Purpose

This document provides instructor demonstrations for the AZ-104 Azure Administrator course.

Things to think about

- Most lessons in this course have a demonstration. Take the time to work through each one and decide which to use. Some of the demonstrations are simple show and tell walk-throughs of the Azure portal; others require scripting skills.
- Consider having the students follow along as you do the demonstration or have one of the students' "drive" and coach them through the steps.
- Consider doing the demonstration first and then using the slides to answer questions and ensure everything was covered.
- Consider the overlap with the formal labs and make the best use of your time.
- These demonstrations are provided for an instructor with Azure administration experience. The steps are at a higher level than the course labs that students will perform.
- These demonstrations provide a minimal set of features to show your students. As time permits feel free to add, show, and discuss more things.
- Use the Word navigation links to easily locate the module, lesson, or demonstration.

Module 01, Administer Identity

Lesson 01, Configure Azure Active Directory

This lesson does not have a formal demonstration.

Lesson 02, Configure User and Group Accounts

In this demonstration, we will explore Azure Active Directory.

Note: Depending on your subscription not all areas of the Azure Active Directory blade will be available.

Review license and domain information

1. Access the Azure portal and navigate to the **Azure Active Directory** blade.
2. On the Overview blade, review the **Tenant information** including license and primary domain.

Explore user accounts

1. Select the **Users** blade.
2. Explain the choices for **New user** and **New guest user**.
3. Select **New user** and discuss the differences between **Create user** and **Invite user**.
4. Create a **New user** reviewing the **Identity**, **Groups and roles**, **Settings**, and **Job Info** parameters.
5. After the user is created, review **Reset password**, **Delete user**, and **Sign-ins**.

Explore group accounts

1. Return to the **Azure Active Directory** page and select the **Groups** blade.
2. Create a **New group** or select an existing group to review.
3. Review information about a group including **Membership type** and **Type**.

Optional - Explore PowerShell for group management

1. Create a new group called Developers.

   New-AzADGroup -DisplayName Developers -MailNickname Developers

2. Retrieve the Developers group ObjectId.

   Get-AzADGroup

3. Retrieve the user ObjectId for the member to add.

   Get-AzADUser

4. Add the user to the group. Replace groupObjectId and userObjectId.

   Add-AzADGroupMember -MemberUserPrincipalName ""myemail@domain.com"" -TargetGroupDisplayName ""MyGroupDisplayName""

5. Verify the members of the group. Replace groupObjectId.

   Get-AzADGroupMember -GroupDisplayName "MyGroupDisplayName"

Module 02, Administer Governance and Compliance

Lesson 01, Configure Subscriptions

This lesson does not have a formal demonstration.

Lesson 02, Configure Azure Policy

In this demonstration, we will work with Azure policies.

Assign a policy

1. Access the Azure portal.
2. Search for and select **Policy**.
3. Select **Assignments** on the left side of the Azure Policy page.
4. Select **Assign Policy** from the top of the Policy - Assignments page.
5. Notice the **Scope** which determines what resources or grouping of resources the policy assignment gets enforced on.
6. Select the **Policy definition ellipsis** to open the list of available definitions. Take some time to review the built-in policy definitions.
7. Search for and select **Allowed locations**. This policy enables you to restrict the locations your organization can specify when deploying resources.
8. Move the **Parameters** tab and using the drop-down select one or more allowed locations.
9. Click **Review + create** and then **Create** to create the policy.

Create and assign an initiative definition

1. Return to the Azure Policy page and select **Definitions** under Authoring.
2. Select **Initiative Definition** at the top of the page.
3. Provide a **Name** and **Description**.
4. **Create new** Category.
5. From the right panel **Add** the **Allowed locations** policy.
6. Add one additional policy of your choosing.
7. **Save** your changes and then **Assign** your initiative definition to your subscription.

Check for compliance

1. Return to the Azure Policy service page.

2. Select **Compliance**.
3. Review the status of your policy and your definition.

Check for remediation tasks

1. Return to the Azure Policy service page.
2. Select **Remediation**.
3. Review any remediation tasks that are listed.

Remove your policy and initiative

1. Return to the Azure Policy service page.
2. Select **Assignments**.
3. Select your **Allowed locations** policy.
4. Click **Delete assignment**.
5. Return to the Azure Policy service page.
6. Select **Initiatives**.
7. Select your new initiative.
8. Click **Delete initiative**.

Lesson 03, Configure Role-Based Access Control

In this demonstration, we will learn about role assignments.

Locate Access Control blade

1. Access the Azure portal and select a resource group. Make a note of what resource group you use.
2. Select the **Access Control (IAM)** blade.
3. This blade will be available for many different resources so you can control access.

Review role permissions

1. Select the **Roles** tab (top).
2. Review the large number of built-in roles that are available.
3. Double-click a role, and then select **Permissions** (top).
4. Continue drilling into the role until you can view the **Read, Write, and Delete** actions for that role.

5. Return to the **Access Control (IAM)** blade.

Add a role assignment

1. Create a user.
2. Select **Add role assignment**.
    - **Role**: *Owner*
    - **Select**: *Managers*
    - **Save** your changes.
3. Select **Check access**.
4. Select the user.
5. Notice the user is part of the Managers group and is an Owner.
6. Notice that you can **Deny assignments**.

Optional - Explore PowerShell commands

1. Open the Azure Cloud Shell.
2. Select the PowerShell drop-down.
3. List role definitions.

   Get-AzRoleDefinition | FT Name, Description

4. List the actions of a role.

   Get-AzRoleDefinition owner | FL Actions, NotActions

5. List role assignments.

   Get-AzRoleAssignment -ResourceGroupName <resource group name>

Module 03, Administer Azure Resources

Lesson 01, Configure Azure Resources with Tools

Demonstration – Azure Portal

In this demonstration, you will explore the Azure portal.

Help and Keyboard Shortcuts

1. Access the Azure Portal.
2. Click the **?** Help and Support icon on the top banner.
3. Select **Launch Guided Tour** and click **Start Tour**. Review the help information.
4. Select **Keyboard Shortcuts** and read through the available shortcuts. Do any seem of interest?
5. Close the Help page, hold G, and press **D** to go your Dashboard.

Customizing your experience

1. Examine the icons next to the Dashboard drop-down. For example, New Dashboard, Upload, Download, Edit, and Clone.
2. Click **New Dashboard**.
3. Practice adding, pinning, moving, resizing, and deleting tiles.
4. Click **Done customizing** to save your edits.
5. Select the **Settings** icon on the top banner. Experiment with different color themes. **Apply** your changes.
6. Practice reordering your **Favorites** list. Do this by holding and dragging list items up or down.
7. Notice how clicking a Favorite takes you to that page.
8. Click the **Cost Management and Billing** blade. **Pin** your Subscription information to your Dashboard.
9. Visit the Dashboard and make any arrangement changes you like.
10. Use the ***search*** textbox at the top of the page.
11. Type ***resource*** and notice context matches are provided.
12. Select **Resource groups** and then click **+ Add**.
13. **Review and create** your first resource group.

Demonstration – Cloud Shell

In this demonstration, we will experiment with the Cloud Shell.

Configure the Cloud Shell

1. Access the **Azure Portal**.
2. Click the **Cloud Shell** icon on the top banner.
3. On the Welcome to the Shell page, notice your selections for Bash or PowerShell. Select **PowerShell**.
4. The Azure Cloud Shell requires an Azure file share to persist files. As you have time, click Learn more to obtain information about the Cloud Shell storage and the associated pricing.
5. Select your **Subscription** and click **Create Storage**.

Experiment with Azure PowerShell

1. Wait for your storage to be created and your account to be initialized.
2. At the PowerShell prompt, type **Get-AzSubscription** to view your subscriptions.
3. Type **Get-AzResourceGroup** to view resource group information.

Experiment with the Bash shell

1. Use the drop-down to switch to the **Bash** shell and confirm your choice.
2. At the Bash shell prompt, type **az account list** to view your subscriptions. Also, try tab completion.
3. Type **az resource list** to view resource information.

Experiment with the Cloud Editor

1. To use the Cloud Editor, type **code .**. You can also select the curly braces icon.
2. Select a file from the left navigation pane. For example, **.profile**.
3. Notice on the editor top banner, selections for Settings (Text Size and Font) and Upload/Download files.
4. Notice on the ellipses (**...**) on the far right for Save, Close Editor, and Open File.
5. Experiment as you have time, then **close** the Cloud Editor.
6. Close the Cloud Shell.

Demonstration – Working with PowerShell

In this demonstration, we will install Azure Az PowerShell module. The Az module is available from a global repository called the *PowerShell Gallery*. You can install the module onto your local machine through the **Install-Module** command. You need an elevated PowerShell shell prompt to install modules from the PowerShell Gallery.

Note: If at any time you receive errors about *running scripts is disabled* be sure to set the execution policy:

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine

Note: You may need to run this code in PowerShell to enable TLSv2:

[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12

Install the Az module

1. Open the **Start** menu, and type **Windows PowerShell**.
2. Right-click the **Windows PowerShell** icon, and select **Run as administrator**.
3. In the **User Account Control** dialog, select **Yes**.
4. Type the following command, and then press Enter. This command installs the module for all users by default. (It's controlled by the scope parameter.) AllowClobber overwrites the previous PowerShell module.

