

Contents

1	AZ-030: Microsoft Azure technologies for AWS architects	4
1.1	Notes on AZ-030 Course & Labs	4
1.2	What are we doing?	4
1.3	How should I use these files relative to the released MOC files?	4
1.4	What about changes to the student handbook?	4
1.5	How do I contribute?	4
1.6	Notes	5
1.6.1	Classroom Materials	5
1.7	It is strongly recommended that MCTs and Partners access these materials and in turn, provide them separately to students. Pointing students directly to GitHub to access Lab steps as part of an ongoing class will require them to access yet another UI as part of the course, contributing to a confusing experience for the student. An explanation to the student regarding why they are receiving separate Lab instructions can highlight the nature of an always-changing cloud-based interface and platform. Microsoft Learning support for accessing files on GitHub and support for navigation of the GitHub site is limited to MCTs teaching this course only.	5
1.8	title: Online Hosted Instructions permalink: index.html layout: home	5
2	Content Directory	5
2.1	Labs	5
2.2	Demos	5
3	AZ-030: Microsoft Azure technologies for AWS architects	5
3.1	Notes on AZ-303 Labs	5
4	Mini-lab: Use the Azure portal	5
4.1	All services	5
4.2	Azure Cloud Shell	6
4.3	Directory and subscription	6
4.4	Notifications pane	7
4.5	Settings	7
4.6	Help pane	7
4.6.1	What's new and other information	8
4.7	Feedback pane	8
4.8	Profile settings	8
5	Mini-lab: Create ARM templates by using the Azure Portal	9
5.1	Generate a template using the portal	9
5.2	Edit and deploy the template	10
5.3	Clean up resources	10
6	Mini-lab: Create ARM templates by using Visual Studio Code	11
6.1	Prerequisites	11
6.2	Open the Quickstart template	11
6.3	Edit the template	11
6.4	Deploy the template	12
6.5	Clean up resources	13
7	Mini-lab: Create VMs inside an availability set	13
7.1	Prerequisites	13
7.2	Create an availability set	14
7.3	Create VMs inside an availability set	14
7.4	Clean up deployment	14
8	Mini-lab: Secure user sign-in events with Azure Multi-Factor Authentication	15
8.1	Create a Conditional Access Policy	15
8.2	Configure the conditions for Multi-Factor Authentication	15
8.3	Test Azure Multi-Factor Authentication	16
9	Mini-lab: Add Guest users to Azure AD in the Azure Portal	16
9.1	Prerequisites	16

9.2	Add Guest users to Azure AD	17
9.3	Assign an App to a Guest user	17
9.4	Accept the Guest user invite	18
10	Mini-lab: Azure Active Directory Seamless Single Sign-On	18
10.1	Prerequisites	18
10.2	Enable Azure AD Connect	18
10.3	Verify Seamless SSO is enabled	19
11	Mini-lab: Enable Azure AD Self-Service Password Reset	19
11.1	Prerequisites	19
11.2	Enable self-service password reset	20
11.3	Select authentication methods and registration options	20
11.4	Configure prompt for registration when users next sign in	20
11.5	Configure notifications and customizations	21
11.6	Test Self-Service Password Reset	21
12	Mini-lab: VNet Peering	21
12.1	Create virtual networks	21
12.2	Configure VNet peering on the first virtual network	22
12.3	Configure a VPN gateway	22
12.3.1	Allow gateway transit	22
12.4	Confirm VNet peering on the second virtual network	23
13	Mini-lab: Create a VM Scale Set in the Azure Portal	23
13.1	Create public load balancer	23
13.2	Create virtual machine scale set	23
13.3	Clean up resources	24
14	Mini-lab: Create a Load Balancer to Load Balance VMs	24
14.1	Create a Load Balancer	24
14.2	Create Load Balancer resources	25
14.3	Create a Backend pool	25
14.4	Create a health probe	25
14.5	Create a Load Balancer rule	25
14.6	Create backend servers	26
14.7	Virtual network and parameters	26
14.8	Create the virtual network	26
14.9	Create virtual machines	27
14.10	Create NSG rule	28
14.11	Install IIS	28
14.12	Test the Load Balancer	29
15	Mini-lab: Run Azure Container Instances	29
15.1	Create a container	29
16	Mini-lab: Deploy Kubernetes with AKS	30
16.1	Create a new resource group	30
16.2	Configure networking	31
16.3	Kubenet networking	31
16.4	Azure Container Networking Interface (CNI) networking	31
16.5	Create the AKS cluster	31
16.6	Test cluster connectivity by using kubectl	32
16.7	Create a Kubernetes namespace for the application	32
17	Mini-lab: Create an App Service Plan	33
17.1	Clean up resources	34
18	Mini-lab: Create an App Service and Web App	34
18.1	Create App Service and web app	34
18.2	Preview web app	35

19 Mini-lab: Deploy staging slots	35
19.1 Add a slot	35
19.2 Swap two slots	36
20 Mini-lab: Create a storage account in the portal	36
21 Mini-lab: Blob Storage	37
21.1 Create a container	37
21.2 Upload a block blob	37
21.3 Download a block blob	38
22 Mini-lab: Create a Shared Access Signature (Portal)	38
22.1 Create an SAS at the service level	38
22.2 Create an SAS at the account level	38
23 Mini-lab: Create an Azure SQL Database single database	38
23.1 Create a single database	39
23.2 Query the database	40
24 Mini-lab: Create an Azure SQL Database Managed Instance	40
24.1 Create a managed instance	40
24.2 Basics	41
24.3 Networking	41
24.4 Additional settings	41
24.5 Review + create	41
25 Lab: Implementing Azure SQL Database-Based Applications	42
25.1 Lab scenario	42
25.2 Objectives	42
25.3 Lab Environment	42
25.4 Lab Files	42
25.5 Exercise 1: Implement Azure SQL Database	42
25.5.0.1 Task 1: Create Azure SQL Database	42
25.5.0.2 Task 2: Connect to and query Azure SQL Database	43
25.5.1 Exercise 2: Implement a .NET Core console app that uses Azure SQL Database as their data store	44
25.5.1.1 Task 1: Identify ADO.NET connection information of Azure SQL Database	44
25.5.1.2 Task 2: Create and configure a .NET Core console app	44
25.5.1.3 Task 3: Test the .NET Core console app	45
25.5.1.4 Task 4: Configure Azure SQL database firewall	46
25.5.1.5 Task 5: Verify the functionality of the .NET Core console app	46
25.5.1.6 Task 6: Remove Azure resources deployed in the lab	46
26 Mini-lab: Azure Security Center	46
27 Mini-lab: Monitor costs with Azure Monitor	47
27.1 Navigate to Cost Analysis	47
27.2 Customize report	47
27.3 Access the demonstration environment	47
27.4 Use the Query Explorer	47
28 Mini-lab: Add users by using Azure Active Directory	47
29 Mini-lab: Add an Azure Role Assignment	48
29.1 Prerequisites	48
29.2 Add a role assignment	48
30 Mini-lab: Enable system-assigned managed identity on an existing VM	49
30.1 Enable managed identity	49
30.2 Sign in to Azure VM using managed identity (PowerShell)	49
31 Mini-lab: Create and manage a policy to enforce compliance	50
31.1 Assign a policy	50

32 Mini-lab: Create and encrypt a Windows virtual machine with the Azure portal	51
32.1 Create a virtual machine	51
32.2 Encrypt the virtual machine	51
32.3 Clean up resources	52
33 Mini-lab: Back up files and folders	52
33.1 Create a Recovery Services vault	52
33.2 Configure the vault	53
33.3 Install and register the agent	53
33.4 Create the backup policy	53
33.5 Backup files and folders	54
33.6 Explore the recover settings	54
33.7 Explore the backup properties	54
33.8 Delete your backup schedule	54

1 AZ-030: Microsoft Azure technologies for AWS architects

1.1 Notes on AZ-030 Course & Labs

This course, AZ-030: Microsoft Azure technologies for AWS architects, is a subset of the AZ-303 course materials, adapted to fit a 4 day course for experienced AWS architects. The course uses mini-labs (located in folder /Instructions/Mini-labs) which give students short and frequent hands on interaction with the Azure Portal and Cloud Shell environments.

To promote further learning students are also encouraged to complete the longer [labs for AZ-303](#), which run 60 to 120 minutes each, after completion of this course.

1.2 What are we doing?

- To support this course, we will need to make frequent updates to the course content to keep it current with the Azure services used in the course. We are publishing the mini-lab instructions on GitHub to allow for open contributions between the course authors and MCTs to keep the content current with changes in the Azure platform.
- We hope that this brings a sense of collaboration to the labs like we've never had before - when Azure changes and you find it first during a live delivery, go ahead and make an enhancement right in the lab source. Help your fellow MCTs.

1.3 How should I use these files relative to the released MOC files?

- The instructor handbook and PowerPoints are still going to be your primary source for teaching the course content.
- These files on GitHub are designed to be used in conjunction with the student handbook, but are in GitHub as a central repository so MCTs and course authors can have a shared source for the latest lab files.
- It will be recommended that for every delivery, trainers check GitHub for any changes that may have been made to support the latest Azure services, and get the latest files for their delivery.

1.4 What about changes to the student handbook?

- We will review the student handbook on a quarterly basis and update through the normal MOC release channels as needed.

1.5 How do I contribute?

- Any MCT can submit a pull request to the code or content in the GitHub repro, Microsoft and the course author will triage and include content and lab code changes as needed.
- You can submit bugs, changes, improvement and ideas. Find a new Azure feature before we have? Submit a new demo!

1.6 Notes

1.6.1 Classroom Materials

1.7 It is strongly recommended that MCTs and Partners access these materials and in turn, provide them separately to students. Pointing students directly to GitHub to access Lab steps as part of an ongoing class will require them to access yet another UI as part of the course, contributing to a confusing experience for the student. An explanation to the student regarding why they are receiving separate Lab instructions can highlight the nature of an always-changing cloud-based interface and platform. Microsoft Learning support for accessing files on GitHub and support for navigation of the GitHub site is limited to MCTs teaching this course only.

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

2 Content Directory

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

2.1 Labs

```
{% assign labs = site.pages | where_exp:"page", "page.url contains '/Instructions/Labs'" %} | Module | Lab | | ---  
| --- | {% for activity in labs %} | {{ activity.lab.module }} | [{{ activity.lab.title }}{% if activity.lab.type %} - {{  
activity.lab.type }}{% endif %}](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-  
for-AWS-architects/{{ site.github.url }}{{ activity.url }}) | {% endfor %}
```

2.2 Demos

```
{% assign demos = site.pages | where_exp:"page", "page.url contains '/Instructions/Demos'" %} | Module  
| Demo | | --- | --- | {% for activity in demos %} | {{ activity.demo.module }} | [{{ activity.demo.title  
}}](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/{{  
site.github.url }}{{ activity.url }}) | {% endfor %}
```

3 AZ-030: Microsoft Azure technologies for AWS architects

3.1 Notes on AZ-303 Labs

To promote further learning students are encouraged to complete the longer [labs for AZ-303](#), which run 60 to 120 minutes each, after completion of this course. This is readme

4 Mini-lab: Use the Azure portal

The Azure portal has several features and services available; let's look at some of the more common areas you'll tend to use. First, take a moment to hover your mouse pointer over each of the icons in the top menu bar for a few seconds each. A tooltip label pop-up should be available for each icon. This label is the name of the menu item. You will use these icons later.

![Screenshot of the Azure portal icon bar](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/5-portal-icon-bar.png)

On a narrower screen the menu may not appear, select the ellipses (...) button.

![Ellipses button icon](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/three-points.png)

4.1 All services

There are several services you will find in the azure portal.

1. Sign in to the [Azure portal](#) with your Azure account.

2. On the top left-hand side of the Azure portal, select **Show portal menu**.

![Screenshot of the portal menu option](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/5-show-portal-menu.png)

3. Select **All services**.
 - Take a couple of minutes to review the list of the many services Azure offers.
4. Select **Virtual machines**. (Alternatively, use the search box in the top left corner of the view).
5. The **Virtual Machines** pane appears. Unless you previously created a virtual machine there will be no results.
6. Select **+ Add**. The **Create a virtual machine** pane appears.
7. Select the **X** in the top right corner to close the **Create a virtual machine** pane.
8. Select the **X** in the top right corner to close the **Virtual machines** pane.
9. Select **Microsoft Azure** in the top left-hand side to get back to the home page.

4.2 Azure Cloud Shell

![Icon representing the Azure Cloud Shell](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/5-cloud-shell-icon.png)

The Azure Cloud Shell allows you to use a command-line interface (CLI) to execute commands in your Azure subscription. You can access it by selecting the icon in the toolbar. You can also navigate to <https://shell.azure.com> to launch a Cloud Shell in the browser independent of the portal.

There are a variety of management and programming tools included in the created environment.

- Azure command-line tools (Azure CLI, AzCopy, etc.)
- Languages / Frameworks including .NET Core, Python, and Java
- Container management support for Docker, Kubernetes, etc.
- Code editors such as vim, emacs, code, and nano
- Build tools (make, maven, npm, etc.)
- Database query tools such as sqlcmd

Working with the CLI:

1. Click the Azure Cloud Shell icon
2. You can choose either a **Bash** or **PowerShell** environment, depending on your personal preferences. Select any
 - You can also change the shell at any time through the language drop-down on the left side of the shell.
3. Now you have accessed the Azure CLI
4. Write

```
az version
```
5. The CLI returns basic information about the Cloud Shell CLI Version in JSON format
6. You can close it now by clicking the **X** on the top right.

4.3 Directory and subscription

![Icon representing the subscription panel](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/5-subscription-icon.png)

1. Select the **Directory + Subscription** (book and filter) icon to show the **Directory + subscription** pane.

This is where you can switch between multiple subscriptions or directories. If you have other Azure directories tied to the same email address, those subscriptions will be available as well.

There is also a link to learn more about directories and subscriptions.

2. Select the **X** in the top right corner to close the **Directory + subscription** pane.

4.4 Notifications pane

1. On the icon bar menu bar, select the **Notifications** (bell) icon. This window lists any pending notifications.

![Screenshot of notifications window](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/5-notifications-pane.png)

2. If any notifications appear, hover your mouse over one of them. Select the **X** that appears in that notification to dismiss it.
3. Select **Dismiss all**. You should have no notifications showing.
4. Select the **X** in the top right corner to close the **Notifications** pane.

