Contents

1 AZ-500 Azure Security (old)

A new version of AZ-500 was released 17 July There is a new lab repository to go with the content update - AZ500X-AzureSecurityTechnologies. Please move to this new version of the course and labs.

- **Are you a MCT?** - Have a look at our GitHub User Guide for MCTs
- **Need to manually build the lab instructions?** - Instructions are available in the MicrosoftLearning/Docker-Build repository

  Be sure to use the MCT Courseware Forum for suggestions or general comments on the course content. Also, bugs and course errors can be reported on the Courseware Support Forum.

To support the new changes, we introduced a new AZ-500 GitHub repository, starting on November 1 2019. At that time, all the AZ-500 labs have been replaced with this repository.

What are we doing?

- We are publishing the lab instructions and lab files on GitHub to allow for interaction between the course authors and MCTs. We hope this will help keep the content current as the Azure platform changes.
- This is an old GitHub repository for the AZ-500, Microsoft Azure Security course.
- Within each repository there are lab guides in the Markdown format in the Instructions folder. If appropriate, there are also additional files that are needed to complete the lab within the Allfiles\Labfiles folder. Not every course has corresponding lab files.
- For each delivery, trainers should download the latest files from GitHub. Trainers should also check the Issues tab to see if other MCTs have reported any errors.
- Lab timing estimates are provided but trainers should check to ensure this is accurate based on the audience.
- To do the labs you will need an internet connection and an Azure subscription. Please read the Instructor Prep Guide for more information on using the Cloud Shell.

How are we doing?

- If as you are teaching these courses, you identify areas for improvement, please use the Issues tab to provide feedback. We will periodically create new files to incorporate the changes.
- When launching Azure Cloud Shell for the first time, you will likely be prompted to create an Azure file share to persist Cloud Shell files. If so, you can typically accept the defaults, which will result in creation of a storage account in an automatically generated resource group. Note that this might happen again if you delete that storage account.
- Before you perform a template based deployments, you might need to register providers that handle provisioning of resource types referenced in the template. This is a one-time operation (per subscription) required when using Azure Resource Manager templates to deploy resources managed by these resource providers (if these resource providers have not been yet registered). You can perform registration from the subscription's Resource Providers blade in the Azure portal or by using Cloud Shell to run Register-AzResourceProvider PowerShell cmdlet or az provider Azure CLI command.

We hope using this GitHub repository brings a sense of collaboration to the labs and improves the overall quality of the lab experience.

1.1 Regards, *Azure Security Courseware Team*

1.2 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-500-Azure-Security/{{ 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-500-Azure-Security/{{ site.github.url }}{{ activity.url }}) | {% endfor %}

3 AZ500 Lab Files

This repository is a collection of setup scripts and templates for AZ500 labs. They compliment the lab exercises.

4 AZ500 Mod2 Lab1 setup

Click **Deploy to Azure**

This will deploy a new app and app service plan that can then be used to demonstrate the scale up options in AZ500 Mod2 Lab 1.

Populate the *site name, service plan name and resource group* with **unique** names.

5 AZ500 Mod2 Lab2 setup

Click **Deploy to Azure**

This will deploy a new blank function app.

Populate the *site name, service plan name and resource group* with **unique** names.

This is the yaml file to support the kubernetes deployment lab

6 AZ500 Mod2 Lab 11 setup

Click **Deploy to Azure**

This will deploy a new resource group with 1 Vnet 2 VMs and 3 Subnets for the AZ500 Mod 2 Lab 11.

The resource group name must be Test-FW-RG

7 SQL Template database deployment

Click **Deploy to Azure**, this will load the template into azure for deployment. You will then need to populate the parameters specified in the lab guide.

The test data that will be loaded into the Db is AdventureworksLT

The firewall rules for the SQL server are set to allow all azure clients to access

8 SQL Template database deployment

Click **Deploy to Azure**, this will load the template into azure for deployment. You will then need to populate the parameters specified in the lab guide.

The test data that will be loaded into the Db is AdventureworksLT

The firewall rules for the SQL server are set to allow all azure clients to access

9 GoDeploy SQL Template database deployment

Click **Deploy to Azure**, this will load the template into azure for deployment. You will then need to populate the parameters specified in the lab guide. Alternatively the parameters are also in this JSON file that you can copy into the template deployment wizard

The test data that will be loaded into the Db is AdventureworksLT

The firewall rules for the SQL server are set to allow all azure clients to access

This repository is a collection of lab manuals for the AZ500 labs. They compliment the AZ500 Azure Security course.

10 Module 1: Lab 1 - Azure AD Privileged Identity Management

Scenario

In this lab, you'll learn how to use Azure Privileged Identity Management (PIM) to enable just-in-time administration and control the number of users who can perform privileged operations. You'll also learn about the different directory roles available as well as newer functionality that includes PIM being expanded to role assignments at the resource level. Lessons include:

- Getting Started with PIM
- PIM Security Wizard
- PIM for Directory Roles
- PIM for Role Resources

The Managing Identities course also covers Azure RBAC and Azure Active Directory. This content has been included here also to provide more context and foundation for the remainder of the course.

10.1 Azure AD Privileged Identity Management

10.2 Exercise 1 - Discover and Manage Azure Resources

10.2.1 Task 1: Lab Setup

This lab requires creating a user that will be used for PIM.

1. In the **Azure Portal** open the **Cloud Shell** in **PowerShell** mode. If prompted click **Create Storage**.
2. Run the following command to authenticate

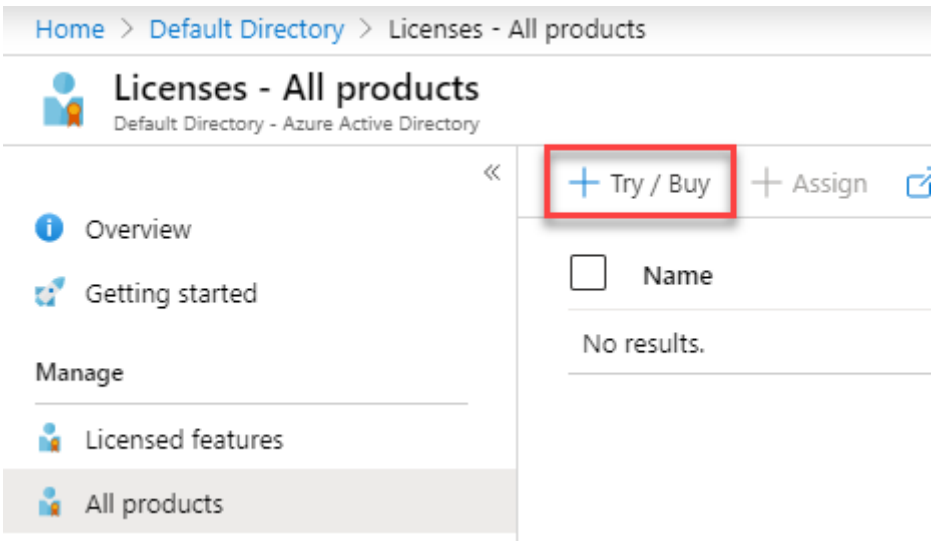
   ```
   Connect-AzureAD
   ```

   Note: If you close your Cloud Shell session you may be required to enter this command again throughout the labs.
3. Run the following PowerShell Commands to create an AD user and password in your default domain

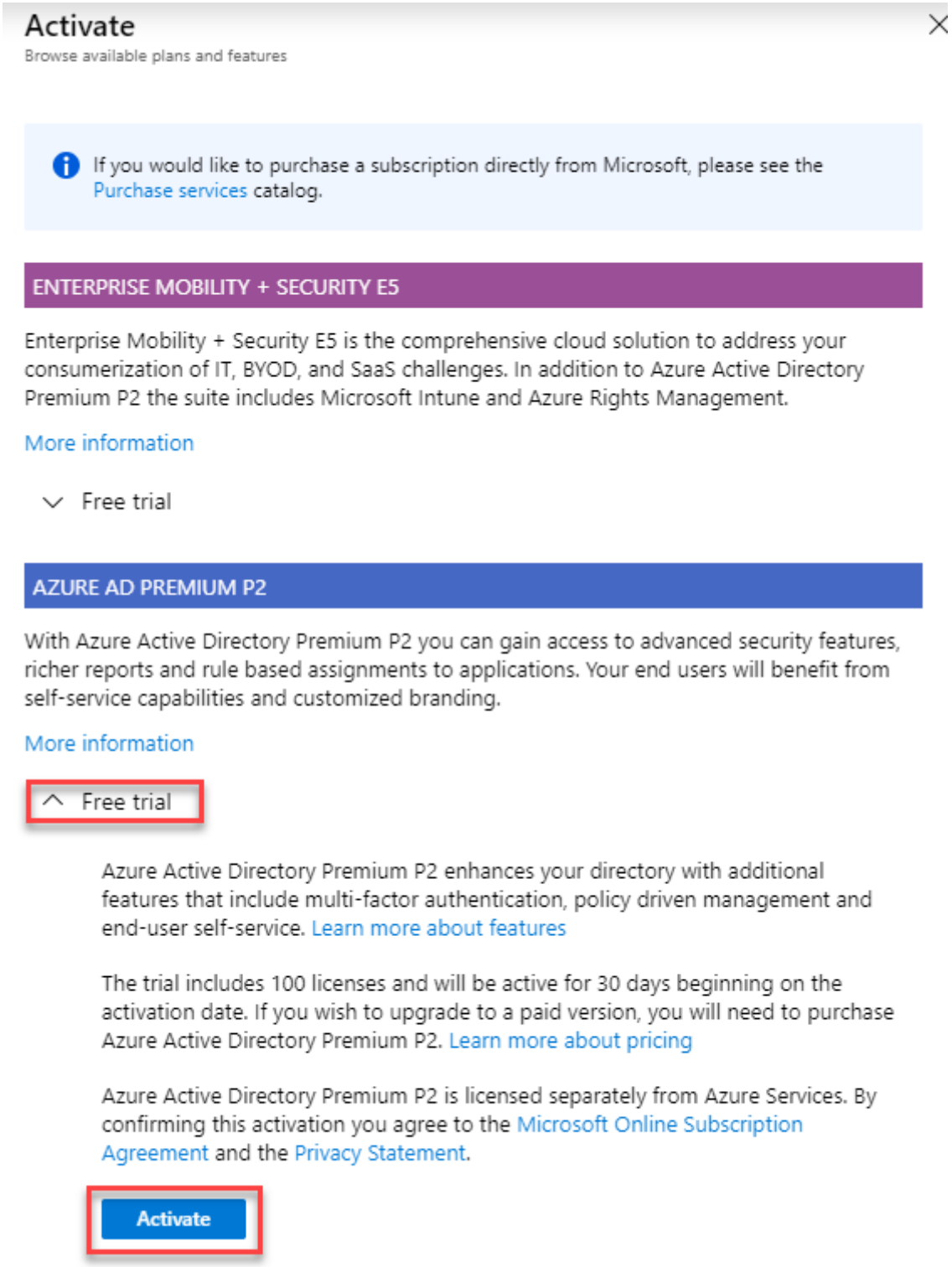
   ```
   $PasswordProfile = New-Object -TypeName Microsoft.Open.AzureAD.Model.PasswordProfile

   $PasswordProfile.Password = "Pa55w.rd"

   $domainObj = get-azureaddomain

   $domain = $domainObj[0].name

    New-AzureADUser -DisplayName "Isabella Simonsen" -PasswordProfile $PasswordProfile -UserPrincipall
   ```

10.2.2 Task 2: Enable Azure AD Premium P2 trial and create a test user.

1. In the Azure Portal, on the Hub menu click **Azure Active Directory**.
2. Select **Licences** then **All Products**.
3. Click **Try / Buy**.



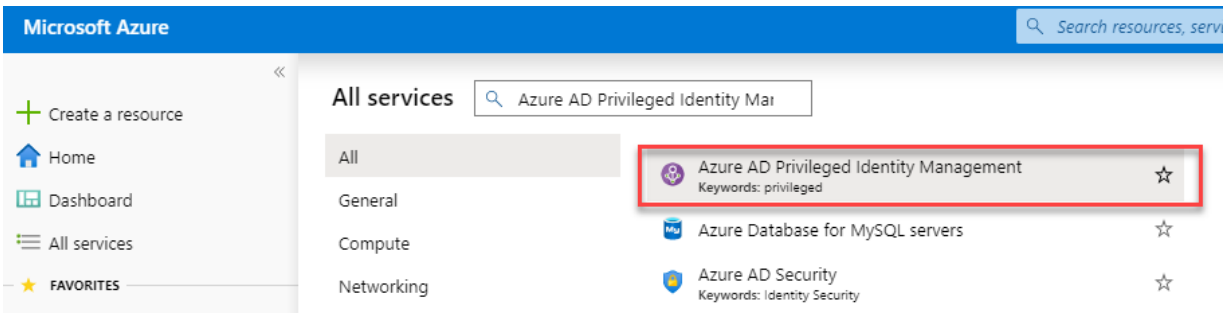
4. Click the drop down arrow and select **Activate** on the **Azure AD Premium P2** product.



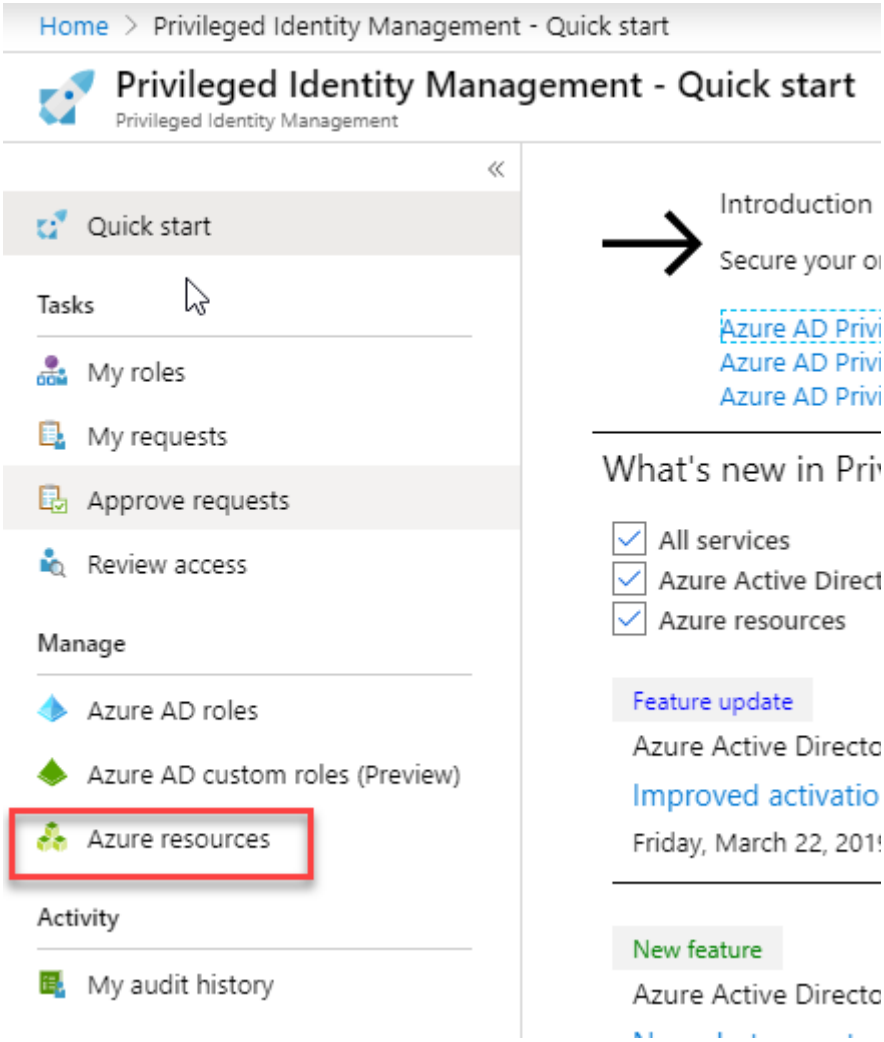
You may need to log out of the Azure portal and log in again for this to refresh

10.2.3 Task 3: Discover resources

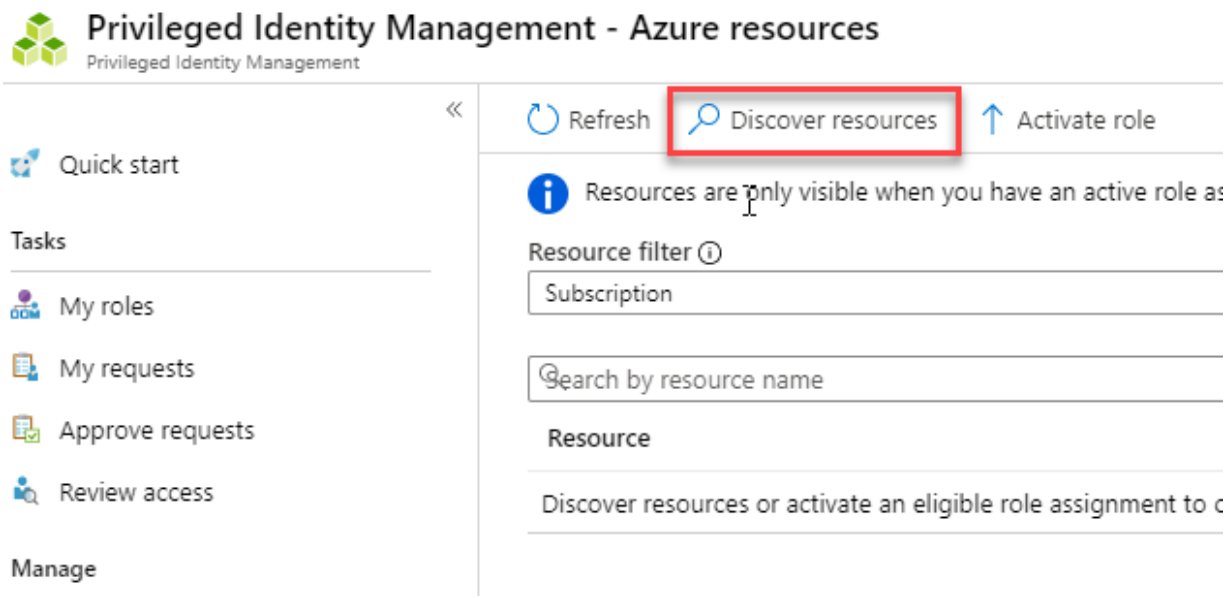
1. In the Azure Portal, click **All services** and search for and select **Azure AD Privileged Identity Management**.



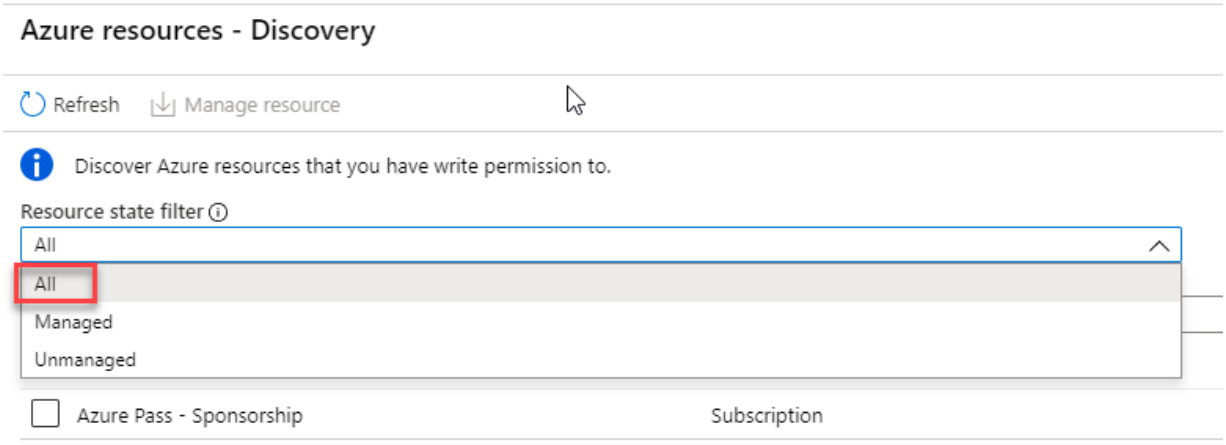
2. Click **Azure resources**.



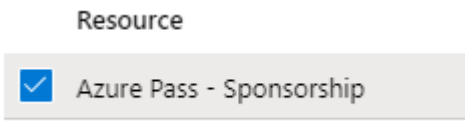
3. Click **Discover resources** to launch the discovery experience.



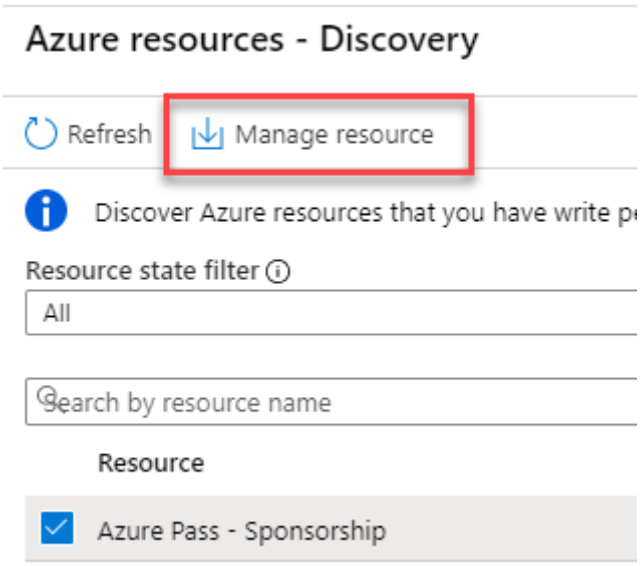
4. On the Discovery pane, use **Resource state filter** and **Select resource type** to filter the management groups or subscriptions you have write permission to. It's probably easiest to start with **All** initially.



5. Add a checkmark next to your Azure subscription.



6. Click **Manage resource** to start managing the selected resources.



7. Click **Yes** when prompted.

   Note: You can only search for and select management group or subscription resources to manage using PIM. When you manage a management group or a subscription in PIM, you can also manage its child resources.

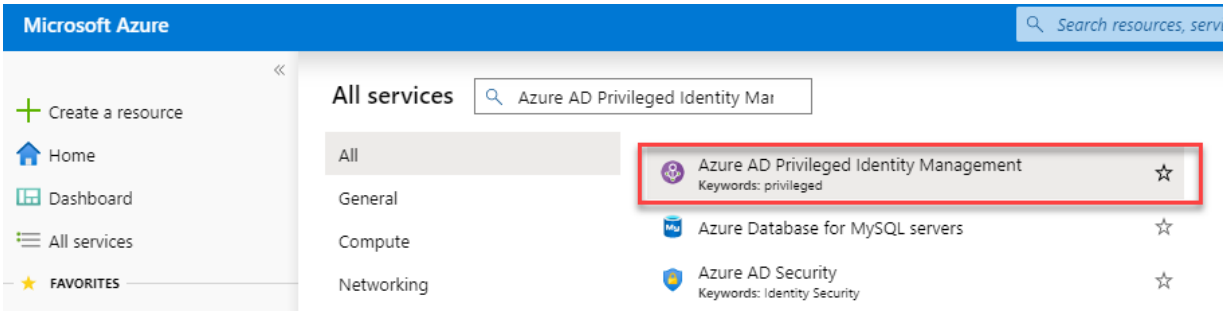
   Note: Once a management group or subscription is set to managed, it can't be unmanaged. This prevents another resource administrator from removing PIM settings.