   Install-Module -Name Az -AllowClobber

Install NuGet (if needed)

1. Depending on the NuGet version you have installed you might get a prompt to download and install the latest version.
2. If prompted, install and import the NuGet provider.

Trust the repository

1. By default, the PowerShell Gallery isn't configured as a trusted repository for PowerShellGet. The first time you use the PowerShell Gallery, you will be prompted.

   You are installing the modules from an untrusted repository. If you trust this repository, change its InstallationPolicy value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from PSGallery'?

2. As prompted, install the modules.

Connect to Azure and view your subscription information

1. Connect to Azure.

Connect-AzAccount

2. When prompted provide your credentials.
3. Verify your subscription information.

   Get-AzSubscription

Create resources

1. Create a new resource group. Provide a different location if you like. The *name* must be unique within your subscription. The *location* determines where the metadata for your resource group will be stored. You use strings like "West US", "North Europe", or "West India" to specify the location; alternatively, you can use single word equivalents, such as westus, northeurope, or westindia. The core syntax is:

   New-AzResourceGroup -name <name> -location <location>

2. Verify your resource group.

   Get-AzResourceGroup

3. Remove your resource group. When prompted, confirm.

   Remove-AzResourceGroup -Name Test

Demonstration – Working with the CLI

In this demonstration, we will install and use the CLI to create resources.

Install the CLI on Windows

You install Azure CLI on the Windows operating system using the MSI installer:

1. Go to https://aka.ms/installazurecliwindows , and in the browser security dialog box, click **Run**.
2. In the installer, accept the license terms, and then click **Install**.
3. In the **User Account Control** dialog, select **Yes**.

Verify Azure CLI installation

1. You run Azure CLI by opening a Bash shell for Linux or macOS, or from the command prompt or PowerShell for Windows.
2. Start Azure CLI and verify your installation by running the version check:

   az --version

Note: Running Azure CLI from PowerShell has some advantages over running Azure CLI from the Windows command prompt. PowerShell provides more tab completion features than the command prompt.

Login to Azure

1. Because you're working with a local Azure CLI installation, you'll need to authenticate before you can execute Azure commands. You do this by using the Azure CLI **login** command:

   az login

2. Azure CLI will typically launch your default browser to open the Azure sign-in page. If this doesn't work, follow the command-line instructions and enter an authorization code at https://aka.ms/devicelogin .

3. After a successful sign in, you'll be connected to your Azure subscription.

Create a resource group

1. You'll often need to create a new resource group before you create a new Azure service, so we'll use resource groups as an example to show how to create Azure resources from the CLI.

2. Azure CLI **group create** command creates a resource group. You must specify a name and location. The *name* must be unique within your subscription.
   The *location* determines where the metadata for your resource group will be stored. You use strings like "West US", "North Europe", or "West India" to specify the location; alternatively, you can use single word equivalents, such as westus, northeurope, or westindia. The core syntax is:

   az group create --name <name> --location <location>

Verify the resource group

1. For many Azure resources, Azure CLI provides a **list** subcommand to view resource details. For example, the Azure CLI **group list** command lists your Azure resource groups. This is useful to verify whether resource group creation was successful:

   az group list

2. To get a more concise view, you can format the output as a simple table:

   az group list --output table

3. If you have several items in the group list, you can filter the return values by adding a **query** option. Try this command:

az group list --query "[?name == '<rg name>']"

Lesson 02, Use Azure Resource Manager

In this demonstration, we will work with the Azure Resource Manager.

Note: Only the Owner and User Access Administrator roles can manage the locks on the resources.

Manage resource groups in the portal

1. Access the Azure portal.
2. Create a resource group. Remember the name of this resource group.
3. In the **Settings** blade for the resource group, select **Locks**.
4. To add a lock, select **Add**. If you want to create a lock at a parent level, select the parent. The currently selected resource inherits the lock from the parent. For example, you could lock the resource group to apply a lock to all its resources.
5. Give the lock a **name** and **lock type**. Optionally, you can add notes that describe the lock.
6. To delete the lock, select the ellipsis and **Delete** from the available options.

Optional - Manage resource groups with PowerShell

1. Access the Cloud Shell.
2. Create the resource lock and confirm your action.

   New-AzResourceLock -LockName <lockName> -LockLevel CanNotDelete -ResourceGroupName <resourceGroupName>

3. View resource lock information. Notice the LockId that will be used in the next step to delete the lock.

   Get-AzResourceLock

4. Delete the resource lock and confirm your action.

   Remove-AzResourceLock -LockName <Name> -ResourceGroupName <Resource Group>

5. Verify the resource lock has been removed.

   Get-AzResourceLock

Lesson 03, Configure Resources with ARM Templates

Demonstration – QuickStart Templates

In this demonstration, we will explore QuickStart templates.

Explore the gallery

1. Start by browsing to the Azure Quickstart Templates gallery . In the gallery you will find several popular and recently updated templates. These templates work with both Azure resources and popular software packages.
2. Browse through the many different types of templates that are available.
3. Are there are any templates that are of interest to you?

Explore a template

1. Let's say you come across the Deploy a simple Windows VM template.

   Note: The **Deploy to Azure** button enables you to deploy the template directly through the Azure portal if you wish.

   Note: Scroll-down to the Use the template **PowerShell** code. You will need the **TemplateURI** in the next demo. **Copy the value**. For example,

   https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-vm-simple-windows/azuredeploy.json

2. Click **Browse on GitHub** to navigate to the template's source code on GitHub.
3. Notice from this page you can also **Deploy to Azure**. Take a minute to view the Readme file. This helps to determine if the template is for you.
4. Click **Visualize** to navigate to the **Azure Resource Manager Visualizer**.
5. Notice the resources that make up the deployment, including a VM, a storage account, and network resources.
6. Use your mouse to arrange the resources. You can also use your mouse's scroll wheel to zoom in an out.
7. Click on the VM resource labeled **SimpleWinVM**.
8. Review the source code that defines the VM resource.
   - The resource's type is **Microsoft.Compute/virtualMachines**.
   - Its location, or Azure region, comes from the template parameter named **location**.
   - The VM's size is **Standard_A2**.

- The computer name is read from a template variable, and the username and password for the VM are read from template parameters.

9. Return to the QuickStart page that shows the files in the template. Copy the link to the azuredeploy.json file.

   Note: You will need the template link in the next demonstration.

Demonstration – Run Templates with PowerShell

In this demonstration, we will create new Azure resources using PowerShell and Resource Manager templates.

Connect to your subscription

1. If you are working with a local install of the PowerShell, you'll need to authenticate before you can execute Azure commands. To do this, open the PowerShell ISE, or a PowerShell console as administrator, and run the following command:

   Connect-AzAccount

2. After successfully signing in, your account and subscription details should display in the PowerShell console window. You must now select either a subscription or context, in which you will deploy your resources. If only one subscription is present it will set the context to that subscription by default. Otherwise you can specify the subscription to deploy resources into by running the following commands in sequence:

   Get-AzContext

   Set-AzContext -subscription < your subscription ID >

Create the resource group

1. You'll often need to create a new resource group before you create a new Azure service or resource. We'll use resource groups as an example to show how to create Azure resources from Azure PowerShell.

2. The Azure PowerShell **New-AzResourceGroup** command creates a resource group. You must specify a name and location. The name must be unique within your subscription, and the location determines where the metadata for your resource group will be stored. You use strings such as West US, North Europe, or West India to specify the location. Alternatively, you can use single word equivalents, such as westus, northeurope, or westindia.

3. Create the resource group into which we will deploy our resources using the following commands.

New-AzResourceGroup -Name < resource group name > -Location < your nearest datacenter >

Deploy the template into the resource group

1. Deploy the template with this command.

   $templateUri = <location of the template from the previous demonstration>

   New-AzResourceGroupDeployment -Name rg9deployment1 -ResourceGroupName rg9 -TemplateUri $templateUri

2. You will be prompted to enter values for:

   - **Adminusername**. For example, azureuser.
   - **Password**. Any compliant password will work, for example Passw0rd0134.
   - **DnsLabelprefix.** This is any unique DNS name, such as your initials and random numbers.

3. To make scripts free of manual input, you can create a .ps1 file, and then enter all the commands and inputs. You could use parameter values in the script to define the *username*, *password* and *dnslabelprefix* values, and then run the PowerShell file without input. Use the file build.ps1 as an example of how you can do this.

Note: In the previous example, we called a publicly available template on GitHub. You could also call a local template or a secure storage location, and you could define the template filename and location as a variable for use in the script. You can also specify the mode of deployment, including incremental or complete.

Verify the template deployed

1. Once you have successfully deployed the template, you need to verify the deployment. To do this, run the following commands:

   Get-AzVM

2. Notice the VM name, then run the following command to obtain additional VM details:

   Get-AzVM -Name < your VM name i.e. SimpleWinVM > -resourcegroupname < your resource group name >

3. You can also list the VMs in your subscription with the **Get-AzVM -Status** command. This can also specify a VM with the **-Name** property. In the following example, we assign it to a PowerShell variable:

   $vm = Get-AzVM -Name < your VM name i.e. SimpleWinVM > -ResourceGroupName < your resource group name >

4. The interesting thing is that this is an object you can interact with. For example, you can take that object, make changes, and then push changes back to Azure with the **Update-AzVM** command:

   $ResourceGroupName = "ExerciseResources"

   $vm = Get-AzVM -Name MyVM -ResourceGroupName $ResourceGroupName

   $vm.HardwareProfile.vmSize = "Standard_A3"

   Update-AzVM -ResourceGroupName $ResourceGroupName -VM $vm

Note: Depending on your datacenter location, you could receive an error related to the VM size not being available in your region. You can modify the vmSize value to one that is available in your region.