4.5 Settings

![Icon representing the settings panel](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/5-settings-icon.png)

1. Select the **Settings** (cog) icon to open the **Portal settings** pane, showing the **General** settings by default.
2. Drop down the **Sign me out when inactive** setting. There you can select a convenient option, for example, selecting **After one hour** of inactivity for automatic sign out.
3. Under **Choose a theme**, select the different colored themes and observe the changes to the portal UI. Leave it set to the one you like the best.
4. Under **High contrast theme**, try the three different options.
5. Select **Enable pop-up notifications**. When this option is checked, notifications will appear as pop-up "toast"-style notifications. They will still show up in the Notifications (bell) icon as well.
6. Select the **Language & region** tab in the settings. Select **Language** and pick **Español**, and then select the **Apply** button. If a **Translate this page** dialog box appears, close the box. The whole portal is now in Spanish.
7. To revert back to English, select the **Settings** (cog) icon in the top menu bar and switch to the **Idioma y región** settings. Select **Idioma** and pick **English**. Select the **Aplicar** button. The portal returns to English.

4.6 Help pane

![Icon representing the help panel](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/5-help-icon.png)

1. Select the **Help** (?) icon to show the **Help** pane.
2. Select the **Help + support** button.

Note: Support requests can only be created using an active paid subscription, therefore, some of these steps may be different the current UI.

1. In the **Help + support** pane, under **Support**, select **New support request**. To create a new support request, you would fill in the information in each of the following sections:
 - **Basics:** the issue type
 - **Problem:** severity of the problem, a summary and description, and any additional information

- **Contact information:** preferred contact method and the information associated with this contact method
2. Select **Create** to lodge the issue.
 3. You can view the status of your support requests by selecting on **All support requests**.

4.6.1 What's new and other information

1. Select the **Help** icon in the top right again, and select **What's new**.
2. Review the features that have recently been released. Also note and explore the other **Help** menu options, such as:
 - Azure roadmap
 - Launch guided tour
 - Keyboard shortcuts
 - Show diagnostics
 - Privacy statement
3. Select the **X** in the top right corner to close the **What's new** pane. You should now be back to the Dashboard.

4.7 Feedback pane

![[Icon representing the feedback panel]](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/5-feedback-icon.png)

If at any point on your Azure journey, you have some feedback or want to do a suggestion to us, you can do the following:

1. Select the **Feedback** (smiley face) icon to open the **Send us feedback** pane.
2. Type your impressions of Azure in the **Tell us about your experience**
3. Select the box that says **Microsoft can email you about your feedback**
4. Select **Submit Feedback**.
5. A **Feedback sent** message will appear, and then close. You should now be back at the Dashboard.

4.8 Profile settings

1. Select on your name in the top right corner of the portal to display profile information:

- Your Name
- Your email
- "My Microsoft account" link
- "Switch Directory" link
- Sign in with another account, or sign out entirely
- An ellipse button. Click for more options

![[Ellipse button]](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/three-point-personal-account.png)

2. Select ".." then **View my bill** to navigate to the **Cost Management + Billing - Invoices** page, which helps you analyze where Azure is generating costs.
3. On the left menu, go to **Cost Management**
4. On the same left menu but now under Cost Management, select **Cost Analysis**
5. Right above the graphs displayed click in **View**

![[Cost view]](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/cost-view.png)

6. From the drop-down menu select **Cost by Service**

Note: Provides cost details on each of your services. If your account is new or if you are using only free services. This page may be empty

1. Select the **X** in the top right corner to close the **Costs by service** pane.
2. Select the **X** in the top right corner to close the **Cost Management + Billing - Invoices** page and return to the home page.

5 Mini-lab: Create ARM templates by using the Azure Portal

In this mini-lab you will learn how to create, edit, and deploy an Azure Resource Manager template by using the Azure portal. This mini-lab shows you how to create an Azure Storage account, but you can use the same process to create other Azure resources.

5.1 Generate a template using the portal

Using the Azure portal, you can configure a resource, for example, an Azure Storage account. Before you deploy the resource, you can export your configuration into a Resource Manager template. You can save the template and reuse it in the future.

1. Sign into the Azure portal: <https://portal.azure.com/>.
2. Select **Create a resource**
3. In **Search the Marketplace**, write *Storage account* and select the option it displays
4. Once in the *Storage account* view, click **Create**
5. Enter the following information:
 - **Resource group:** Select **Create new** and specify a resource group name of your choice.
 - **Storage account name:** Give your storage account a unique name. The storage account name must be unique across all of Azure. If you get an error message saying, "The storage account name is already taken", try using `<your name>storage<Today's date in MMDD>`, for example, *jackstorage1016*.
 - You can use the default values for the rest of the properties.

Note: Some of the exported templates require some edits before you can deploy them.

6. Click **Review + create** at the bottom left of the screen.

Note: Do *not* select **Create** in the next step.

7. Select **Download a template for automation** at the bottom right of the screen. The portal shows the generated template:
 - The main pane shows the template. It is a JSON file with six top-level elements: `schema`, `contentVersion`, `parameters`, `variables`, `resources`, and `output`.
 - There are six **parameters** defined. One of them is called **storageAccountName**.

In the next section, edit the template to use a generated name for the storage account.

- In the template, one Azure **resource** is defined. The type is `Microsoft.Storage/storageAccounts`. Note how the resource is defined and the definition structure.
8. Select **Download** from the top of the screen (under "Template").
 9. Open the downloaded zip file, there are 2 files (**parameters.JSON** and **template.JSON**). Save both files to your computer.

In the next section, use a template deployment tool to edit the template.

10. Select the **Parameters** tab to get to the values you provided. Write down these values. You will need them in the next section when you deploy the template.

- Example:

![[Parameters Template]](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/template-parameters.png)

11. Go back to the home view pressing the **Microsoft Azure** label on the top left of the window

5.2 Edit and deploy the template

The Azure portal can be used to perform some basic template editing by using a portal tool called *Template Deployment*. To edit a more complex template, consider using Visual Studio Code which provides richer editing functionality.

Tip: Azure requires that each Azure service has a unique name. The deployment fails if you enter a storage account name that already exists. To avoid this issue, you can use the template function `uniqueString()` to generate a unique storage account name.

1. In the Azure portal, select **Create a resource**.
2. In **Search the Marketplace**, type **template deployment**, and select the option it displays. (**Template deployment (deploy using custom templates)**).
3. Select **Create**.
4. Select **Build your own template in the editor** to open the editor.
5. Select **Load file** from the menu below *Edit template*, and then select the *template.json* file you downloaded in the last section.
6. Make the following three changes to the template:
 - Remove the **storageAccountName** parameter from the **parameters** element.
 - Add one variable called **storageAccountName** as shown below to the **variables** element. The example below will generate a unique Storage Account name:
`"storageAccountName": "[concat(uniqueString(subscription().subscriptionId), 'storage')]"`
 - Update the name element of the **resources** (where the **Microsoft.Storage/storageAccounts** is located) to use the newly defined variable instead of the parameter:
`"name": "[variables('storageAccountName')]"`,
7. Select **Save**.
8. A form will appear
9. In the **BASICS** section of the form that appears, select the resource group you created in the last section.
10. In the **SETTINGS** section of the form, enter the values from the parameters you wrote down in Step 8 of the previous section. Here is a screenshot of a sample deployment:

![[Azure Resource Manager templates deployment with the fields filled in using sample information.]](/home/ll/Azure_clone/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/azure-resource-manager-template-tutorial-deploy.png)

11. Accept the terms and conditions and then select **Purchase**.
12. Select the bell icon (notifications) from the top of the screen to display the deployment status.

Wait until the deployment is completed.

13. Once the deployment is complete, select **Go to resource group** from the notification pane. The information should reveal that the deployment status was successful, and there is only one storage account in the resource group. The storage account name is a unique string generated by the template.

5.3 Clean up resources

When the Azure resources are no longer needed, clean up the resources you deployed by deleting the resource group.

1. From the options on top, click **Delete resource group**

![Options bar on top of resource group](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/delete-resource-group-option.png)

2. You'll be asked to **TYPE THE RESOURCE GROUP NAME** to confirm the action.

![Delete confirmation window](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/delete-confirmation.png)

3. Now you'll be able to click the button **Delete** on the bottom of the window
4. Wait for the notifications pane to indicate that the resource has been deleted
5. We are done!

6 Mini-lab: Create ARM templates by using Visual Studio Code

Azure Resource Manager (ARM) is a service for Azure that provides a management layer that enables you to create, update, and delete resources in your Azure account.

In this mini-lab, you will learn how to use Visual Studio Code and the Azure Resource Manager (ARM) Tools extension to create and edit Azure Resource Manager templates. You can create Resource Manager templates in Visual Studio Code without the extension, but the extension provides autocomplete options that simplify template development.

It's often easier and better to begin building your ARM template based on one of the existing Quickstart templates available on the [Azure Quickstart Templates](#) site.

This mini-lab is based on the [Create a standard storage account](#) template.

6.1 Prerequisites

You will need:

- Visual Studio Code. You can download a copy here: <https://code.visualstudio.com/>.
- Resource Manager Tools extension.

Follow these steps to install the Resource Manager Tools extension:

1. Open Visual Studio Code.
2. Press **CTRL+SHIFT+X** to open the Extensions pane.
3. Search for **Azure Resource Manager Tools**, and then select **Install**.
4. Select **Reload** to finish the extension installation.

6.2 Open the Quickstart template

1. Go to the following address and copy the contents of the file

<https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-storage-account-creation>

2. From Visual Studio Code, select **File > New File**.
3. Paste the previously copied code into the file
4. Select **File > Save As...** to save the file
5. Save the file as *azuredeploy.json* to your local computer.

6.3 Edit the template

Add one more element into the outputs section to show the storage URI.

1. Add the following code to the **outputs** property of the *azuredeploy.json*

```
"storageUri": {  
  "type": "string",  
  "value": "[reference(variables('storageAccountName')).primaryEndpoints.blob]"  
},
```

When you are done, the outputs section looks like:

```
"outputs": {
  "storageAccountName": {
    "type": "string",
    "value": "[variables('storageAccountName')]"
  },
  "storageUri": {
    "type": "string",
    "value": "[reference(variables('storageAccountName')).primaryEndpoints.blob]"
  }
}
```

TIP: Be careful that between `storageAccountName` and our just added `storageUri`, there is a colon (,)!

If you copied and pasted the code inside Visual Studio Code, try to retype the **value** element to experience the IntelliSense capability of the Resource Manager Tools extension.

![Resource Manager template visual studio code IntelliSense](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/manager-templates-visual-studio-code-intellisense.png)

2. Select **File>Save** to save the file.

6.4 Deploy the template

There are many methods for deploying templates, you will be using the Azure Cloud Shell.

1. Sign in to the [Azure Cloud shell](#).
 - If you are prompted with the message **"You have no storage mounted"** maintain the default selection and click **Create storage** (The creation of the storage may take a few seconds)
2. Wait for the terminal to complete and it displays: `*YourName*@Azure:~$`
3. Choose the **PowerShell** environment in the upper left corner.
4. Restarting the shell is required when you switch. Click on the **Confirm** button
5. Wait for the terminal to complete and it displays: `PS /home/*YourName*>`
6. Select the **Upload/download files** icon, and then select **Upload**.

![Image showing the location of the Upload file button in the interface](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/portal-cloud-shell-upload-file-powershell.png)

7. Select the file you saved in the previous section (**azuredeploy.json**).
8. The confirmation is displayed on the bottom right of the window
 - To confirm that your file has uploaded successfully, run the following command
9. From the Cloud shell, run the following commands.

```
$resourceGroupName = Read-Host -Prompt "Enter the Resource Group name"
```

- This command will ask you to enter a name for your resource group (for example "MyRS")

```
$location = Read-Host -Prompt "Enter the location (i.e. centralus)"
```

- This command will as you to enter the location for the resource group (for example westus, eastus or centralus)

```
New-AzResourceGroup -Name $resourceGroupName -Location "$location"
```

- This command will show you the configuration you just entered

```
New-AzResourceGroupDeployment -ResourceGroupName $resourceGroupName -TemplateFile "$HOME/azuredepl
```

- This command will create a new Azure `ResourceGroupDeployment` based on the values previously gave (this make take a few seconds)

Update the template file name if you save the file to a name other than **azuredeploy.json**.

- The following screenshot shows a sample deployment:

![Azure portal Cloud shell deploy template with the variables and commands from above highlighted](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/azure-portal-cloud-shell-deploy-template-powershell.png)

- The storage account name and the storage URL in the outputs section are highlighted on the screenshot.

6.5 Clean up resources

When the Azure resources are no longer needed, clean up the resources you deployed by deleting the resource group.

1. Go to your Azure account
2. On the search bar on top, write **Resource groups** and from the **services** select the one named "Resource groups"
3. Find the name of the resource group you just created
4. Click on the name
5. From the options on top, click **Delete resource group**

![Options bar on top of resource group](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/delete-resource-group-option.png)

6. You'll be asked to **TYPE THE RESOURCE GROUP NAME** to confirm the action (it's the name under **Home** on the top left of the window)

![Delete confirmation window](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 01/../../Linked_Image_Files/delete-confirmation.png)

7. Now you'll be able to click the button **Delete** on the bottom of the window
8. Wait for the notifications panel to indicate that the resource has been deleted
9. We are done!

7 Mini-lab: Create VMs inside an availability set

7.1 Prerequisites

Prior to this mini-lab we need to set the environment variable `AdminPassword` before hand. To do that:

1. Launch Cloud Shell from the top navigation of the Azure portal.

![Azure portal top navigation, with Cloud Shell icon highlighted](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 02/../../Linked_Image_Files/icon.png)

- If the icon is not displayed on the top menu bar on a narrower screen, select the ellipses (...) button.

![Ellipses button icon](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 02/../../Linked_Image_Files/three-points.png)

- If you are prompted with the message "**You have no storage mounted**" maintain the default selection and click **Create storage** (The creation of the storage may take a few seconds)

2. Once the Cloud Shell opens, check that the environment drop-down from the left-hand side of shell window says **Bash**.

![Environment drop-down, displaying Bash.](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 02/../../Linked_Image_Files/select_Bash_env)

- If it does not say "Bash", click and select **Bash**. Confirm

3. Once the bash is ready, enter the following command using your own password

```
AdminPassword="myStr0ngPW%%"
```

It is recommended that you change *myStr0ngPW%%* to a secret value of your preference, the password length must be between 12 and 72 characters, and have 1 lower case character, 1 upper case character, 1 number and 1 special character.