10.3 Exercise 2 - Assign Directory Roles

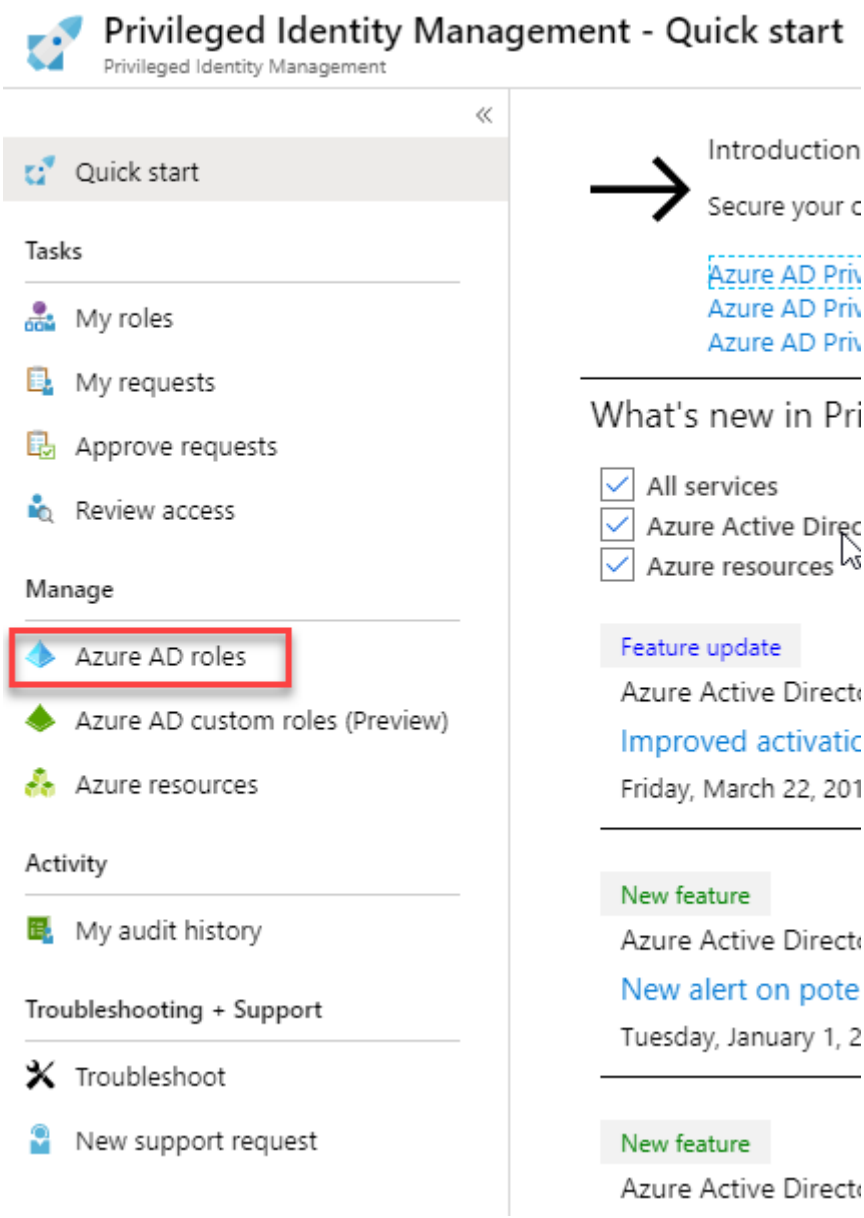
10.3.1 Task 1: Make a user eligible for a role

In the following task you will make a user eligible for an Azure AD directory role.

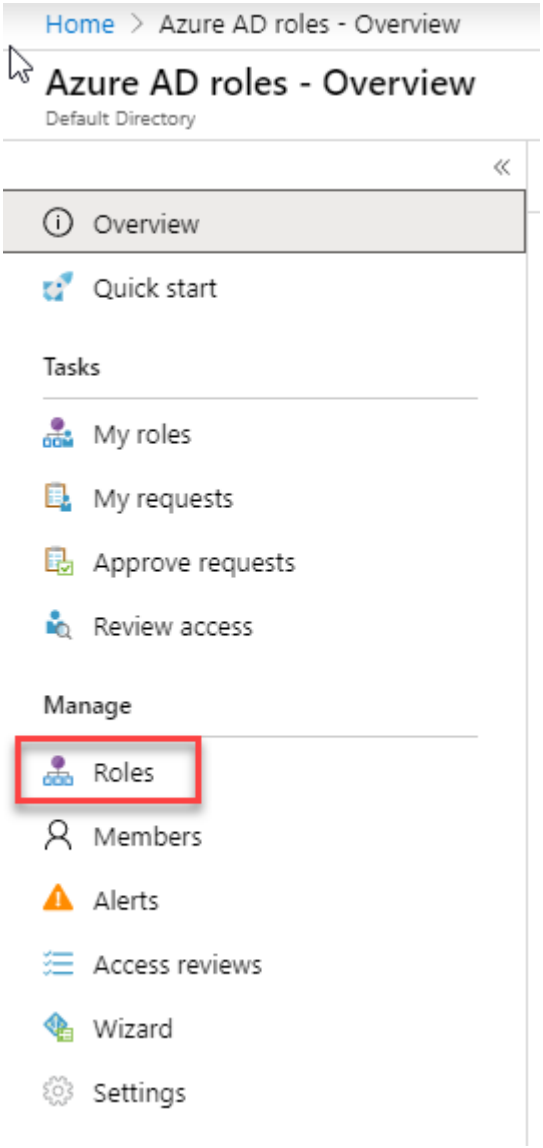
1. Sign in to Azure portal
2. In the Azure Portal, click **All services** and search for and select **Azure AD Privileged Identity Management**.



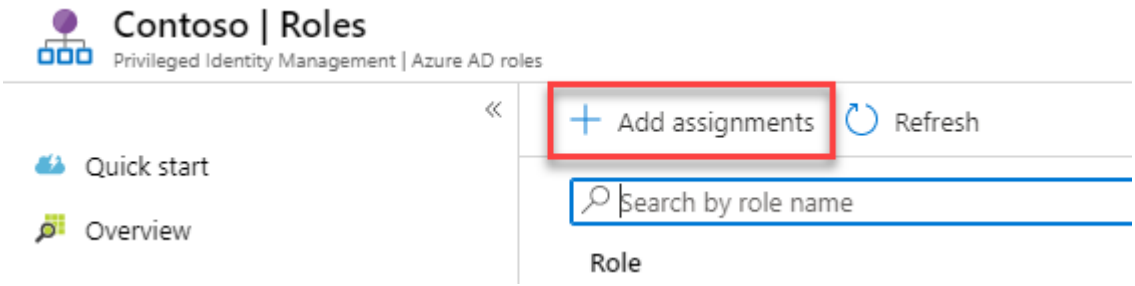
3. Select **Azure AD Roles**.



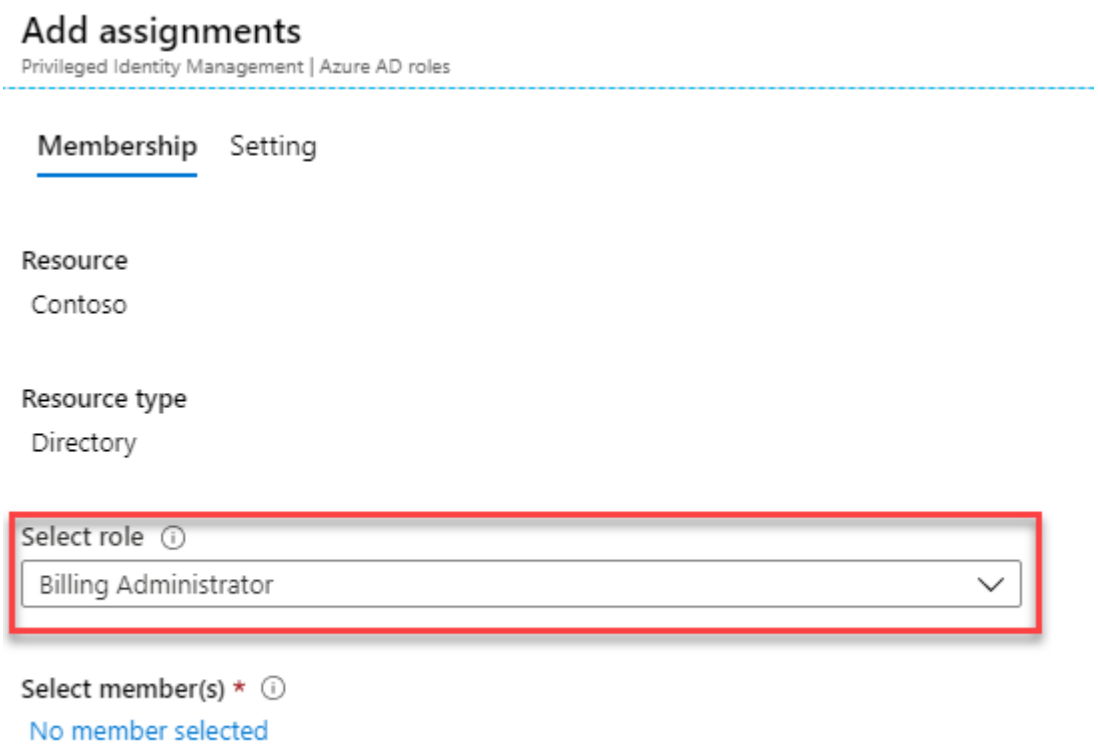
4. Click **Roles**.



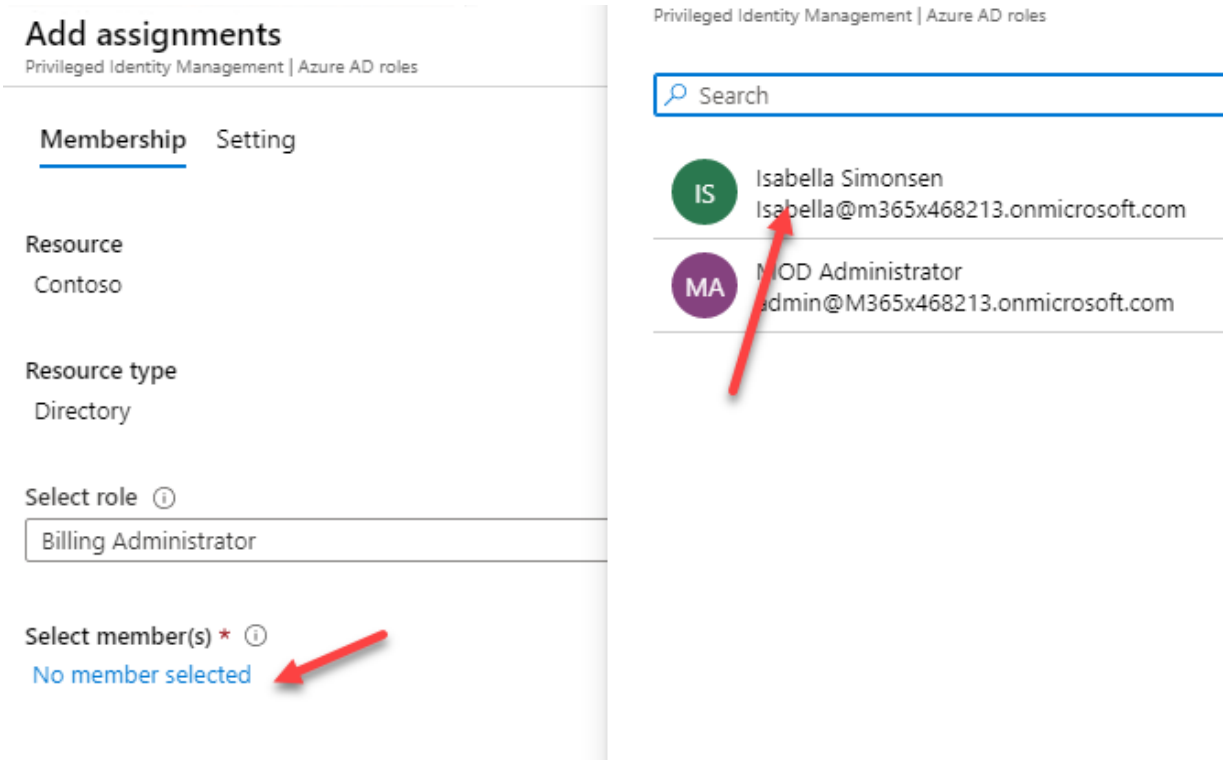
5. Click **Add assignments** to open Add managed members.



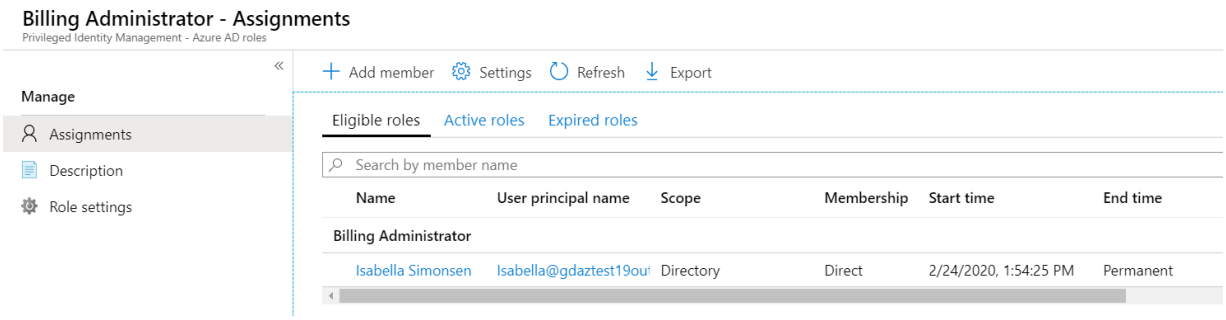
6. Click the **Select role** dropdown and select **Billing Administrator**.



7. Click **No member selected**, select **Isabella** and then click **Select**.



8. On the Add assignments blade, click **Next**.
9. Click **Assign** to add the user to the role.
10. Select the **Billing Administrator** role.
11. Review the added assignment.



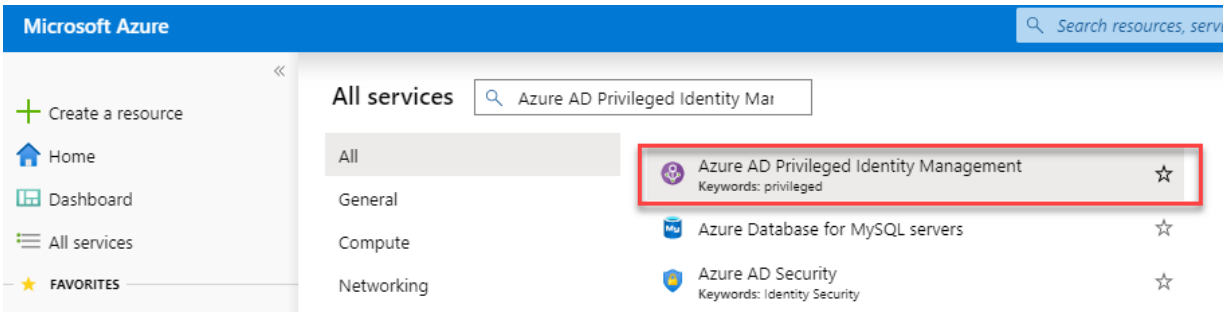
12. When the role is assigned, the user you selected will appear in the members list as **Eligible** for the role.

10.4 Exercise 3 - Activate and Deactivate PIM Roles

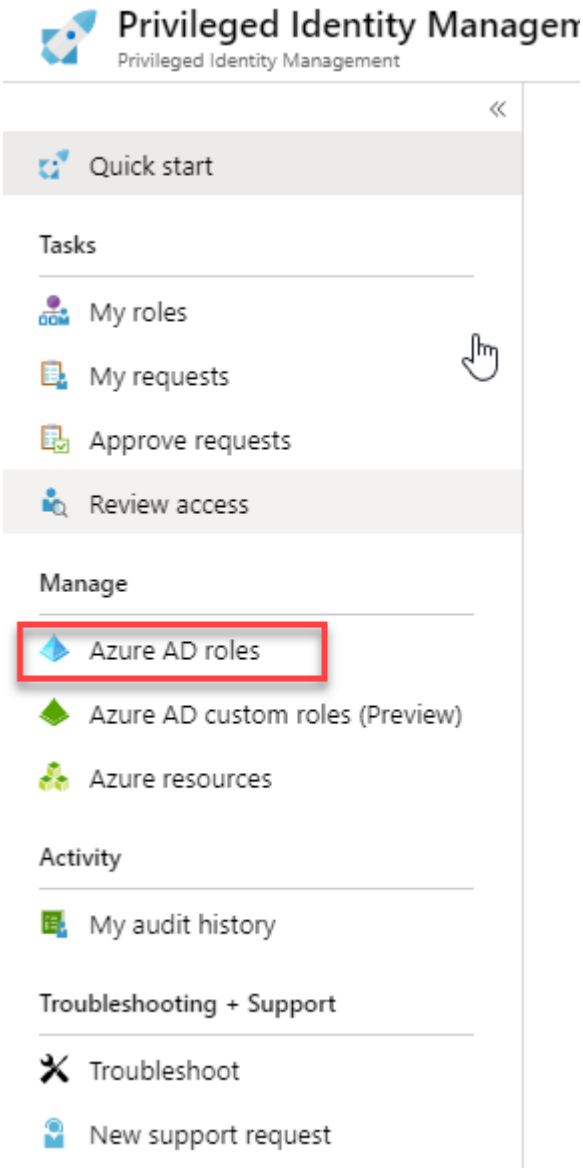
10.4.1 Task 1: Activate a role

When you need to take on an Azure AD directory role, you can request activation by using the **My roles** navigation option in PIM.

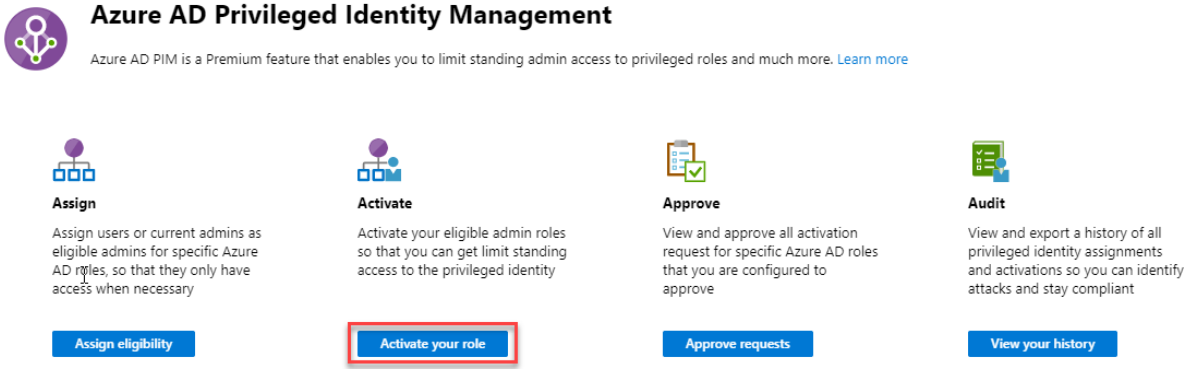
1. Open an **In Private** browsing session and navigate to `https://portal.azure.com` and login as **Isabella** using her UPN. example Isabella@myaad.onmicrosoft.com with the password **Pa55w.rd**. When prompted change Isabella's password.
2. In the Azure Portal, click **All services** and search for and select **Azure AD Privileged Identity Management**.



3. Click **Azure AD roles**.

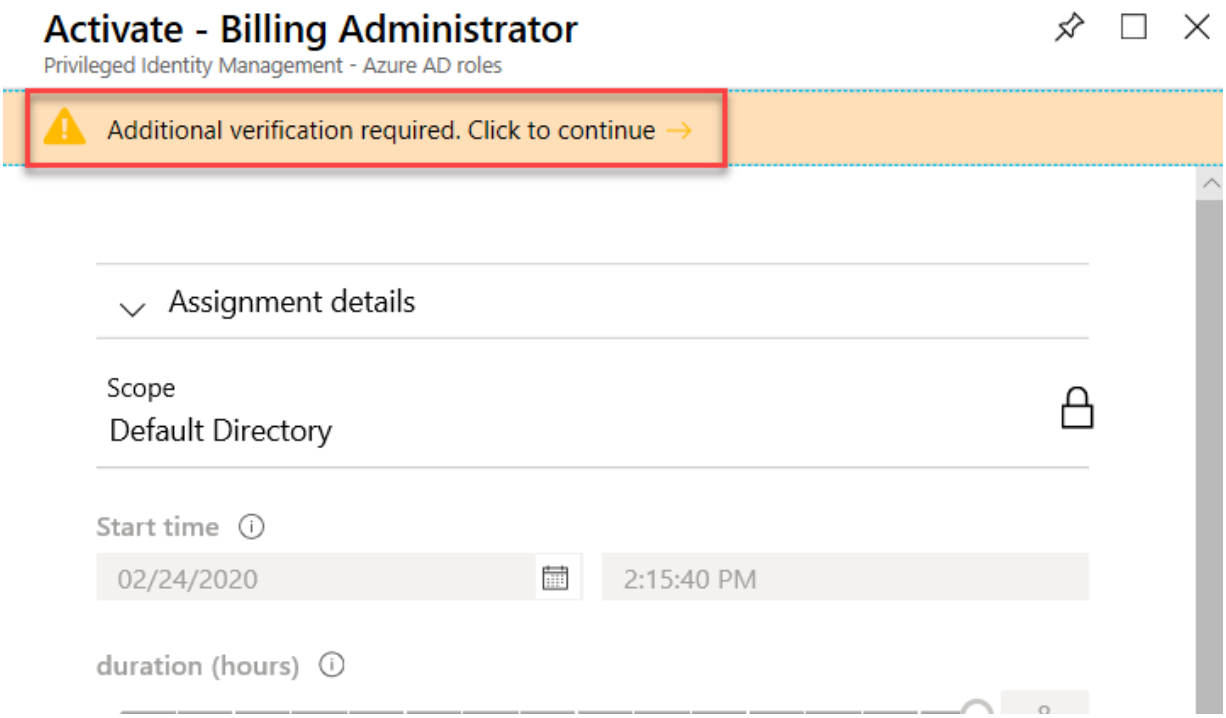


4. On the **Quick start** blade click **Activate your role**.

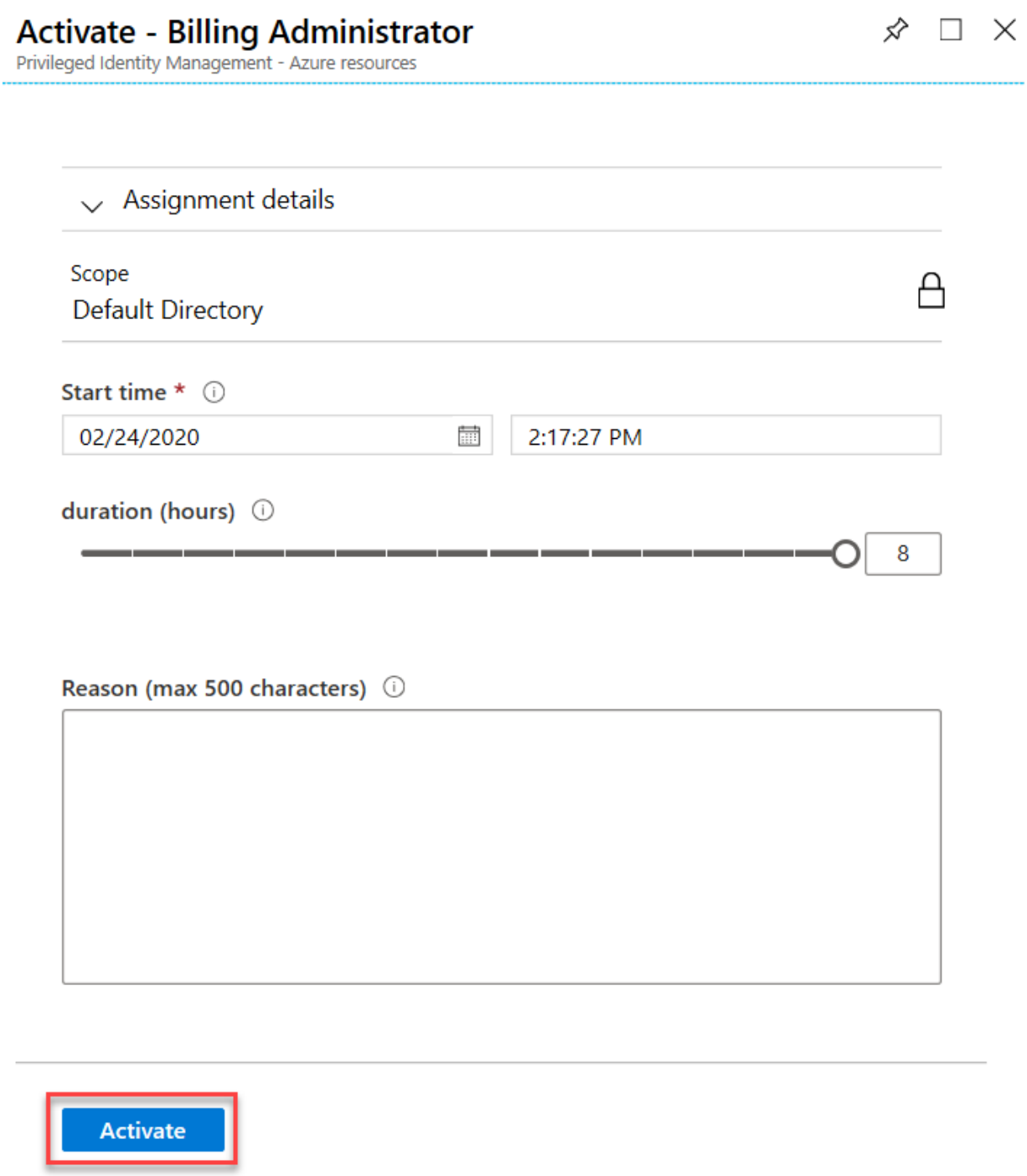


5. On the Billing Administrator role, scroll to the right and click **Activate**.

6. Click **Additional verification required. Click to continue**. You only have to authenticate once per session. Run through the wizard to authenticate Isabella.