Note: PowerShell's interactive mode is appropriate for one-off tasks. In our example, we'll likely use the same resource group for the lifetime of the project, which means that creating it interactively is reasonable. Interactive mode is often quicker and easier for this task than writing a script and then executing it only once.

Module 04, Administer Virtual Networking

Lesson 01, Configure Virtual Networks

In this demonstration, you will create virtual networks.

Note: You can use the suggested values for the settings, or your own custom values if you prefer.

Create a virtual network in the portal

1. Sign in the to the Azure portal and search for **Virtual Networks**.
2. On the Virtual Networks page, click **Add**.
    - **Name**: *myVNet1*.
    - **Address**:*10.1.0.0/16*.
    - **Subscription**: Select your subscription.
    - **Resource group**: Select new or choose an existing resource group
    - **Location** - Select your location
    - **Subnet** - Enter *mySubnet1*.
    - **Subnet - Address range**: *10.1.0.0/24*
3. Leave the rest of the default settings and select **Create**.
4. Verify your virtual network was created.

Optional - Create a virtual network using PowerShell

1. Create a virtual network. Use values as appropriate.

   $myVNet2 = New-AzVirtualNetwork -ResourceGroupName myResourceGroup -Location EastUS -Name myVNet2 -AddressPrefix 10.0.0.0/16

2. Verify your new virtual network information.

   Get-AzVirtualNetwork -Name myVNet2

3. Create a subnet. Use values as appropriate.

   $mySubnet2 = Add-AzVirtualNetworkSubnetConfig -Name mySubnet2 -AddressPrefix 10.0.0.0/24 -VirtualNetwork $myVNet2

4. Verify your new subnet information.

   Get-AzVirtualNetworkSubnetConfig -Name mySubnet2 -VirtualNetwork $myVNet2

5. Associate the subnet to the virtual network.

$mySubnet2 | Set-AzVirtualNetwork

6. Return to the portal and verify your new virtual network with subnet was created.

Lesson 02, Configure Network Security Groups

In this demonstration, you will explore NSGs and service endpoints.

Access the NSGs blade

1. Access the Azure Portal.
2. Search for and access the **Network Security Groups** blade.
3. If you have virtual machines, you may already have NSGs. Notice the ability to filter the list.

Add a new NSG

1. **+ Add** a network security group.
    - **Name**: *select a unique name*
    - **Subscription**: *select your subscription*
    - **Resource Group**: *create new or select an existing resource group*
    - **Location**: *your choice*
    - Click **Create**
2. Wait for the new NSG to deploy.

Explore inbound and outbound rules

1. Select your new NSG.
2. Notice the NSG can be associated with subnets and network interfaces (summary information above the rules).
3. Notice the three inbound and three oubound NSG rules.
4. Under **Settings** select **Inbound security rules**.
5. Notice you can use **Default rules** to hide the default rules.
6. **+ Add** a new inbound security rule.
7. Click **Basic** to change to the Advanced mode.
8. Use the **Service** drop-down to review the predefined services that are available.

9. When you make a service selection (like HTTPS) the port range (like 443) is automatically populated. This makes it easy to configure the rule.
10. Use the Information icon next to the Priority label to learn how to configure the priority.
11. Exit the rule without making any changes.
12. As you have time, review adding an outbound security rule.

Lesson 03, No demonstration

Lesson 04, Configure Azure DNS

In this demonstration, you will explore Azure DNS.

Note: There is a DNS lab for the student.

Create a DNS zone

1. Access the Azure Portal.
2. Search for the **DNS zones** service.
3. On the **Create DNS zone** blade enter the following values, and **Create** the new DNS zone.
    - **Name**: *contoso.internal.com*
    - **Subscription**: *<your subscription>*
    - **Resource group**: Select or create a resource group
    - **Location**: Select your Location
4. Wait for the DNS zone to be created.
5. You may need to **Refresh** the page.

Add a DNS record set

1. Select **+Record Set**.
2. Use the **Type** drop-down to view the different types of records.
3. Notice how the required information changes as you change record types.
4. Change the **Type** to **A** and enter these values.
    - **Name**: *ARecord*
    - **IP Address**: *1.2.3.4*
5. Notice you can add other records.

6. Click **OK** to save your record.
7. **Refresh** the page to observe the new record set.
8. You will need the resource group name.

Optional - Use PowerShell to view DNS information

1. Open the Cloud Shell.
2. Get information about your DNS zones. Notice the name servers and number of record sets.

   Get-AzDnsZone -Name "contoso.internal.com" -ResourceGroupName <resourcegroupname>

3. Get information about your DNS record set.

   Get-AzDnsRecordSet -ResourceGroupName <resourcegroupname> -ZoneName contoso.internal.com

View your name servers

1. Access the Azure Portal and your DNS zone.
2. Review the Name Server information. There should be four name servers.
3. Open the Cloud Shell.
4. Use PowerShell to confirm your NS records.

   # Retrieve the zone information

   $zone = Get-AzDnsZone –Name contoso.internal.com –ResourceGroupName <resourcegroupname>

   # Retrieve the name server records

   Get-AzDnsRecordSet –Name “@” –RecordType NS –Zone $zone

Test the resolution

1. Continue in the Cloud Shell.
2. Use a Name Server in your zone to review records.

   nslookup arecord.contoso.internal.com <name server for the zone>

3. Nslookup should provide the IP address for the record.

Explore DNS metrics

1. Return to the Azure portal.
2. Select a DNS zone, and then select **Metrics**.
3. Use the **Metrics** drop-down to view the different metrics that are available.
4. Select **Query Volume**. If you have been using nslookup, there should be queries.
5. Use the **Line Chart** drop-down to observe other chart types, like Area Chart, Bar Chart, and Scatter Chart.

Note: For more information, Nslookup

Module 05, Administer Intersite Connectivity

Lesson 01, Configure VNet Peering

Note: For this demonstration you will need two virtual networks.

Configure VNet peering on the first virtual network

1. In the **Azure portal**, select the first virtual network.
2. Under **Settings**, select **Peerings**.
3. Select **+ Add**.
    - Provide a **Peering link name** for **This** virtual network peering. For example, VNet1toVNet2.
    - Provide a **Peering link name** for the **Remote** virtual network peering. For example, VNet2toVNet1.
    - In the **Virtual network** drop-down, select the **Remote virtual network** you would like to peer with ensuring you also select the correct **Subscription**.
    - Use the informational icons to review the **Traffic to remote virtual network Traffic forwarded from remote virtual network**, and **Virtual network gateway or Route Server** settings. If you do not have a VPN Gateway, those settings will be greyed out.
    - Click **Add** to save your settings.
4. On the **Peerings** page, discuss the **Peering Status**.

Confirm VNet peering on the second virtual network

1. In the **Azure portal**, select the second virtual network
2. Under **Settings**, select **Peerings**.
3. Notice that a peering has automatically been created. The name is what you provided when the first virtual network peering was configured.
4. Notice that the **Peering Status** is **Connected**.
5. Discuss how the settings could be changed.
6. **Cancel** your changes.

Lesson 02, Configure VPN Gateway

In this demonstration, we will explore virtual network gateways.

Note: This demonstration works best with two virtual networks with subnets.

Explore the Gateway subnet blade

1. For one of your virtual networks, select the **Subnets** blade.
2. Select **+ Gateway subnet**. Notice the name of the subnet cannot be changed. Notice the **address range** of the gateway subnet. The address must be contained by the address space of the virtual network.
3. Remember each virtual network needs a gateway subnet.
4. Close the Add gateway subnet page. You do not need to save your changes.

Explore the Connected Devices blade

1. For the virtual network, select the **Connected Devices** blade.
2. After a gateway subnet is deployed it will appear on the list of connected devices.

Explore adding a virtual network gateway

1. Search for **Virtual network gateways**.
2. Click **+ Add**.
3. Review each setting for the virtual network gateway.
4. Use the Information icons to learn more about the settings.
5. Notice the **Gateway type**, **VPN type**, and **SKU**.
6. Notice the need for a **Public IP address**.
7. Remember each virtual network will need a virtual network gateway.
8. Close the Add virtual network gateway. You do not need to save your changes.

Explore adding a connection between the virtual networks

1. Search for **Connections**.
2. Click **+ Add**.
3. Notice the **Connection type** can be VNet-to-VNet, Site-to-Site (IPsec), or ExpressRoute.
4. Provide enough information, so you can click the **Ok** button.
5. On the **Settings** page, notice that you will need to select the two different virtual networks.
6. Read the Help information on the **Establish bidirectional connectivity** checkbox.
7. Notice the **Shared key (PSK)** information.

8. Close the Add connection page. You do not need to save your changes.

Lesson 03, Configure ExpressRoute and Virtual WAN

This lesson does not have a formal demonstration.