4. Confirm that you have set your environment variable with the following command:

```
echo $AdminPassword
```

7.2 Create an availability set

1. To create a resource group, we are going to run the following command:

```
az group create --name myResourceGroup --location eastus
```

2. Create a managed availability set by running the following command:

```
az vm availability-set create --resource-group myResourceGroup --name myAvailabilitySet --platform-fault-domain-count 2 --platform-update-domain-count 2
```

- This may take a few seconds

7.3 Create VMs inside an availability set

VMs must be created within the availability set to make sure that they are correctly distributed across the hardware. You can't add an existing VM to an availability set after it's created.

When you create a VM with `az vm create`, you use the `--availability-set` parameter to specify the name of the availability set.

1. Create two virtual machines by running the following command:

```
for i in `seq 1 2`; do
az vm create \
  --resource-group myResourceGroup \
  --name myVM$i \
  --availability-set myAvailabilitySet \
  --vnet-name MyVnet --subnet subnet1 \
  --image debian \
  --admin-password $AdminPassword \
  --admin-username azureuser \
  --no-wait
done
```

2. It takes a few minutes to create and configure both VMs. When it finishes, you will have two virtual machines distributed across the underlying hardware.
3. On the search bar at the top of the window, write **Resource Groups** and select the service of that name
4. Once on the resource groups view, select **myResourceGroup**
5. From the resources contained by **myResourceGroup**. Select **myAvailabilitySet**
6. Observe how the VMs are distributed across the two fault and update domains.

![Azure portal UI, showing the new availability set.](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 02/../../Linked_Image_Files/myRes)

7.4 Clean up deployment

1. Run the following command to remove the resource group, VM, and all related resources: `az group delete --name myResourceGroup --yes`

- This may take a while, wait for the console to finish

8 Mini-lab: Secure user sign-in events with Azure Multi-Factor Authentication

Multi-factor authentication (MFA) is a process where a user is prompted during a sign-in event for additional forms of identification. The user might be prompted to enter a code on their cellphone or to provide a fingerprint scan. Requiring a second form of authentication increases security because the additional factor is not easy for an attacker to obtain or duplicate.

Azure Multi-Factor Authentication and Conditional Access policies give the flexibility to enable MFA for users during specific sign-in events.

Prerequisites

It is expected that this lab will be run as an **instructor demo** since in order to perform this mini-lab you need the following resources and privileges:

- A working Azure AD tenant with Azure AD Premium or trial license enabled.
- An account with global administrator privileges.
- A non-administrator user with a password you know, such as testuser. You will test the end-user Azure Multi-Factor Authentication experience using this account in this mini-lab.
- A group that the non-administrator user is a member of, such as MFA-Test-Group. You enable Azure Multi-Factor Authentication for this group in this mini-lab.

8.1 Create a Conditional Access Policy

The recommended way to enable and use Azure Multi-Factor Authentication is with Conditional Access policies. Conditional Access lets you create and define policies that react to sign-in events and request additional actions before a user is granted access to an application or service.

![Overview diagram of how Conditional Access works to secure the sign-in process](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 03/../../Linked_Image_Files/demo)

In this mini-lab you will create a basic Conditional Access policy to prompt for MFA when a user signs in to the Azure portal.

First, create a Conditional Access policy and assign your test group of users as follows:

1. Sign in to the [Azure portal](#) using an account with global administrator permissions.
2. Search for and select **Azure Active Directory**, then choose **Security** on the left menu
3. Select **Conditional Access**, also on the left menu, and then choose **+ New policy**.
4. Enter a name for the policy, such as *MFA Pilot*.
5. Under **Assignments**, choose **Users and groups**, then **Select users and groups**.
6. Check the box for **Users and groups**, then **Select**.
7. Browse for and select your Azure AD group, such as *MFA-Test-Group*, then choose **Select**.

[Picture 3](#)

8. To apply the Conditional Access policy for the group, select **Done**.

8.2 Configure the conditions for Multi-Factor Authentication

With the Conditional Access policy created and a test group of users assigned, you can now define the cloud apps or actions that trigger the policy. These cloud apps or actions are the scenarios you decide require additional processing, such as to prompt for MFA.

Configure the Conditional Access policy to require MFA when a user signs in to the Azure portal.

1. Select **Cloud apps or actions**. You can choose to apply the Conditional Access policy to *All cloud apps* or *Select apps*.

2. On the **Include** page, choose **Select apps**.
3. Choose **Select**, then browse the list of available sign-in events that can be used.
4. Choose **Microsoft Azure Management** so the policy applies to sign-in events to the Azure portal.
5. To apply the select apps, choose **Select**, then **Done**.

![Select the Microsoft Azure Management app to include in the Conditional Access policy](/home/ll/Azure_clone/Azure_030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 03/../../Linked_Image_Files/)

6. Access controls let you define the requirements for a user to be granted access, such as needing an approved client app or using a device that's Hybrid Azure AD joined. Configure the access controls to require MFA during a sign-in event to the Azure portal.
7. Under **Access controls**, choose **Grant**, then make sure that **Grant access** is selected.
8. Check the box for **Require multi-factor authentication**, then choose **Select**.

Conditional Access policies can be set to Report-only if you want to observe how the configuration would impact users, or set to Off if you don't want to use the policy right now. Because a test group of users was targeted for this demonstration, let's enable the policy and then test Azure Multi-Factor Authentication.

9. Set the *Enable policy* to **On**.
10. To apply the *Conditional Access policy*, select **Create**.

8.3 Test Azure Multi-Factor Authentication

To view the Conditional Access policy and Azure Multi-Factor Authentication, sign in to a resource that doesn't require MFA as follows:

1. Open a new browser window in InPrivate or incognito mode and browse to <https://account.activedirectory.windowsazure.com/>
2. Sign in with your non-administrator test user, such as testuser. There's no prompt for you to complete MFA.
3. Close the browser window.

Now sign in to the Azure portal. Because the Azure portal was configured in the Conditional Access policy to require additional verification, you will get an Azure Multi-Factor Authentication prompt.

4. Open a new browser window in InPrivate or incognito mode and browse to <https://portal.azure.com>.
5. Sign in with your non-administrator test user, such as testuser. You're required to register for and use Azure Multi-Factor Authentication. Follow the prompts to complete the process and verify that you have successfully signed in to the Azure portal.

![Follow the browser prompts and then on your registered multi-factor authentication prompt to sign in](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 03/../../Linked_Image_Files/demo_conditional_access_image4.png)

9 Mini-lab: Add Guest users to Azure AD in the Azure Portal

You can invite anyone to collaborate with your organization by adding them to your directory as a guest user. Then you can either send an invitation email that contains a redemption link or send a direct link to an app you want to share.

Guest users can sign in with their own work, school, or social identities.

In this mini-lab, you'll add a new guest user to Azure AD and send an invitation.

9.1 Prerequisites

To complete the scenario in this mini-lab you need:

- A role that allows you to create users in your tenant directory, like the Global Administrator role or any of the limited administrator directory roles.
- A valid email account that you can add to your tenant directory, and that you can use to receive the test invitation email.

9.2 Add Guest users to Azure AD

1. Sign in to the [Azure portal](#) as an Azure AD administrator.
2. In the left pane, select **Azure Active Directory**.
3. On the **Manage** left panel on the left, select **Users**.

![Screenshot showing where to select the Users option](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 03/../../Linked_Image_Files/guest

4. Select **New guest user** on the top bar

[Screenshot showing where to select the New guest user option](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 03/../../Linked_Image_Files/)

5. On the **New user** page, select **Invite user** and add the guest user's information.

![Screenshot showing where to select the New guest user option](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 03/../../Linked_Image_Files/)

- *Identity*

- **Name:** The first and last name of the guest user.
- **Email address (required):** The email address of the guest user.

- *Personal message (optional)*: Include a personal welcome message to the guest user.

- *Groups and roles*

- **Groups:** You can add the guest user to one or more existing groups, or you can do it later.
- **Directory role:** If you require Azure AD administrative permissions for the user, you can add them to an Azure AD role.

- *Settings*

- **Block sign in (optional):** Allows you to block an user to sign-in without deleting the profile
- **Usage location (optional):** The physical location of the user

- *Job info*

- Job title (optional)
- Department (optional)

6. Select **Invite** to automatically send the invitation to the guest user. A notification appears in the upper right with the message Successfully invited user.
7. After you send the invitation, the user account is automatically added to the directory as a guest.

9.3 Assign an App to a Guest user

Add the *Active Directory for GitHub Enterprise* app to your test tenant and assign the test guest user to the app.

1. Sign in to the Azure portal as an Azure AD administrator.
2. In the left pane, select **Enterprise applications**.

- Alternatively, you can search for it on the search box on top of the page

3. Select **New application**.

! [Screenshot showing the Add from the gallery search box](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 03/../../Linked_Image_Files/)

4. Under **Add from the gallery**, search for **GitHub**, and then select **Active Directory for GitHub Enterprise**.

[Screenshot showing the Add from the gallery search box](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 03/../../Linked_Image_Files/)

5. Select **Add** (or Create). It will show you a new window

![Screenshot showing the Add from the gallery search box](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 03/../../Linked_Image_Files/)

6. Under the Manage left pane, select Single **sign-on**, and under **Single Sign-on Mode**, select **Password-based Sign-on**, and click **Save**.

7. Under the **Manage** left pane, select **Users and groups > Add user**

8. Click **Users**, a right pane will appear, click the user you want to add, and then Users click **Select**.

![Screenshot showing Add user to group.](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 03/../../Linked_Image_Files/guest_user_ima)

9. Select **Assign**.

9.4 Accept the Guest user invite

Now sign in to the guest user email account to view the invitation.

1. Sign in to your test guest user's email account.
2. In the inbox, find the "You're invited" email.
3. In the email body, select Get Started. A Review permissions page opens in the browser.
4. Select **Accept Invitation**. The Access Panel opens, which lists the applications the guest user can access.

![Screenshot showing the Review permissions page](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 03/../../Linked_Image_Files/guest_)

10 Mini-lab: Azure Active Directory Seamless Single Sign-On

Azure Active Directory (Azure AD) Seamless Single Sign-On (Seamless SSO) automatically signs in users when they are on their corporate desktops that are connected to your corporate network. Seamless SSO provides your users with easy access to your cloud-based applications without needing any additional on-premises components.

10.1 Prerequisites

It is expected that this lab will be run as an **instructor demo** since the following prerequisites need to be in place prior to this mini-lab:

- **Set up your Azure AD Connect server:** If you use Pass-through Authentication as your sign-in method, no additional prerequisite check is required. If you use password hash synchronization as your sign-in method, and if there is a firewall between Azure AD Connect and Azure AD, ensure that:
 - You use version 1.1.644.0 or later of Azure AD Connect.
 - If your firewall or proxy permits DNS allow lists, allow the connections to the *.msapproxy.net URLs over port 443.
- **Set up domain administrator credentials:** You need to have domain administrator credentials for each Active Directory forest that:
 - You synchronize to Azure AD through Azure AD Connect.
 - Contains users you want to enable for Seamless SSO.

10.2 Enable Azure AD Connect

1. Enable Seamless SSO through [Azure AD Connect](#).

- If you're doing a fresh installation of Azure AD Connect, choose the [custom installation path](#). On the **User sign-in** page, check the **Enable single sign on option**.

![Azure AD Connect: User sign-in](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 04/../../Linked_Image_Files/SSO_demo_imag)

- If you already have an installation of Azure AD Connect, select the **Change user sign-in** page in Azure AD Connect, and then select **Next**.
 ![Azure AD Connect: Change the user sign-in](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 04/../../Linked_Image_Files/S)
- 2. Continue through the wizard until you get to the **Enable single sign on** page. Provide domain administrator credentials for each Active Directory forest that:
 - You synchronize to Azure AD through Azure AD Connect.
 - Contains users you want to enable for Seamless SSO.
- 3. After completion of the wizard, Seamless SSO is enabled on your tenant.

10.3 Verify Seamless SSO is enabled

Follow procedure below to verify that you have enabled Seamless SSO correctly:

1. Sign in to the [Azure Active Directory administrative center](#) with the global administrator credentials for your tenant.
2. Select **Azure Active Directory** in the left pane.
3. Select **Azure AD Connect**.
4. Verify that the Seamless single sign-on feature appears as *Enabled*.

![Azure portal: Azure AD Connect pane](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 04/../../Linked_Image_Files/SSO_demo_ima)
 Important

Seamless SSO creates a computer account named AZUREADSSOACC in your on-premises Active Directory (AD) in each AD forest. The AZUREADSSOACC computer account needs to be strongly protected for security reasons. Only Domain Admins should be able to manage the computer account. Ensure that Kerberos delegation on the computer account is disabled and that no other account in Active Directory has delegation permissions on the AZUREADSSOACC computer account. Store the computer account in an Organization Unit (OU) where they are safe from accidental deletions and where only Domain Admins have access.

11 Mini-lab: Enable Azure AD Self-Service Password Reset

Azure Active Directory (Azure AD) self-service password reset (SSPR) gives users the ability to change or reset their password with no administrator or help desk involvement. If a user's account is locked or they forget their password, they can follow prompts to unblock themselves and get back to work. This ability reduces help desk calls and loss of productivity when a user can't sign in to their device or an application.

This topic demonstrates how to enable self-service password reset.

- Enable self-service password reset for a group of Azure AD users
- Configure authentication methods and registration options
- Test the SSPR process as a user

11.1 Prerequisites

It is expected that this lab will be run as an **instructor demo** since to complete this mini-lab, you need the following resources and privileges:

- A working Azure AD tenant with at least a trial license enabled.
- An account with Global Administrator privileges.
- A non-administrator user with a password you know, such as testuser. You test the end-user SSPR experience using this account in this mini-lab.
- A group that the non-administrator user is a member of, such as SSPR-Test-Group. You enable SSPR for this group in this mini-lab.

11.2 Enable self-service password reset

Azure AD lets you enable SSPR for **None**, **Selected**, or **All users**. This granular ability lets you choose a subset of users to test the SSPR registration process and workflow. When you're comfortable with the process and can communicate the requirements with a broader set of users, you can select additional groups of users to enable for SSPR. Or, you can enable SSPR for everyone in the Azure AD tenant.

In this mini-lab, configure SSPR for a set of users in a test group. In the following example, we will use the "SSPR-Test-Group" group. Provide your Azure AD group as needed:

1. Sign in to the [Azure portal](#) using an account with global administrator permissions.
2. Search for and select **Azure Active Directory**, then choose **Password reset** from the menu on the left-hand side.
3. From the **Properties** page, under the option **Self service** password reset enabled, choose **Select group**.
4. Search and select your Azure AD group, such as *SSPR-Test-Group*, then choose **Select**.

Enable self-service password reset

Nested groups are supported as part of a wider deployment of SSPR. Make sure that the users in the group(s) you choose have the appropriate licenses assigned. Currently, there is no validation process for these licensing requirements.