7. Once returned to the Azure Portal, enter an activation reason and click **Activate**.



By default, roles do not require approval unless configured explicitly in settings.

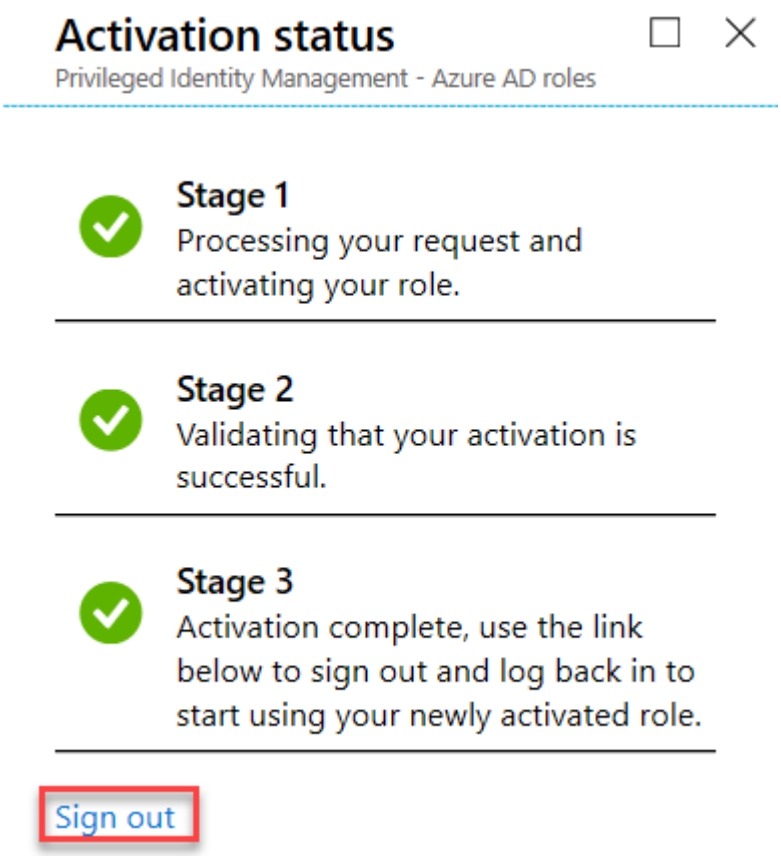
If the role does not require approval, it is activated and added to the list of active roles. If you want to use the role right away, follow the steps in the next section.

If the role requires approval to activate, a notification will appear in the upper right corner of your browser informing you the request is pending approval.

10.4.2 Task 2: Use a role immediately after activation

When you activate a role in PIM, it can take up to 10 minutes before you can access the desired administrative portal or perform functions within a specific administrative workload. To force an update of your permissions, use the **Application access** page as described in the following steps.

1. Click **Sign Out**.



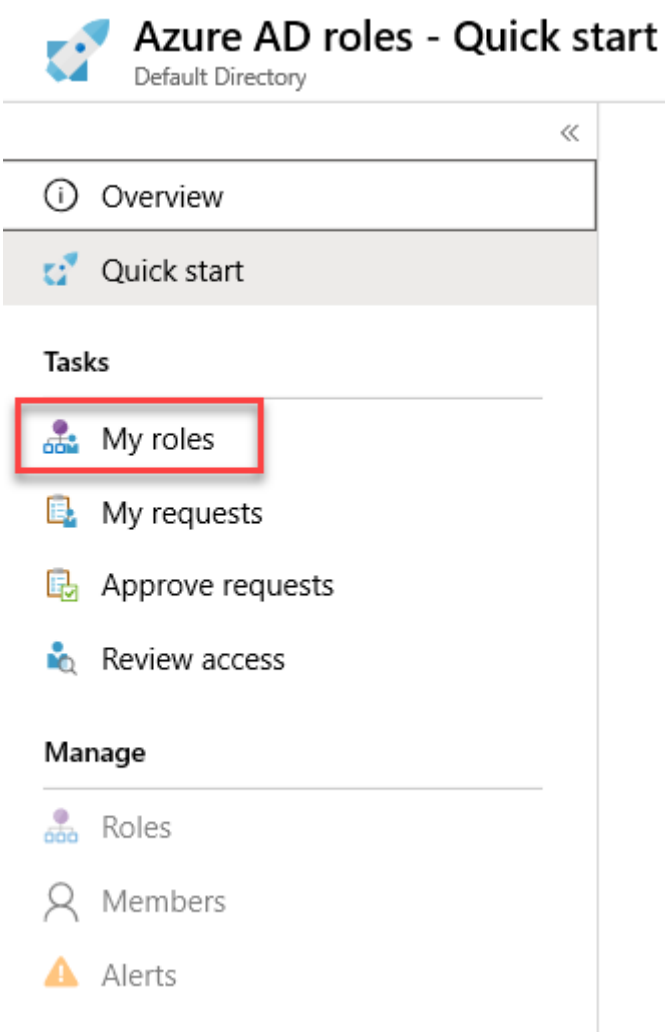
2. Log back in as Isabella.

10.4.3 Task 3: Deactivate a role

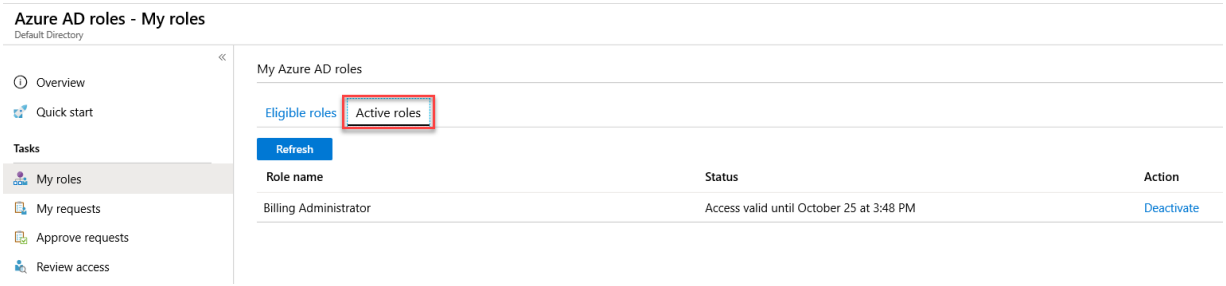
Once a role has been activated, it automatically deactivates when its time limit (eligible duration) is reached.

If you complete your administrator tasks early, you can also deactivate a role manually in Azure AD Privileged Identity Management.

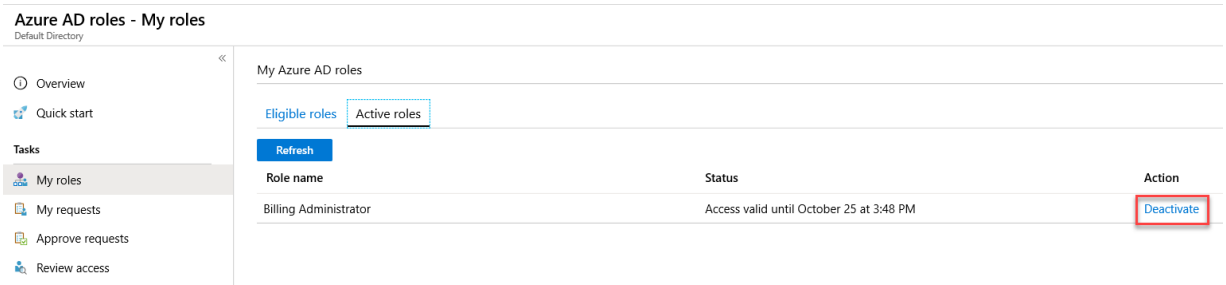
1. Still signed in as **Isabella**, open Azure AD Privileged Identity Management.
2. Click **Azure AD roles**.
3. Click **My roles**.



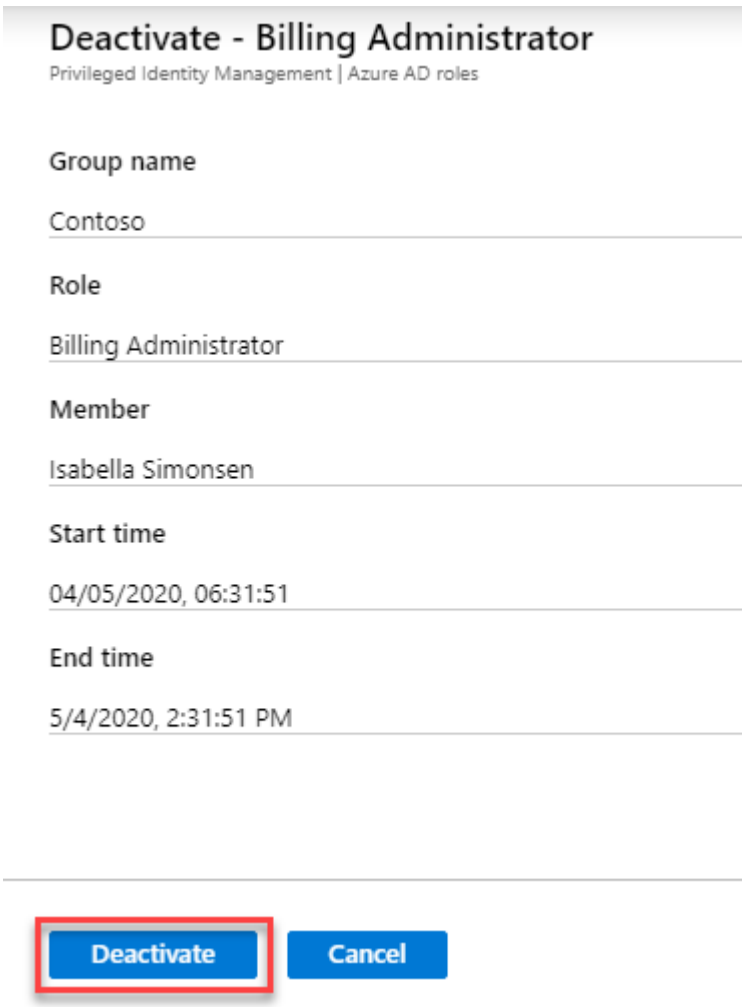
4. Click **Active roles** to see your list of active roles.



5. Find the role you're done using and then click **Deactivate**.



6. Click **Deactivate** again.



10.4.4 Task 4: Cancel a pending request

If you do not require activation of a role that requires approval, you can cancel a pending request at any time.

1. **Open Azure AD Privileged Identity Management**.
2. Click **Azure AD roles**.
3. Click **Pending requests**.
4. For the role that you want to cancel, click the **Cancel** button.

Note: The cancel button in this task is greyed out as the request was approved.

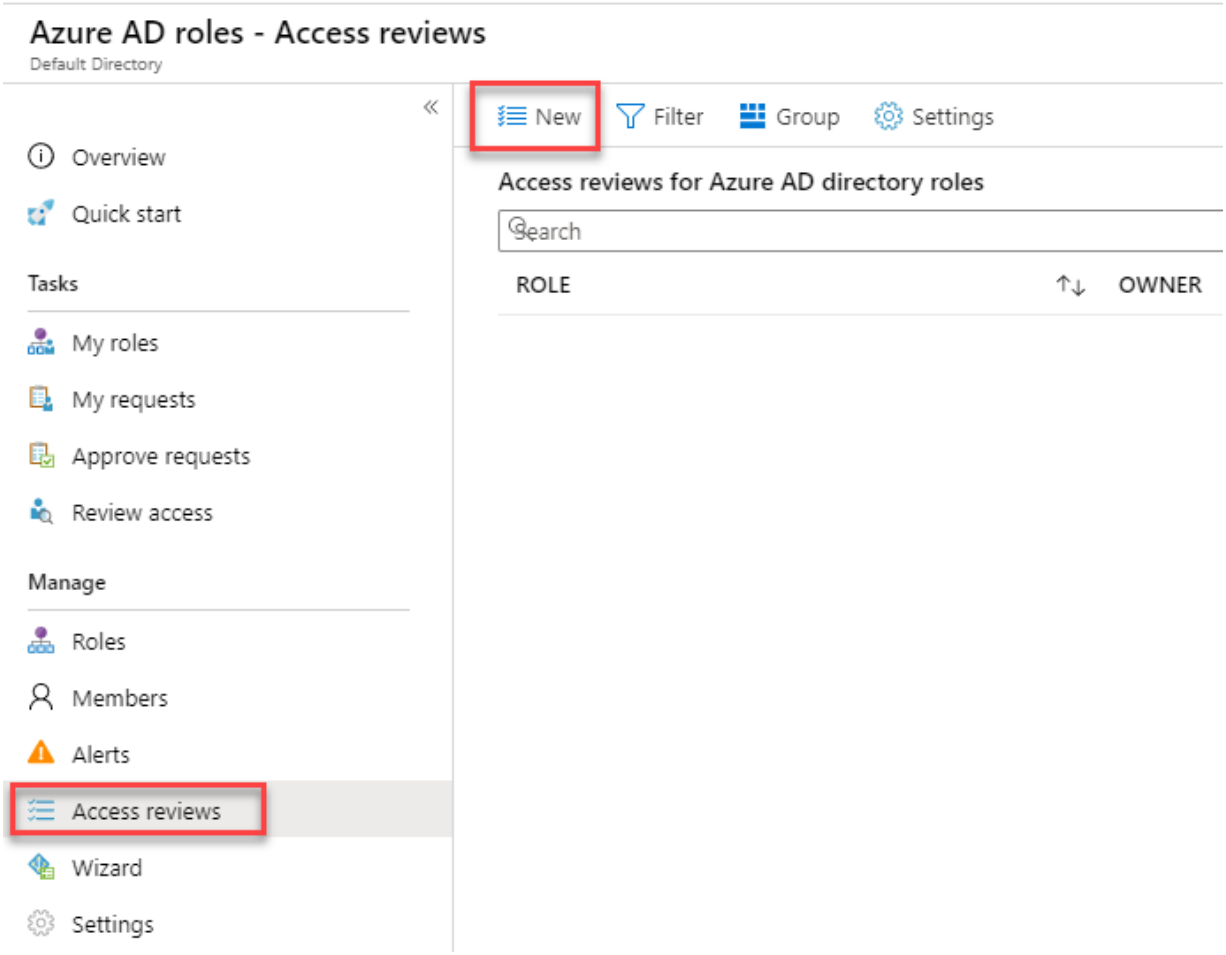
When you click Cancel, the request will be cancelled. To activate the role again, you will have to submit a new request for activation.

10.5 Exercise 4 - Directory Roles (General)

10.5.1 Task 1: Start an access review for Azure AD directory roles in PIM

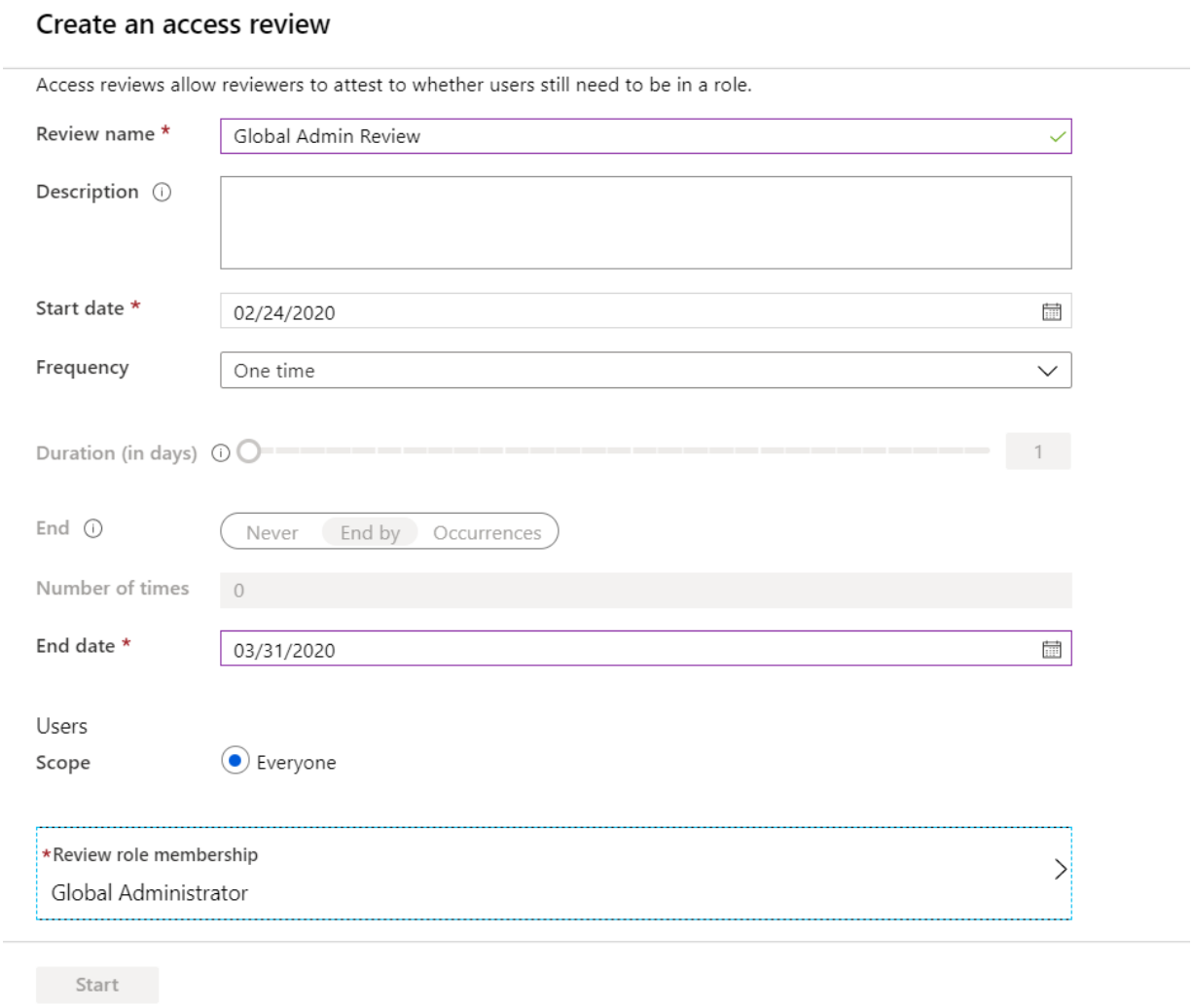
Role assignments become ”stale” when users have privileged access that they don't need anymore. In order to reduce the risk associated with these stale role assignments, privileged role administrators or global administrators should regularly create access reviews to ask admins to review the roles that users have been given. This task covers the steps for starting an access review in Azure AD Privileged Identity Management (PIM).

1. Return back to the browser that is logged in as your Global Admin Account.
2. From the PIM application main page click **Azure AD Roles** under the **Manage** section click **Access reviews** and click > **New**.



3. Enter the following details and click **Start**:

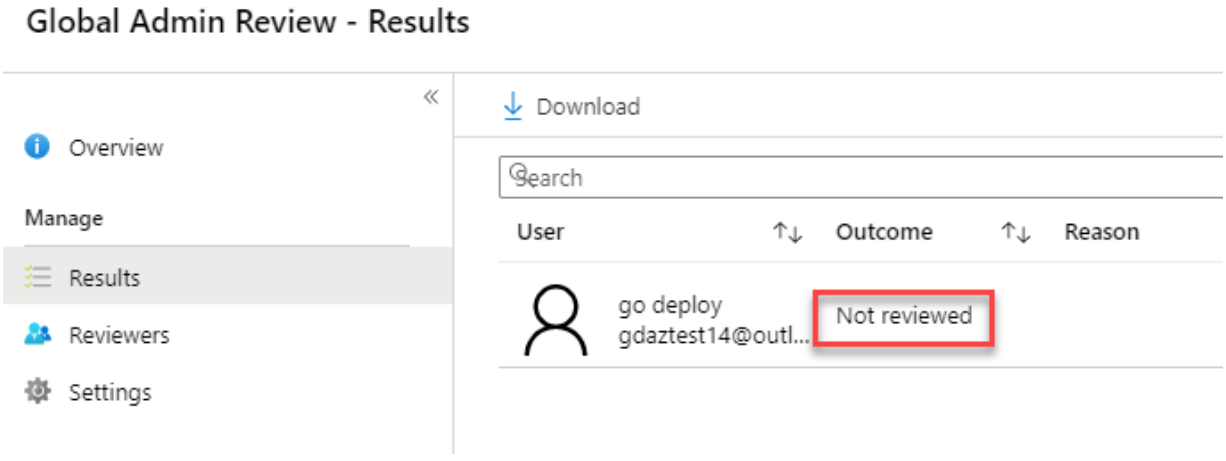
- Review name: **Global Admin Review**
- Start Date: **Today's Date**
- Frequency: **One time**
- End Date: **End of next month**
- Review role membership: **Global Administrator**
- Reviewers: **Select your account**



4. Once the review has completed and has a status of Active, click on the **Global Admin Review**.

   Note: You may have to refresh your browser.

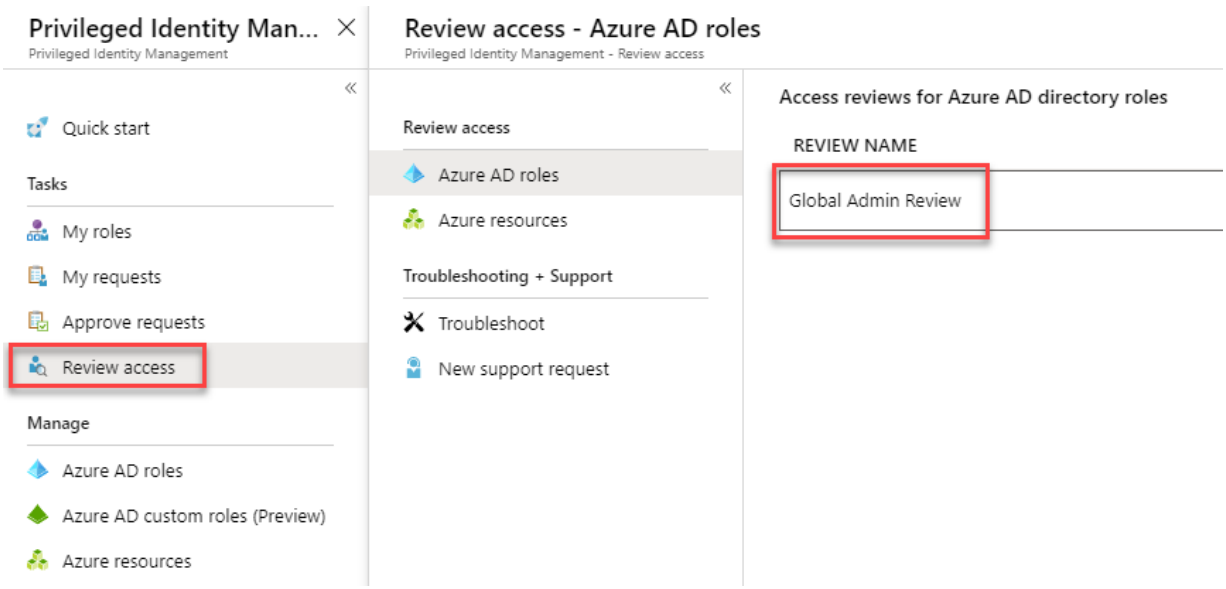
5. Select **Results** and see the outcome of **Not reviewed**.