Module 06, Administer Network Traffic

Lesson 01, Configure Network Routing and Endpoints

In this demonstration, we will learn how to create a route table, define a custom route, and associate the route with a subnet.

Note: This demonstration requires a virtual network with at least one subnet.

Create a routing table

1. Access the Azure portal.
2. Navigate to **Route tables**.
3. Select **+ Create**.
    - **Name**: *myRouteTablePublic*
    - **Subscription**: *select your subscription*
    - **Resource group**: *create or select a resource group*
    - **Region**: *select your location*
    - **Virtual network gateway route propagation**: *Enabled*
4. Select **Create**.
5. Wait for the new routing table to be deployed.

Add a route

1. Select your new routing table, and then select **Routes**.
2. Select **+ Add**.
    - **Name**: *ToPrivateSubnet*
    - **Address prefix**: *10.0.1.0/24*
    - **Next hop type**: *Virtual appliance*
    - **Next hop address**: *10.0.2.4*
3. Read the information and ensure you have IP forwarding enabled on your virtual appliance. You can enable this by navigating to the respective network interface's IP address settings.
4. Select **Create**.
5. Wait for the new route to be deployed.

Associate a route table to a subnet

1. Navigate to the subnet you want to associate with the routing table.
2. Select **Route table**.
3. Select your new routing table, **myRouteTablePublic**.
4. **Save** your changes.

Use PowerShell to view your routing information

1. Open the Cloud Shell.
2. View information about your new routing table.

   Get-AzRouteTable

3. Verify the **Routes** and **Subnet** information is correct.

Lesson 02, Configure Azure Load Balancer

This lesson does not have a formal demonstration. There is a Quickstart: Create a public load balancer - Azure portal - Azure Load Balancer | Microsoft Docs.

Lesson 03, Configure Azure Application Gateway

This lesson does not have a formal demonstration. There is a Quickstart: Direct web traffic using the portal - Azure Application Gateway | Microsoft Docs

Module 07, Administer Azure Storage

Lesson 01, Configure Storage Accounts

In this demonstration, we will create a storage accounts, upload a file, and secure the file endpoint.

Create a storage account in the portal

1. In the Azure portal, select **All services**. In the list of resources, type Storage Accounts. As you begin typing, the list filters based on your input. Select **Storage Accounts**.
2. On the Storage Accounts window that appears, choose **Add**.
3. Select the **subscription** in which to create the storage account.
4. Under the Resource group field, select **Create new**. Enter a name for your new resource group.
5. Enter a **name** for your storage account. The name you choose must be unique across Azure. The name also must be between 3 and 24 characters in length and can include numbers and lowercase letters only.
6. Select a **location** for your storage account or use the default location.
7. Leave these fields set to their default values:
    - Deployment model: **Resource Manager**
    - Performance: **Standard**
    - Account kind: **StorageV2 (general-purpose v2)**
    - Replication: **Locally redundant storage (LRS)**
    - Access tier: **Hot**
8. Select **Review + Create** to review your storage account settings and create the account.
9. Select **Create**.
10. If you have time, review the PowerShell and CLI code at the end of this demonstration.

Upload a file to the storage account

1. Within the Storage Account, create a **file share**, and **upload** a file.
2. For the Storage Account, use the **Shared Access Signature** blade to **Generate SAS and connection string**.
3. Use Storage Explorer and the connection string to access the file share.
4. Ensure you can view your uploaded file.

Note: This part of the demonstration requires a virtual network with a subnet.

Create a subnet service endpoint

1. Select your virtual network, and then select a subnet in the virtual network.
2. Under **Service Endpoints**, view the **Services** drop-down and the different services that can be secured with an endpoint.
3. Check the **Microsoft.Storage** option.
4. **Save** your changes.

Secure the storage to the service endpoint

1. Return to your **storage account**.
2. Select **Firewalls and virtual networks**.
3. Change to **Selected networks**.
4. Add existing virtual network, verify your subnet with the new service endpoint is listed.
5. **Save** your changes.

Test the storage endpoint

1. Return to the Storage Explorer.
2. **Refresh** the storage account.
3. You should now have an access error similar to this one:

Optional - Create a storage account using PowerShell

Use the following code to create a storage account using PowerShell. Swap out the storage types and names to suit your requirements.

```
Get-AzLocation | select Location

$location = "westus"

$resourceGroup = "storage-demo-resource-group"

New-AzResourceGroup -Name $resourceGroup -Location $location

New-AzStorageAccount -ResourceGroupName $resourceGroup -Name "storagedemo"
-Location $location -SkuName Standard_LRS -Kind StorageV2
```

Create a storage account using Azure CLI (optional)

Use the following code to create a storage account using Azure CLI. Change the storage types and names to suit your requirements.

```
az group create --name storage-resource-group --location westus

az account list-locations --query "[].{Region:name}" --out table

az storage account create --name storagedemo --resource-group storage-resource-group --location westus --sku Standard_LRS --kind StorageV2
```

Note: If you plan to use the storage account in other scenarios be sure to return the account to **All networks** in the **Firewalls and virtual networks** blade.

Lesson 02, Configure Blob Storage

In this demonstration, you will explore blob storage.

Note: This demonstration requires a storage account.

Create a container

1. Navigate to a storage account in the Azure portal.
2. In the left menu for the storage account, scroll to the **Blob service** section, then select **Blobs**.
3. Select the **+ Container** button.
4. Type a **Name** for your new container. The container name must be lowercase, must start with a letter or number, and can include only letters, numbers, and the dash (-) character.
5. Set the level of public access to the container. The default level is Private (no anonymous access).
6. Select **OK** to create the container.

Upload a block blob

1. In the Azure portal, navigate to the container you created in the previous section.
2. Select the container to show a list of blobs it contains. Since this container is new, it won't yet contain any blobs.
3. Select the **Upload** button to upload a blob to the container.
4. Expand the **Advanced** section.
5. Notice the **Authentication type**, **Blob type**, **Block size**, and the ability to **Upload to a folder**.
6. Notice the default **Authentication type** is SAS.

7. Browse your local file system to find a file to upload as a block blob and select **Upload**.
8. Upload as many blobs as you like in this way. You'll observe that the new blobs are now listed within the container.

Download a block blob

You can download a block blob to display in the browser or save to your local file system.

1. Navigate to the list of blobs that you uploaded in the previous section.
2. Right-click the blob you want to download and select **Download**.

Lesson 03, Configure Storage Security

In this demonstration, we will create a shared access signature.

Note: This demonstration requires a storage account, with a blob container, and an uploaded file.

Create a SAS at the service level

1. Sign into the Azure portal.
2. Locate the storage account you want to work with and open it. Drill down to your blob container.
3. Click the file you would like to provide access to.
4. Select the **Generate SAS** tab.
5. Configure the shared access signature using the following parameters:
   - **Permissions**: Read
   - **Start and expiry date/time**: Today's date to start, 1 year out for expiry
   - **Allowed protocols**: HTTPS
   - **Signing key**: Key1
6. Copy the **Blob Server SAS URL** and paste the URL into a browser.
7. Verify the blob file displays.
8. Review the different URL parameters.

Create a SAS at the account level

1. Return to your storage account.
2. Click **Shared access signature**.
3. Notice you can configure a variety of services, resource types, and permissions.
4. Click **Generate SAS and connection string**.
5. Review the connection string, SAS token, and URL information that is provided.

Lesson 04, Configure Azure Files and File Sync

In this demonstration, we will work with files shares and snapshots.

Note: These steps require a storage account.

Create a file share and upload a file

1. Access your storage account and click **Files**.
2. Click **+ File share** and give your new file share a **Name** and a **Quota**.
3. After your file share is created **Upload** a file.
4. Notice the ability to **Add a directory**, **Delete share**, and edit the **Quota**.

Manage snapshots

1. Access your file share.
2. Select **Create Snapshot**.
3. Select **View Snapshots** and verify your snapshot was created.
4. Click the snapshot and verify it includes your uploaded file.
5. Click the file that is part of the snapshot and review the **File properties**.
6. Notice the choices to **Download** and **Restore** the snapshot file.
7. Access the file share and delete the file you previously uploaded.
8. **Restore** the file from the snapshot.