5. To enable SSPR for the selected users, select **Save**.

11.3 Select authentication methods and registration options

When users need to unlock their accounts or reset their password, they're prompted for an additional confirmation method. This additional authentication factor makes sure that only approved SSPR events are completed.

You can choose which authentication methods to allow based on the registration information the user provides.

1. On the **Authentication methods** page from the menu on the left-hand side, set the **Number of methods required to reset** to *1*.

To improve security, you can increase the number of authentication methods required for SSPR.

2. Choose the **Methods available to users** that your organization wants to allow. For this mini-lab, check the boxes to enable the following methods:
 - *Mobile app notification*
 - *Mobile app code*
 - *Email*
 - *Mobile phone*
 - *Office phone*
3. To apply the authentication methods, select **Save**.

11.4 Configure prompt for registration when users next sign in

Before users can unlock their account or reset a password, they must register their contact information. This contact information is used for the different authentication methods configured in the previous steps.

An administrator can manually provide this contact information, or users can go to a registration portal to provide the information themselves. In this mini-lab, configure the users to be prompted for registration when they next sign in.

1. On the **Registration** page from the menu on the left-hand side, select **Yes** for **Require users to register when signing in**.
2. It's important that contact information is kept up to date. If the contact information is outdated when an SSPR event is started, the user may not be able to unlock their account or reset their password.
3. Set **Number of days before users are asked to reconfirm their authentication information** to *180*.

4. To apply the registration settings, select **Save**.

11.5 Configure notifications and customizations

To keep users informed about account activity, you can configure e-mail notifications to be sent when an SSPR event happens. These notifications can cover both regular user accounts and admin accounts. For admin accounts, this notification provides an additional layer of awareness when a privileged administrator account password is reset using SSPR.

1. On the **Notifications** page from the menu on the left-hand side, configure the following options:
 - Set **Notify users on password resets option** to *Yes*.
 - Set **Notify all admins when other admins reset their password** to *Yes*.
2. To apply the notification preferences, select **Save**.

11.6 Test Self-Service Password Reset

With SSPR enabled and configured, test the SSPR process with a user that's part of the group you selected in the previous section, such as Test-SSPR-Group. In the following example, the testuser account is used. Provide your user account that's part of the group you enabled for SSPR in the first section of this mini-lab.

Note When you test the self-service password reset, use a non-administrator account. Admins are always enabled for self-service password reset and are required to use two authentication methods to reset their password.

1. For the manual registration process, open a new browser window in InPrivate or incognito mode, and browse to <https://aka.ms/ssprsetup>. Users should be directed to this registration portal when they next sign in.
2. Sign in with a non-administrator test user, such as **testuser**, and register your authentication method's contact information.
3. Once complete, select the button marked **Looks good** and close the browser window.
4. Open a new browser window in InPrivate or incognito mode, and browse to <https://aka.ms/sspr>.
5. Enter your non-administrator test user's account information, such as **testuser**, the characters from the CAPTCHA, and then select **Next**.

![Enter user account information to reset the password](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 04/../../Linked_Image_Files/how_t

6. Follow the verification steps to reset your password. When complete, you should receive an e-mail notification that your password was reset.

12 Mini-lab: VNet Peering

Note: For this mini-lab you will need two virtual networks.

12.1 Create virtual networks

1. Sign in to the Azure portal at <https://portal.azure.com>
2. Search "Virtual network" on the searchbox on top of the page and click on the resource of that name. A new view will appear
3. On this new view, click the + **Add** button
4. Select your subscription and the resource group on which you'd like your virtual networks to reside
 - If you don't have a resource group, click on "Create new" and give it any name you like
5. Give your virtual network a name (for example: "virtualnetwork1"), and select any region you like
6. Click on **Review + Create**
7. Wait a few seconds

8. Click on **Create**
9. Repeat this process for `virtualnetwork2` in order to have 2 virtual networks available

12.2 Configure VNet peering on the first virtual network

1. Select the first virtual network.
2. On the search bar right under the virtual network's name, search for "**Peerings**" and select the "**Peerings**" option.
3. Select **+ Add**. This will take you to a new view
 - Provide a **name** for the first virtual network peering. For example, VNet1toVNet2.
 - In the **Virtual network** drop-down, select the second virtual network you would like to peer with.
 - Note the region, this will be needed when you configure the VPN gateway.
 - Provide a name for the second virtual network peering. For example, VNet2toVNet1.
 - Use the informational icons to review the network access, forwarded traffic, and gateway transit settings.
 - Check the box for **Allow gateway transit**. Note the error that the virtual network does not have a gateway.
 - Make sure the **Allow gateway transit** check box is not selected.
 - Click **OK** to save your settings.

12.3 Configure a VPN gateway

1. In the **Azure portal**, search for **virtual network gateways**.
2. Select **+ Add**.
 - Provide a **name** for your virtual network gateway. For example, VNet1Gateway.
 - Ensure the gateway is in the same region as the first virtual network.
 - In the **virtual network** drop-down select the first virtual network.
 - In the **Public IP address** area, **Create new** and give the IP address a name.
 - Click **Create and review**. Address any validation errors.
 - Click **Create**.
3. Wait a few moments until the gateway is successfully created (a notification is displayed)

12.3.1 Allow gateway transit

1. In the **Azure portal**, return to your first virtual network.
2. On the **Overview** blade, notice the new **Connected device** for your VPN gateway.
3. Select the gateway and notice you can perform a health check and review access statistics.
4. Return to the previous page (clicking on the **X** on the top right) and on the search bar right under the virtual network's name, write "**Peerings**" again and click on the option that appears.
 - Select the peering
 - Enable **Allow gateway transit**. Notice the previous error has been resolved.
 - Notice after making this selection, **Use remote gateways** is disabled.
5. **Save** your changes (with "enable gateway transit" checkbox checked)

12.4 Confirm VNet peering on the second virtual network

1. In the **Azure portal**, select the second virtual network.
2. On the search bar right under the virtual network's name, write "**Peerings**" and click on the option that appears.
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. Click the peering.
 - Notice that ****use remote gateways**** cannot be selected.
 - Use the informational icon to review the **Use remote gateways** setting.
6. **Discard** your changes.

13 Mini-lab: Create a VM Scale Set in the Azure Portal

13.1 Create public load balancer

Create a public load balancer using the portal. The name and public IP address you create are automatically configured as the load balancer's front end.

1. In the search box, type *load balancer*. Under **Marketplace** in the search results, pick **Load balancer**.
2. In the **Basics** tab of the Create load balancer page, enter or select the following information:

Setting	Value
Subscription	Select your subscription.
Resource group	Select Create new and type myVMSSResourceGroup in the text box.
Name	myLoadBalancer
Region	Select East US.
Type	Select Public.
SKU	Select Standard.
Public IP address	Select Create new.
Public IP address name	MyPip
Assignment	Static
Availability zone	Zone-redundant

3. Select **Review + create**.
4. Select **Create**.

![Create a load balancer](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 06/../../Linked_Image_Files/create_a_scale_set_image1.png)

13.2 Create virtual machine scale set

You can deploy a scale set with a Windows Server image or Linux image such as RHEL, CentOS, Ubuntu, or SLES.

1. Type *Scale set* in the search box. In the results, under **Marketplace**, select **Virtual machine scale sets**. The **Create a virtual machine scale set** page will open.
2. In the **Basics** tab, under **Project details**, make sure the correct subscription is selected and then choose **Create new** resource group. Type *myVMSSResourceGroup* for the name and then select OK .
3. Type *myScaleSet* as the name for your scale set.
4. In **Region**, select a region that is close to your area.
5. Leave the default value of **ScaleSet VMs for Orchestrator**.
6. Select a marketplace image for **Image**. In this example, we have chosen Ubuntu Server 18.04 LTS.

7. Enter your desired username and select which authentication type you prefer.

- A **Password** must be at least 12 characters long and meet three out of the four following complexity requirements: one lower case character, one upper case character, one number, and one special character.
- If you select a Linux OS disk image, you can instead choose **SSH public key**. Only provide your public key, such as `~/.ssh/id_rsa.pub`. You can use the Azure Cloud Shell from the portal to create and use SSH keys.

![[Create a virtual machine scale set]](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 06/../../Linked_Image_Files/create_a_scale_set_in

8. Select **Next** to move to the other pages. As you explore each tab, review the attributes of the VMs in the scale set being created.
9. Review the **Instance** and **Disks** tabs, but leave the default values in place.
10. On the **Networking** page, under **Load balancing**, select **Yes** to put the scale set instances behind a load balancer.
11. In **Load balancing options**, select **Azure load balancer**.
12. In **Select a load balancer**, select **myLoadBalancer** that you created earlier.
13. For **Select a backend pool**, select **Create new**, type *myBackendPool*, and select **Create**.
14. Inspect the **Scaling** tab and the **Management** tab default values
15. Select **Review + create**.
16. Select **Create** to deploy the scale set.

13.3 Clean up resources

When no longer needed, delete the resource group, scale set, and all related resources. To do so, select the resource group for the scale set and then select **Delete**.

14 Mini-lab: Create a Load Balancer to Load Balance VMs

Sign in to the Azure portal at <https://portal.azure.com>.

14.1 Create a Load Balancer

In this mini-lab, you create a Load Balancer that helps load balance virtual machines. You can create a public Load Balancer or an internal Load Balancer. When you create a public Load Balancer, you must also create a new public IP address that is configured as the frontend (named as *LoadBalancerFrontend* by default) for the Load Balancer.

1. Select **+ Create a resource**, type *load balancer*.

![[Create a Standard Load Balancer]](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 07/../../Linked_Image_Files/create-standard-load-balancer.png).

2. Select **Create**.

![[Create a Standard Load Balancer, select Create.]](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 07/../../Linked_Image_Files/create-load-balancer-start.png).

3. In the **Basics** tab of the **Create load balancer** page, enter or select the following information, accept the defaults for the remaining settings, and then select **Review + create**:

![[Create a Standard Load Balancer configuration.]](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 07/../../Linked_Image_Files/load_balancer_3.png)

Setting	Value
Subscription	Select your subscription.
Resource group	Select Create new and type <i>myResourceGroupSLB</i> in the text box.
Name	<i>myLoadBalancer</i>
Region	Select West Europe .
Type	Select Public .
SKU	Select Standard or Basic . Microsoft recommends Standard for production workloads.
Public IP address	Select Create new . If you have an existing Public IP you would like to use, select Use existing .
Public IP address name	Type <i>myPublicIP</i> in the text box. Use -SKU Basic to create a Basic Public IP. Basic Public IPs are not available in all regions.
Availability zone	Type <i>Zone-redundant</i> to create a resilient Load Balancer. To create a zonal Load Balancer, select a zone.

Important This mini-lab assumes that Standard SKU is chosen during the SKU selection process above.

14.2 Create Load Balancer resources

In this section, you will configure Load Balancer settings for a backend address pool, a health probe, and specify a balancer rule.

14.3 Create a Backend pool

To distribute traffic to the VMs, a backend address pool contains the IP addresses of the virtual (NICs) connected to the Load Balancer. Create the backend address pool *myBackendPool* to include virtual machines for load-balancing internet traffic.

1. Select **All services** from the left-hand menu, select **All resources**, and then select **myLoadBalancer** from the resources list.
2. Under **Settings**, select **Backend pools**, then select **Add**.
3. On the **Add a backend pool** page, for name, type *myBackendPool* as the name for your backend pool, then select **Add**.

14.4 Create a health probe

To allow the Load Balancer to monitor the status of your app, you use a health probe. The health probe dynamically adds or removes VMs from the Load Balancer rotation based on their response to health checks. Create a health probe *myHealthProbe* to monitor the health of the VMs.

1. Select **All services** from the left-hand menu, select **All resources**, then select **myLoadBalancer** from the resources list.
2. Under **Settings**, select **Health probes**, then select **Add**.

Setting	Value
Name	Enter <i>myHealthProbe</i> .
Protocol	Select HTTP .
Port	Enter <i>80</i> .
Interval	Enter <i>15</i> for number of Interval in seconds between probe attempts.
Unhealthy threshold	Select 2 for number of Unhealthy threshold or consecutive probe failures that must occur before

3. Select **OK**.

14.5 Create a Load Balancer rule

A Load Balancer rule is used to define how traffic is distributed to the VMs. You define the frontend IP configuration for the incoming traffic and the backend IP pool to receive the traffic, along with the required source and destination port. Create a Load Balancer rule *myLoadBalancerRuleWeb* for listening to port 80 in the frontend *FrontendLoadBalancer* and sending load-balanced network traffic to the backend address pool *myBackendPool*, also using port 80.

1. Select **All services** in the left-hand menu, select **All resources**, then select **myLoadBalancer** from the

resources list.

2. Under **Settings**, select **Load balancing rules**, then select **Add**.

3. Use these values to configure the load balancing rule:

Setting	Value
Name	Enter <i>myHTTPRule</i> .
Protocol	Select TCP .
Port	Enter <i>80</i> .
Backend port	Enter <i>80</i> .
Backend pool	Select myBackendPool .
Health probe	Select myHealthProbe .

4. Leave the rest of the defaults and then select **OK**.

14.6 Create backend servers

In this section, you will create a virtual network, create three virtual machines for the backend pool of the Load Balancer, and then install IIS on the virtual machines to help test the Load Balancer.

14.7 Virtual network and parameters

In this section you'll need to replace the following parameters in the steps with the information below:

Parameter	Value
resource-group-name	myResourceGroupSLB
virtual-network-name	myVNet
region-name	West Europe
IPv4-address-space	10.1.0.0\16
subnet-name	myBackendSubnet
subnet-address-range	10.1.0.0\24

14.8 Create the virtual network

In this section, you will create a virtual network and subnet.

1. On the upper-left side of the screen, select **Create a resource > Networking > Virtual network** or search for *Virtual network* in the search box.
2. In **Create virtual network**, enter or select this information on the **Basics** tab:

Setting	Value
Project Details	
Subscription	Select your Azure subscription
Resource Group	Select Create new , enter resource-group-name , then select OK , or select an existing resource-g
Instance details **	
Name	Enter virtual-network-name
Region	Select region-name

3. Select the **IP Addresses** tab or select the **Next: IP Addresses** button at the bottom of the page.
4. In the **IP Addresses** tab, enter this information:

Setting	Value
IPv4 address space	Enter IPv4-address-space

5. Under **Subnet name**, select the word **default**.
6. In **Edit subnet**, enter this information:

Setting	Value
Subnet name	Enter subnet-name
Subnet address range	Enter subnet-address-range

7. Select **Save**.
8. Select the **Review + create** tab or select the **Review + create** button.
9. Select **Create**.