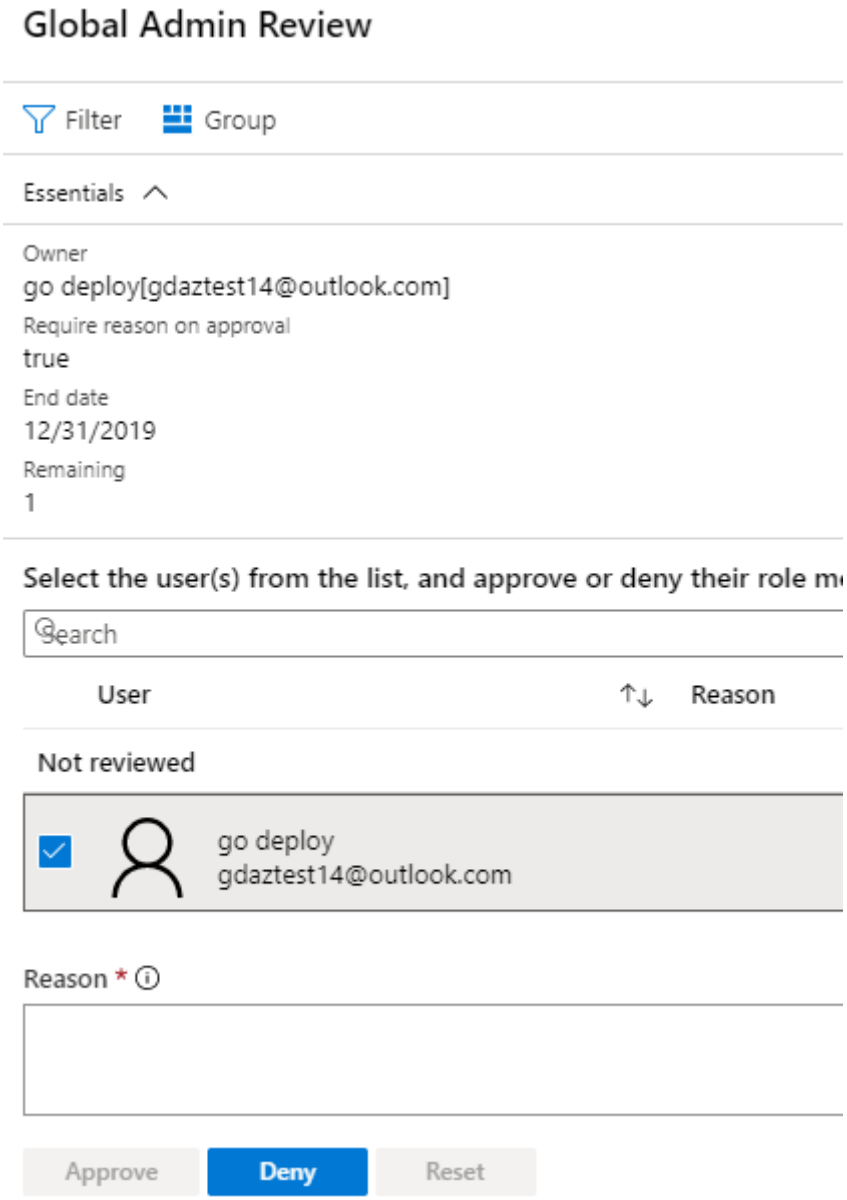
10.5.2 Task 2: Approve or deny access

When you approve or deny access, you're just telling the reviewer whether you still use this role or not. Choose Approve if you want to stay in the role, or Deny if you don't need the access anymore. Your status won't change right away, until the reviewer applies the results. Follow these steps to find and complete the access review:

1. In the PIM application, select **Review access**.
2. Select the **Global Admin Review**.



3. Since you created the review, you appear as the only user in the review. Select the check mark next to your name.



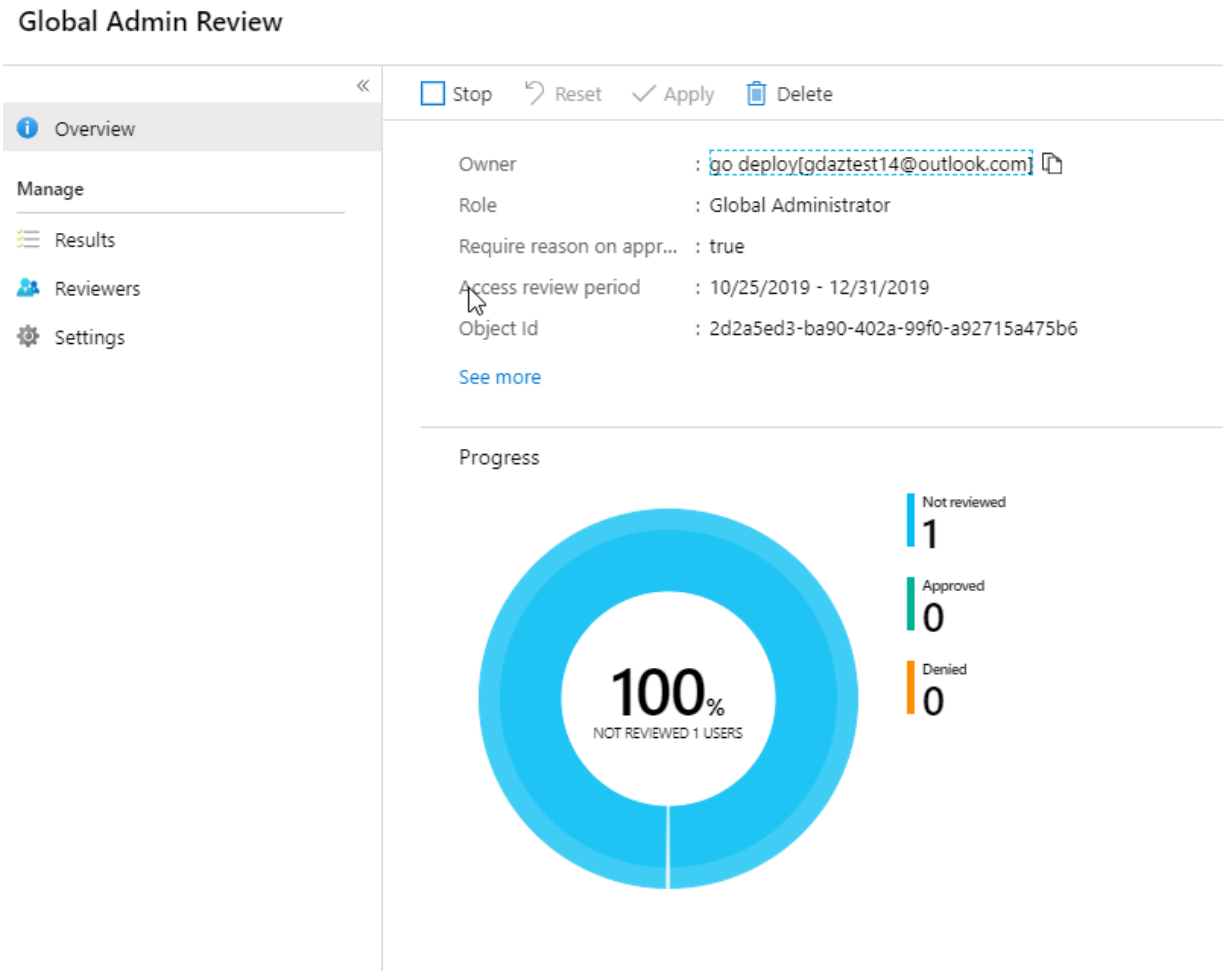
4. Close the **Review Azure AD roles** blade.

10.5.3 Task 3: Complete an access review for Azure AD directory roles in PIM

Privileged role administrators can review privileged access once an access review has been started. Azure AD Privileged Identity Management (PIM) will automatically send an email prompting users to review their access. If a user did not get an email, you can send them the instructions in how to perform an access review.

After the access review period is over, or all the users have finished their self-review, follow the steps in this task to manage the review and see the results.

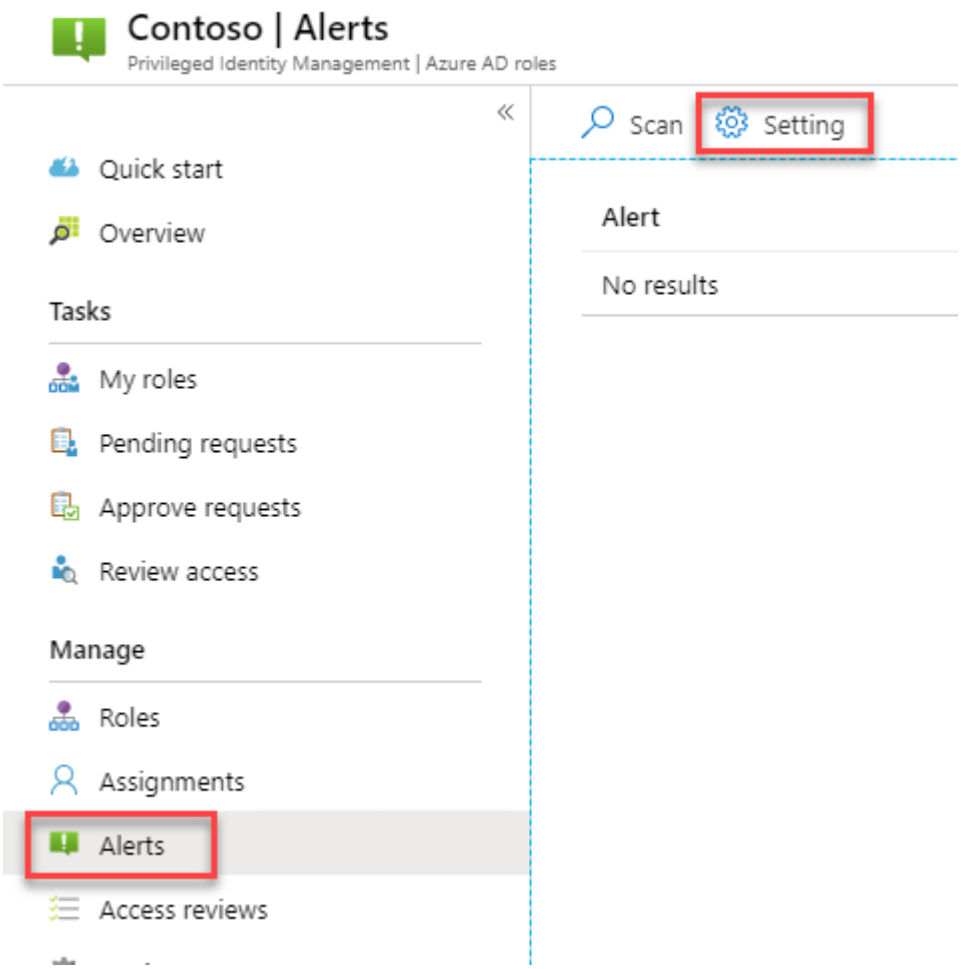
1. Go to the Azure portal and select the **Azure AD Privileged Identity Management**.
2. Select **Azure AD Roles**.
3. Select the **Access reviews**.
4. Select the Global Admin Review.
5. Review the blade.



10.5.4 Task 4: Configure security alerts for Azure AD directory roles in PIM

You can customize some of the security alerts in PIM to work with your environment and security goals. Follow these steps to open the security alert settings:

1. Open **Azure AD Privileged Identity Management**.
2. Click **Azure AD roles**.
3. Click **Alerts** and then **Setting**.

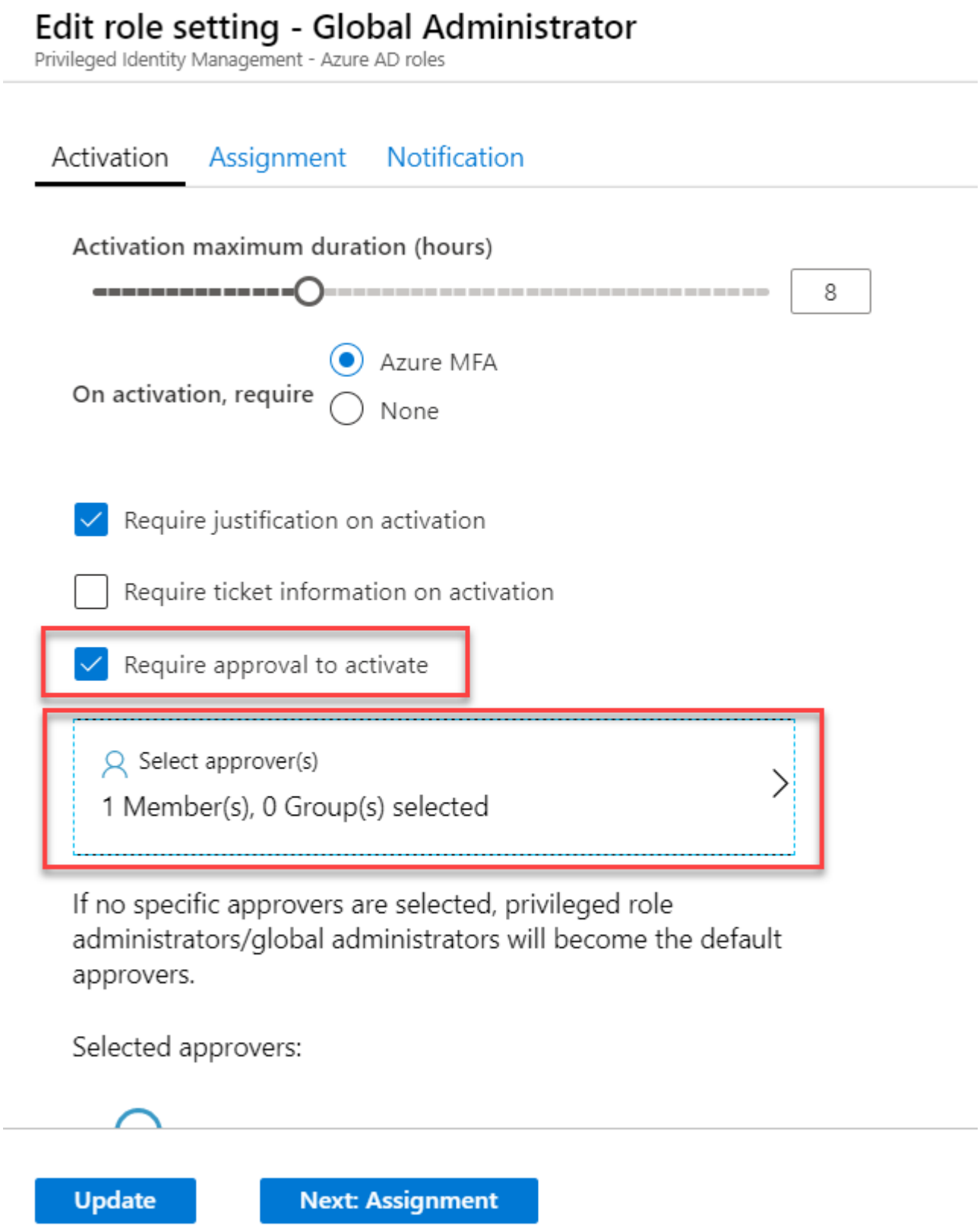


4. Click an alert name to see the settings for the preconfigured alerts.

10.6 Exercise 5 - PIM Resource Workflows

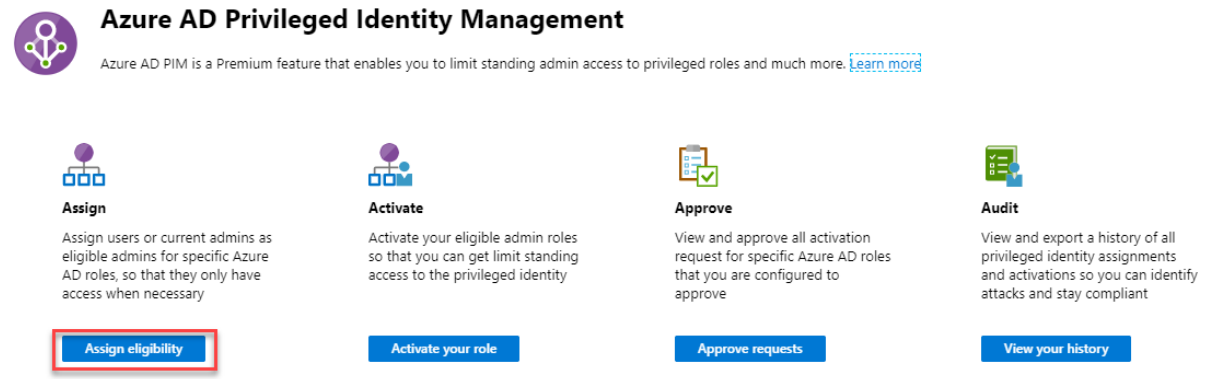
10.6.1 Task 1: Configure the Global Administrator role to require approval.

1. Open **Azure AD Privileged Identity Management**.
2. Click **Azure AD roles**.
3. Click **Roles** and select **Global Administrator**.
4. Click on **Role settings**.
5. On the **Role setting** blade, click on **Edit**.
6. Scroll down and select **Require Approval** and select your account as the approver then click **Select**.
7. On the **Edit role setting – Global Administrator blade**, click **Update**.

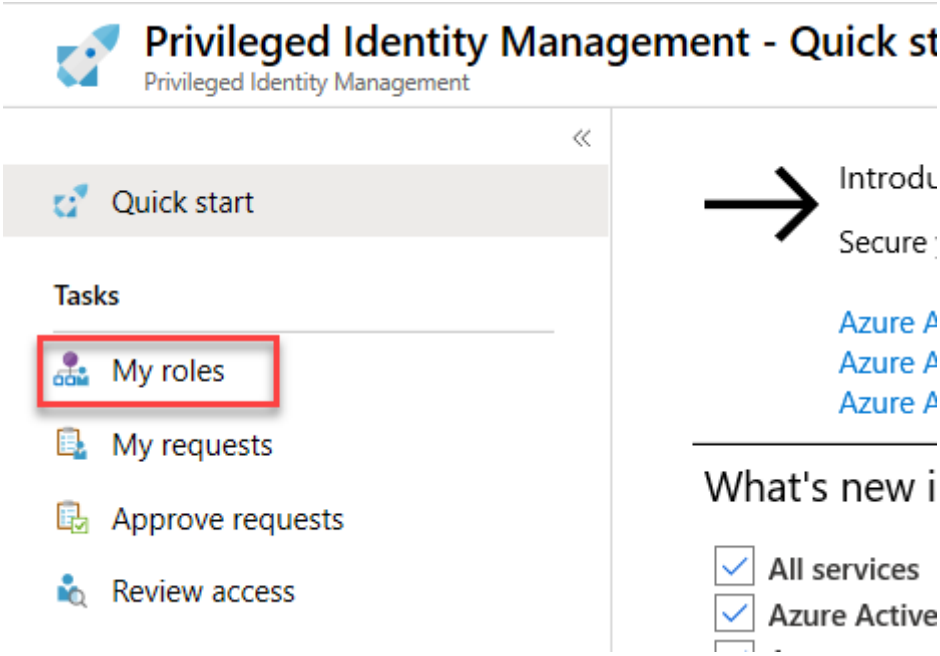


10.6.2 Task 2: Enable Isabella for Global Administrator privileges.

1. Open **Azure AD Privileged Identity Management**.
2. Click **Azure AD roles**.
3. On the **Quick Start** blade, select **Assign eligibility**.

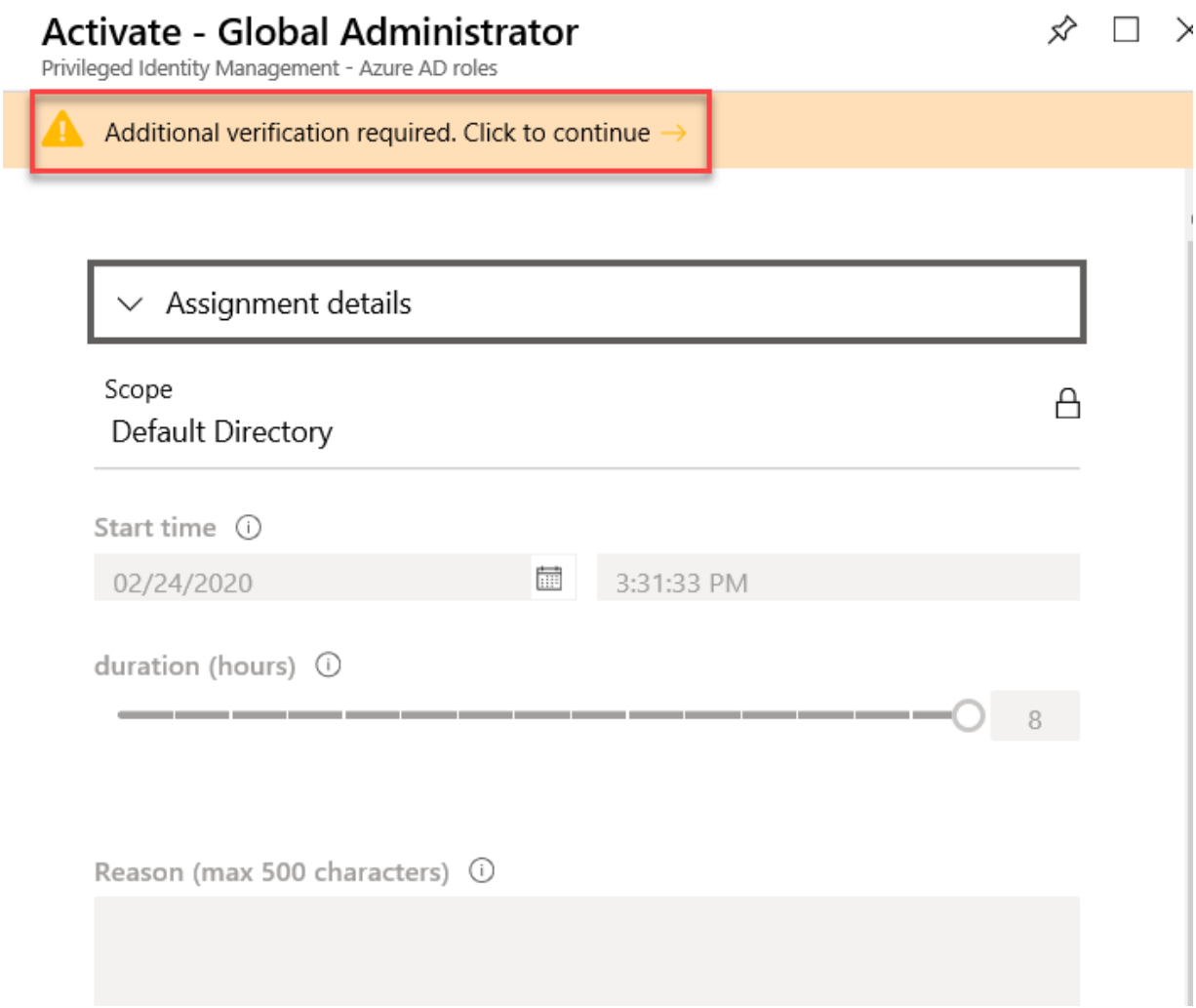


4. Select **Global Administrator** and click + **Add assignments**.
5. On the **Add assignments** blade, under **Select member(s) *** click **No member selected**, select **Isabella** and click **Select** > **Next** and then **Assign**.
6. Open an in Private Browsing session and login to portal.azure.com as Isabella.
7. Open **Azure AD Privileged Identity Management**.
8. Select **My Roles**.

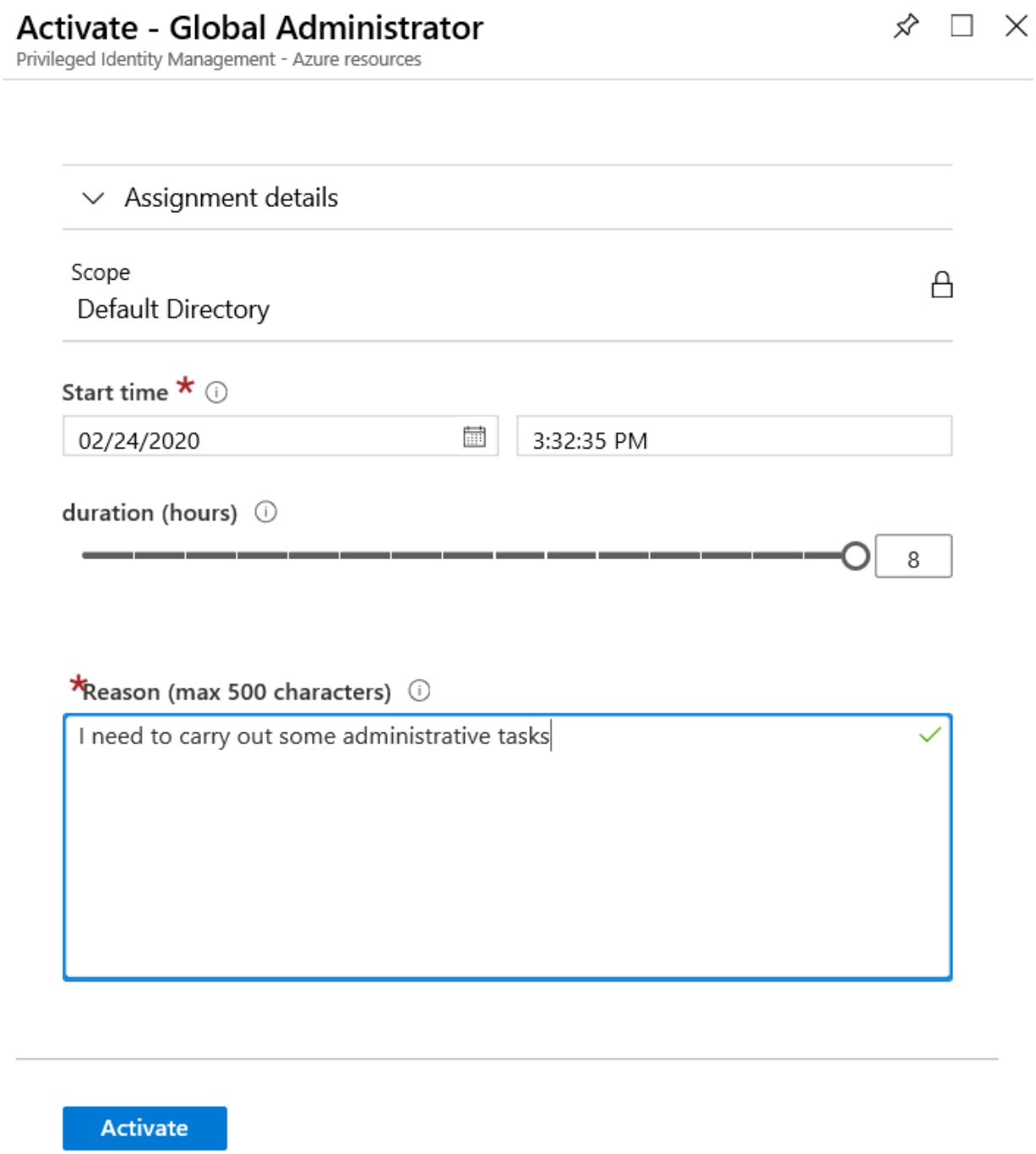


9. **Activate** the Global Administrator Role.

