_s contains "Session
host:" | where hostpoolName_s == "az140-21-hp1" | project
TimeStampUTC = TimeGenerated, TimeStampLocal = TimeStamp_s,
```

```
HostPool = hostpoolName_s, LineNumAndMessage = logmessage_s,
AADTenantId = TenantId
```

1. In the query window, paste the following query, click **Run** to display any scaling related errors and warnings:

```
kql WVDTenantScale_CL | where logmessage_s contains "ERROR:"
or logmessage_s contains "WARN:" | project TimeStampUTC =
TimeGenerated, TimeStampLocal = TimeStamp_s, HostPool =
hostpoolName_s, LineNumAndMessage = logmessage_s, AADTenantId
= TenantId
```

Exercise 3: Stop and deallocate Azure VMs provisioned in the lab

The main tasks for this exercise are as follows:

1. Stop and deallocate Azure VMs provisioned in the lab

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

Task 1: Deallocate Azure VMs provisioned 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 in this lab:

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

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

```
powershell Get-AzVM -ResourceGroup 'az140-21-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.

Table of Contents

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. 7

{{ activity.lab.title }}{% if activity.lab.type %} - {{ activity.lab.type }}{% endif %} 1

{{ activity.demo.title }} 1

Azure portal 1