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

Student lab manual

Lab dependencies

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

Estimated Time

150 minutes

> **Note**: Provisioning of a Microsoft Entra DS involves about 90-minute wait time.

Lab scenario

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

Objectives

After completing this lab, you will be able to:

- Implement a Microsoft Entra DS domain
- Configure the Microsoft Entra 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**.

4. In the Azure portal, in the PowerShell session of the **Cloud Shell**, run the following to register the **Microsoft.Compute** resource provider, in case it's not registered:

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

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

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

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

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

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

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

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

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

8. Review the output of the command executed in the previous step and ensure that you have at least **20** available vCPUs in the **Standard DSv3 Family** 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 **Azure Pass – Sponsorship | Usage + quotas** blade, select the following drop down arrows from the top search bar:

Setting	Value
Search	**Standard DSv3**
All locations	**Clear all**, and then check *your location*
Resource provider	**Microsoft.Compute**

4. In the returned **Standard DSv3 Family vCPUs** item, select the pencil icon, **Edit**.

5. In the **Quota Details** blade, in the **New limit** column text box, type **30**, and then select **Save and continue**.

6. Allow the quota request to complete. After a few moments, the **Quota Details** blade will specify the request has been approved and Quota increased. Close the **Quota Details** blade.

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

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

The main tasks for this exercise are as follows:

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

Task 1: Create and configure a Microsoft Entra user account for administration of Microsoft Entra 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 Microsoft Entra tenant associated with the Azure subscription.

2. In the web browser displaying the Azure portal, navigate to the **Overview** blade of the Microsoft Entra tenant and, in the vertical menu on the left side, in the **Manage** section, click **Properties**.

3. On the **Properties** blade of your Microsoft Entra tenant, at the very bottom of the blade, select the **Manage Security defaults** link.

4. On the **Enable Security defaults** blade, if needed, select **No**, select the **My organization is using Conditional Access** checkbox, and select **Save**.

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

6. 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**.

7. From the Cloud Shell pane, run the following to sign in to your Microsoft Entra tenant:

   ```
   Connect-AzureAD
   ```

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

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

9. From the Cloud Shell pane, run the following to create Microsoft Entra users that will be granted elevated privileges (replace the `<password>` placeholder with a random, complex password):

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

   ```
   $passwordProfile = New-Object -TypeName
   Microsoft.Open.AzureAD.Model.PasswordProfile
   $passwordProfile.Password = '<password>'
   $passwordProfile.ForceChangePasswordNextLogin = $false
   New-AzureADUser -AccountEnabled $true -DisplayName 'aadadmin1' -
   PasswordProfile $passwordProfile -MailNickName 'aadadmin1' -
   UserPrincipalName "aadadmin1@$aadDomainName"
   New-AzureADUser -AccountEnabled $true -DisplayName 'wvdaadmin1' -
   PasswordProfile $passwordProfile -MailNickName 'wvdaadmin1' -
   UserPrincipalName "wvdaadmin1@$aadDomainName"
   ```