10. Verify Isabella's identity using the wizard.



11. Once you are returned to the **Activate - Global Administrator** blade, enter the justification **I need to carry out some administrative tasks** and click **Activate**.



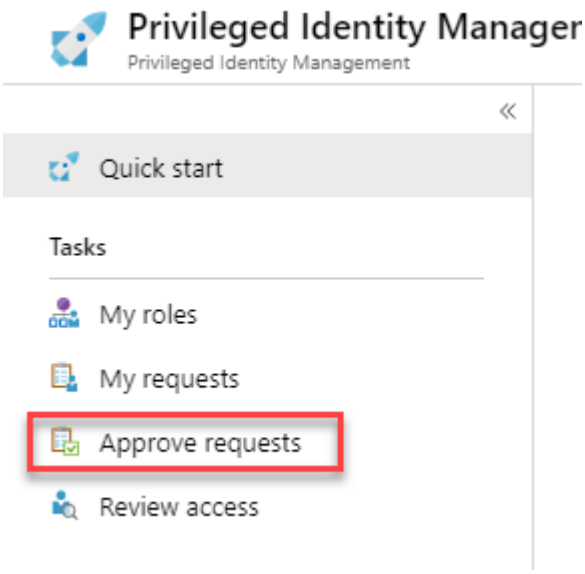
10.6.3 Task 3: Approve or deny requests for Azure resource roles in PIM

With Azure AD Privileged Identity Management (PIM), you can configure roles to require approval for activation, and choose one or multiple users or groups as delegated approvers. Follow the steps in this article to approve or deny requests for Azure resource roles.

10.6.3.1 View pending requests

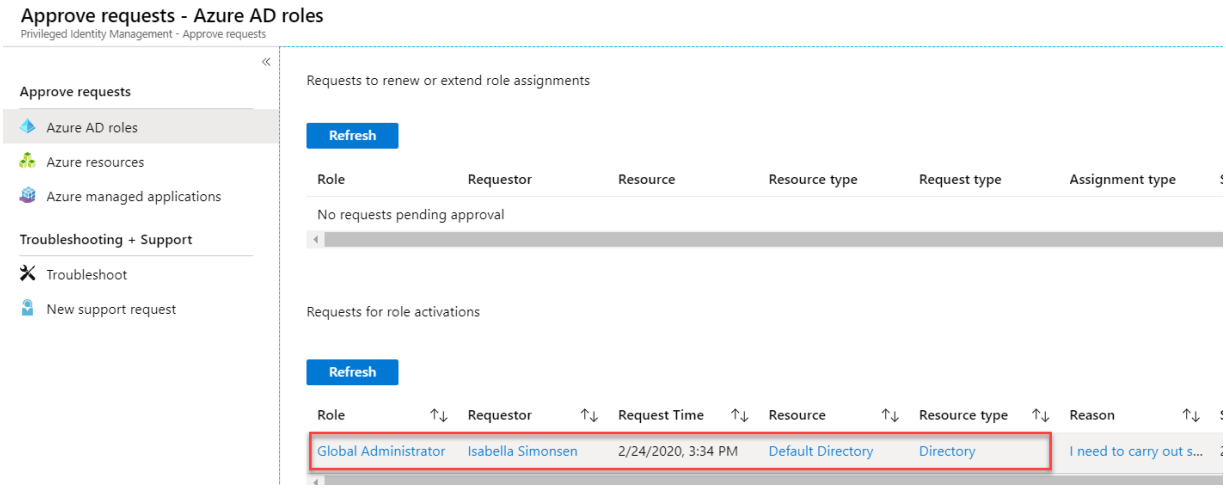
As a delegated approver, you'll receive an email notification when an Azure resource role request is pending your approval. You can view these pending requests in PIM.

1. Switch back to the browser you are signed in with your Global Administrative account.
2. Open **Azure AD Privileged Identity Management**.
3. Click **Approve requests**.

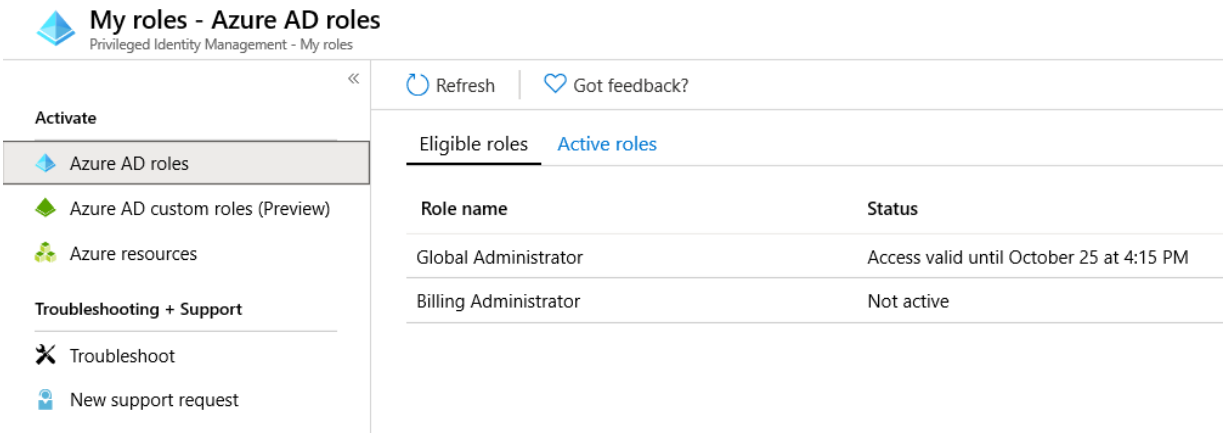


Note: You may need to refresh your browser to see the request.

4. Click the request from Isabella and enter the justification **Granted for this task** and click **Approve**.



5. Switch back to the In Private Browsing session where Isabella is signed in and click My Roles and click the Active roles tab. Note the status.



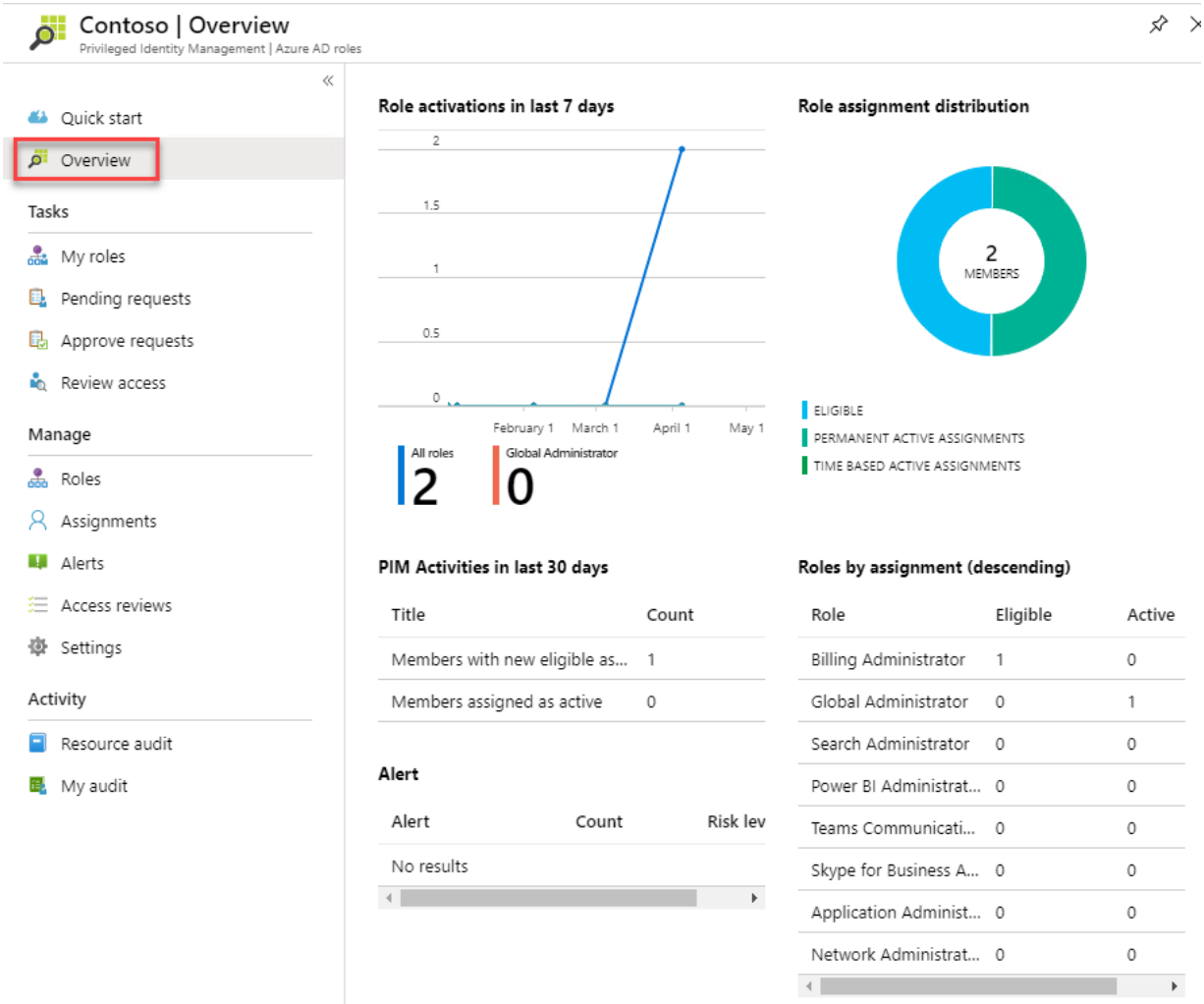
10.7 Exercise 6 - View audit history for Azure AD roles in PIM

You can use the Azure Active Directory (Azure AD) Privileged Identity Management (PIM) audit history to see all the role assignments and activations within the past 30 days for all privileged roles. If you want to see the full audit history of activity in your directory, including administrator, end user, and synchronization activity, you can use the Azure Active Directory security and activity reports.

10.7.1 Task 1: View audit history

Follow these steps to view the audit history for Azure AD roles.

1. Open **Azure AD Privileged Identity Management**.
2. Click **Azure AD roles**.
3. Click to see the charts available.



4. Click **Resource audit**.

   Depending on your audit history, a column chart is displayed along with the total activations, max activations per day, and average activations per day.

   At the bottom of the page, a table is displayed with information about each action in the available audit history. The columns have the following meanings:

Column	Description
Time	When the action occurred.
Requestor	User who requested the role activation or change. If the value is **Azure System**, check the Azure audit histo
Action	Actions taken by the requestor. Actions can include Assign, Unassign, Activate, Deactivate, or AddedOutside
Member	User who is activating or assigned to a role.
Role	Role assigned or activated by the user.
Reasoning	Text that was entered into the reason field during activation.
Expiration	When an activated role expires. Applies only to eligible role assignments.

5. To sort the audit history, click the **Time**, **Action**, and **Role** buttons.

10.7.2 Task 2: Filter audit history

1. At the top of the audit history page, use the filter options to filter the results.

Results: You have now completed this lab.

11 Module 1 - Lab 2: Key Vault (Implementing Secure Data by setting up Always Encrypted)

Scenario

This module includes the following tasks:

- Azure confidential computing
- Azure Azure Key Vault

11.1 Exercise 1: Introduction to Azure Key Vault

Scenario

In this lab, you will get started with Azure Key Vault to create a hardened container (a vault) in Azure, to store and manage cryptographic keys and secrets in Azure. First you will use Azure PowerShell. Then you will store a password as a secret that could then be used with an Azure application.

11.1.1 Task 1: Download SQL Server Management Studio

1. To download the latest version of SQL Management Studio required for this lab visit the following link and select download SQL Management Studio `https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-ma`

   Note: You do not need to wait for the SQL Management Studio to install before continuing.

11.1.2 Task 2: Use PowerShell to create a Key Vault

In this exercise, you will use PowerShell to create an Azure Key Vault.

1. Start PowerShell by clicking **Start > PowerShell**
2. Use the following command to authenticate to Azure using the account for your Azure subscription.

   ```
   Login-AzAccount
   ```

3. Create a new Resource Group

   ```
   New-AzResourceGroup -Name 'KeyVaultPSRG' -Location 'eastus'
   ```

4. Create a key vault in the resource group. **The VaultName must be unique therefore change to something unique.**

   ```
   New-AzKeyVault -VaultName '<keyvault name>' -ResourceGroupName 'KeyVaultPSRG' -Location 'eastus'
   ```

   Note: The output of this shows important pieces of information: Vault Name in this case that is KeyVaultPS and the Vault URI: `https://KeyVaultPS.vault.azure.net`

5. In the Azure Portal open the **KeyVaultPSRG** Resource Group.
6. Click on the Key Vault name to examine what you have created.

   Note: For all future instructions replace KeyVaultPS with the name of your Key Vault.

7. Click **Access Policies > + Add Access Policy**
8. Select **Key, Secret and Certificate Management** from **Configure from template (optional)**
9. Click **Select Principal** and search for and then click on your account, then click on **Select**
10. Click **Add** and then **Save**

11.1.3 Task 3: Add a key and secret to Key Vault

1. Return to the PowerShell window.
2. Add a software-protected key to the Key Vault using this command. Be sure to change the placeholder text to your vault name.

   ```
   $key = Add-AZKeyVaultKey -VaultName '<YourVaultName>' -Name 'MyLabKey' -Destination 'Software'
   ```

3. Move back to **KeyVaultPS** in the Azure portal. Click **Keys** under Settings in the left navigation pane.
4. Click **MyLabKey**
5. Click the Current Version.
6. Examine the information about the key you created.

   Note: You can always reference this key by using its URI. To get the most current version, just reference `https://keyvaultps.vault.azure.net/keys/MyLabKey/` or if need be the exact version: `https://keyvaultps.vault.azure.net/keys/MyLabKey/da1a3a1efa5dxxxxxxxxxxxxxd53c5959e`

7. Move back to the PowerShell window. To display the current version of the key, enter the following command.

   ```
   $Key.key.kid
   ```

8. To view the Key you just created you can use the Get-AzureKeyVaultKey cmdlet. Be sure to change the placeholder text to your vault name.

   ```
   Get-AZKeyVaultKey -VaultName '<YourVaultName>'
   ```

11.1.4 Task 4: Add a Secret to Key Vault

1. Next, you will add a secret to the **KeyVaultPS**. To do this, add a variable named **$secretvalue** using the following code.

   ```
   $secretvalue = ConvertTo-SecureString 'Pa55w.rd1234' -AsPlainText -Force
   ```

2. Next add the secret to the Vault with this command. Be sure to change the placeholder text to your vault name.

   ```
   $secret = Set-AZKeyVaultSecret -VaultName 'YourVaultName' -Name 'SQLPassword' -SecretValue $secret
   ```

3. Move back to the Azure Portal on **KeyVaultPS** and click **Secrets**
4. Click the Secret **SQLPassword**
5. Click the current version
6. Examine the Secret that you created

   Note: You can always reference this key by using its URI. To get the most current version just reference `https://keyvaultps.vault.azure.net/secrets/SQLPassword` or if need be the exact version: `https://keyvaultps.vault.azure.net/secrets/SQLPassword/c5aada85d3acxxxxxxxxxxe8701efafcf3`

7. Click the **Show secret value** button -- notice that the password appears.
8. To view the Secret, use the Get-AzureKeyVaultSecret cmdlet. Be sure to change the placeholder text to your vault name.

   ```
   Get-AZKeyVaultSecret -VaultName 'YourVaultName'
   ```

11.1.5 Task 5: Enable a Client Application

You will enable your client application to access the Azure SQL Database service. This will be done by setting up the required authentication and acquiring the Application ID and Secret that you will need to authenticate your application. These steps will be accomplished in the Azure portal.

1. Open the Azure portal and navigate to Azure Active Directory.
2. Click **App Registrations** under **Manage** in the left navigation pane.
3. **Click + New registration**

4. Provide the name **sqlApp** for your application. Under **Redirect URI (optional)**, select **Web**, and for the SIGN-ON URL type **`https://sqlapp`**
5. Click **Register**.
6. Once the App Registration is complete click on **sqlApp** if it does not automatically appear.
7. Copy your Application (client) ID as you will need it later.
8. Click **Certificates & secrets**
9. Click **+ New client secret**
10. In the **Description** section, enter **Key1** for the description. Select **1 year** from the **Expires** list, then click **Add**
11. Copy the Key1 value as you will need it later. If you close and reopen the blade, the value will show as hidden.

11.1.6 Task 6: Add a Key Vault Policy allowing the application access to the Key Vault.

1. In the **Azure portal** open your **Resource Group** created at the beginning of the lab
2. Select the **Azure Key** vault
3. Click **Access Policies**
4. Select the account associated with your Azure subscription
5. In the **Key Permissions** drop down select **Select All** to highlight all permissions
6. Select **Save**

   Important! You must click save otherwise the permissions will not be committed
7. Run the following Powershell in the **Powershell ISE** to set the sqlApp key permissions replacing the placeholder text with **your account details**

```
$subscriptionName = '[Azure_Subscription_Name]'
$applicationId = '[Azure_AD_Application_ID]'
$resourceGroupName = '[Resource_Group_with_KeyVault]'
$location = '[Azure_Region_of_KeyVault]'
$vaultName = '[KeyVault_Name]'

Login-AzAccount

Set-AZKeyVaultAccessPolicy -VaultName $vaultName -ResourceGroupName $resourceGroupName -ServicePri
```

11.1.7 Task 7: Use Key Vault to Encrypt Data with Azure SQL Database

Scenario

In this task, you will create a blank Azure SQL Database, connect to it with SQL Server Management Studio and create a table. You will then encrypt two data columns using an autogenerated key from the Azure Key Vault. Then you will create a Console application using Visual Studio to Load data into the Encrypted Columns and then access that data securely using a connection string that accesses the key via Key Vault.

1. From the Azure Portal click **+ Create a resource> Databases > SQL Database**
2. Provide the following details on the SQL Database blade and click **Create**.
   - Resource Group: (create new) **SQLEncryptRG**
   - Database Name: **medical**
   - Server: **Create new**
     - Server name: [**Unique Server Name**]
     - Server Admin Login: **demouser**
     - Password: **Pa55w.rd1234**
     - Location: [**same location as KeyVaultPS**]

- Then click **OK**

- Pricing Tier: Standard S0

3. Once everything above is configured, select **Review + create,** then **Create**
4. Once the SQL Database is deployed, open it in the Azure Portal to locate and then copy the **ADO.NET Connection String**.

Note: When you save the connection string for future use, be sure to replace {your_username} with **demouser** and {your_password} with **Pa55w.rd1234**.

11.1.8 Task 8: Create a Table in the SQL Database

1. Use the Azure portal to locate the Server name where the Medical Database is located and copy the name.
2. On this same blade click Set Server firewall.
3. Next click + **Add client IP** and then click **Save**.
4. Open SQL Server Management Studio. Connect to the Server using these properties for the **Connect to Server** dialog.
   - Server Type: **Database Engine**
   - Server Name: [**found on the Database Overview Blade**]
   - Authentication: **SQL Server Authentication**
   - Login: **demouser**
   - Password: **Pa55w.rd1234**

11.1.9 Task 9: Create and Encrypt a Table

1. In SQL Server Management Studio expand **Databases > Right-click medical > New Query**.
2. Paste the following code into the query window and click Execute

```
CREATE TABLE [dbo].[Patients](

[PatientId] [int] IDENTITY(1,1),

[SSN] [char](11) NOT NULL,

[FirstName] [nvarchar](50) NULL,

[LastName] [nvarchar](50) NULL,

[MiddleName] [nvarchar](50) NULL,

[StreetAddress] [nvarchar](50) NULL,

[City] [nvarchar](50) NULL,

[ZipCode] [char](5) NULL,

[State] [char](2) NULL,

[BirthDate] [date] NOT NULL

PRIMARY KEY CLUSTERED ([PatientId] ASC) ON [PRIMARY] );
```

3. After the table is created successfully, expand **medical > tables > right-click dbo.Patients** and select **Encrypt Columns**.
4. Click **Next**.
5. On the Column Selection Screen check **SSN** and **Birthdate**. Then set the Encryption Type for SSN to **Deterministic** and for Birthdate **Randomized**. Click **Next**.

6. On the Master Key Configuration page on the Select the Key store provider, click **Azure Key Vault.** Click **Sign in** and authenticate. Select your Azure Key Vault. Click **Next**.
7. On the Run Settings screen click **Next** and then **Finish** to Proceed with the encrypting.
8. When the encryption process is complete, click **Close** and expand **medical > security > Always Encrypted Keys** and note that now there are keys found.

11.1.10 Task 10: Build a Console Application to work with Encrypted Columns

1. Open Visual Studio 2019 and Sign in using your Azure account.
2. Click **File > New > Project**
3. Next select **C# > Console App (.NET Framework)** and provide the name **OpsEncrypt** in the location **C:** and then click **Create**.
4. **Right-Click** the **OpsEncrypt** project > click **Properties**.
5. Change the **Target Framework** to **.NET Framework 4.7.2.** Click **Yes** when prompted to change the **Target Framework.**
6. Install the following **NuGet** packages by going to **Tools > NuGet Package Manager > Package Manager Console.**

   ```
   Install-Package Microsoft.SqlServer.Management.AlwaysEncrypted.AzureKeyVaultProvider
   ```

   ```
   Install-Package Microsoft.IdentityModel.Clients.ActiveDirectory
   ```
7. Open the **program.cs** file in notepad from Allfiles\Labs\Mod1_Lab02 and copy the code.
8. Replace the code in **Program.cs** in Visual Studio with the code you just copied.
9. Locate the **Connection string, clientId, and clientSecret** settings in the Main method and replace them with the values that you copied from the previous steps.
10. **Click** the **Start Button** in **Visual Studio**.
11. **The Console Application** will **Build** and then start. First it will ask for your password, then the app will add data to the database.
    - Server Password: **Pa55w.rd1234**
12. **Leave** the **Console Application Running** and move to the **SQL Management Studio**. Right-Click the medical database and click **New Query**.
13. Run the following query to see the data that was loaded into the database is encrypted.

    ```
    SELECT FirstName, LastName, SSN, BirthDate FROM Patients;
    ```
14. Now, move back to the console application where you will be asked to **Enter** a **Valid SSN**. This will query the encrypted column for the data. Notice that with the key called from the Key Vault, now the data is unencrypted and shown to the console window.

    ```
    999-99-0003
    ```
15. To **Exit** you press enter.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results : You have now completed this Lab.