Optional - Create a file share (PowerShell)

1. Gather the storage account name and the storage account key.

   Get-AzStorageAccount | fl *name*

   Get-AzStorageAccount -ResourceGroupName "YourResourceGroupName" -Name "YourStorageAccountName"

2. Retrieve an access key for your storage account.

   $storageAccountKeys = Get-AzStorageAccountKey -ResourceGroupName $resourceGroupName -Name $storageAccountName

3. Create a context for your storage account and key. The context encapsulates the storage account name and account key.

   $storageContext = New-AzStorageContext -StorageAccountName "YourStorageAccountName" -StorageAccountKey $storageAccountKeys[0].value

4. Create the file share. The name of your file share must be all lowercase.

   $share = New-AzStorageShare "YourFileShareName" -Context $storageContext

Optional - Mount a file share (PowerShell)

Note: Run the following commands from a regular (not an elevated) PowerShell session to mount the Azure file share. Remember to replace <your-resource-group-name>, <your-storage-account-name>, <your-file-share-name>, and desired-drive-letter with the proper information.

```
$resourceGroupName = "your-resource-group-name"

$storageAccountName = "your-storage-account-name"

$fileShareName = "your-file-share-name"

# These commands require you to be logged into your Azure account, run Login-AzAccount if you haven't already logged in.

$storageAccount = Get-AzStorageAccount -ResourceGroupName $resourceGroupName -Name $storageAccountName

$storageAccountKeys = Get-AzStorageAccountKey -ResourceGroupName $resourceGroupName -Name $storageAccountName

$fileShare = Get-AzStorageShare -Context $storageAccount.Context | Where-Object

{
  $_.Name -eq $fileShareName -and $_.IsSnapshot -eq $false
}
if ($fileShare -eq $null) {
  throw [System.Exception]::new("Azure file share not found")
}

# The value given to the root parameter of the New-PSDrive cmdlet is the host address for the storage account, storage-account.file.core.windows.net for Azure Public Regions. $fileShare.StorageUri.PrimaryUri.Host is used because non-Public Azure regions, such as sovereign clouds or Azure Stack deployments, will have different hosts for Azure file shares (and other storage resources).

$password = ConvertTo-SecureString -String $storageAccountKeys[0].Value -AsPlainText -Force

$credential = New-Object System.Management.Automation.PSCredential -ArgumentList "AZURE\$($storageAccount.StorageAccountName)", $password
```

New-PSDrive -Name desired-drive-letter -PSProvider FileSystem -Root "\\$($fileShare.StorageUri.PrimaryUri.Host)\$($fileShare.Name)" -Credential $credential -Persist

When finished, you can dismount the file share by running the following command:

Remove-PSDrive -Name desired-drive-letter

Lesson 05, Configure Storage with Tools

Demonstration – Storage Explorer

Note: If you have an older version of the Storage Explorer, be sure to upgrade.

Note: For the demonstration we will only configure a basic storage account connection.

In this demonstration, we will review several common Azure Storage Explorer tasks.

Download and install Storage Explorer

Note: Storage Explorer is available through the portal, if you prefer to use that for the demonstration.

1. Download and install Azure Storage Explorer
   - https://azure.microsoft.com/features/storage-explorer/
2. After the installation, launch the tool.
3. Review the Release Notes and menu options.

Connect to an Azure subscription

1. In Storage Explorer, select **Manage Accounts**, second icon top left. This will take you to the Account Management Panel.
2. The left pane now displays all the Azure accounts you've signed in to. To connect to another account, select **Add an account**.
3. If you want to sign into a national cloud or an Azure Stack, click on the Azure environment dropdown to select which Azure cloud you want to use.
4. Once you have chosen your environment, click the **Sign in...** button.
5. After you successfully sign in with an Azure account, the account and the Azure subscriptions associated with that account are added to the left pane.
6. Select the Azure subscriptions that you want to work with, and then select **Apply**.
7. The left pane displays the storage accounts associated with the selected Azure subscriptions.

Note: This next section requires an Azure storage account.

Attach an Azure storage account

1. Access the Azure portal, and your storage account.
2. Explore the choice for **Storage Explorer**.
3. Select **Access keys** and read the information about using the keys.
4. To connect in Storage Explorer, you will need the **Storage account name** and **Key1** information.
5. In Storage Explorer, **Add an account**.
6. Paste your account name in the Account name text box, paste your account key (the key1 value from the Azure portal) into the Account key text box, and then select **Next**.
7. Verify your storage account is available in the navigation pane. You may need to refresh the page.
8. Right-click your storage account and notice the choices including **Open in portal**, **Copy primary key**, and **Add to Quick Access**.

Generate a SAS connection string for the account you want to share

1. In **Storage Explorer**, right-click the storage account you want share, and then select **Get Shared Access Signature**.
2. Specify the time frame and permissions that you want for the account, and then click the **Create** button.
3. Next to the Connection String text box, select **Copy** to copy it to your clipboard, and then click **Close**.

Attach to a storage account by using a SAS Connection string

1. In **Storage Explorer**, open the **Connect Dialog**.
2. Choose **Use a connection string** and then click **Next**.
3. Paste your connection string into the **Connection string:** field. The **Display name:** field should populate. Click the **Next** button.
4. Verify the information is correct and select **Connect**.
5. After the storage account has successfully been attached, the storage account is displayed in the **Local and Attached** node with **(SAS)** appended to its name.

Demonstration – AzCopy

In this demonstration, we will explore AzCopy.

Install the AzCopy tool

1. Download your version of AZCopy - Get started with AZCopy
2. Install and launch the tool.

Explore the help

1. View the help.

   azcopy /?

2. Scroll to the top of the Help information and read about the **Common options**, like: source, destination, source key, and destination key.
3. Scroll down the **Samples** section. We will be trying several of these examples. Are any of these examples particularly interesting to you?

Download a blob from Blob storage to the file system

Note: This example requires an Azure storage account with blob container and blob file. You will also need to capture parameters in a text editor like Notepad.

1. Access the Azure portal.
2. Access your storage account with the blob you want to download.
3. Select **Access keys** and copy the **Key Key1** value. This will be the *sourcekey:* value.
4. Drill down to the blob of interest, and view the file **Properties**.
5. Copy the **URL** information. This will be the *source:* value.
6. Locate a local destination directory. This will be the *dest:* value. A filename is also required.
7. Construct the command using your values.

   azcopy /source:sourceURL /dest:destinationdirectoryandfilename /sourcekey:"key"

8. If you have errors, read them carefully and make corrections.
9. Verify the blob was downloaded to your local directory.

Upload files to Azure blob storage

Note: The example continues from the previous example and requires a local directory with files.

1. The *source:* for the command will be a local directory with files.

2. The *dest:* will the blob URL used in the previous example. Be sure to remove the filename, just include the storage account and container.
3. The *destkey:* will the key used in the previous example.
4. Construct the command using your values.

   azcopy /source:source /dest:destinationcontainer /destkey:key
5. If you have errors, read them carefully and make corrections.
6. Verify your local files were copied to the Azure container.
7. Notice there are switches to recurse subdirectories and pattern match.

Module 08, Administer Azure Virtual Machines

Lesson 01, Configure Virtual Machines

Demonstration – Create Virtual Machines in the portal

In this demonstration, we will create and access a Windows virtual machine in the portal.

Create the virtual machine

Note: These steps only cover a few virtual machine parameters. Feel free to explore and cover other areas.

1. Choose **Create a resource** in the upper left-hand corner of the Azure portal.
2. In the search box above the list of Azure Marketplace resources, search for **Windows Server 2016 Datacenter**. After locating the image, click **Create**.
3. In the **Basics** tab, under **Project details**, make sure the correct subscription is selected and then choose to **Create new** resource group. Type *myResourceGroup* for the name.
4. Under **Instance details**, type *myVM* for the **Virtual machine name** and choose *East US* for your **Location**. Leave the other defaults.
5. Under **Administrator account**, provide a username, such as *azureuser* and a password. The password must be at least 12 characters long and meet the defined complexity requirements.
6. Under **Inbound port rules**, choose **Allow selected ports** and then select **RDP (3389)** and **HTTP** from the drop-down.
7. Move to the **Management** tab, and under **Monitoring** turn **Off** Boot Diagnostics. This will eliminate validation errors.
8. Leave the remaining defaults and then select the **Review + create** button at the bottom of the page. Wait for the validation, then click **Create**.

Connect to the virtual machine

Create a remote desktop connection to the virtual machine. These directions tell you how to connect to your VM from a Windows computer. On a Mac, you need to install an RDP client from the Mac App Store.

1. Select the **Connect** button on the virtual machine properties page.
2. In the **Connect to virtual machine** page, keep the default options to connect by DNS name over port 3389 and click **Download RDP file**.
3. Open the downloaded RDP file and select **Connect** when prompted.

4. In the **Windows Security** window, select **More choices** and then **Use a different account**. Type the username as localhost\username, enter password you created for the virtual machine, and then select **OK**.
5. You may receive a certificate warning during the sign-in process.
   Select **Yes** or **Continue** to create the connection.

Install web server

1. To observe your VM in action, install the IIS web server. Open a PowerShell prompt on the VM and run the following command:

   Install-WindowsFeature -name Web-Server -IncludeManagementTools

2. After IIS has installed, close the RDP connection to the VM.

View the IIS welcome page

1. In the portal, select the VM and in the overview of the VM, use the **Click to copy** button to the right of the public IP address to copy it and paste it into a browser tab.
2. The default IIS welcome page will open.

Note: When no longer needed, you can delete the resource group, virtual machine, and all related resources. To do so, select the resource group for the virtual machine, select **Delete**, then confirm the name of the resource group to delete.

Demonstration – Connect to Linux Virtual Machines

In this demonstration, we will create a Linux machine and access the machine with SSL.

Note: Ensure port 22 is open for the connection to work.

Create the SSH Keys

1. Download the PuTTY tool. This will include PuTTYgen - https://putty.org/.
2. Once installed, open the **PuTTYgen** program.
3. In the **Parameters** option group choose **RSA**.
4. Click the **Generate** button.
5. Move your mouse around the blank area in the window to generate some randomness.
6. Copy the text of the **Public key for pasting into authorized keys file**.
7. Optionally you can specify a **Key passphrase** and then **Confirm passphrase.** You will be prompted for the passphrase when you authenticate to the VM with your private SSH key. Without a passphrase, if someone obtains your private key, they can sign in to any

VM or service that uses that key. We recommend you create a passphrase. However, if you forget the passphrase, there is no way to recover it.