10. From the Cloud Shell pane, run the following to assign the Global Administrator role to the first of the newly created Microsoft Entra users:

```
$aadUser = Get-AzureADUser -ObjectId "aadadmin1@$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.

11. From the Cloud Shell pane, run the following to identify the user principal name of the newly created Microsoft Entra user:

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

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

12. Close the Cloud Shell pane.

13. Within the Azure portal, search for and select **Subscriptions** and, from the **Subscriptions** blade, select the Azure subscription you are using in this lab.

14. On the blade displaying properties of your Azure subscription, select **Access control (IAM)**, select **Add**, and then select **Add role assignment**.

15. On the **Add role assignment** blade, select **Owner** and then click **Next**

16. Click the **+Select members** hyperlink.

17. In the **Select Members** blade, select the **aadadmin1** item, and then click the **Select** button, and then click **Next**.

18. In the **Review + assign** blade, select the **Review + Assign** button.

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

Task 2: Deploy a Microsoft Entra DS instance by using the Azure portal

1. From your lab computer, in the Azure portal, search for and select **Microsoft Entra Domain Services** and, from the **Microsoft Entra Domain Services** blade, select **+ Create**. This will open the **Create Microsoft Entra Domain Services** blade.

2. On the **Basics** tab of the **Create Microsoft Entra 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

Setting	Value
Resource group	Select Create new **az140-11a-RG**
DNS domain name	**adatum.com**
Region	the name of the region where you want to host your AVD deployment
SKU	**Standard**

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

3. On the **Networking** tab of the **Create Microsoft Entra Domain Services** blade, next to the **Virtual network** drop-down list, select **Create new**.

4. On the **Create virtual network** blade, assign the following settings and select **OK**:

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

5. Back on the **Networking** tab of the **Create virtual network** blade, select **Next** (leave others with their existing values).

6. On the **Administration** tab of the **Create Microsoft Entra Domain Services** blade, accept the default settings and select **Next**.

7. On the **Synchronization** tab of the **Create Microsoft Entra Domain Services** blade, ensure that **All** is selected and then select **Next**.

8. On the **Security Settings** tab of the **Create Microsoft Entra Domain Services** blade, accept the default settings and select **Next**.

9. On the **Tags** tab of the **Create Microsoft Entra Domain Services** blade, accept the default settings and select Next

10. On the **Review + create** tab of the **Create Microsoft Entra Domain Services** blade, select **Create**.

11. Review the notification regarding settings that you will not be able to change following creation of the Microsoft Entra DS domain and select **OK**.

> **Note**: The settings that you will not be able to change following provisioning of an Microsoft Entra 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 Microsoft Entra DS deployment

1. From your lab computer, in the Azure portal, search for and select **Microsoft Entra Domain Services** and, from the **Microsoft Entra Domain Services** blade, select the **adatum.com** entry to navigate to the newly provisioned Microsoft Entra DS instance.

2. On the **adatum.com** blade of the Microsoft Entra DS instance, click the warning starting with **Configuration issues for your managed domain were detected**.

3. On the **adatum.com | Configuration diagnostics** 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 Microsoft Entra DS instance and, in the **Required configuration steps** section, review the information regarding the Microsoft Entra DS password hash synchronization.

   > **Note**: Any existing cloud-only users that need to be able to access Microsoft Entra 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.

7. From your lab computer, in the Azure portal, open a **PowerShell** session in the **Cloud Shell** pane.

8. From the PowerShell session in the Cloud Shell pane, run the following to identify the objectID attribute of the Microsoft Entra **aadadmin1** user account:

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

9. 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 (replace the `<password>` placeholder with a random, complex password):

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

   ```
   $password = ConvertTo-SecureString '<password>' -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.

10. Repeat the previous two steps to reset the password for the **wvdaadmin1** user account.

Exercise 2: Configure the Microsoft Entra 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. Deploy Azure Bastion
3. Review the default configuration of the Microsoft Entra DS domain
4. Create AD DS users and groups that will be synchronized to Microsoft Entra 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:

   ```
   $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
   ```

2. From your lab computer, locate the **\\AZ-140\AllFiles\Labs\01\az140-11_azuredeploycl11a.parameters.json** parameters file and open the file using Visual Studio Code.

3. At line 21, locate the value of the domainPassword parameter. Update the existing password in the parameter file to use the password that you set earlier in this lab for the **aadadmin1** user account, and then **Save** the file.

4. 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.

5. 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 Azure Virtual Desktop client and join it to the Microsoft Entra DS domain:

   ```
   $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: Deploy Azure Bastion

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

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

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

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

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