12 Module 1: Lab 3: Using Multi-Factor Authentication for Secure Access

12.1 Exercise 1: MFA Authentication Pilot (Require MFA for specific apps with Azure Active Directory conditional access)

12.1.1 Task 1: Create your conditional access policy

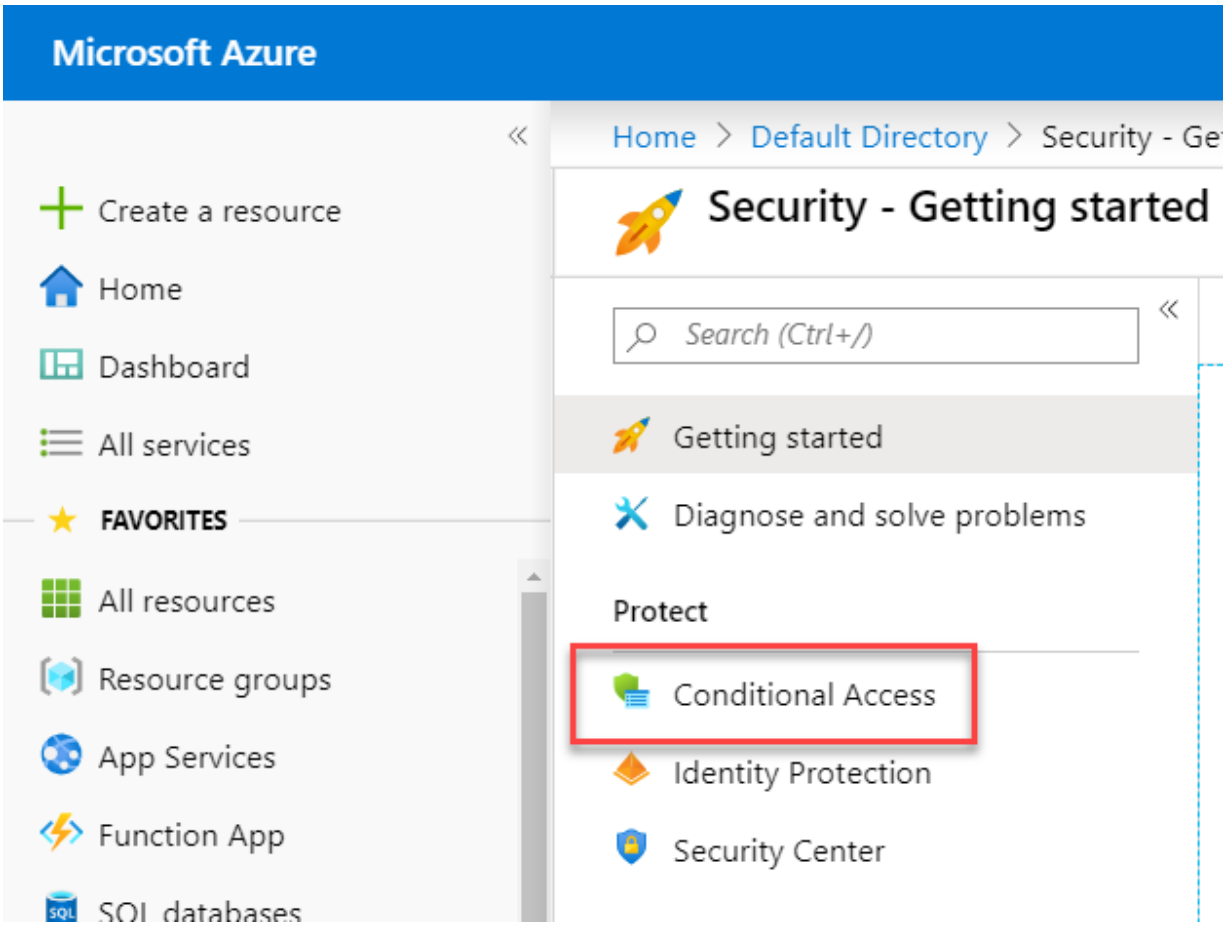
This section shows how to create the required conditional access policy. The scenario uses:

- The Azure portal as placeholder for a cloud app that requires MFA.
- Your sample user to test the conditional access policy.

In your policy, set:

Setting	Value
Users and groups	Isabella Simonsen
Cloud apps	Microsoft Azure Management
Grant access	Require multi-factor authentication

1. Sign in to the Azure Portal.
2. In the Azure portal, on the hub menu, click **Azure Active Directory**.
3. In the **Manage** section, click **Properties**, and click **Manage Security defaults**.
4. Change the **Enable Security defaults** option to **No**. Under reason, select **My organisation is using Conditional Access**, and click **Save**.
5. In the **Manage** section, click **Security**.
6. In the **Security** blade, click **Conditional access**.



7. On the **Conditional Access** page, in the toolbar on the top, click **New Policy**.

Note: if this is greyed out, refresh the browser session.

8. On the **New** page, in the **Name** textbox, type **Require MFA for Azure portal access**.
9. In the **Assignment** section, click **Users and groups**.
10. On the **Users and groups** page, perform the following steps:
    a. Click **Select users and groups**, and then select **Users and groups**.
    b. Click **Select**.
    c. On the **Select** page, select **Isabella Simonsen**, and then click **Select**.
    d. On the **Users and groups** page, click **Done**.
11. Click **Cloud apps or actions**.
12. On the **Cloud apps** page, perform the following steps:
    a. Click **Select apps**.
    b. Click **Select**.
    c. On the **Select** page, select **Microsoft Azure Management**, and then click **Select**.
    d. On the **Cloud apps** page, click **Done**.
13. In the **Access controls** section, click **Grant**.
14. On the **Grant** page, perform the following steps:
    1. Select **Grant access**.
    2. Select **Require multi-factor authentication**.
    3. Click **Select**.
15. In the **Enable policy** section, click **On**.
16. Click **Create**.

12.1.2 Task 2: Evaluate a simulated sign-in

Now that you have configured your conditional access policy, you probably want to know whether it works as expected. As a first step, use the conditional access what if policy tool to simulate a sign-in of your test user. The simulation estimates the impact this sign-in has on your policies and generates a simulation report.

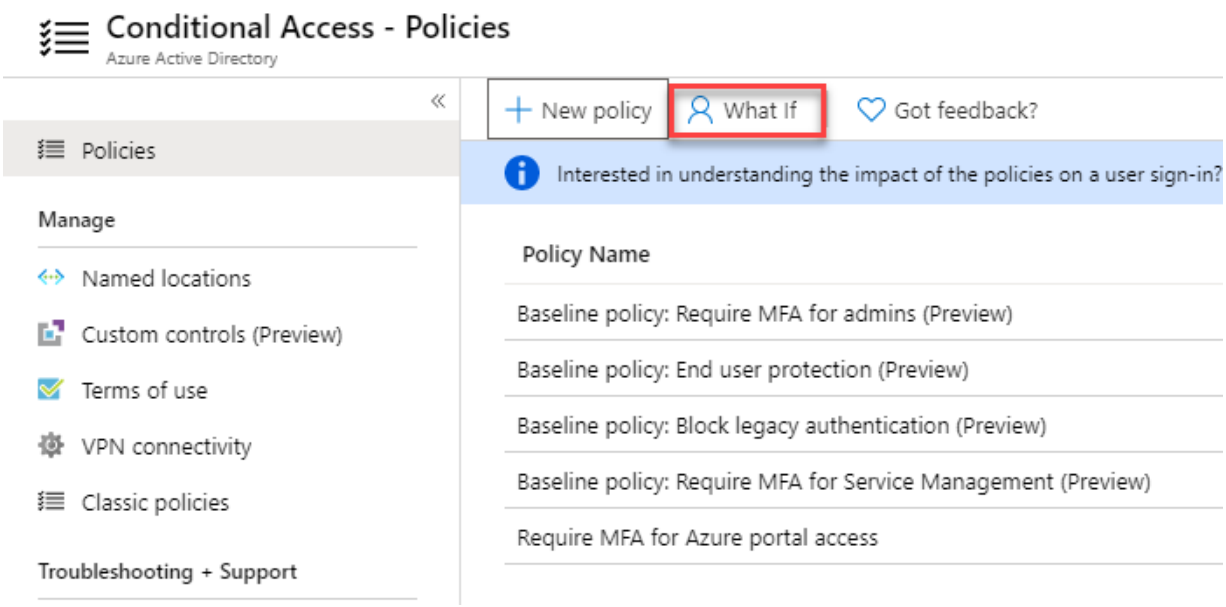
To initialize the what if policy evaluation tool, set:

- **Isabella Simonsen** as user
- **Microsoft Azure Management** as cloud app

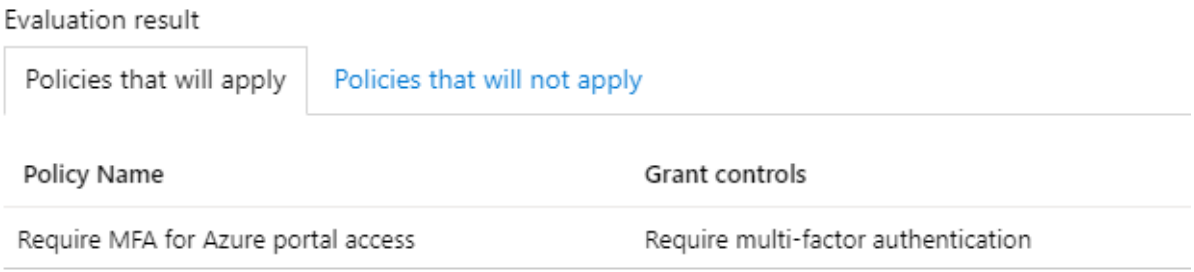
Clicking **What If** creates a simulation report that shows:

- **Require MFA for Azure portal access** under **Policies that will apply**
- **Require multi-factor authentication** as **Grant Controls**.

1. On the Conditional access - Policies page, in the menu on the top, click **What If**.



2. Click **Users**, select **Isabella Simonsen**, and then click **Select**.
3. To select a cloud app, perform the following steps:
   a. Click **Cloud apps or actions**.
   b. On the **Cloud apps page**, click **Select apps**.
   c. Click **Select**.
   d. On the **Select** page, select **Microsoft Azure Management**, and then click **Select**.
   e. On the cloud apps page, click **Done**.
4. Click **What If**.
5. Note the result, Require MFA for Azure portal access.



12.1.3 Task 3: Test your conditional access policy

In the previous section, you have learned how to evaluate a simulated sign-in. In addition to a simulation, you should also test your conditional access policy to ensure that it works as expected.

To test your policy, try to sign-in to the Azure portal `https://portal.azure.com` using your **Isabella Simonsen** test account. You should see a dialog that requires you to set your account up for additional security verification.

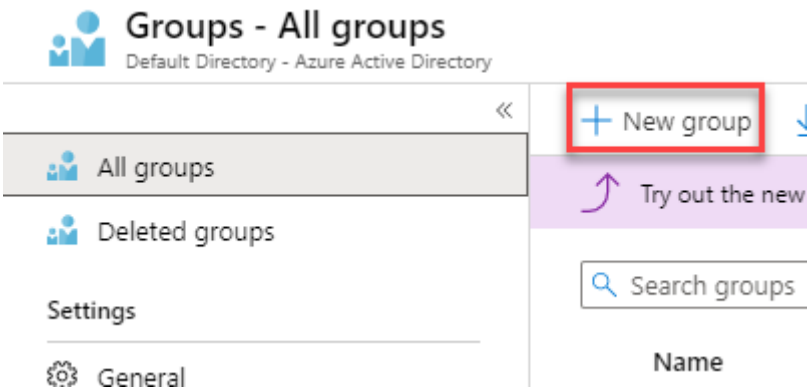
12.2 Exercise 2: MFA Conditional Access (Complete an Azure Multi-Factor Authentication pilot roll out)

In this lab, you walk you through configuring a conditional access policy enabling Azure Multi-Factor Authentication (Azure MFA) when logging in to the Azure portal. The policy is deployed to and tested on a specific group of pilot users. Deployment of Azure MFA using conditional access provides significant flexibility for organizations and administrators compared to the traditional enforced method.

- Enable Azure Multi-Factor Authentication
- Test Azure Multi-Factor Authentication

12.2.1 Task 1: Enable Azure Multi-Factor Authentication

1. Return to the the Azure portal that is logged in as your Global Admin account.
2. On the Hub menu click **Azure Active Directory**,
3. Click **Groups** and click + **New group**.



4. Enter the following information then click **Create**:
   - Group type; **Security**
   - Group Name: **MFA Pilot**
   - Group description: **MFA Pilot Group**
   - Membership type: **Assigned**
   - Members: Select **Isabella**

5. Browse to **Azure Active Directory**, click **Security** and select **Conditional access** on the **Security** Blade.
6. Select + **New policy**
7. Name your policy **MFA Pilot**
8. Under **users and groups**, select the **Select users and groups** check box

- Select your pilot group **MFA Pilot**
- Click **Select**

9. Under **Cloud apps or actions**, select the **Select apps** radio button
    - The cloud app for the Azure portal is **Microsoft Azure Management**
    - Click **Select**
    - Click **Done**
10. Skip the **Conditions** section
11. Under **Grant**, make sure the **Grant access** radio button is selected
    - Check the box for **Require multi-factor authentication**
    - Click **Select**
    - Click **Done**
12. Skip the **Session** section
13. Set the **Enable policy** toggle to **On**
14. Click **Create**

12.2.2 Task 2: Test Azure Multi-Factor Authentication

To prove that your conditional access policy works, you test logging in to a resource that should not require MFA and then to the Azure portal that requires MFA.

1. Open a new browser window in InPrivate or incognito mode and browse to `https://account.activedirectory.window`
    - Log in with the Isabella account. You should not ask you to complete MFA.
    - Close the browser window.
2. Open a new browser window in InPrivate or incognito mode and browse to `https://portal.azure.com`
    - Log in with the Isabella account. You should now be required to register for and use Azure Multi-Factor Authentication.
    - Close the browser window.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results : You have now completed this lab.

13 Module 1: Lab 4: App Registration

Scenario

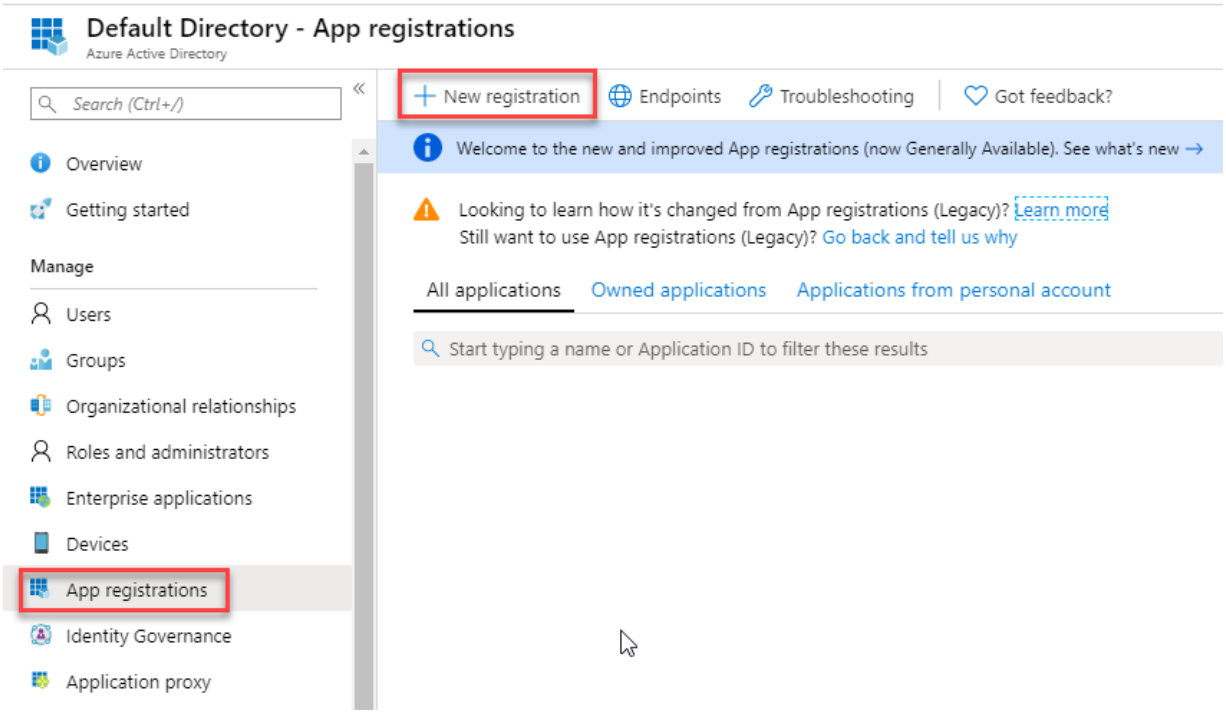
Enterprise developers and software-as-a-service (SaaS) providers can develop commercial cloud services or line-of-business applications that can be integrated with Microsoft identity platform to provide secure sign-in and authorization for their services.

This lab shows you how to add and register an application using the App registrations experience in the Azure portal so that your app can be integrated with the Microsoft identity platform.

13.1 Exercise 1: Application Registration

13.1.1 Task 1: Register a new application using the Azure portal

1. Sign in to the Azure portal.
2. In the left-hand navigation pane, select the **Azure Active Directory** service, and then select **App registrations > New registration**.

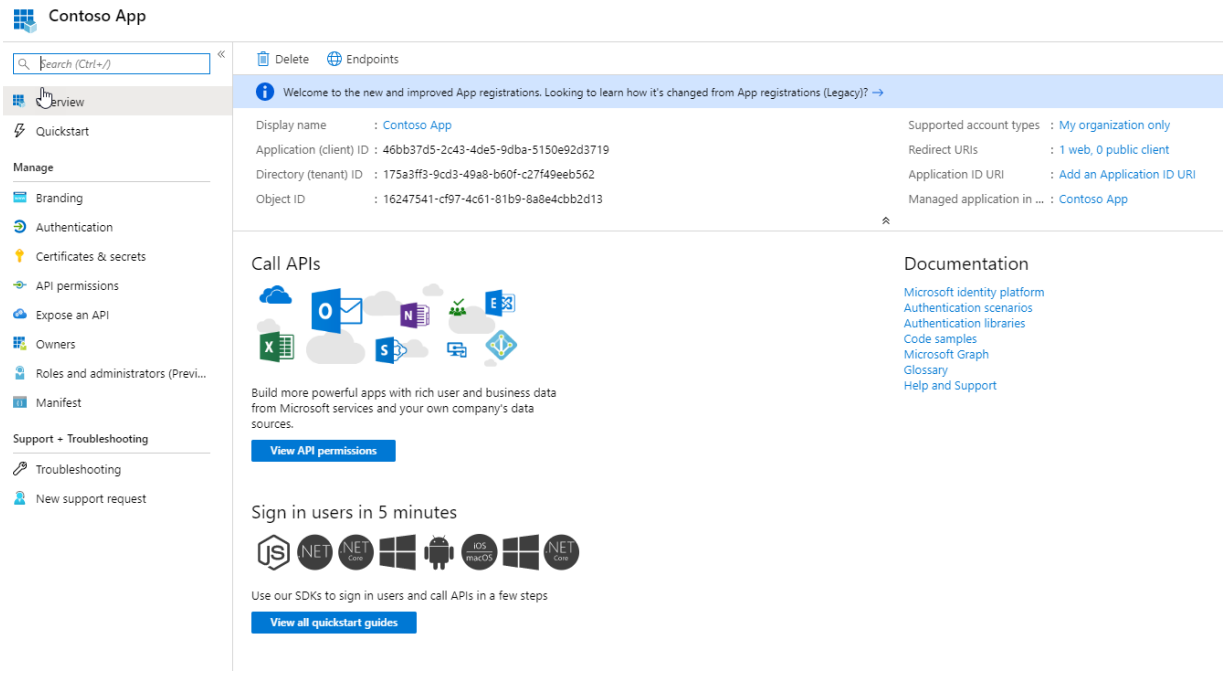


3. When the **Register an application** page appears, enter your application's registration information:
   - **Name** - Enter: **Contoso App**
   - **Supported account types** - Select **Accounts in this organizational directory only** (Read the options below).

Supported account types	Description
Accounts in this organizational directory only	Select this option if you're building
Accounts in any organizational directory	Select this option if you would like t
Accounts in any organizational directory and personal Microsoft accounts	Select this option to target the wide

   - **Redirect URI (optional)** - Select **Web**, and then enter `https://app.contoso.com`
     - For web applications, provide the base URL of your app. For example, `http://localhost:31544` might be the URL for a web app running on your local machine. Users would use this URL to sign in to a web client application.
     - For public client applications, provide the URI used by Azure AD to return token responses. Enter a value specific to your application, such as `myapp://auth`.

1. When finished, select **Register**.
2. Azure AD assigns a unique application (client) ID to your app, and you're taken to your application's **Overview** page. To add additional capabilities to your application, you can select other configuration options including branding, certificates and secrets, API permissions, and more.



Results: You have now completed this lab.

14 Module 1: Lab 5 - Application Service Principal

Scenario

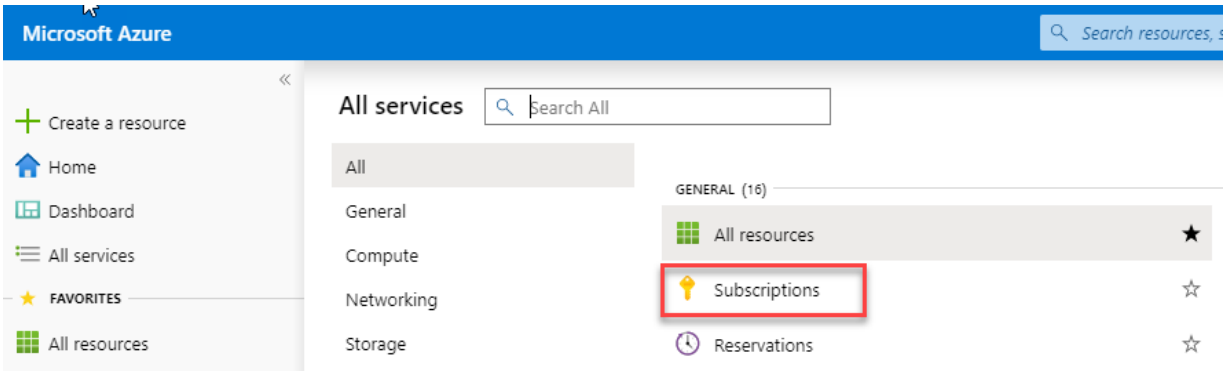
This lab shows you how to create a new Azure Active Directory (Azure AD) application and service principal that can be used with role-based access control. When you have code that needs to access or modify resources, you can create an identity for the app. This identity is known as a service principal. You can then assign the required permissions to the service principal. This lab shows you how to use the portal to create the service principal. It focuses on a single-tenant application where the application is intended to run within only one organization. You typically use single-tenant applications for line-of-business applications that run within your organization.

14.1 Exercise 1: Use the portal to create a service principal that can access resources

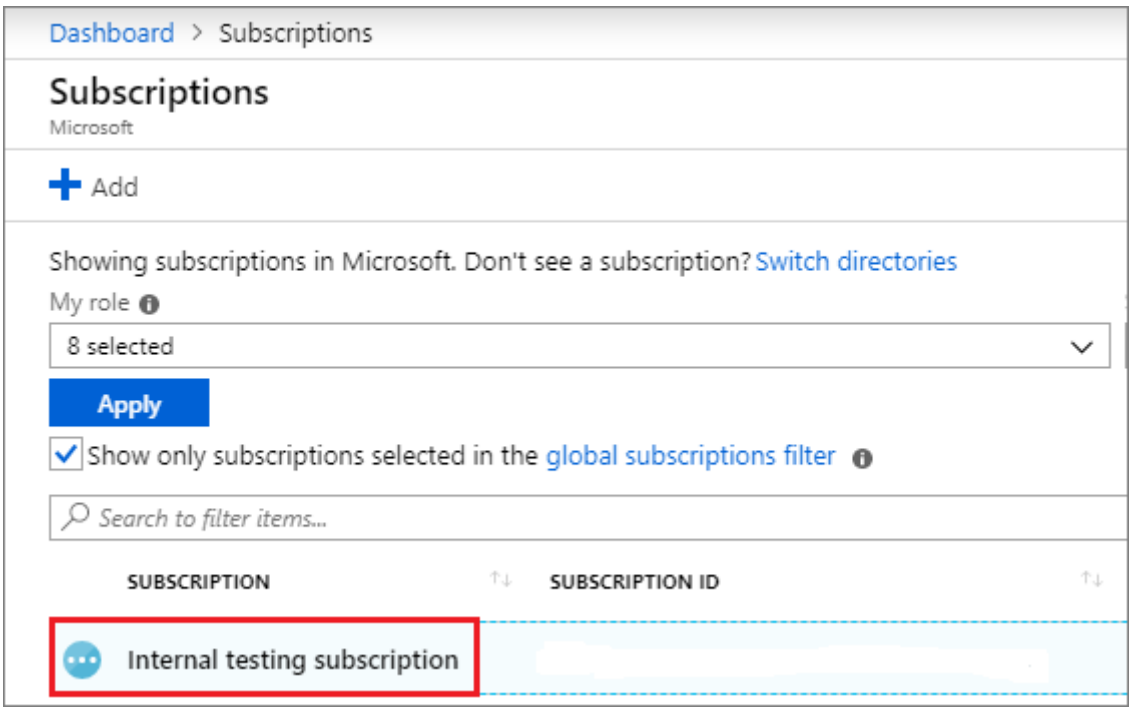
You can set the scope at the level of the subscription, resource group, or resource. Permissions are inherited to lower levels of scope. For example, adding an application to the Reader role for a resource group means it can read the resource group and any resources it contains.