8. Click **Save private key**.
9. Choose a location and filename and click **Save**. You will need this file to access the VM.

Create the Linux machine and assign the public SSH key

1. In the portal create a Linux machine of your choice.
2. Choose **SSH Public Key** for the **Authentication type** (instead of **Password**).
3. Provide a **Username**.
4. Paste the public SSH key from PuTTY into the **SSH public key** text area. Ensure the key validates with a checkmark.
5. Create the VM. Wait for it to deploy.
6. Access the running VM.
7. From the **Overview** blade, click **Connect**.
8. Remember your login information including user and public IP address.

Access the server using SSH

1. Open the **PuTTY** tool.
2. Enter **username@publicIpAddress** where username is the value you assigned when creating the VM and publicIpAddress is the value you obtained from the Azure portal.
3. Specify **22** for the **Port**.
4. Choose **SSH** in the **Connection Type** option group.
5. Navigate to **SSH** in the Category panel, then click **Auth**.
6. Click the **Browse** button next to **Private key file for authentication**.
7. Navigate to the private key file saved when you generated the SSH keys and click **Open**.
8. From the main PuTTY screen click **Open.**
9. You will now be connected to your server command line.

Lesson 02, Configure Virtual Machine Availability

In this demonstration, we will explore virtual machine scaling options.

Note: This demonstration requires a virtual machine scale set. If you need help with this, review Quickstart: Create a virtual machine scale set in the Azure portal .

Note: This demonstration is based on Exercise - Configure a virtual machine scale set .

Create a scale out rule

1. Access the Azure Portal select the virtual machine scale set you want to explore.
2. Under **Settings** select **Scaling**.
3. On the **Configuration** tab, review the purpose of scaling.
4. Review how **Manual scale** is used and discuss how to change the **Instance count**.
5. Select **Custom autoscale**.
    - Discuss how the default scale condition is executed when none of the other scale condition(s) match.
    - In the **Default** scale rule, discuss the difference between **Scale based on a metric** and **Scale to a specific instance count**.
6. Ensure that the **Scale mode** is set to **Scale based on a metric**. Then select **+ Add a rule**.
7. Create a rule with a Criteria and Action. The Criteria is when the CPU Percentage is over 75% for 10 minutes. The Action is to increase the instance count by 1.
    - Metric name: **Percentage CPU**
    - Operator: **Greater than**
    - Threshold: **75**
    - Duration: **10**
    - Operation: **Increase count by**
    - Instance count: **1**
8. After your rule is added, discuss how additional rules (like a scale in rule) could be used to optimize the deployment.

Create a scale in rule

1. Ensure that the **Scale mode** is set to **Scale based on a metric**. Then select **+ Add a rule**.
2. Create a rule with a Criteria and Action. The Criteria is when the CPU Percentage is less than 50% for 10 minutes. The Action is to decrease the instance count by 1.
    - Metric name: **Percentage CPU**
    - Operator: **Less than**
    - Threshold: **50**
    - Duration: **10**
    - Operation: **Decrease count by**
    - Instance count: **1**

3. Your default scale condition now contains two scale rules. One rule scales the number of instances out. Another rule scales the number of instances back in.

Lesson 03, Configure Virtual Machine Extensions

In this demonstration, we will explore Custom Script Extensions.

Note: This scenario requires a Windows virtual machine in the running state.

Verify the Web Server feature is available

1. Connect (RDP) to your Windows virtual machine and open a PowerShell prompt.
2. Run this command and verify the Web Server feature status is **Available** but not Installed.

   Get-WindowsFeature -name Web-Server

Create a PowerShell script file to install the Web Server

1. Create a file **Install_IIS.ps1** on your local machine.
2. Edit the file and add this command:

   Install-WindowsFeature -Name Web-Server

Configure an Extension in the Portal to run the script

1. In the Azure Portal, access your virtual machine, and select **Extensions**.
2. Click **+ Add**. Take a minute to review the many different extensions that are available.
3. Locate the **Custom Script Extension** resource, select, and click **Create**.
4. Browse to your PowerShell script and upload the file. There will be a notification that the file was uploaded.
5. Click **OK**.
6. Select your **CustomScriptExtension**.
7. Click **View detailed status** and verify provisioning succceeded.

Verify the Web Server was installed

1. Return to your virtual machine RDP session.
2. Verify the Web Server role was installed. This may take a couple of minutes.

   Get-WindowsFeature -name Web-Server

Note: You could also use the PowerShell **Set-AzVmCustomScriptExtension** command to deploy the extension. You would need to upload the script to blob container and use the URI. We will do this in the next demonstration.

Module 09, Configure PaaS Compute Options

Lesson 01, Configure Azure App Service Plans

In this demonstration, we will create and work with Azure App Service plans.

Create an App Service Plan

1. Sign-in to the Azure portal .
2. Search for and select **App Service Plans**.
3. Click **+ Add** to create a new App Service plan.
    - Subscription: **Choose your subscription**
    - Resource Group: **myRGAppServices (create new)**
    - Nam: **AppServicePlan1**
    - Operating System: **Windows**
    - Region: **East US**
4. Click **Review + Create** and then **Create**.
5. Wait for your new App Service plan to deploy.

Review Pricing Tiers

1. Locate your new App Service plan.
2. Under **Settings**, click **Scale up (App Service Plan)**.
3. Notice there are three tiers: **Dev/Test**, **Production**, and **Isolated**.
4. Click each tier and review the included features and included hardware.
5. How do the tiers compare?

Review autoscaling

1. Under **Settings** click **Scale out (App Service Plan)**.
2. Notice the default is **Manual scale**.
3. Notice you can specify an **instance count** depending on your App Service plan selection.
4. Click **Custom autoscale**.
5. Notice two scale modes: **Scale based on a metric** and **Scale to a specific instance count**.
6. Click **Add a rule** to automatically add an instance when the CPU percentages is greater than 80% for 10 minutes.

- Time aggregation: **Average**
- Metric name: **CPU percentage**
- Operator: **Greater than**
- Threshold: **80**
- Duration: **10 minutes**
- Operation: **Increase count by**
- Instance count: **1**
- Cool down: **5 minutes**

7. **Add** your rule changes.
8. Review the **Instance limits**: **Minimum**, **Maximum**, and **Default**.
9. Notice that you can add a **Schedule** and **Specify start/end dates** and **Repeat specific days**.
10. Do you see how you can create different App Service plans for your apps?

Lesson 02, Configure Azure App Services

In this demonstration, we will create a new web app that runs a Docker container. The container displays a Welcome message.

Create a Web App

Azure App Service is a collection of four services, all of which are built to help you host and run web applications. The four services (Web Apps, Mobile Apps, API Apps, and Logic Apps) look different, but in the end they all operate in very similar ways. Web Apps are the most used of the four services, and this is the service that we will be using in this lab.

In this task, you will create an Azure App Service Web App.

1. Sign-in to the Azure portal .
2. From the **All services** blade, search for and select **App Services**, and click **+ Add**
3. On the **Basics** tab of the **Web App** blade, specify the following settings (replace **xxxx** in the name of the web app with letters and digits such that the name is globally unique). Leave the defaults for everything else, including the App Service Plan.

Setting	Value
Subscription	**Choose your subscription**
Resource Group	**myRGWebApp1** (create new)
Name	**myLinuxWebAppxxxx** (unique)
Publish	**Docker Container**
Operating System	**Linux**
Region	**East US** (ignore any service plan availability warnings)

4. Click **Next > Docker** and configure the container information. The startup command is optional and not needed in this exercise.

Setting	Value
Options	**Single container**
Image Source	**Quickstart**
Sample	**Python Hello World**

5. Click **Review + create**, and then click **Create**.

Test the Web App

In this task, we will test the Web App.

1. Wait for the Web App to deploy.
2. From **Notifications** click **Go to resource**.

3. On the **Overview** blade, locate the **URL** entry.
4. Click on the **URL** to open the new browser tab and display the "Hello World, App Service!" page.
5. Switch back to the **Overview** blade of your web app and notice that it includes several charts. If you repeat step 4 a few times, you should be able to see corresponding telemetry being displayed in the charts. This includes number of requests and average response time.

Configure Deployment Slots

In this task, we will configure Deployment Slots for the Web App.

1. From the Web App blade, click **Deployment Slots**.
2. On the **Deployment Slots** blade, click **+ Add Slot**
3. From the **Add a slot** blade, configure the following settings.
    - Name: **DEVELOPMENT**
    - Clone Settings From: **myLinuxWebAppXXXX**
4. Click **Add**.
5. If the **Add a slot** blade remains open, click **Close**.
6. From the **Deployment Slots** blade, notice the **Names**, their **Status**, and the **Traffic %** of each Deployment Slot.
7. Click the newly created Deployment Slot **mylinuxwebappXXXX-DEVELOPMENT**. This will take you to the **Overview** blade of the new Deployment Slot.
8. From the **Overview** blade of the DEVELOPMENT Deployment Slot, locate the **URL** entry.
9. Click on the **URL** to open the new browser tab and display the "Hello World, App Service!" page.