14.9 Create virtual machines

Public IP SKUs and Load Balancer SKUs must match. For Standard Load Balancer, use VMs with Standard IP addresses in the backend pool. In this section, you will create three VMs (*myVM1*, *myVM2* and *myVM3*) with a Standard public IP address in three different zones (*Zone 1*, *Zone 2*, and *Zone 3*) that are later added to the backend pool of the Load Balancer that was created earlier. If you selected Basic, use VMs with Basic IP addresses.

1. On the upper-left side of the portal, select **Create a resource > Compute > Windows Server 2019 Datacenter**.
2. In **Create a virtual machine**, type or select the following values in the **Basics** tab:
 - **Subscription > Resource Group**: Select **myResourceGroupSLB**.
 - **Instance Details > Virtual machine name**: Type *myVM1*.
 - **Instance Details > Region** > Select **West Europe**.
 - **Instance Details > Availability Options** > Select **Availability zones**.
 - **Instance Details > Availability zone** > Select **1**.
 - **Administrator account** > Enter the **Username**, **Password** and **Confirm password** information.
 - Select the **Networking** tab, or select **Next: Disks**, then **Next: Networking**.
3. On the **Networking** tab, make sure the following are selected:
 - **Virtual network**: *myVnet*
 - **Subnet**: *myBackendSubnet*
 - **Public IP** > Select **Create new**. In the **Create public IP address** window, for **SKU** select **Standard**, and for **Availability zone** select **Zone-redundant**, and then select **OK**. If you created a Basic Load Balancer, select **Basic**. Microsoft recommends using Standard SKU for production workloads.
 - To create a new network security group (NSG), a type of firewall, under **Network Security Group**, select **Advanced**.
1. In the **Configure network security group** field, select **Create new**.
2. Type *myNetworkSecurityGroup* and select **OK**.
 - To make the VM a part of the Load Balancer's backend pool, complete the following steps:
 - In **Load Balancing**, for **Place this virtual machine behind an existing load balancing solution?**, select **Yes**.
 - In **Load balancing settings**, for **Load balancing options**, select **Azure load balancer**.
 - For **Select a load balancer**, *myLoadBalancer*.
 - Select the **Management** tab, or select **Next > Management**.
4. In the **Management** tab, under **Monitoring**, set **Boot diagnostics** to **Off**.
5. Select **Review + create**.
6. Review the settings, then select **Create**.
7. Follow steps 2 through 6 to create two additional VMs with the following values and all the other settings the same as *myVM1*:

Setting	VM 2	VM 3
Name	<i>myVM2</i>	<i>myVM3</i>
Availability zone	2	3
Public IP	Standard SKU	Standard SKU
Public IP - Availability zone	Zone redundant	Zone redundant
Network security group	Select the existing <i>myNetworkSecurity Group</i>	Select the existing <i>myNetworkSecurity Group</i>

14.10 Create NSG rule

In this section, you create a network security group rule to allow inbound connections using HTTP.

1. Select **All services** in the left menu. Select **All resources**, then from the resources list select **myNetworkSecurityGroup** which is located in the **myResourceGroupSLB** resource group.
2. Under **Settings**, select **Inbound security rules**, then select **Add**.
3. Enter these values for the inbound security rule named **myHTTPRule** to allow for an inbound HTTP connections using port **80**:
 - **Source:** *Service Tag*
 - **Source service tag:** *Internet*
 - **Destination port ranges:** *80*
 - **Protocol:** *TCP*
 - **Action:** *Allow*
 - **Priority:** *100*
 - **Name:** *myHTTPRule*
 - **Description:** *Allow HTTP*
4. Select **Add**.
5. Repeat the steps for the inbound RDP rule, if needed, with the following differing values:
 - **Destination port ranges:** *Type 3389.*
 - **Priority:** *Type 200.*
 - **Name:** *Type MyRDPRule.*
 - **Description:** *Type Allow RDP.*

14.11 Install IIS

1. Select **All services** in the left menu, select **All resources**, then from the resources list select **myVM1** which is located in the *myResourceGroupSLB* resource group.
2. On the **Overview** page, select **Connect** to RDP into the VM.
3. Log into the VM with the credentials that you provided during the creation of this VM. This launches a remote desktop session with virtual machine - *myVM1*.
4. On the server desktop, navigate to **Windows Administrative Tools>Windows PowerShell**.
5. In the PowerShell Window, run the following commands to install the IIS server, remove the default iisstart.htm file, and add a new iisstart.htm file that displays the name of the VM:

```
# install IIS server role
```

```
Install-WindowsFeature -name Web-Server -IncludeManagementTools
```

```
# remove default htm file
```

```
remove-item C:\inetpub\wwwroot\iisstart.htm
```

```
# Add a new htm file that displays server name
```

```
Add-Content -Path "C:\inetpub\wwwroot\iisstart.htm" -Value $("Hello World from " + $env:computername)
```

6. Close the RDP session with *myVM1*.
7. Repeat steps 1 through 6 to install IIS and the updated iisstart.htm file on *myVM2* and *myVM3*.

14.12 Test the Load Balancer

1. Find the public IP address for the Load Balancer on the **Overview** screen. Select **All services** in the left-hand menu, select **All resources**, and then select **myPublicIP**.
2. Copy the public IP address, and then paste it into the address bar of your browser. The default page of IIS Web server is displayed on the browser.

![IIS Web server](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 07/../../Linked_Image_Files/load-balancer-test.png)

The Load Balancer distributes traffic across all three VMs, you can customize the default page of each VM's IIS Web server and then force-refresh your web browser from the client machine.

15 Mini-lab: Run Azure Container Instances

In this lab you create a container in Azure and expose it to the Internet with a fully qualified domain name (FQDN).

Azure Container Instances is useful for scenarios that can operate in isolated containers, including simple applications, task automation, and build jobs. Here are some of the benefits:

- **Fast startup:** Launch containers in seconds.
- **Per second billing:** Incur costs only while the container is running.
- **Hypervisor-level security:** Isolate your application as completely as it would be in a VM.
- **Custom sizes:** Specify exact values for CPU cores and memory.
- **Persistent storage:** Mount Azure Files shares directly to a container to retrieve and persist state.
- **Linux and Windows:** Schedule both Windows and Linux containers using the same API.

For scenarios where you need full container orchestration, including service discovery across multiple containers, automatic scaling, and coordinated application upgrades, we recommend Azure Kubernetes Service (AKS).

15.1 Create a container

1. Sign in to the Azure portal at <https://portal.azure.com> with your Azure subscription.
2. Open the Azure Cloud Shell from the Azure portal using the Cloud Shell icon.

![Picture 7](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 08/../../Linked_Image_Files/demo_Azure_containers_image1.png)

3. Create a new resource group with the name **learn-deploy-aci-rg** so that it will be easier to clean up these resources when you are finished with the module. If you choose a different resource group name, remember it for the rest of the exercises in this module. You also need to choose a region in which you want to create the resource group, for example **East US**.

```
az group create --name learn-deploy-aci-rg --location eastus
```

You create a container by providing a name, a Docker image, and an Azure resource group to the **az container create** command. You can optionally expose the container to the Internet by specifying a DNS name label. In this example, you deploy a container that hosts a small web app. You can also select the location to place the image - you'll use the **East US** region, but you can change it to a location close to you.

4. You provide a DNS name to expose your container to the Internet. Your DNS name must be unique. For learning purposes, run this command from Cloud Shell to create a Bash variable that holds a unique name.

```
DNS_NAME_LABEL=aci-demo-$(cat /dev/urandom | tr -dc 'a-z0-9' | fold -n 32 | tr -d '\n' | fold -n 1 | xargs | sha256sum | tr -d '-' | tr -d '\n')
```

5. Run the following **az container create** command to start a container instance.

```
az container create \
  --resource-group learn-deploy-aci-rg \
  --name mycontainer \
  --image microsoft/aci-helloworld \
  --ports 80 \
  --dns-name-label $DNS_NAME_LABEL \
  --location eastus
```

`$DNS_NAME_LABEL` specifies your DNS name. The image name, **microsoft/aci-helloworld**, refers to a Docker image hosted on Docker Hub that runs a basic Node.js web application.

6. When the `az container create` command completes, run `az container show` to check its status.

```
az container show \
  --resource-group learn-deploy-aci-rg \
  --name mycontainer \
  --query "{FQDN:ipAddress.fqdn,ProvisioningState:provisioningState}" \
  --out table
```

Observe the container's fully qualified domain name (FQDN) and its provisioning state. Here's an example.

```
FQDN ProvisioningState
```

```
-----
```

```
aci-demo.eastus.azurecontainer.io Succeeded
```

If your container is in the **Creating** state, wait a few moments and run the command again until the **Succeeded** state is displayed.

7. From a browser, navigate to your container's FQDN and observe that it is running.

![Screenshot of the sample Node.js container app running in a browser.](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 08/../../Linked_Image_Files/demo)

16 Mini-lab: Deploy Kubernetes with AKS

In this mini-lab, you will:

- Create a new resource group.
- Configure cluster networking.
- Create an Azure Kubernetes Service cluster.
- Connect to the Kubernetes cluster by using `kubectl`.
- Create a Kubernetes namespace.

16.1 Create a new resource group

You'll first need to create a resource group for your resources to deploy into.

1. Sign in to Azure Cloud Shell with your Azure account. Select the Bash version of Cloud Shell.

[Azure Cloud Shell](#)

2. You need to choose a region where you want to create a resource group, for example, **East US**. If you select a different value, remember it for the rest of the exercises in this module. You may need to redefine the value between Cloud Shell sessions. Run the following commands to record these values in Bash variables.

```
REGION_NAME=eastus
RESOURCE_GROUP=aksworkshop
SUBNET_NAME=aks-subnet
VNET_NAME=aks-vnet
```

You can check each value using the `echo` command, for example, `echo $REGION_NAME`.

3. Create a new resource group with the name `**aksworkshop`. Deploy all resources created in these exercises in this resource group. A single resource group makes it easier to clean up the resources after you finish the module.

```
az group create \  
  --name $RESOURCE_GROUP \  
  --location $REGION_NAME
```

16.2 Configure networking

There are two network models to choose from when deploying an AKS cluster. The first model is *Kubenet networking*, and the second is *Azure Container Networking Interface (CNI) networking*.

16.3 Kubenet networking

Kubenet networking is the default networking model in Kubernetes. With Kubenet networking, nodes get assigned an IP address from the Azure virtual network subnet. Pods receive an IP address from a logically different address space to the Azure virtual network subnet of the nodes.

Network address translation (NAT) is then configured so that the pods can reach resources on the Azure virtual network. The source IP address of the traffic is translated to the node's primary IP address and then configured on the nodes. Note that pods receive an IP address that's "hidden" behind the node IP.

16.4 Azure Container Networking Interface (CNI) networking

With Azure Container Networking Interface (CNI), the AKS cluster is connected to existing virtual network resources and configurations. In this networking model, every pod gets an IP address from the subnet and can be accessed directly. These IP addresses must be unique across your network space and calculated in advance.

Some of the features you'll use require you to deploy the AKS cluster by using the *Azure Container Networking Interface networking* configuration.

Below, you will create the virtual network for your AKS cluster. You will use this virtual network and specify the networking model when you deploy the cluster.

1. First, create a virtual network and subnet. Pods deployed in your cluster will be assigned an IP from this subnet. Run the following command to create the virtual network.

```
az network vnet create \  
  --resource-group $RESOURCE_GROUP \  
  --location $REGION_NAME \  
  --name $VNET_NAME \  
  --address-prefixes 10.0.0.0/8 \  
  --subnet-name $SUBNET_NAME \  
  --subnet-prefix 10.240.0.0/16
```

2. Next, retrieve and store the subnet ID in a Bash variable by running the command below.

```
SUBNET_ID=$(az network vnet subnet show \  
  --resource-group $RESOURCE_GROUP \  
  --vnet-name $VNET_NAME \  
  --name $SUBNET_NAME \  
  --query id -o tsv)
```

16.5 Create the AKS cluster

With the new virtual network in place, you can create your new cluster. There are two values you need to know before running the `az aks create` command. The first is the version of the latest, non-preview Kubernetes version available in your selected region, and the second is a unique name for your cluster.

1. To get the latest, non-preview Kubernetes version you use the `az aks get-versions` command. Store the value that returns from the command in a Bash variable named `VERSION`. Run the command below to retrieve and store the version number.

```
SUBNET_ID=$(az network vnet subnet show \  
  --resource-group $RESOURCE_GROUP \  
  --vnet-name $VNET_NAME \  
  --name $SUBNET_NAME \  
  --query id -o tsv)
```

```
--vnet-name $VNET_NAME \  
--name $SUBNET_NAME \  
--query id -o tsv)
```

2. The AKS cluster name must be unique. Run the following command to create a Bash variable that holds a unique name.

```
AKS_CLUSTER_NAME=aksworkshop-$RANDOM
```

3. Run the following command to output the value stored in `$AKS_CLUSTER_NAME`. Note this for later use. You'll need it to reconfigure the variable in the future, if necessary.

```
echo $AKS_CLUSTER_NAME
```

4. Run the following `az aks create` command to create the AKS cluster running the latest Kubernetes version. This command can take a few minutes to complete.

```
az aks create \  
--resource-group $RESOURCE_GROUP \  
--name $AKS_CLUSTER_NAME \  
--vm-set-type VirtualMachineScaleSets \  
--load-balancer-sku standard \  
--location $REGION_NAME \  
--kubernetes-version $VERSION \  
--network-plugin azure \  
--vnet-subnet-id $SUBNET_ID \  
--service-cidr 10.2.0.0/24 \  
--dns-service-ip 10.2.0.10 \  
--docker-bridge-address 172.17.0.1/16 \  
--generate-ssh-keys
```

16.6 Test cluster connectivity by using kubectl

`kubectl` is the main Kubernetes command-line client you use to interact with your cluster and is available in Cloud Shell. A cluster context is required to allow `kubectl` to connect to a cluster. The context contains the cluster's address, a user, and a namespace. Use the `az aks get-credentials` command to configure your instance of `kubectl`.

1. Retrieve the cluster credentials by running the command below.

```
az aks get-credentials \  
--resource-group $RESOURCE_GROUP \  
--name $AKS_CLUSTER_NAME
```

2. Observe everything that was deployed by listing all the nodes in your cluster. Use the `kubectl get nodes` command to list all the nodes.

```
kubectl get nodes
```

a list of your cluster's nodes is displayed. Here's an example:

NAME	STATUS	ROLES	AGE	VERSION
aks-nodepool11-24503160-vmss000000	Ready	agent	1m	v1.15.7
aks-nodepool11-24503160-vmss000001	Ready	agent	1m	v1.15.7
aks-nodepool11-24503160-vmss000002	Ready	agent	1m	v1.15.7

16.7 Create a Kubernetes namespace for the application

A namespace in Kubernetes creates a logical isolation boundary. Names of resources must be unique within a namespace, but not across namespaces. If you don't specify the namespace when you work with Kubernetes resources, the default namespace is implied.