14.1.1 Task 1: Assign the application to a role

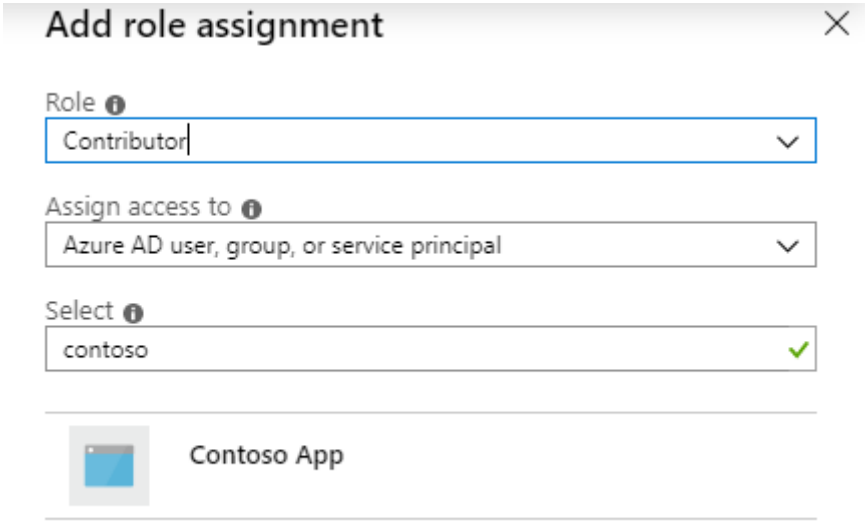
1. Navigate to the level of scope you wish to assign the application to. For example, to assign a role at the subscription scope, select **All services** and **Subscriptions**.



2. Select your subscription.



3. Select **Access control (IAM)**.
4. Select **Add** and select **Add role assignment**.
5. Select the Contributer role you wish to assign to the application. To allow the application to execute actions like **reboot**, **start** and **stop** instances, select the **Contributor** role. By default, Azure AD applications aren't displayed in the available options. To find your application, search for the name **Contoso App** and select it.



6. Select **Save** to finish assigning the role. You see your application in the list of users assigned to a role for that scope.

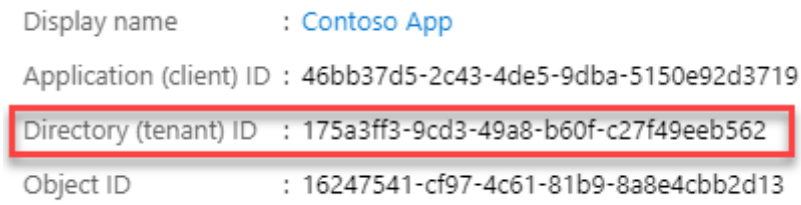
Your service principal is set up. You can start using it to run your scripts or apps. The next section shows how to get values that are needed when signing in programmatically.

14.1.2 Task 2: Get values for signing in

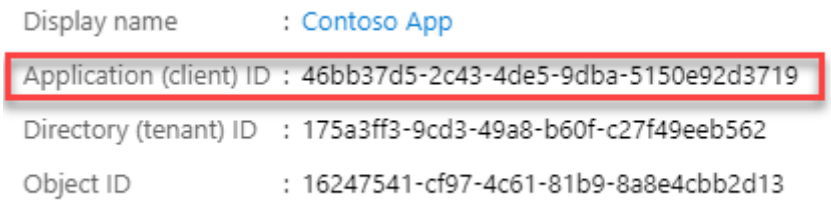
When programmatically signing in, you need to pass the tenant ID with your authentication request. You also need the ID for your application and an authentication key. To get those values, use the following steps:

1. Select **Azure Active Directory**.

2. From **App registrations** in Azure AD, select the **Contoso App** application.
3. Copy the **Directory (tenant) ID** and store it in your application code.



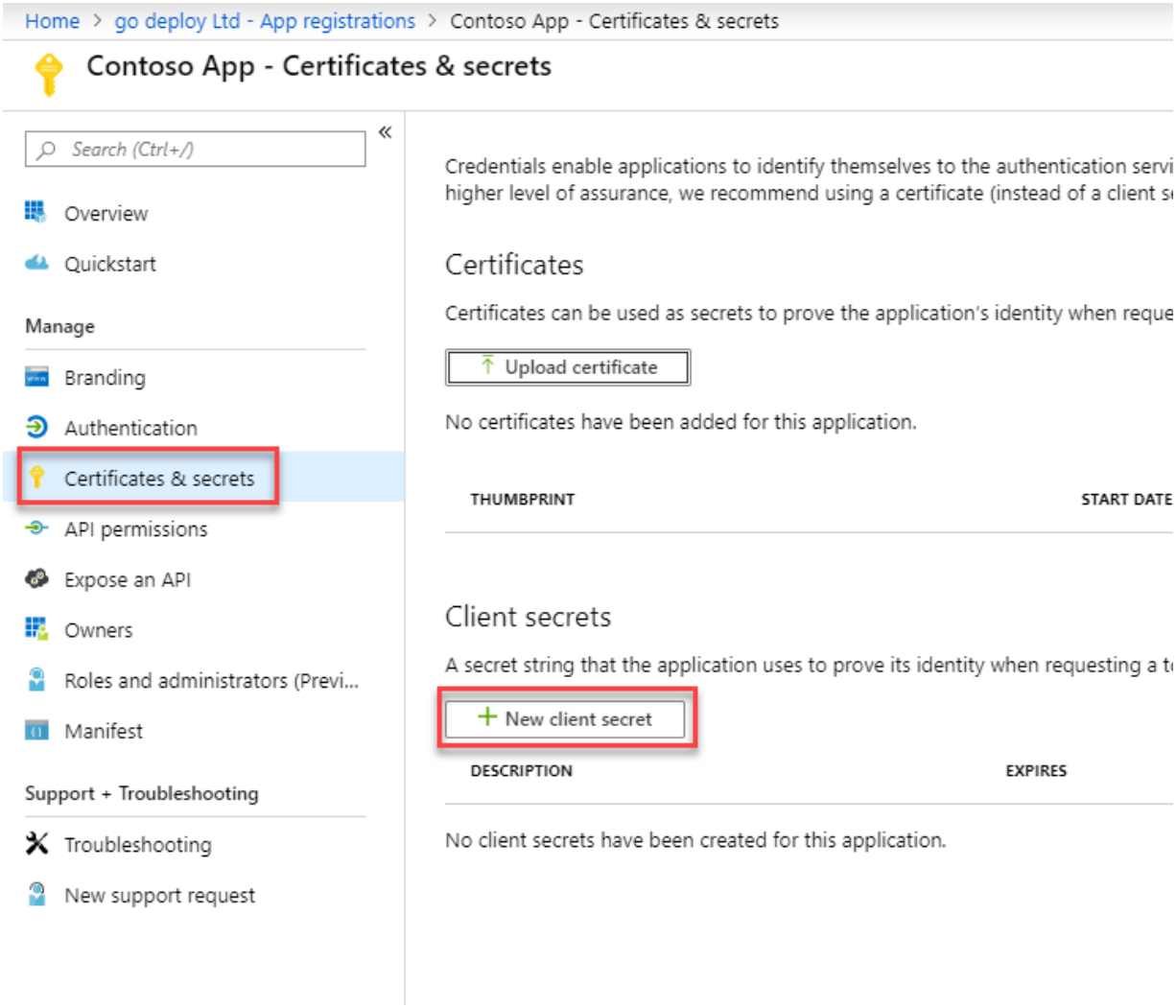
4. Copy the **Application (client) ID** and store it in your application code.



14.1.3 Task 3: Create a new application secret

You can choose a certificate or an application secret. In this task you will create an application secret.

1. Select **Certificates & secrets**.
2. Select **Client secrets -> New client secret**.



3. Provide a description of the secret, and a duration. When done, select **Add**.

After saving the client secret, the value of the client secret is displayed. Copy this value because you aren't able to retrieve the key later. You provide the key value with the application ID to sign in as the application. Store the key value where your application can retrieve it.

14.1.4 Task 4: Check Azure AD permissions

1. Select **Azure Active Directory**.
2. Select **User settings**.
3. Check the **App registrations** setting. This value can only be set by an administrator. If set to **Yes**, any user in the Azure AD tenant can register an app.

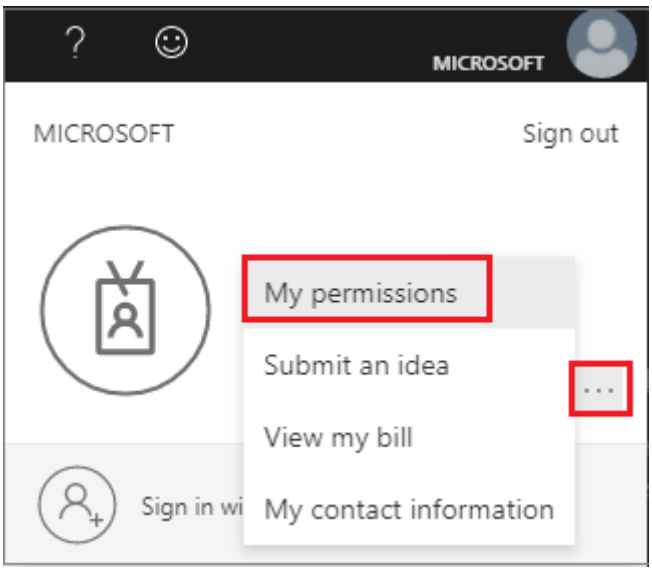
If the app registrations setting is set to **No**, only users with an administrator role may register these types of applications.

14.1.5 Task 5: Check Azure subscription permissions

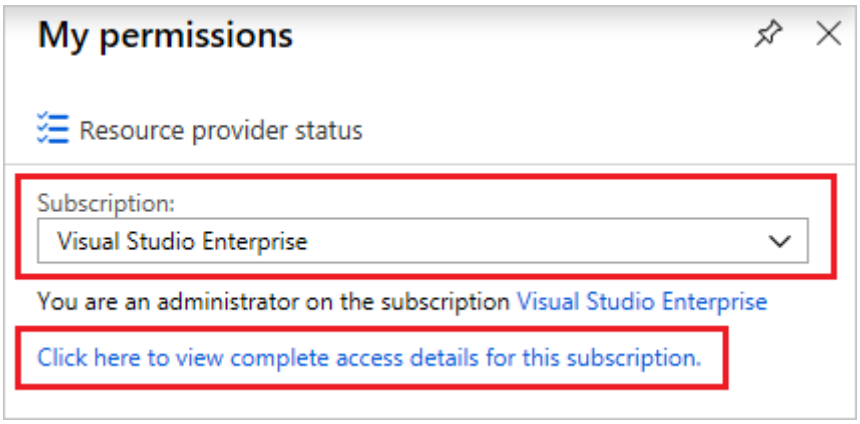
In your Azure subscription, your account must have `Microsoft.Authorization/*/Write` access to assign an AD app to a role. This action is granted through the Owner role or User Access Administrator.

To check your subscription permissions:

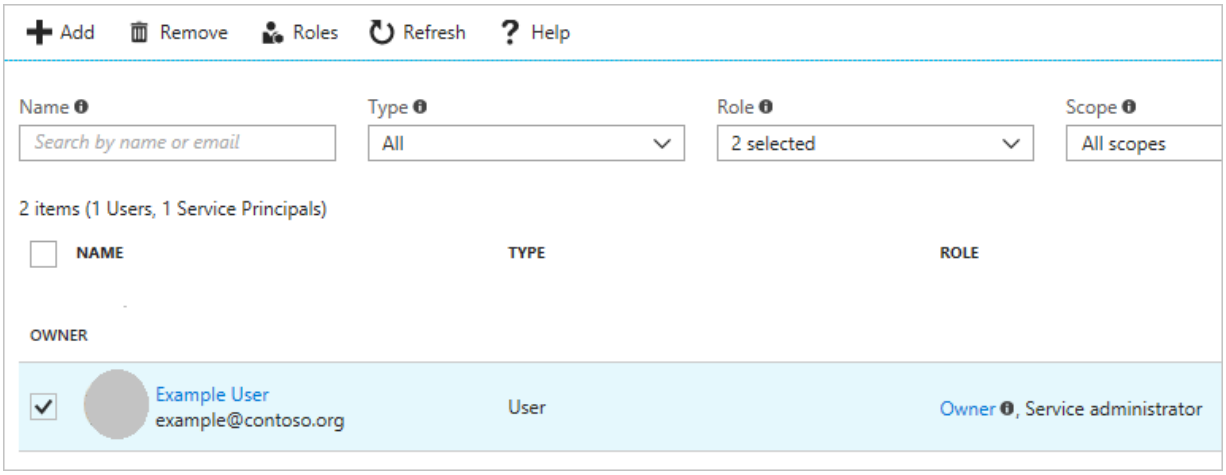
1. Select your account in the upper right corner, and select **... -> My permissions**.



2. From the drop-down list, select the subscription you want to create the service principal in. Then, select **Click here to view complete access details for this subscription**.



3. Select **Role assignments** to view your assigned roles and determine if you have adequate permissions to assign an AD app to a role. If not, you would ask your subscription administrator to add you to User Access Administrator role. In the following image, the user is assigned to the Owner role, which means that user has adequate permissions.



Results: You have now completed this lab.

15 Module 1: Lab 6: Manage Identity and Access

Scenario

In this module, you'll learn about Role-Based Access Control as the foundation to organizing and managing an organization's administrative access based on the principle of least privilege. You will also review Azure Active Directory concepts, as well as gain insight into the threat landscape and security risks that are exposed to IT organizations through breach of privileged access. Lessons include:

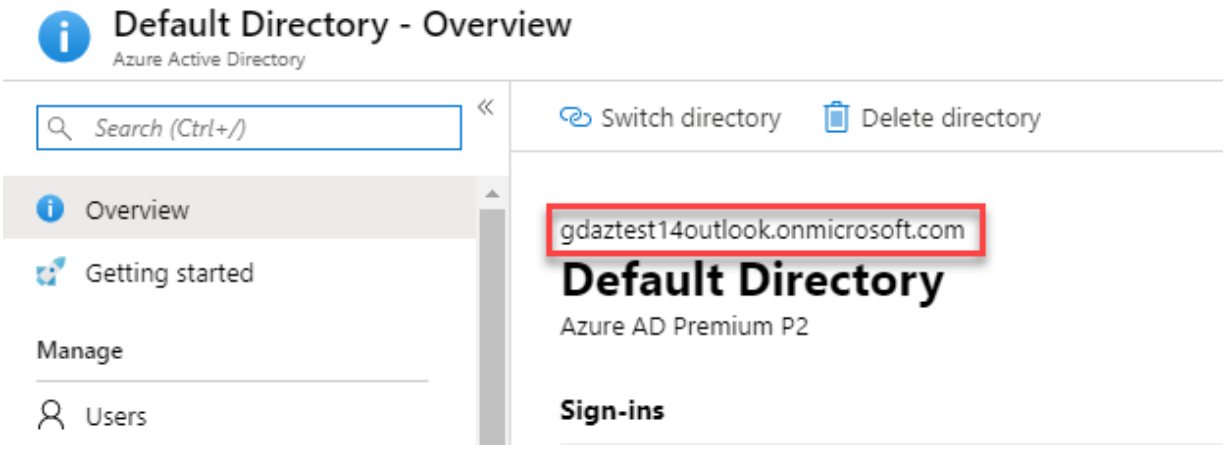
- Role-Based Access Control
- Azure Active Directory (Refresher)
- Protecting Privileged Access in the Environment

16 Lab 5: Introduction to Identity Protection in Azure

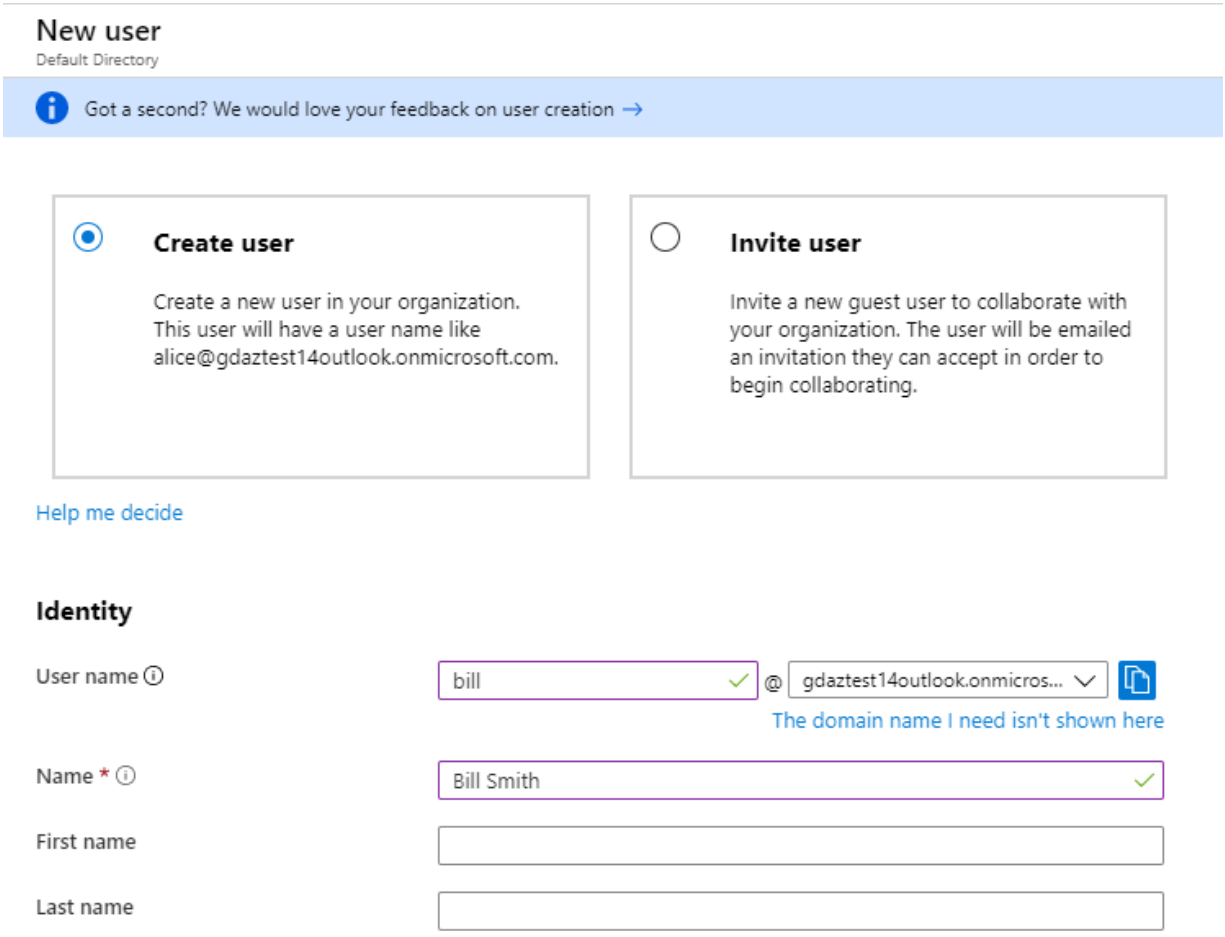
16.1 Exercise 1: Role-Based Access Control

16.1.1 Task 1: Create a User

1. Sign in to the Azure portal `https://portal.azure.com/`
2. Select **Azure Active Directory** and on the overview blade note down your tenant domain.



3. Select **Users**, and then select **New user**.
4. On the **User** page, fill out the blade with the following information:
   - **User name**: bill
   - **Name**: Bill Smith

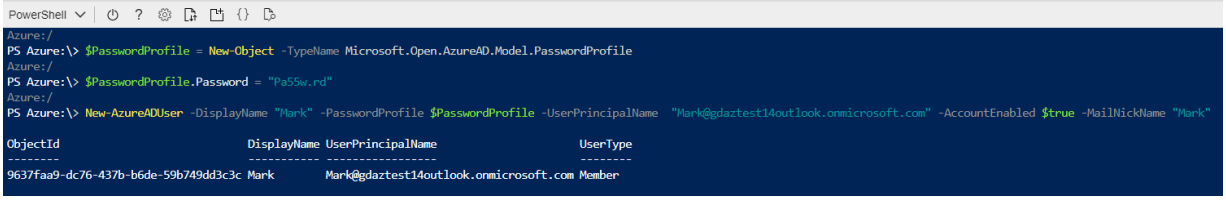


5. Show the auto-generated password provided in the **Password** box. You'll need to give this password to the user for the initial sign-in process.
6. Select **Create**.

   The user is created and added to your Azure AD tenant.
7. Launch **Azure Cloud Shell** by clicking on the PowerShell icon at the top of the Azure Portal and select PowerShell if prompted then run the following command to connect to AzureAD:

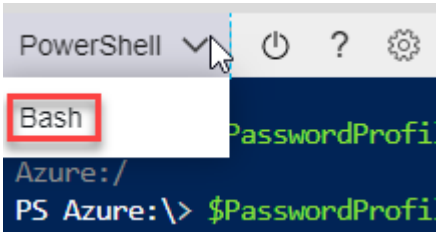
   ```
   Connect-AzureAD
   ```
8. **Enter the following commands** to create a user in the PS cloud shell **replacing yourdomain** with your domain noted down erlier

   ```
   $PasswordProfile = New-Object -TypeName Microsoft.Open.AzureAD.Model.PasswordProfile

   $PasswordProfile.Password = "Pa55w.rd"

   New-AzureADUser -DisplayName "Mark" -PasswordProfile $PasswordProfile      -UserPrincipalName "Mark
   ```



9. Run the following command to get a list of the users in Azure AD

   ```
   Get-AzureADUser
   ```
10. Change the Azure cloud shell to azure CLI mode with Bash by using the drop down menu



11. Enter the following command in **azure CLI** to create a user in Azure CLI **replacing yourdomain** with the domain you noted earlier.

```
az ad user create --display-name Tracy --password Pa55w.rd --user-principal-name Tracy@yourdomain.
```

You should now have 5 users in your Azure AD

16.1.2 Task 2: Create Groups In Portal, PowerShell, and CLI

1. In the Azure Portal click **Azure Active Directory** on the **Azure AD blade** click **Groups** and select **New group**.
2. Fill in the details with the following details:
   - **Group Type**: Security
   - **Group Name**: Senior Admins Group
3. In the Members section, click No Members link and search for and select Bill then click **Select**.
4. Click **Create**
5. Launch the **Cloud Shell** in **Bash** mode by clicking the Cloud Shell icon at the top of the Azure Portal.
6. In the **Cloud Shell** enter the following commands:

```
az login
```

```
az ad group create --display-name ServiceDesk --mail-nickname ServiceDesk
```

7. Change the Cloud Shell to **PowerShell** and enter the following command:

```
New-AzureADGroup -DisplayName "Junior Admins" -MailEnabled $false -SecurityEnabled $true -MailNickl
```

8. Exit the **Cloud Shell**.
9. In the **Active Directory blade** click **Groups** and confirm you have **5** groups

Name	Object Id	Group Type
JA Junior Admins	3ce819ea-a8bc-47ad-9650-5b2be1...	Security
MP MFA Pilot	283c65eb-0705-42a9-a1ca-867191...	Security
SA Senior Admins Group	6bb50553-0f6d-490d-8b0e-80c554...	Security
SE ServiceDesk	131b57a3-f06a-47b5-8c11-fb28887...	Security

16.2 Exercise 2: Practice - RBAC

16.2.1 Task 1: Create a resource group

1. In the navigation list, choose **Resource groups**.
2. Choose **Add** to open the **Resource group** blade.
3. For **Resource group name**, enter **myRBACrg**
4. Select your subscription and the location of **East US**.
5. Choose **Review + create** then **Create** to create the resource group.