   Note: The process of cloning the Web App settings to the new Deployment Slot, includes cloning the base Docker Image from the initial deployment.
10. Click the **X** in the top right corner of the DEVELOPMENT Deployment Slot blade. This will return you to the **Deployment Slots** blade of the **myLinuxWebAppXXXX** Web App.

Configure Backup

1. From the Web App blade, click **Backups**.
2. On the **Backups** blade, click **Configure**. This will open up the **Backup Configuration** blade.

3. From the **Backup Configuration** blade, under **Backup Storage**, click **Storage not configured** to configure a Storage Account for backups.
4. On the **Storage accounts** blade, click **+ Storage account**.
5. From the **Create storage account** blade, configure the following settings.

Setting	Value
Name	**webappxxxxstorage** (unique)
Account kind	**Storage (general purpose v1)**
Performance	**Standard**
Replication	**Locally-redundant storage (LRS)**
Location	**(US) East US)**

6. Click **OK**.
7. On the **Storage accounts** blade, click the Storage Account, **webappxxxxstorage**, that you created in the previous step.
8. From the **Containers** blade, click **+ Container**, enter **backups** for the name of the New Container, and set the **Public access level** to **Private (no anonymous access)**.
9. Click **OK**.
10. From the **Containers** blade, click **backups**, and click **Select** to choose the newly created Container. This will take you back to the **Backup Configuration** blade.
11. On the **Backup Configuration** blade, click **On** next to **Scheduled backup**, and configure the following settings.

Setting	Value
Backup Every	**1 Hours**

Setting	Value
Start backup schedule from	**Configure custom start time**
Retention (Days)	**30**
Keep at least one backup	**Yes**

12. Click **Save**.

Lesson 03, Configure Azure Container Instances

In this demonstration we create, configure, and deploy a container by using Azure Container Instances (ACI) in the Azure Portal. The container is a Welcome to ACI web application that displays a static HTML page.

Create a container instance

In this task, we will create a new container instance for the web application.

1. Sign-in to the Azure portal.
2. From the **All services** blade, search for and select **Container instances** and then click **+ Add, + Create, + New**.
3. Provide the following Basic details for the new container instance (leave the defaults for everything else):
    - Subscription: **Use default supplied**
    - Resource group: **Create new resource group**
    - Container name: **mycontainer**
    - Region: **<your choice>**
    - Image source: **Docker Hub or other registry**
    - Image type: **Public**
    - Image: **microsoft/aci-helloworld**
    - OS type: **Linux**
    - Size: **Leave at the default**
4. Configure the Networking tab (replace **xxxxx** with letters and digits such that the name is globally unique). Leave all other settings at their default values.

- DNS name label: **mycontainerdnsxxxxx**
- Your container will be publicly reachable at dns-name-label.region.azurecontainer.io . If you receive a **DNS name label not available** error message following the deployment, specify a different DNS name label (replacing the xxxxx) and re-deploy.

5. Click **Review and Create** to start the automatic validation process.
6. Click **Create** to create the container instance.
7. Monitor the deployment page and the **Notifications** page.

Verify deployment of the container instance

In this task, we verify that the container instance is running by ensuring that the welcome page displays.

1. After the deployment is complete, click the **Go to resource** link the deployment blade or the link to the resource in the Notification area.
2. On the **Overview** blade of **mycontainer**, ensure your container **Status** is **Running**.
3. Locate the Fully Qualified Domain Name (FQDN).
4. Copy the container's FQDN into a new web browser tab and press **Enter**. The Welcome page should display.

Note: To avoid additional costs, you can remove this resource group. Search for resource groups, click your resource group, and then click **Delete resource group**. Verify the name of the resource group and then click **Delete**. Monitor the **Notifications** to see how the delete is proceeding.

Lesson 04, Configure Azure Kubernetes Services

In this demonstration, we wil deploy an Azure Kubernetes Service.

Create a Kubernetes service

1. Sign-in to the Azure portal .
2. Search for and select **Kubernetes services**, and then click **+ Add**.
3. On the Basics page, configure the following options and then select **Next: Scale**.

- **Project details**: Select an Azure Subscription, then select or create an Azure Resource group, such as **myResourceGroup**.
- **Cluster details**: Enter a Kubernetes cluster name, such as **myAKSCluster**. Select a Region, Kubernetes version, and DNS name prefix for the AKS cluster.

- **Primary node pool**: Select a VM Node size for the AKS nodes. The VM size can't be changed once an AKS cluster has been deployed. - Select the number of nodes to deploy into the cluster. For this demonstration, set Node count to 1. Node count can be adjusted after the cluster has been deployed.

4. On the **Scale** page, review and keep the default options. At the bottom of the screen, click **Next: Authentication**.
5. On the **Authentication** page, configure the following options:
    - Create a new service principal by leaving the Service Principal field with (new) default service principal. Or you can choose Configure service principal to use an existing one. If you use an existing one, you will need to provide the SPN client ID and secret.
    - Enable the option for Kubernetes role-based access controls (RBAC). This will provide more fine-grained control over access to the Kubernetes resources deployed in your AKS cluster.
6. By default, **Basic networking** is used, and Azure Monitor for containers is enabled. Click **Review + create** and then **Create** when validation completes.
7. It takes a few minutes to create the AKS cluster.

Connect to the cluster

1. To manage a Kubernetes cluster, you use **kubectl**, the Kubernetes command-line client. The kubectl client is pre-installed in the Azure Cloud Shell.
2. Open the **Cloud Shell**, select the **Bash** shell.
3. Connect to the cluster, downloads your credentials, and configure the Kubernetes CLI to use them.

    az aks get-credentials --resource-group myResourceGroup --name myAKSCluster
4. Verify the connection to your clusterand return a list of the cluster nodes. Make sure that the status of the nodes is Ready.

    kubectl get nodes

Run the application

Note: You will need an Kubernetes manifest file for the next steps. Navigate to the Quickstart - Deploy an AKS cluster in the portal .

1. In the cloud shell, use either the **nano azure-vote.yaml** or **vi azure-vote.yaml** command to create a file named azure-vote.yaml.

2. Copy the YAML definition from the Quickstart page. Be sure to save your changes.
3. Deploy the application.

   kubectl apply -f azure-vote.yaml

4. Ensure there are no errors and the output shows the Deployments and Services created successfully.

Test the application

1. When the application runs, a Kubernetes service exposes the application front end to the internet. This process can take a few minutes to complete.
2. Continue in the cloud shell to monitor the progress of the deployment.

   kubectl get service azure-vote-front --watch

3. Wait until the EXTERNAL-IP address changes from pending to an actual public IP address. Use Ctrl + C to break out of the command.
4. To see the Azure Vote app in action, open a web browser to the external IP address of your service.
5. Return to the Azure portal and your myAKSCluster resource.
6. Under **Monitoring** choose **Insights**. Review the available information.
7. As you have time review other areas of the cluster.

Module 10, Administer Data Protection

Note: There are several demonstrations in this lesson. You should select those that are most appropriate for you audience.

Lesson 01, Configure File and Folder Backups

Demonstration - Backup Azure File Shares

In this demonstration, we will explore backing up a file share in the Azure portal.

Configure a storage account with file share

Note: If you already have a storage account and file share, you can skip this step.

1. In the Azure portal, search for **Storage Accounts**.
2. **Add** a new storage account.
3. Provide the storage account information (your choice).
4. Click **Review + create** and then **Create**.
5. Access your storage account, and click **Files**.
6. Click **+ File share** and give your new file share a **Name** and a **Quota**.
7. After your file share is created **Upload a file**.

Create a Recovery Services vault

1. In the Azure portal, type Recovery Services and click **Recovery Services vaults**.
2. Click **Add**.
3. Provide a **Name**, **Subscription**, **Resource group**, and **Location**.
4. Your new vault should be in the same location as the file share.
5. Click **Create**. It can take several minutes for the Recovery Services vault to be created. Monitor the status notifications in the upper right-hand area of the portal. Once your vault is created, it appears in the list of Recovery Services vaults.
6. If after several minutes the vault is not added, click **Refresh**.

Configure file share backup

1. Open your recovery services vault.
2. Click **Backup** and create a new backup instance.
3. From the **Where is your workload running?** drop-down menu, select **Azure**.
4. From the **What do you want to backup?** menu, select **Azure FileShare**.

5. Click **Backup**.
6. From the list of Storage accounts, **select a storage account**, and click **OK**. Azure searches the storage account for files shares that can be backed up. If you recently added your file shares, allow a little time for the file shares to appear.
7. From the File Shares list, **select one or more of the file shares** you want to backup, and click **OK.**
8. On the Backup Policy page, choose **Create New backup policy** and provide Name, Schedule, and Retention information. Click **OK**.
9. When you are finished configuring the backup click **Enable backup**.

Verify the file share backup

1. Explore the **Backup items** blade. There is information on backed up items and replicated items.
2. Explore the **Backup policies** blade. You can add or delete backup policies.
3. Explore the **Backup jobs** blade. Here you can review the status of your backup jobs.

Demonstration – Backup Files and Folders

In this demonstration, we will step through the process to backup and restore files and folders from Windows to Azure.

Note: This demonstration assumes you have not used the Azure Backup Agent before and need a complete installation.