```
$resourceGroupName = 'az140-11a-RG'
$vnet = Get-AzVirtualNetwork -ResourceGroupName $resourceGroupName -Name
'az140-aadds-vnet11a'
$subnetConfig = Add-AzVirtualNetworkSubnetConfig `
  -Name 'AzureBastionSubnet' `
  -AddressPrefix 10.10.254.0/24 `
  -VirtualNetwork $vnet
$vnet | Set-AzVirtualNetwork
```

4. Close the Cloud Shell pane.

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

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

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

Setting	Value
Region	the same Azure region to which you deployed the resources in the previous task of this exercise
Tier	**Basic**
Virtual network	**az140-aadds-vnet11a**
Subnet	**AzureBastionSubnet (10.10.254.0/24)**
Public IP address	**Create new**
Public IP name	**az140-aadds-vnet11a-ip**

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

   > **Note**: Wait for the deployment to complete before you proceed to the next task of this exercise. The deployment might take about 5 minutes.

Task 3: Review the default configuration of the Microsoft Entra DS domain

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

> **Note**: You have the option of populating this group with existing Microsoft Entra user accounts when you provision a Microsoft Entra DS instance.

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

   ```
   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
   ```

2. Close the Cloud Shell pane.

3. 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.

4. On the **az140-cl-vm11a** blade, select **Connect**, in the drop-down menu, select **Bastion**, on the **Bastion** tab of the **az140-cl-vm11a**, provde the following credentials and select **Connect**:

5. When prompted, sign in as the **aadadmin1** user using its principal name you identified earlier in this lab and the password you set for this user account when creating it earlier in the lab.

6. Within the Bastion 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:

   ```
   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.

7. Within the Bastion 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**.

8. 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 Microsoft Entra tenant associated with the Azure subscription hosting the deployment of Microsoft Entra DS instance. The **AADDC Users** organizational unit also includes the **AAD DC Administrators** group synchronized from the same Microsoft Entra tenant, along with its group membership. This membership cannot be modified directly within the Microsoft Entra DS domain, but instead, you have to manage it within the Microsoft Entra DS tenant. Any changes are automatically synchronized with the replica of the group hosted in the Microsoft Entra DS domain.

   Hint: If the **Active Directory Users and Computers** does not list out any domain related content then Right-Click on the **Active Directory Users and Computers** and select **Change Domain** and choose the domain **Adatum**.

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

9. 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 Microsoft Entra DNS domain name and is not modifiable.

10. 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 4: Create AD DS users and groups that will be synchronized to Microsoft Entra DS

1. Within the Bastion 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 the password you set earlier in this lab as its password.

2. In the Azure portal, open 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**.

4. From the PowerShell session in the Cloud Shell pane, run the following to sign in to authenticate to your Microsoft Entra tenant:

   ```
   Connect-AzureAD
   ```

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

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

6. From the PowerShell session in the Cloud Shell pane, run the following to create the Microsoft Entra user accounts you will use in the upcoming labs (replace the `<password>` placeholder with a random, complex password):

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

   ```
   $passwordProfile = New-Object -TypeName
   Microsoft.Open.AzureAD.Model.PasswordProfile
   $passwordProfile.Password = '<password>'
   $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"
   }
   ```

7. From the PowerShell session in the Cloud Shell pane, run the following to create a Microsoft Entra group named **az140-wvd-aadmins** and add to it the **aadadmin1** and **wvdaadmin1** user accounts:

```
$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
$userObjectId = (Get-AzureADUser -Filter "MailNickName eq 'wvdaadmin1'").ObjectId
Add-AzureADGroupMember -ObjectId $az140wvdaadmins.ObjectId -RefObjectId $userObjectId
```

8. From the Cloud Shell pane, repeat the previous step to create Microsoft Entra groups for users that you will use in the upcoming labs and add to them previously created Microsoft Entra user accounts:

> **Note**: Note: Because of the limited size of the Clipboard on the virtual machine, not all the listed cmdlets will copy over correctly. Open up Notepad on the virtual machine and copy all of the cmdlets to it by using the Type Text, Type Clipboard Text construct that is part of the Lightning Bolt control. Once you have ensured that all of the cmdlets are in Notepad, cut and paste them in blocks into the Cloud Shell and run them.

```
$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
```

9. Close the Cloud Shell pane.

10. Within the Bastion 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 Microsoft Entra 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.

11. Navigate back to the Microsoft Entra 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.

12. Within the Bastion 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.