Create a namespace for your ratings application.

1. List the current namespaces in the cluster.

```
kubectl get namespace
```

Observe the list of namespaces similar to this output.

NAME	STATUS	AGE
default	Active	1h
kube-node-lease	Active	1h
kube-public	Active	1h
kube-system	Active	1h

2. Use the `kubectl create namespace` command to create a namespace for the application called **ratingsapp**.

```
kubectl create namespace ratingsapp
```

Confirmation that the namespace was created is displayed.

```
namespace/ratingsapp created
```

17 Mini-lab: Create an App Service Plan

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 at <https://portal.azure.com>.
2. Search for and select **App Service Plans**.
3. Click **+** **Add** to create a new App Service plan.

Setting	Value
Subscription	Choose your subscription
Resource Group	myRGAppServices (create new)
Name	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**.

Note: This rule will add an instance when the CPU percentages is greater than 80% for 10 minutes.

Setting	Value
Time aggregation	Average
Metric name	CPU percentage
Operator	Greater than
Threshold	80

Setting	Value
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. Observe how you can create different App Service plans for your apps.

17.1 Clean up resources

When the Azure resources are no longer needed, clean up the resources you deployed. You may want to save environment for the demo in the next lesson (Deploy staging slots).

18 Mini-lab: Create an App Service and Web App

In this demonstration, you will use the Azure portal to create a web app and an Azure App Service.

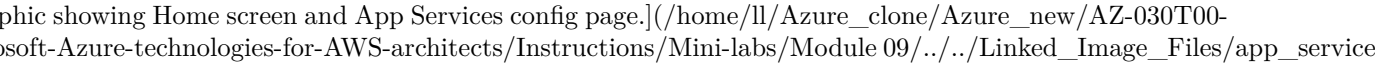
18.1 Create App Service and web app

1. Sign in to the Azure portal at <https://portal.azure.com>.
2. On the Azure portal menu, or from the **Home** page, select **App Services**.

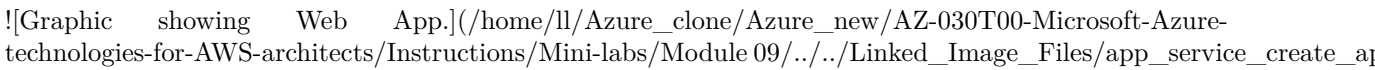
3. From the **App Services** page, select **Create app service**.

4. From the **Web App**, complete the following values:

Field	Value	Details
Subscription	Select your subscription	The web app you are creating must belong to a resource group. Here, you s
Resource Group	Select from the menu	The resource group to which the web app will belong. All Azure resources m
Name	Enter a unique name	The name of your web app. This name will be part of the app's URL: appn
Publish	Code	The method you will use to publish your application. When publishing your
Runtime stack	.NET Core 3.1 (LTS)	The platform on which your application runs. Your choice may affect wheth
Operating System	Linux	The operating system used on the virtual servers that run your app.
Region	Central US	The geographical region from which your app will be hosted.
Linux Plan	Leave default	The name of the App Service plan that will power your app. By default, the
Sku and size	Default	The pricing tier of the plan being created. This determines the performance



4. Select **Review and Create** to navigate to the review page, then select **Create** to create the web app.



Note: It can take a few seconds to get your web app created and ready for your use.

The portal will display the deployment page where you can view the status of your deployment.

18.2 Preview web app

Once the app is ready, navigate to the new app in the Azure portal:

1. On the Azure portal menu or from the **Home page**, select **All resources**.
2. Select the App Service for your web app from the list. Make sure to select the App Service and not the App Service plan.

![Graphic App Services listing.](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 09/../../Linked_Image_Files/app_service_create_ap

The portal displays the **App Services** overview page.

![Graphic App Services listing.](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 09/../../Linked_Image_Files/app_service_create_ap

3. To preview your new web app's default content, select its URL at the top right. The placeholder page that loads indicates that your web app is up and running and ready to receive deployment of your app's code.

![Screenshot showing the newly created App Service in a browser.](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 09/../../Linked_Image_Files

19 Mini-lab: Deploy staging slots

Deploying your application to a non-production slot has the following benefits:

- You can validate app changes in a staging deployment slot before swapping it with the production slot.
- Deploying an app to a slot first and swapping it into production makes sure that all instances of the slot are warmed up before being swapped into production. This eliminates downtime when you deploy your app. The traffic redirection is seamless, and no requests are dropped because of swap operations. You can automate this entire workflow by configuring [auto swap](#) when pre-swap validation isn't needed.
- After a swap, the slot with previously staged app now has the previous production app. If the changes swapped into the production slot aren't as you expect, you can perform the same swap immediately to get your "last known good site" back.

19.1 Add a slot

The app must be running in the **Standard**, **Premium**, or **Isolated** tier in order for you to enable multiple deployment slots.

1. Sign in to the Azure portal at <https://portal.azure.com>. Search for and select **App Services** and select your app.

![Search for App Services](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 09/../../Linked_Image_Files/search-for-app-services.png)

2. In the left pane, select **Deployment slots > Add Slot**.

![Add a new deployment slot](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 09/../../Linked_Image_Files/qgaddnewdeploymentsl

NOTE: If the app isn't already in the **Standard**, **Premium**, or **Isolated** tier, you receive a message that indicates the supported tiers for enabling staged publishing. At this point, you have the option to select **Upgrade** and go to the **Scale** tab of your app before continuing.

3. In the **Add a slot** dialog box, give the slot a name, and select whether to clone an app configuration from another deployment slot. Select **Add** to continue.

![Configuration source](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 09/../../Linked_Image_Files/configuration-source-1.png)

You can clone a configuration from any existing slot. Settings that can be cloned include app settings, connection strings, language framework versions, web sockets, HTTP version, and platform bitness.

4. After the slot is added, select **Close** to close the dialog box. The new slot is now shown on the **Deployment slots** page. By default, **Traffic %** is set to 0 for the new slot, with all customer traffic routed to the production slot.
5. Select the new deployment slot to open that slot's resource page.

![Deployment slot title](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 09/../../Linked_Image_Files/staging-title.png)

The staging slot has a management page just like any other App Service app. You can change the slot's configuration. To remind you that you're viewing the deployment slot, the app name is shown as **<app-name>/<slot-name>**, and the app type is **App Service (Slot)**. Notice the slot displayed as a separate app in your resource group with the same designations.

The new deployment slot has no content, even if you clone the settings from a different slot. For example, you can publish to this slot with Git. You can deploy to the slot from a different repository branch or a different repository.

19.2 Swap two slots

You can swap deployment slots on your app's **Deployment slots** page and the **Overview** page.

Important: Before you swap an app from a deployment slot into production, make sure that production is your target slot and that all settings in the source slot are configured exactly as you want to have them in production.

To swap deployment slots:

1. Go to your app's **Deployment slots** page and select **Swap**.

![Swap button](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 09/../../Linked_Image_Files/swap-button-bar.png)

The **Swap** dialog box shows settings in the selected source and target slots that will be changed.

2. Select the desired **Source** and **Target** slots. Usually, the target is the production slot. Also, select the **Source Changes** and **Target Changes** tabs and verify that the configuration changes are expected. When you're finished, you can swap the slots immediately by selecting **Swap**.

![Complete swap](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 09/../../Linked_Image_Files/swap-immediately.png)

3. When you're finished, close the dialog box by selecting **Close**.

20 Mini-lab: Create a storage account in the portal

Every storage account must belong to an Azure resource group. A resource group is a logical container for grouping your Azure services. When you create a storage account, you have the option to either create a new resource group or use an existing resource group. This article shows how to create a new resource group.

A general-purpose v2 storage account provides access to all of the Azure Storage services: blobs, files, queues, tables, and disks. The steps outlined here create a general-purpose v2 storage account, but the steps to create any type of storage account are similar.

To create a general-purpose v2 storage account in the Azure portal, follow these steps:

1. On the Azure portal menu, 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 as shown in the following image.

![Screenshot showing how to create a resource group in the portal](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 10/../../Linked_Image_Files/resource-group-for-storage.png)

- Next, 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.
- Select a location for your storage account, or use the default location.
- Leave these fields set to their default values:

Field	Value
Deployment model	Resource Manager
Performance	Standard
Account kind	StorageV2 (general-purpose v2)
Replication	Read-access geo-redundant storage (RA-GRS)
Access tier	Hot

- Select **Review + Create** to review your storage account settings and create the account.
- Select **Create**.

Note: We will use the storage account created during this demo later in this module, in other demos/walkthroughs.

21 Mini-lab: Blob Storage

In this mini-lab, you will explore blob storage.

Note: This mini-lab requires a storage account.

21.1 Create a container

- Navigate to a storage account in the Azure portal.
- In the left menu for the storage account, scroll to the **Blob service** section, then select **Blobs**.
- Select the **+ Container** button.
- 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.
- Set the level of public access to the container. The default level is Private (no anonymous access).
- Select **OK** to create the container.

21.2 Upload a block blob

- In the Azure portal, navigate to the container you created in the previous section.
- Select the container to show a list of blobs it contains. Since this container is new, it won't yet contain any blobs.
- Select the **Upload** button to upload a blob to the container.
- Expand the **Advanced** section.
- Notice the **Authentication type**, **Blob type**, **Block size**, and the ability to **Upload to a folder**.
- Notice the default **Authentication type** is SAS.
- Browse your local file system to find a file to upload as a block blob and select **Upload**.
- Upload as many blobs as you like in this way. You'll observe that the new blobs are now listed within the container.

21.3 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**.

22 Mini-lab: Create a Shared Access Signature (Portal)

In this mini-lab, we will create a Shared Access Signature (SAS).

Note: This mini-lab requires a storage account with a blob container and an uploaded file.

22.1 Create an SAS at the service level

1. Sign in to the Azure portal at <https://portal.azure.com>.
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 that you learned about in the lesson.

22.2 Create an 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.

23 Mini-lab: Create an Azure SQL Database single database

In this mini-lab, you will use the Azure portal to create an Azure SQL Database single database. You will then query the database using Query editor in the Azure portal.

A single database is the quickest and simplest deployment option for Azure SQL Database. You can manage a single database within a SQL Database server, which is inside an Azure resource group in a specified Azure region. In this mini-lab, you will create a new resource group and SQL server for the new database.

You can create a single database in the *provisioned* or *serverless* compute tier. A provisioned database is pre-allocated a fixed amount of compute resources, including CPU and memory, and uses one of two purchasing models. This mini-lab creates a provisioned database using the vCore-based purchasing model.

23.1 Create a single database

In this step, you will create an Azure SQL Database server and a single database that uses AdventureWorksLT sample data. You can create the database by using Azure portal menus and screens, or by using an Azure CLI or PowerShell script in the Azure Cloud Shell.

To create a resource group, SQL server, and single database in the Azure portal:

1. Sign in to the Azure portal at <https://portal.azure.com>.
2. From the Search bar, search for and select **Azure SQL**.
3. On the Azure SQL page, select **Add**.

![Add to Azure SQL](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 12/../../Linked_Image_Files/demo_sql_image1.png)

4. On the **Select SQL deployment option** page, select the **SQL databases** tile, with **Single database** under **Resource** type. You can view more information about the different databases by selecting **Show details**.
5. Select **Create**.

![Create single database](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 12/../../Linked_Image_Files/demo_sql_image2.png)

6. On the **Basics** tab of the Create SQL database form, under **Project details**, select the correct Azure Subscription if it isn't already selected.
7. Under **Resource group**, select **Create new**, enter *myResourceGroup*, and select **OK**.
8. Under **Database details**, for **Database name** enter *mySampleDatabase*.
9. For **Server**, select **Create new** and fill out the New server form as follows:
 - **Server name**: Enter *mysqlserver* and some characters for uniqueness.
 - **Server admin login**: Enter *azureuser*.
 - **Password**: Enter a password that meets requirements, and enter it again in the **Confirm password** field.
 - **Location**: Drop down and choose a location, such as **(US) East US**.

Select **OK**.

![New server](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 12/../../Linked_Image_Files/demo_sql_image3.png)

Record the server admin login and password so you can log in to the server and databases. If you forget your login or password, you can get the login name or reset the password on the **SQL server** page after database creation. To open the **SQL server** page, select the server name on the database **Overview** page.

10. Under **Compute + storage**, if you want to reconfigure the defaults, select **Configure database**.

On the **Configure** page, you can optionally:

- Change the **Compute tier** from **Provisioned** to **Serverless**.
- Review and change the settings for **vCores** and **Data max size**.
- Select **Change configuration** to change the hardware generation.

After making any changes, select **Apply**.

11. Select **Next: Networking** at the bottom of the page.

![New SQL database - Basic tab](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 12/../../Linked_Image_Files/demo_sql_image4.png)

12. On the **Networking** tab, under **Connectivity method**, select **Public endpoint**.
13. Under **Firewall rules**, set **Add current client IP address** to **Yes**.
14. Select **Next: Additional** settings at the bottom of the page.

![Networking tab](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 12/../../Linked_Image_Files/demo_sql_image5.png)

15. On the **Additional settings** tab, in the **Data source** section, for **Use existing data**, select **Sample**.
16. Select **Review + create** at the bottom of the page.

![Additional settings tab](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 12/../../Linked_Image_Files/demo_sql_image6.png)

17. After reviewing settings, select **Create**.

23.2 Query the database

Once your database is created, you can use the built-in Query editor in the Azure portal to connect to the database and query the data.

1. In the portal, search for and select **SQL databases** and then select your database from the list.
2. On the **SQL Database** page for your database, select **Query editor** in the left menu.
3. Enter your server admin login information, and select **OK**.

![Sign in to Query editor](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 12/../../Linked_Image_Files/demo_sql_image7.png)

4. Enter the following query in the Query editor pane.

```
SELECT TOP 20 pc.Name as CategoryName, p.name as ProductName
```

```
FROM SalesLT.ProductCategory pc
```

```
JOIN SalesLT.Product p
```

```
ON pc.productcategoryid = p.productcategoryid;
```

5. Select **Run**, and then review the query results in the **Results** pane.

![Query editor results](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 12/../../Linked_Image_Files/demo_sql_image8.png)

6. Close the **Query editor** page and select **OK** when prompted to discard your unsaved edits.

24 Mini-lab: Create an Azure SQL Database Managed Instance

This mini-lab walks you through how to create an Azure SQL Database managed instance in Azure portal.

