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 Virtual Desktop (AD DS)**

Estimated Time

60 minutes

Lab scenario

You need to automate deployment of Azure 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 Virtual Desktop host pools and hosts by using PowerShell
- Add hosts to the Azure 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 Virtual Desktop host pools and session hosts by using PowerShell

The main tasks for this exercise are as follows:

1. Prepare for deployment of Azure Virtual Desktop host pool by using PowerShell
2. Create a Azure Virtual Desktop host pool by using PowerShell
3. Perform a template-based deployment of an Azure VM running Windows 11 Enterprise by using PowerShell
4. Add an Azure VM running Windows 11 Enterprise as a session host to the Azure Virtual Desktop host pool by using PowerShell
5. Verify the deployment of the Azure Virtual Desktop session host

Task 1: Prepare for deployment of Azure 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 **Connect via Bastion**.
4. When prompted, provide the following credentials and select **Connect**:

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

5. Within the Bastion session to **az140-dc-vm11**, start **Windows PowerShell ISE** as administrator.
6. Within the Bastion 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**):

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

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

7. Within the Bastion 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.
8. Within the Bastion 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**.
9. 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**
Subnet address range	**10.0.3.0/24**

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

1. Within the Bastion 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:

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

2. Within the Bastion 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:

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

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

```
$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.

4. Within the Bastion 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**:

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

5. Within the Bastion 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:

```
$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 11 Enterprise by using PowerShell

1. From your lab computer, navigate to the deployed storage account. On the File Share blade, select the **az140-22-profiles** file share.

2. Select **Upload** and upload both the lab files **\\AZ-140\AllFiles\Labs\02\az140-24_azuredeployhp3.json** and **\\AZ-140\AllFiles\Labs\02\az140-24_azuredeployhp3.parameters.json** to file share.

3. Within the Bastion session to **az140-dc-vm11**, open File Explorer and navigate to the previously configured Z:, or the drive letter assigned to the connection to the File Share. Copy the uploaded deployment files to **C:\AllFiles\Labs\02**.

4. Within the Bastion session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to deploy an Azure VM running Windows 11 Enterprise (multi-session) that will serve as a Azure Virtual Desktop session host in the host pool you created in the previous task:

```
$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 5-10 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.

5. Within the Bastion 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:

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

Task 4: Add an Azure VM running Windows 11 Enterprise as a host to the Azure Virtual Desktop host pool by using PowerShell

1. Within the Bastion 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 **Connect**
3. Ensure **Connecting using** says **Private IP address | 10.0.3.4**
4. From within the **Native RDP** section, select **Download RDP File**.
5. If prompted, select **Keep**, and then click the downloaded **az140-24-p3-0.rdp** file.
6. When prompted, sign in with the following credentials:

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

7. Within the Remote Desktop session to **az140-24-p3-0**, start **Windows PowerShell ISE** as administrator.
8. 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:

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

> **Note** take care using the [T] construct to copy over the PowerShell cmdlets. In some instances, the text copied over can be incorrect, such as the $ sign showing as a 4 number character. You will need to correct these before issuing the cmdlet. Copy over to the PowerShell ISE **Script** pane, make the corrections there, and then highlight the corrected text and press **F8** (**Run Selection**).

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

```
$webClient = New-Object System.Net.WebClient
$wvdAgentInstallerURL =
'https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RWrmXv'
$wvdAgentInstallerName = 'WVD-Agent.msi'
$webClient.DownloadFile($wvdAgentInstallerURL,"$labFilesFolder/$wvdAgentInst
allerName")
$wvdBootLoaderInstallerURL =
'https://query.prod.cms.rt.microsoft.com/cms/api/am/binary/RWrxrH'
$wvdBootLoaderInstallerName = 'WVD-BootLoader.msi'
$webClient.DownloadFile($wvdBootLoaderInstallerURL,"$labFilesFolder/$wvdBoot
LoaderInstallerName")
```

10. From the **Administrator: Windows PowerShell ISE** console, run the following to install the latest version of the Az PowerShell module, when prompted to install the NuGet provider, press **Y**:

```
Install-Module -Name Az -Force
```

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

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

```
Connect-AzAccount
```

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

13. Within the Remote Desktop session to **az140-24-p3-0**, from the **Administrator: Windows PowerShell ISE** console, run the following to generate the token necessary to join this session host to the pool you provisioned earlier in this exercise:

```
$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.

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

```
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
```

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

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

16. Within the Remote Desktop session to **az140-24-p3-0**, right-click **Start**, in the right-click menu, select **Shut down or sign out** and, in the cascading menu, click **Sign out**.

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

1. Switch to the lab computer, in the web browser displaying the Azure portal, search for and select **Azure Virtual Desktop**, on the **Azure Virtual Desktop** blade, select **Host pools** and, on the **Azure 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 Bastion session to **az140-dc-vm11**, from the **Administrator: Windows PowerShell ISE** console, run the following to create a Remote App group:

```
$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/$resourceGroupName/providers/
Microsoft.DesktopVirtualization/hostPools/$hostPoolName"-Location $location
```

2. Within the Bastion 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:

```
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**.

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

```
$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 -FriendlyName $name -Filepath $filePath -IconPath
$iconPath -IconIndex 0 -CommandLineSetting 'DoNotAllow' -ShowInPortal:$true
```

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

```
$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'
```

5. 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**.

6. On the **az140-24-hp3 | Application groups** blade, in the list of application groups, select the **az140-24-hp3-Office365-RAG** entry.

7. 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:

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

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

```
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.