Create a Recovery Services vault

1. In the Azure portal, type Recovery Services and click **Recovery Services vaults**.
2. Click **Add**.
3. Provide a **Name**, **Subscription**, **Resource group**, and **Location**.
4. Click **Create**. It can take several minutes for the Recovery Services vault to be created. Monitor the status notifications in the upper right-hand area of the portal. Once your vault is created, it appears in the list of Recovery Services vaults.
5. If after several minutes you don't observe your vault, click **Refresh**.

Configure the vault

1. For your recovery services vault, click **Backup**.
2. From the **Where is your workload running?** drop-down menu, select **On-premises**.
3. From the **What do you want to backup?** menu, select **Files and folders**. Notice your other choices.

4. Click **Prepare infrastructure**.
5. Click **Download Agent for Windows Server or Windows Client**. A pop-up menu prompts you to run or **save** MARSAgentInstaller.exe.
6. By default, the MARSagentinstaller.exe file is saved to your **Downloads** folder. When the installer completes, a pop-up asking if you want to run the installer, or open the folder. You **don't need** to install the agent yet. You can install the agent after you have downloaded the vault credentials.
7. Return to your recovery services vault, check the box **Already downloaded or using the latest recovery services agent**.
8. Click **Download**. After the vault credentials finish downloading, a pop-up asking if you want to open or **save** the credentials. Click **Save**. If you accidentally click **Open**, let the dialog that attempts to open the vault credentials, fail. You cannot open the vault credentials. Proceed to the next step. The vault credentials are in the **Downloads** folder.

Note: You must have the latest version of the MARS agent. Versions of the agent below 2.0.9083.0 must be upgraded by uninstalling and reinstalling the agent.

Install and register the agent

1. Locate and double-click the **MARSagentinstaller.exe** from the **Downloads** folder (or other saved location). The installer provides a series of messages as it extracts, installs, and registers the Recovery Services agent.
2. To complete the wizard, you need to:
    - Choose a location for the installation and cache folder.
    - Provide your proxy server info if you use a proxy server to connect to the internet.
    - Provide your user name and password details if you use an authenticated proxy.
    - If prompted, install any missing software.
    - Provide the downloaded vault credentials
    - Enter and save the encryption passphrase in a secure location.
3. Wait for the server registration to complete. This could take a couple of minutes.
4. The agent is now installed and your machine is registered to the vault. You're ready to configure and schedule your backup.

Create the backup policy

1. Open the **Microsoft Azure Recovery Services** agent. You can find it by searching your machine for Microsoft Azure Recovery Services.

2. If this is the first time you are using the agent there will be a **Warning** to create a backup policy. The backup policy is the schedule when recovery points are taken, and the length of time the recovery points are retained.
3. Click **Schedule Backup** to launch the Schedule Backup Wizard.
    - Read the **Getting Started** page.
    - **Add items** to include files and folders that you want to protect. Select just a few sample files. Notice you can exclude files from the backup.
    - Specify the **backup schedule**. You can schedule daily (at a maximum rate of three times per day) or weekly backups.
    - Select your **retention policy** settings. The retention policy specifies the duration for which the backup is stored. Rather than just specifying a "flat policy" for all backup points, you can specify different retention policies based on when the backup occurs. You can modify the daily, weekly, monthly, and yearly retention policies to meet your needs.
    - Choose your **initial backup type page** as **Automatically**. Notice there is a choice for offline backup.
    - **Confirm** your choices and **Finish** the wizard.

Backup files and folders

1. Click **Back Up Now** to complete the initial sending over the network.
2. In the wizard, confirm your settings, and then click **Back Up**.
3. You may **Close** the wizard. It will continue to run in the background.
4. The **Status** of your backup will show on the first page of the agent.
5. You can **View Details** for more information.

Explore the recover settings

1. Click **Recover data**.
2. Walkthrough the wizard making selections based on your backup settings.
3. Notice your choices to restore from the current server or another server.
4. Notice you can backup individual files and folders or an entire volume.
5. Select a volume and **Mount** the drive. This can take a couple of minutes.
6. Verify the mounted volume can be accessed in **File Explorer** and that your backup files are available.
7. **Unmount** the drive.

Explore the backup properties

1. Click **Change Properties**.
2. Explore the different tabs.

3. On the **Encryption** tab you can change the passphrase.
4. On the **Proxy Configuration** tab you can add proxy information.
5. On the **Throttling** tab you can enable internet bandwidth usage throttling. Throttling controls how network bandwidth is used during data transfer. This control can be helpful if you need to back up data during work hours but do not want the backup process to interfere with other Internet traffic. Throttling applies to back up and restore activities.

Delete your backup schedule

1. Click **Schedule Backup**.
2. In the wizard, select **Stop using this backup schedule and delete all the stored backups**.
3. Verify your choices and click **Finish**.
4. You will be prompted for a recovery services vault security pin.
5. In the Azure portal locate your recovery services vault.
6. Select **Properties** and then Security PIN **Generate**.
7. Copy the PIN into the Backup agent to finish deleting the schedule.

Lesson 02, Configure Virtual Machine Backups

In this demonstration, we will schedule a daily backup of a virtual machine to a Recovery Services vault.

Note: This demonstration requires a virtual machine and a recovery service vault.

Enable a backup on a virtual machine

1. In the Azure portal select the virtual machine you would like to backup.
2. In the **Operations** section, choose **Backup**. The Enable backup window opens.
3. Select **Create new** and provide a name for the new vault, such as *myRecoveryServicesVault*.
4. If not already selected, choose **Use existing**, then select the resource group of your VM from the drop-down menu.
5. Discuss how, by default, the vault is set for Geo-Redundant storage. This storage redundancy level ensures that your backup data is replicated to a secondary Azure region that's hundreds of miles away from the primary region.
6. Discuss how you can create and use policies to define when a backup job runs and how long the recovery points are stored. The default protection policy runs a backup job each day and retains recovery points for 30 days.
7. To accept the default backup policy values, select **Enable Backup**.
8. It takes a few moments to create the Recovery Services vault.

Start a backup job and monitor the progress

1. Discuss how you can start a backup at any time, rather than wait for the default policy to run the job at the scheduled time. This first backup job creates a full recovery point. Each backup job after this initial backup creates incremental recovery points. Incremental recovery points are storage and time-efficient, as they only transfer changes made since the last backup.
2. On the **Backup** window for your VM, select **Backup now**.
3. Accept the backup retention policy of 30 days.
4. To start the job, select **Backup.**
5. In the Backup window for your VM, review the status of the backup and number of completed restore points.
6. Once the VM backup job is complete, information on the **Last backup time**, **Latest restore point**, and **Oldest restore point** is shown.
7. Point out the **Stop Backup** selection.

Module 11, Administer Monitoring

Lesson 01, Configure Azure Monitor

This lesson does not have a formal demonstration.

Lesson 02, Configure Azure Alerts

In this demonstration, we will create an alert rule.

Create an alert rule

1. In Azure portal, click on **Monitor**. The Monitor blade consolidates all your monitoring settings and data in one view.
2. Click **Alerts** then click **+ New alert rule**. As most resource blades also have Alerts in their resource menu under Monitoring, you could create alerts from there as well.

Explore alert targets

1. Click **Select** under Target, to select a target resource that you want to alert on. Use **Subscription** and **Resource type** drop-downs to find the resource you want to monitor. You can also use the search bar to find your resource.
2. If the selected resource has metrics you can create alerts on, Available signals on the bottom right will include metrics. You can view the full list of resource types supported for metric alerts in this article.
3. Click **Done** when you have made your selection.

Explore alert conditions

1. Once you have selected a target resource, click on **Add condition**.
2. You will observe a list of signals supported for the resource, select the metric you want to create an alert on.
3. Optionally, refine the metric by adjusting Period and Aggregation. If the metric has dimensions, the Dimensions table will be presented.
4. Observe a chart for the metric for the last 6 hours. Adjust the **Show history** drop-down.
5. Define the **Alert logic**. This will determine the logic which the metric alert rule will evaluate.
6. If you are using a static threshold, the metric chart can help determine what might be a reasonable threshold. If you are using a Dynamic Thresholds, the metric chart will display the calculated thresholds based on recent data.
7. Click **Done**.
8. Optionally, add another criteria if you want to monitor a complex alert rule.

Explore alert details

1. Fill in Alert details like **Alert Rule Name**, **Description** and **Severity**.
2. Add an action group to the alert either by selecting an existing action group or creating a new action group.
3. Click **Done** to save the metric alert rule.

Lesson 03, Configure Log Analytics

In this demonstration, you will work with the Log Analytics query language.

Access the demonstration environment

1. Access the Log Analytics Querying Demonstration page.
2. This page provides a live demonstration workspace where you can run and test queries.

Use the Query Explorer

1. Select **Query Explorer** (top right).
2. Expand **Favorites** and then select **All Syslog records with errors**.
3. Notice the query is added to the editing pane. Notice the structure of the query.
4. **Run** the query. Explore the records returned.
5. As you have time experiment with other **Favorites** and **Saved Queries**.

Note: Is there a particular query your students are interested in seeing?

Lesson 04, Configure Network Watcher

This lesson does not have a formal demonstration. The Module 06 Implement Traffic Management lab included the Network Watcher agent.