Sign in to the Azure portal at <https://portal.azure.com>.

24.1 Create a managed instance

The following steps show you how to create a managed instance:

1. Select **Azure SQL** on the left menu of Azure portal. If **Azure SQL** is not in the list, select **All services** and then enter *Azure SQL* in the search box.
2. Select **+Add** to open the **Select SQL deployment option** page. You can view additional information about an Azure SQL Database managed instance by selecting **Show details** on the **Managed instances** tile.
3. Select **Create**.

![Create a managed instance](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 12/../../Linked_Image_Files/demo_managed_sql_image1.png)

4. Use the tabs on the **Create Azure SQL Database Managed Instance** provisioning form to add required and optional information. The following sections describe these tabs.

24.2 Basics

- Fill out mandatory information required on the **Basics** tab.

!["Basics" tab for creating a managed instance](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 12/../../Linked_Image_Files/demo_managed_sql_

- Select **Configure Managed Instance** to size compute and storage resources and to review the pricing tiers. Use the sliders or text boxes to specify the amount of storage and the number of virtual cores. When you're finished, select **Apply** to save your selection.

![Managed instance form](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 12/../../Linked_Image_Files/demo_managed_sql_image3.png)

- To review your choices before you create a managed instance, you can select **Review + create**. Or, configure networking options by selecting **Next: Networking**.

24.3 Networking

- Fill out optional information on the **Networking** tab. If you omit this information, the portal will apply default settings.

!["Networking" tab for creating a managed instance](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 12/../../Linked_Image_Files/demo_managed_sql_

Use the table below as a reference for information required at this tab.

Setting	Suggested value
Virtual network	Select either Create new virtual network or a valid virtual network and
Connection type	Choose between a proxy and a redirect connection type.
Public endpoint	Select Enable.
Allow access from (if Public endpoint is enabled)	Select one of the options.

- Select **Review + create** to review your choices before you create a managed instance. Or, configure more custom settings by selecting **Next: Additional settings**.

24.4 Additional settings

- Fill out optional information on the **Additional settings** tab. If you omit this information, the portal will apply default settings.

!["Additional settings" tab for creating a managed instance](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 12/../../Linked_Image_Files/demo_mana

Use the table below as a reference for information required at this tab.

Setting	Suggested value
Collation	Choose the collation that you want to use for your
Time zone	Select the time zone that your managed instance w
Use as failover secondary	Select Yes.
Primary managed instance (if Use as failover secondary is set to Yes)	Choose an existing primary managed instance that

24.5 Review + create

5. Select the **Review + create** tab to review your choices before you create the managed instance.

![Tab for reviewing and creating a managed instance](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 12/../../Linked_Image_Files/demo_mana

6. Select **Create** to start provisioning the managed instance.

25 Lab: Implementing Azure SQL Database-Based Applications

25.1 Lab scenario

Adatum Corporation has a number two tier applications with .NET Core-based front end and SQL Server-based backend. The Adatum Enterprise Architecture team is exploring the possibility of implementing these applications by leveraging Azure SQL Database as the data tier. Given intermittent, unpredictable usage of the existing SQL Server backend and relatively high tolerance for latency built into the front-end apps, Adatum is considering the serverless tier of Azure SQL Database.

Serverless is a compute tier for individual Azure SQL Databases instances that automatically scales compute based on workload demand and bills for compute used per second. The serverless compute tier is also capable of automatically pausing databases during inactive periods when only storage is billed and automatically resumes databases when activity returns.

The Adatum Enterprise Architecture team is also interested in evaluating network-level security provided by the Azure SQL Databases, in order to ensure that it is possible to restrict inbound connections to specific ranges of IP addresses, in scenarios where the apps must be able to connect from its on-premises locations without relying on hybrid connectivity via Site-to-Site VPN or ExpressRoute.

To accomplish these objectives, the Adatum Architecture team will test Azure SQL Database-based applications, including:

- Implementing serverless tier of Azure SQL Database
- Implementing .NET Core console apps that use Azure SQL Database as their data store

25.2 Objectives

After completing this lab, you will be able to:

- Implement serverless tier of Azure SQL Database
- Configure .NET Core-based console apps that use Azure SQL Database as their data store

25.3 Lab Environment

Windows Server admin credentials

- User Name: **Student**
- Password: **Pa55w.rd1234**

Estimated Time: 60 minutes

25.4 Lab Files

- None

25.5 Exercise 1: Implement Azure SQL Database

The main tasks for this exercise are as follows:

1. Create Azure SQL Database
2. Connect to and query Azure SQL Database

25.5.0.1 Task 1: Create Azure SQL Database

1. From your lab computer, start a web browser, navigate to the [Azure portal](#), and sign in by providing credentials of a user account with the Owner role in the subscription you will be using in this lab.
2. In the Azure portal, search for and select **SQL database** and, on the **SQL databases** blade, select **+** **Add**.
3. On the **Basics** tab of the **Create SQL Database** blade, specify the following settings (leave others with their default values):

Setting	Value
Subscription	the name of the Azure subscription you will be using in this lab
Resource group	the name of a new resource group az30303a-labRG
Database name	az30303a-db1

- Directly below the **Server** drop down list, select the **Create new** and, on the **New server** blade, specify the following settings and select **OK** (leave others with their default values):

Setting	Value
Server name	any valid, globally unique name
Server admin login	sqladmin
Password	Pa55w.rd1234
Location	the name of an Azure region where you can provision SQL databases
Allow Azure services to access server	<i>Select the checkbox</i>

- Next to the **Compute + storage** label, select the **Configure database** link.
- On the **Configure** blade, select **Serverless**, review the corresponding hardware configuration and auto-pause delay settings, leave the **Enable auto-pause** checkbox enabled, and select **Apply**.
- Back on the **Basics** tab of the **Create SQL Database** blade, select **Next: Networking** >.
- On the **Networking** tab of the **Create SQL Database** blade, specify the following settings (leave others with their default values):

Setting	Value
Connectivity method	Public endpoint
Allow Azure services and resources to access this server	Yes
Add current client IP address	No

- Select **Next: Additional settings** >.
- On the **Additional settings** tab of the **Create SQL Database** blade, specify the following settings (leave others with their default values):

Setting	Value
Use existing data	Sample
Enable advanced data security	Not now

- Select **Review + create** and then select **Create**.

Note: Wait for the SQL database to be created. Provisioning should take about 2 minutes.

25.5.0.2 Task 2: Connect to and query Azure SQL Database

- In the Azure portal, search for and select **SQL database** and, on the **SQL databases** blade, select the entry representing the newly created **az30303a-db1** Azure SQL database.
- On the SQL database blade, select **Query editor (preview)**.
- In the **SQL Server authentication** section, in the **Password** textbox, type **Pa55w.rd1234** and select **OK**.
- In the **Query editor (preview)** pane, on the **Query 1** tab, enter the following query and select **Run**:

```
SELECT TOP 20 pc.Name as CategoryName, p.name as ProductName
FROM SalesLT.ProductCategory pc
JOIN SalesLT.Product p
ON pc.productcategoryid = p.productcategoryid;
```

5. Review the **Results** tab to verify that the query completed successfully.

25.5.1 Exercise 2: Implement a .NET Core console app that uses Azure SQL Database as their data store

The main tasks for this exercise are as follows:

1. Identify ADO.NET connection information of Azure SQL Database
2. Create and configure a .NET Core console app
3. Test the .NET Core console app
4. Configure Azure SQL database firewall
5. Verify the functionality of the .NET Core console app
6. Remove Azure resources deployed in the lab

25.5.1.1 Task 1: Identify ADO.NET connection information of Azure SQL Database

1. In the Azure portal, on the blade of the Azure SQL database you deployed in the previous exercise, in the **Settings** section, select **Connection strings**.
2. On the **ADO.NET** tab, note the ADO.NET connection string for SQL authentication.

25.5.1.2 Task 2: Create and configure a .NET Core console app

1. In the Azure portal, open the **Cloud Shell** pane by selecting on the toolbar icon directly to the right of the search textbox.
2. If prompted to select either **Bash** or **PowerShell**, select **Bash**.

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**.

3. From the Cloud Shell pane, run the following to create a new folder named **az30303a1** and set it as your current directory:

```
mkdir az30303a1
cd az30303a1/
```

4. From the Cloud Shell pane, run the following to create a new app project file for a .NET Core-based app based on the desktop template:

```
dotnet new console
```

5. In the Cloud Shell pane, use the built in editor to open and modify the **az30303a1.csproj** file by adding the following XML element between the <Project> tags:

```
<ItemGroup>
  <PackageReference Include="System.Data.SqlClient" Version="4.6.0" />
</ItemGroup>
```

6. Save and close the **az30303a1.csproj** file.
7. In the Cloud Shell pane, use the built in editor to open and modify the **Program.cs** file by replacing its content with the following code:

```
using System;
using System.Data.SqlClient;
using System.Text;

namespace sqltest
{
    class Program
    {
        static void Main(string[] args)
        {
            try
```

```

    {
        SqlConnectionStringBuilder builder = new SqlConnectionStringBuilder();
        builder.ConnectionString = "<your_ado_net_connection_string>";

        using (SqlConnection connection = new SqlConnection(builder.ConnectionString))
        {
            Console.WriteLine("\nQuery data example:");
            Console.WriteLine("=====\\n");

            connection.Open();
            StringBuilder sb = new StringBuilder();
            sb.Append("SELECT TOP 20 pc.Name as CategoryName, p.name as ProductName ");
            sb.Append("FROM [SalesLT].[ProductCategory] pc ");
            sb.Append("JOIN [SalesLT].[Product] p ");
            sb.Append("ON pc.productcategoryid = p.productcategoryid;");
            String sql = sb.ToString();

            using (SqlCommand command = new SqlCommand(sql, connection))
            {
                using (SqlDataReader reader = command.ExecuteReader())
                {
                    while (reader.Read())
                    {
                        Console.WriteLine("{0}{1}", reader.GetString(0), reader.GetString(1));
                    }
                }
            }
        }
        catch (SqlException e)
        {
            Console.WriteLine(e.ToString());
        }
        Console.WriteLine("\nDone. Press enter.");
        Console.ReadLine();
    }
}

```

8. Leave the editor window open.
9. In the Azure portal, on the blade displaying the connection strings for the **az30303a-db1** database, copy the ADO.NET connection string.
10. Switch back to the editor window and replace the placeholder **<your_ado_net_connection_string>** with the value of the connection string you copied in the previous step.
11. In the connection string you copied into the editor window, replace the placeholder **{your_password}** with **Pa55w.rd1234**.
12. Save and close the **Program.cs** file.

25.5.1.3 Task 3: Test the .NET Core console app

1. From the Cloud Shell pane, run the following to compile the newly created .NET Core-based console app:


```
dotnet restore
```
2. From the Cloud Shell pane, run the following to execute the newly created .NET Core-based console app:


```
dotnet run
```
3. Note that the execution of the console app will trigger an error.

Note: This is expected, since the connection from IP address assigned to the virtual machine running the Cloud Shell session must be explicitly allowed.

25.5.1.4 Task 4: Configure Azure SQL database firewall

1. From the Cloud Shell pane, run the following to identify the public IP address of the virtual machine running the Cloud Shell session:

```
curl -s checkip.dyndns.org | sed -e 's/.*Current IP Address: //' -e 's/<.*$//'
```

2. In the Azure portal, on the blade displaying the connection strings for the **az30303a-db1** database, select **Overview** and, in the toolbar, select **Set server firewall**.
3. On the **Firewall settings** blade, set the following entries and select **Save**:

Setting	Value
Rule name	cloudshell
Start IP	the IP address you identified earlier in this task
End IP	the IP address you identified earlier in this task

Note: Obviously this is meant for the lab purposes only, since that IP address will change after you restart the Cloud Shell session.

25.5.1.5 Task 5: Verify the functionality of the .NET Core console app

1. From the Cloud Shell pane, run the following to execute the newly created .NET Core-based console app:

```
dotnet run
```
2. Note that the execution of the console app will this time be successful and that it returns the same results as those displayed in the query editor within the Azure portal SQL database blade.

25.5.1.6 Task 6: Remove Azure resources deployed in the lab

1. From the Cloud Shell pane, run the following to list the resource group you created in this exercise:

```
az group list --query "[?starts_with(name,'az30303')].name" --output tsv
```

Note: Verify that the output contains only the resource group you created in this lab. This group will be deleted in this task.

2. From the Cloud Shell pane, run the following to delete the resource group you created in this lab

```
az group list --query "[?starts_with(name,'az30303')].name" --output tsv | xargs -L1 bash -c 'az g
```

3. Close the Cloud Shell pane.

26 Mini-lab: Azure Security Center

Azure Security Center provides unified security management and threat protection across your hybrid cloud workloads. To enable Security Center on your Azure subscription, follow these steps:

1. Sign into the [Azure portal](#).
2. From the portal's menu, select **Security Center**.

Security Center's overview page opens.

![Security Center's overview dashboard](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 13/../../Linked_Image_Files/security-center-overview.png)

Security Center – Overview provides a unified view into the security posture of your hybrid cloud workloads, enabling you to discover and assess the security of your workloads and to identify and mitigate risk. Security Center automatically, at no cost, enables any of your Azure subscriptions not previously onboarded by you or another subscription user.

Within minutes of launching Security Center the first time, you may see:

- Insights Recommendations for ways to improve the security of your connected resources.
- Prioritized security alerts along with the information you need to quickly investigate the problem and remediate an attack

- An inventory of your resources that are now being assessed by Security Center, along with the security posture of each.
- A Secure Score measure of the security posture of your subscriptions

Tip: To enable Security Center on all subscriptions within a management group, see [Enable Security Center on multiple Azure subscriptions](#).

27 Mini-lab: Monitor costs with Azure Monitor

For this mini-lab, you must have previously-running and currently-running services during the billing period. Data may not be available if it is less than 24 hours since resources were deployed.

27.1 Navigate to Cost Analysis

1. Sign in to the Azure portal at <https://portal.azure.com>.
2. Search for **Cost Management** and select **Cost Management + Billing** from the list of results.
3. Under **Cost Management**, select and review **Cost Analysis**

Note this is similar to the **Cost Management + Billing Overview**.

27.2 Customize report

1. Customize the display to show actual and forecast costs. Change the granularity and select **Area** to change the chart type.

![[Screenshot of Azure Monitor cost report, with chart type list highlighted.]](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 13/../../Linked_Image_Files/

2. For additional charts (such as **Service name**, **Location**, and **Resource group name**) use filters on these charts to display results for different data.

![[Screenshot of Azure Monitor cost report, with filter list highlighted.]](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 13/../../Linked_Image_Files/

Mini-lab: Log Analytics

In this mini-lab, you will work with the Log Analytics query language.

27.3 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.

27.4 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 also **Saved Queries**.

Is there a particular query you are interested in?

28 Mini-lab: Add users by using Azure Active Directory

Add new users or delete existing users from your Azure Active Directory (Azure AD) organization. To add or delete users, you must be a User administrator or Global administrator.

You can create a new user using the Azure Active Directory portal. Follow these steps:

1. Sign in to the Azure portal at <https://portal.azure.com> as a User administrator for the organization.
2. Search for and select *Azure Active Directory* from any page.

3. Select **Users** and then select **New user**.

![Graphic showing the Azure Active Directory page in the Azure portal. All users and New user are highlighted.](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 14/../../Linked_Image_Files/AAD_User_NewUser.png)

4. On the **User** page, enter information for this user:

- **Name.** Required. The first and last name of the new user. For example, *TesterAAD*.
- **User name.** Required. The user name of the new user. For example, *TesterAAD@contoso.com*. The domain part of the user name must use either the initial default domain name, *.onmicrosoft.com*, or a custom domain name, such as *contoso.com*.
- **Job info:** You can add more information about the user here, or do it later. For more information about adding user info, see [How to add or change user profile information](#).

5. Copy the autogenerated password provided in the **Password** box. You'll need to give this password to the user to sign in for the first time.

6. Select **Create**.

The user is created and added to your Azure AD organization.

29 Mini-lab: Add an Azure Role Assignment

29.1 Prerequisites

To add or remove role assignments, you must have:

- `Microsoft.Authorization/roleAssignments/write` and `Microsoft.Authorization/roleAssignments/delete` permissions, such as *User Access Administrator* or *Owner*.

Access control (IAM) is the blade that you use to assign roles to grant access to Azure resources. It's also known as identity and access management and appears in several locations in the Azure portal. The following shows an example of the Access control (IAM) blade for a subscription.

![Access control (IAM) blade for a subscription](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 14/../../Linked_Image_Files/demo_RBAC_image1)

To be the most effective with the Access control (IAM) blade, it helps if you can answer the following three questions when you are trying to assign a role:

1. Who needs access?

Who refers to a user, group, service principal, or managed identity. This is also called a *security principal*.

2. What role do they need?

Permissions are grouped together into roles. You can select from a list of several [built-in roles or you can use your own custom roles.

3. Where do they need access?

Where refers to the set of resources that the access applies to. Where it can be a management group, subscription, resource group, or a single resource such as a storage account. This is called the *scope*.

29.2 Add a role assignment

In Azure RBAC, to grant access to an Azure resource, you add a role assignment. Follow these steps to assign a role.

1. In the Azure portal, click **All services** and then select the scope that you want to grant access to.
2. Click the specific resource for that scope.
3. Click **Access control (IAM)**.
4. Click the **Role assignments** tab to view the role assignments at this scope.

![Access control (IAM) and Role assignments tab](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 14/../../Linked_Image_Files/demo_RBAC_image2)

5. Click **Add > Add role assignment**.

If you don't have permissions to assign roles, the **Add role assignment** option will be disabled.

![Add menu](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/labs/Module 14/../../Linked_Image_Files/demo_RBAC_image3.png)

The **Add role assignment** pane opens.

![Add role assignment pane](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 14/../../Linked_Image_Files/demo_RBAC_image4.png)

6. In the **Role** drop-down list, select a role such as **Virtual Machine Contributor**.
7. In the **Select** list, select a user, group, service principal, or managed identity. If the security principal is not found in the list, you can type in the **Select** box to search the directory for display names, email addresses, and object identifiers.
8. Click **Save** to assign the role.

After a few moments, the security principal is assigned the role at the selected scope.

![Add role assignment saved](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 14/../../Linked_Image_Files/demo_RBAC_image5.png)

30 Mini-lab: Enable system-assigned managed identity on an existing VM

30.1 Enable managed identity

To enable system-assigned managed identity on a VM that was originally provisioned without it, your account needs the [Virtual Machine Contributor](#) role assignment. No additional Azure AD directory role assignments are required.

1. Sign in to the Azure portal at <https://portal.azure.com> using an account associated with the Azure subscription that contains the VM.
2. Navigate to the desired Virtual Machine and select **Identity**.
3. Under **System assigned, Status**, select **On** and then click **Save**.

![Configuration page screenshot](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 14/../../Linked_Image_Files/create-windows-vm-portal-configuration-blade.png)

30.2 Sign in to Azure VM using managed identity (PowerShell)

Traditionally, in order to access secured resources under its own identity, a script client would need to:

- Be registered and consented with Azure AD as a confidential/web client application.
- Sign in under its service principal using the app's credentials (which are likely embedded in the script).

With managed identities for Azure resources, your script client no longer needs to do either, as it can sign in under the managed identities for Azure resources service principal.

The following script demonstrates how to:

1. Sign in to Azure AD under the VM's managed identity for Azure resources service principal.
2. Call an Azure Resource Manager cmdlet to get information about the VM. PowerShell takes care of managing token use for you automatically.

```
Add-AzAccount -identity
```

```
# Call Azure Resource Manager to get the service principal ID for the VM's managed identity for Az
$vmInfoPs = Get-AzVM -ResourceGroupName <RESOURCE-GROUP> -Name <VM-NAME>
$spID = $vmInfoPs.Identity.PrincipalId
echo "The managed identity for Azure resources service principal ID is $spID"
```

31 Mini-lab: Create and manage a policy to enforce compliance

In this min-lab, you will learn to use Azure Policy to do some of the more common tasks related to creating, assigning, and managing policies across your organization, such as:

- Assign a policy to enforce a condition for resources you create in the future
- Create and assign an initiative definition to track compliance for multiple resources
- Resolve a non-compliant or denied resource
- Implement a new policy across an organization

31.1 Assign a policy

The first step in enforcing compliance with Azure Policy is to assign a policy definition. A policy definition defines under what condition a policy is enforced and what effect to take. In this example, assign the built-in policy definition called *Inherit a tag from the resource group if missing* to add the specified tag with its value from the parent resource group to new or updated resources missing the tag.

1. Go to the Azure portal to assign policies. Search for and select **Policy**.

![Search for Policy in the search bar](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 14/../../Linked_Image_Files/Demonstration_Policy_image.png)

2. Select **Assignments** on the left side of the Azure Policy page. An assignment is a policy that has been assigned to take place within a specific scope.

![Select Assignments from Policy Overview page](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 14/../../Linked_Image_Files/Demonstration_Policy_image.png)

3. Select **Assign Policy** from the top of the **Policy - Assignments** page.

![Assign a policy definition from Assignments page](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 14/../../Linked_Image_Files/Demonstration_Policy_image.png)

4. On the **Assign Policy** page and **Basics** tab, select the **Scope** by selecting the ellipsis and selecting either a management group or subscription. Optionally, select a resource group. A scope determines what resources or grouping of resources the policy assignment gets enforced on. Then select **Select** at the bottom of the **Scope** page.
5. Resources can be excluded based on the Scope. Exclusions start at one level lower than the level of the **Scope**. **Exclusions** are optional, so leave it blank for now.
6. Select the **Policy definition** ellipsis to open the list of available definitions. You can filter the policy definition **Type** to *Built-in* to view all and read their descriptions.
7. Select **Inherit a tag from the resource group if missing**. Select **Select** at the bottom of the **Available Definitions** page once you have found and selected the policy definition.

![Use search filter to locate a policy](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 14/../../Linked_Image_Files/Demonstration_Policy_image.png)

8. The **Assignment name** is automatically populated with the policy name you selected, but you can change it. For this example, leave *Inherit a tag from the resource group if missing*. You can also add an optional **Description**. The description provides details about this policy assignment.
9. Leave **Policy enforcement** as *Enabled*. When *Disabled*, this setting allows testing the outcome of the policy without triggering the effect.
10. **Assigned by** is automatically filled based on who is logged in.
11. Select the **Parameters** tab at the top of the wizard.
12. For **Tag Name**, enter *Environment*.
13. Select the **Remediation** tab at the top of the wizard.
14. Leave **Create a remediation task** unchecked. This box allows you to create a task to alter existing resources in addition to new or updated resources.

15. **Create a Managed Identity** is automatically checked since this policy definition uses the modify effect. **Permissions** is set to *Contributor* automatically based on the policy definition.
16. Select the **Review + create** tab at the top of the wizard.
17. Review your selections, then select **Create** at the bottom of the page.

32 Mini-lab: Create and encrypt a Windows virtual machine with the Azure portal

Azure virtual machines (VMs) can be created through the Azure portal. The Azure portal is a browser-based user interface to create VMs and their associated resources. In this mini-lab, you will use the Azure portal to deploy a Windows virtual machine (VM) running Ubuntu 18.04 LTS, create a key vault for the storage of encryption keys, and encrypt the VM.

If you don't have an Azure subscription, create a [free account](#) before you begin.

32.1 Create a virtual machine

1. Sign in to the Azure portal at <https://portal.azure.com>.
2. Choose **Create a resource** in the upper left corner of the Azure portal.
3. On the New page, under Popular, select **Windows Server 2016 Datacenter**.
4. On the **Basics** tab, under **Project details**, make sure the correct subscription is selected and then choose to **Create new resource group**. Enter *myResourceGroup* as the name.
5. For **Virtual machine name**, enter *MyVM*.
6. For **Region**, select your region (e.g., *East US*).
7. Make sure the **Size** is *Standard D2s v3*.
8. Under **Administrator account**, select **Password**. Enter a user name and a password.

![ResourceGroup creation screen](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 15/../../Linked_Image_Files/portal-qswindows-vm-creation.png)

Warning: The **Disks** tab features an **Encryption Type** field under **Disk options**. This field is used to specify encryption options for Managed Disks + CMK, not for Azure Disk Encryption. To avoid confusion, we suggest you skip the **Disks** tab entirely while completing this tutorial.

9. Select the **Management** tab and verify that you have a Diagnostics Storage Account. If you have no storage accounts, select **Create New**, give your new account a name, and select **Ok**

![ResourceGroup creation screen](/home/ll/Azure_clone/Azure_new/AZ-030T00-Microsoft-Azure-technologies-for-AWS-architects/Instructions/Mini-labs/Module 15/../../Linked_Image_Files/portal-qsvm-creation-storage.png)

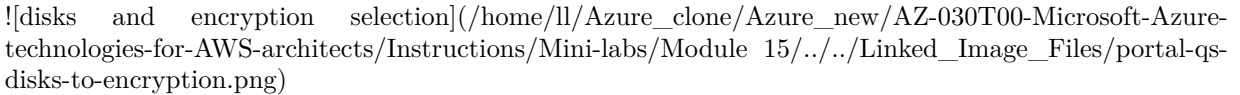
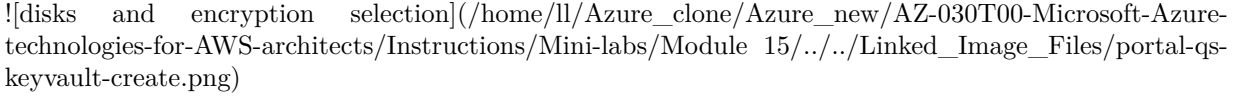
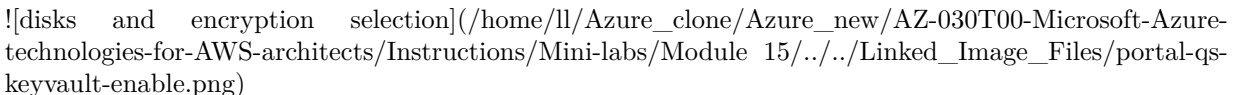
10. Select **Review + Create**.
11. On the **Create a virtual machine** page, you can find the details about the VM you are about to create. When you are ready, select **Create**.

It will take a few minutes for your VM to be deployed. When the deployment is finished, move on to the next section.

32.2 Encrypt the virtual machine

Warning: In order to make sure the encryption secrets don't cross regional boundaries, Azure Disk Encryption needs the Key Vault and the VMs to be co-located in the same region. Create and use a Key Vault that is in the same region as the VM to be encrypted.

1. When the VM deployment is complete, select **Go to resource**.
2. On the left-hand sidebar, select **Disks**.

3. On the Disks screen, select **Encryption**.

4. On the encryption screen, under **Disks to encrypt**, choose **OS and data disks**.
5. Under **Encryption settings**, choose **Select a key vault and key for encryption**.
6. On the **Select key from Azure Key Vault** screen, select **Create New**.

7. On the **Create key vault** screen, ensure that the Resource Group is the same as the one you used to create the VM.
8. Give your key vault a name. Every key vault across Azure must have a unique name.
9. On the **Access Policies** tab, check the **Azure Disk Encryption for volume encryption** box.

10. Select **Review + create**.
11. After the key vault has passed validation, select **Create**. This will return you to the **Select key from Azure Key Vault** screen.
12. Leave the **Key** field blank and choose **Select**.
13. At the top of the encryption screen, select **Save**. A popup will warn you that the VM will reboot. select **Yes**.

32.3 Clean up resources

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.

33 Mini-lab: Back up files and folders

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

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

33.1 Create a Recovery Services vault

1. Sign in to the Azure portal at <https://portal.azure.com>. In the portal, type Recovery Services and select **Recovery Services vaults**.
2. Select **Add**.
3. Provide a **Name**, **Subscription**, **Resource group**, and **Location**.
4. Select **Create**. It can take several minutes for the Recovery Services vault to be created. Monitor the status notifications in the upper-right 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, select **Refresh**.

33.2 Configure the vault

1. For your recovery services vault, select **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. Select **Prepare infrastructure**.
5. Select **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. Select **Download**. After the vault credentials finish downloading, a pop-up asking if you want to open or **save** the credentials. Select **Save**. If you accidentally select **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.

33.3 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.

33.4 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. Select **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. Note 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.

33.5 Backup files and folders

1. Select **Back Up Now** to complete the initial sending over the network.
2. In the wizard, confirm your settings, then select **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.

33.6 Explore the recover settings

1. Select **Recover data**.
2. Walk through 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.

33.7 Explore the backup properties

1. Select **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 backup and restore activities.

33.8 Delete your backup schedule

1. Select **Schedule Backup**.
2. In the wizard, select **Stop using this backup schedule and delete all the stored backups**.
3. Verify your choices and select **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.