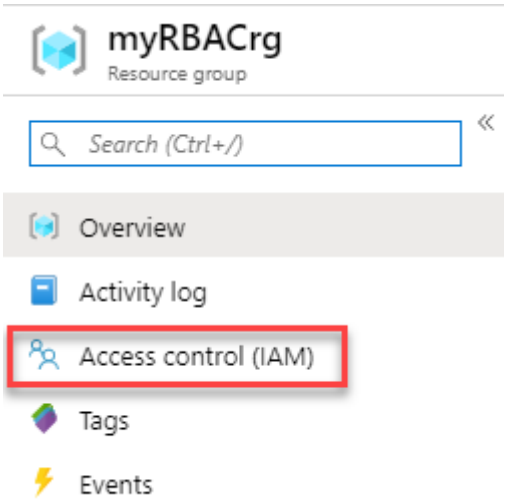
6. Choose **Refresh** to refresh the list of resource groups.

The new resource group appears in your resource groups list.

16.2.2 Task 2: Grant access

In RBAC, to grant access, you create a role assignment.

1. In the list of **Resource groups**, choose the new **myRBACrg** resource group.
2. Choose **Access control (IAM)** to see the current list of role assignments.



3. Choose **Add** to open the **Add role assignment** pane.

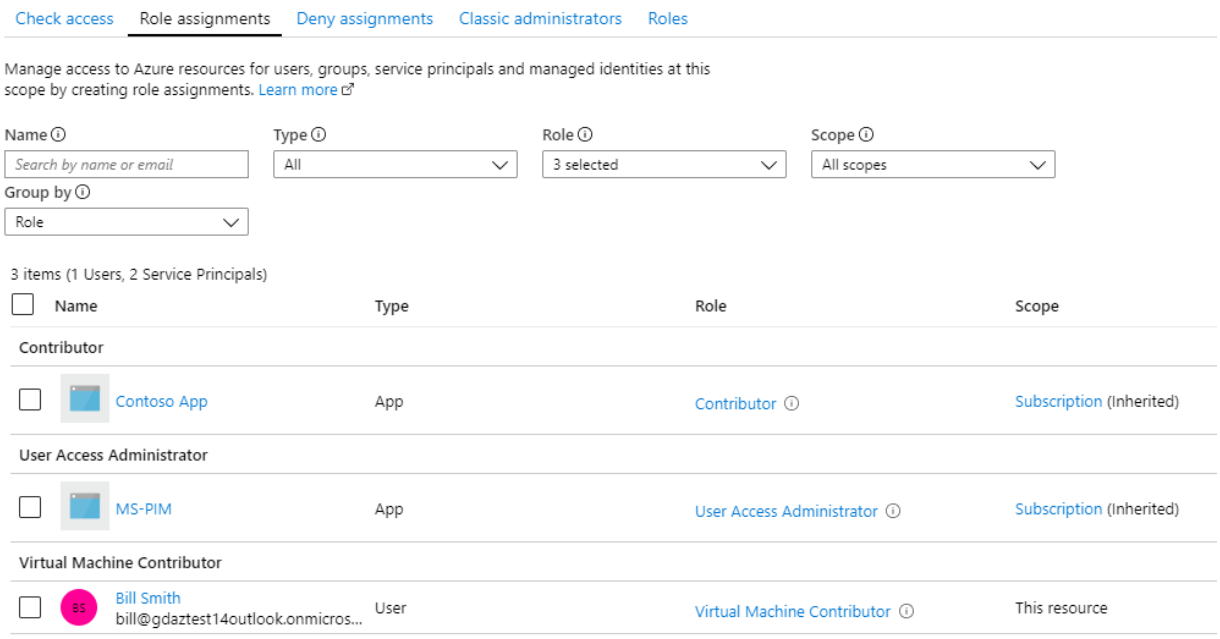
   If you don't have permissions to assign roles, you won't see the **Add** option.
4. In the **Role** drop-down list, select **Virtual Machine Contributor**.
5. In the **Select** list, select **Bill Smith**.
6. Choose **Save** to create the role assignment.

After a few moments, the user is assigned the Virtual Machine Contributor role at the myRBACrg resource group scope.

16.2.3 Task 3: Remove access

In RBAC, to remove access, you remove a role assignment.

1. Click the Role Assignments tab.
2. In the list of role assignments, add a checkmark next to user with the Virtual Machine Contributor role.



3. Choose **Remove**.
4. In the remove role assignment message that appears, choose **Yes**.

16.3 Exercise 3: Role-based Access Control (RBAC) using PowerShell

In this exercise you use PowerShell to :

- Use the `Get-AzRoleAssignment` command to list the role assignments
- Use the `Remove-AzResourceGroup` command to remove access

16.3.1 Task 1: Grant access

To grant access for the user, you use the New-AzRoleAssignment command to assign a role. You must specify the security principal, role definition, and scope.

1. Launch the **Cloud Shell PowerShell**.
2. Get the ID of your subscription using the **`Get-AzSubscription`** command.

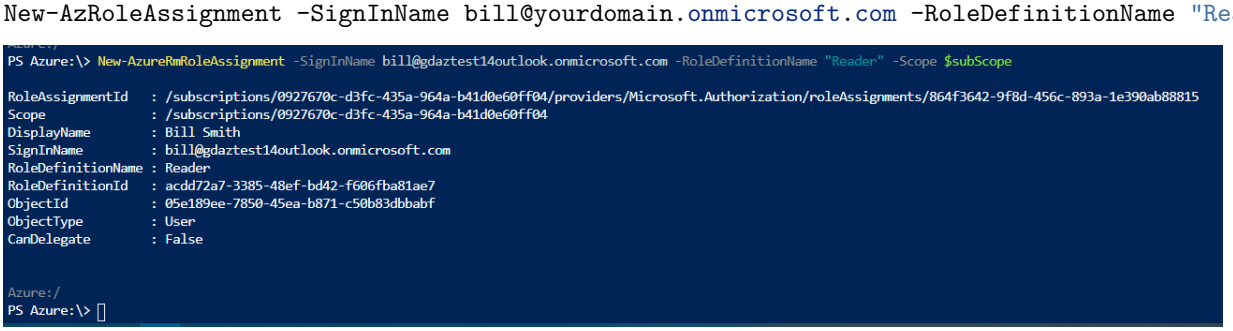
   ```
   Get-AzSubscription
   ```

3. Save the subscription scope in a variable replacing the 000000's with your subscription ID.

   ```
   $subScope = "/subscriptions/00000000-0000-0000-0000-000000000000"
   ```

4. Assign the Reader role to the user at the subscription scope by using the following command (replacing your domain with the tenant domain you noted earlier):

   ```
   New-AzRoleAssignment -SignInName bill@yourdomain.onmicrosoft.com -RoleDefinitionName "Reader" -Sco
   ```



5. Assign the Contributor role to the user at the resource group scope using the following command:

   ```
   New-AzRoleAssignment -SignInName bill@yourdomain.onmicrosoft.com -RoleDefinitionName "Contributor"
   ```

16.3.2 Task 2: List access

1. To verify the access for the subscription, use the Get-AzRoleAssignment command to list the role assignments use the following command:

   ```
   Get-AzRoleAssignment -SignInName bill@yourdomain.onmicrosoft.com -Scope $subScope
   ```

   ```
   PS Azure:\> Get-AzureRmRoleAssignment -SignInName bill@gdaztest14outlook.onmicrosoft.com -Scope $subScope

   RoleAssignmentId   : /subscriptions/0927670c-d3fc-435a-964a-b41d0e60ff04/providers/Microsoft.Authorization/roleAssignments/864f3642-9f8d-456c-893a-1e390ab88815
   Scope              : /subscriptions/0927670c-d3fc-435a-964a-b41d0e60ff04
   DisplayName        : Bill Smith
   SignInName         : bill@gdaztest14outlook.onmicrosoft.com
   RoleDefinitionName : Reader
   RoleDefinitionId   : acdd72a7-3385-48ef-bd42-f606fba81ae7
   ObjectId           : 05e189ee-7850-45ea-b871-c50b83dbbabf
   ObjectType         : User
   CanDelegate        : False


   Azure:/
   PS Azure:\>
   ```

   In the output, you can see that the Reader role has been assigned to the RBAC Tutorial User at the subscription scope.

2. To verify the access for the resource group, use the Get-AzRoleAssignment command to list the role assignments using the following command:

   ```
   Get-AzRoleAssignment -SignInName bill@yourdomain.onmicrosoft.com      -ResourceGroupName "myRBACrg"
   ```

In the output, you can see that both the Contributor and Reader roles have been assigned to the RBAC Tutorial User. The Contributor role is at the myRBACrg resource group scope and the Reader role is inherited at the subscription scope.

16.3.3 Task 3: Remove access

To remove access for users, groups, and applications, use `Remove-AzRoleAssignment` to remove a role assignment.

1. Use the following command to remove the Contributor role assignment for the user at the resource group scope.

   ```
   Remove-AzRoleAssignment -SignInName bill@yourdomain.onmicrosoft.com -RoleDefinitionName "Contribut
   ```

2. Use the following command to remove the Reader role assignment for the user at the subscription scope.

   ```
   Remove-AzRoleAssignment -SignInName bill@yourdomain.onmicrosoft.com -RoleDefinitionName "Reader" -
   ```

3. Remove the resource group by running the following command (When prompted to confirm press Y and press enter):

   ```
   Remove-AzResourceGroup -Name "myRBACrg"
   ```

4. Close the **Cloud Shell**.

Results: You have now completed this lab.

17 Module 1: Lab 7: Azure Policy

In this 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

17.1 Exercise 1: Using Azure Policy

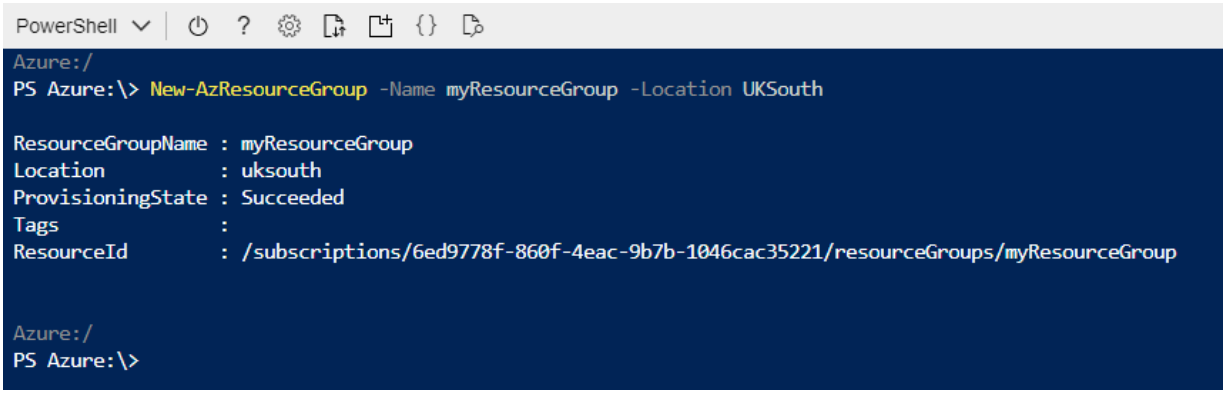
In this exercise, you will learn the basics of using Azure Policy. You will use a built-in policy to restrict which Azure regions can be used. You will then verify that the policy is working.

17.1.1 Task 1: Create an Azure Policy Assignment

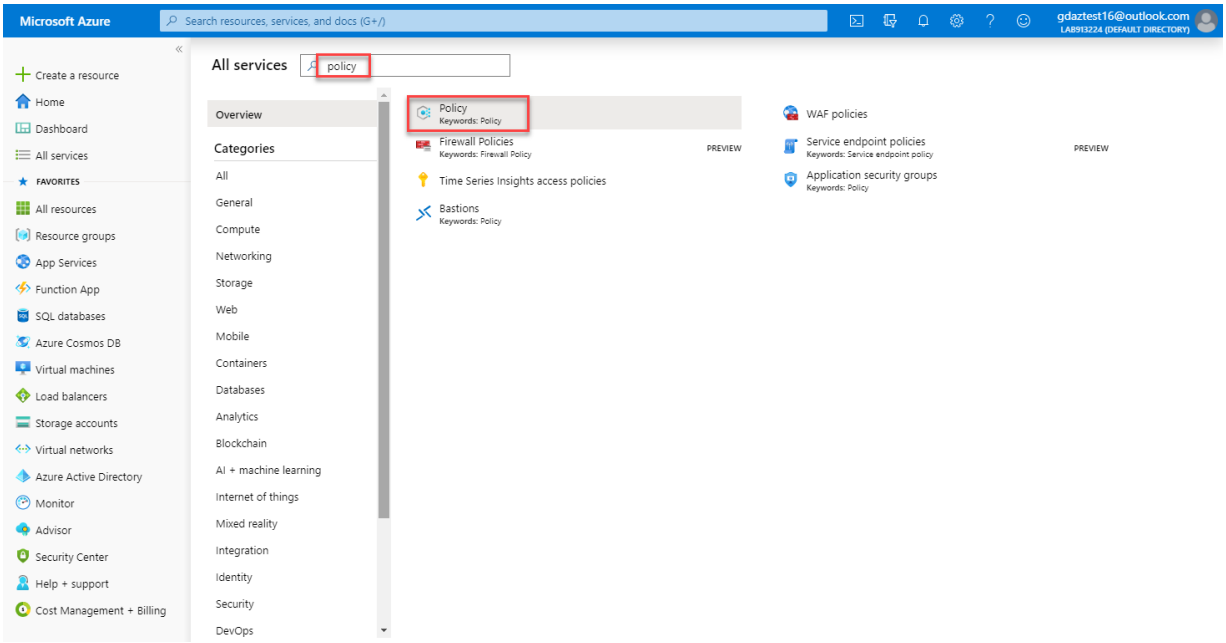
In this task, you will first browse the built-in policy definitions using the Azure portal. You will then create a policy assignment using one of the built-in definitions to restrict which Azure Regions can be used. This policy will be scoped to the Test-RG resource group.

1. Log in to the Azure portal at `https://portal.azure.com` using your Azure subscription credentials.
2. Open the Cloud Shell in PowerShell Mode and run the following command to create a Resource Group in the UK South region.

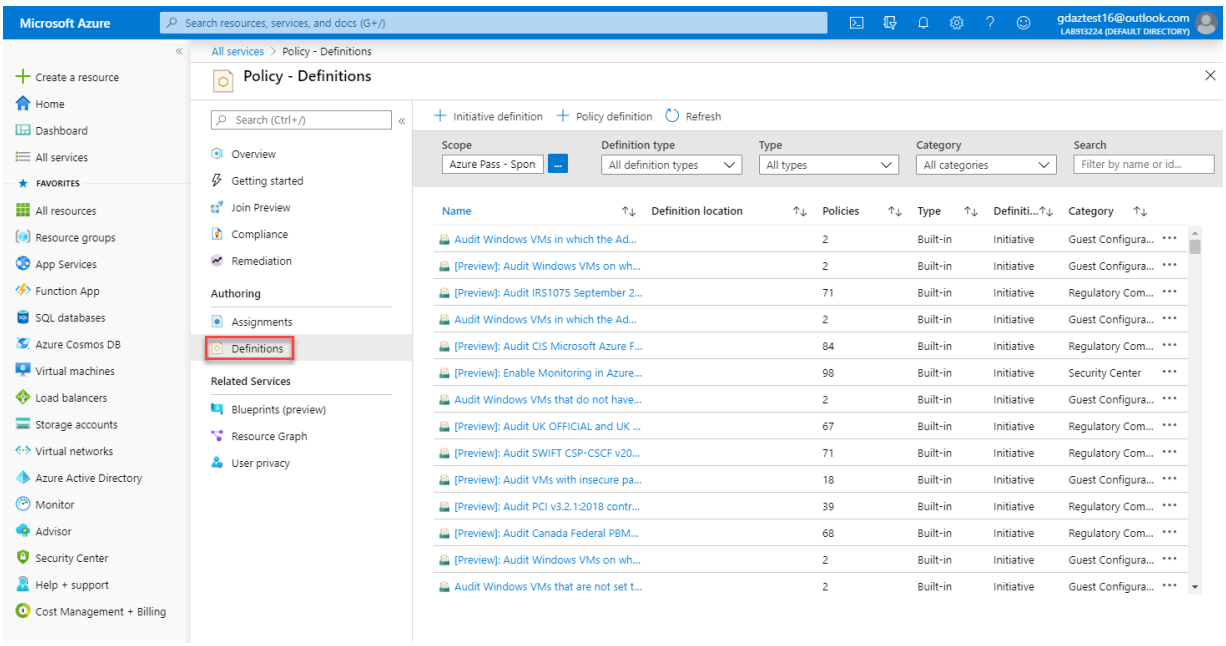
   ```
   New-AzResourceGroup -Name myResourceGroup -Location UKSouth
   ```



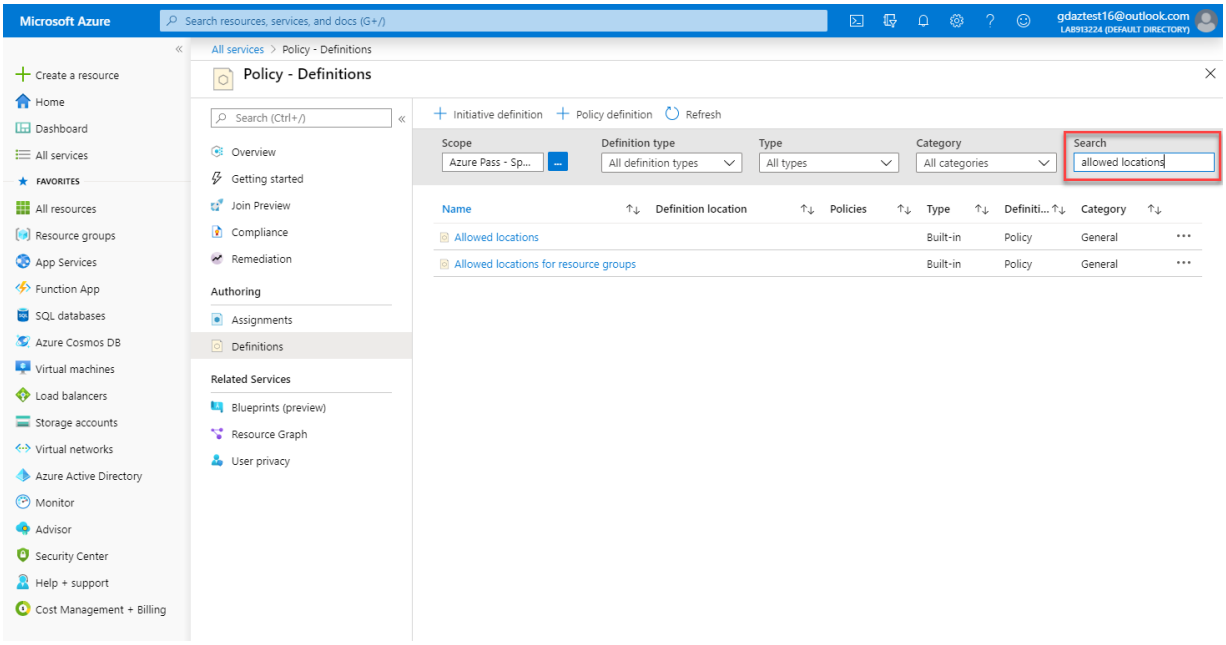
3. In the Hub menu, click on **All services**. Enter **Policy** into the search box, and click on the **Policy** service.



4. The Azure Policy blade will open. Click on **Definitions** to show the list of available policy definitions.



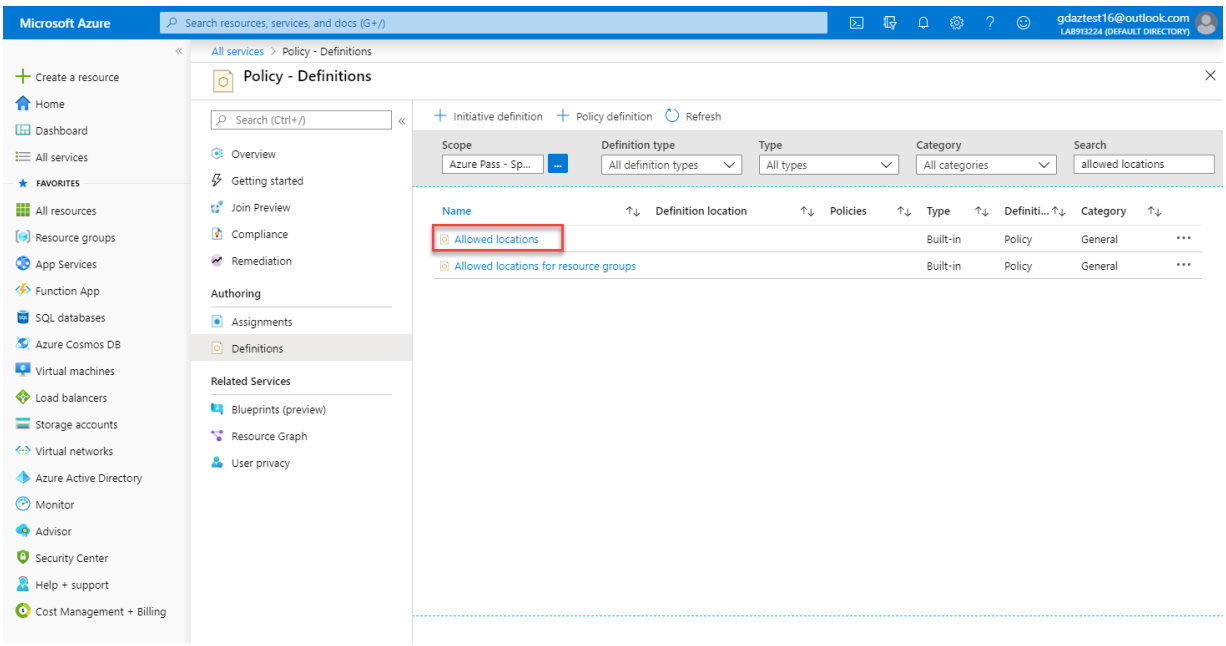
5. Use the filters and search to find the policy definition called **Allowed locations**.



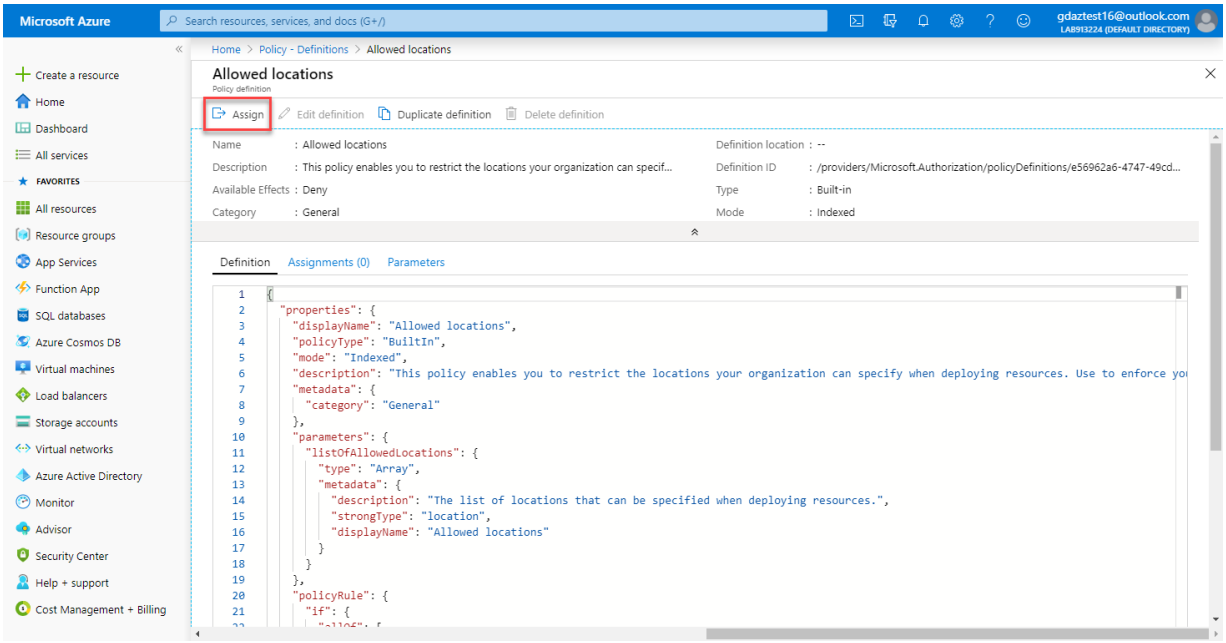
Note: This policy only restricts resource locations, not resource group locations. There is a separate policy for 'Allowed locations for resource groups'.

6. Click on the **Allowed locations** policy definition to open the definition details view.

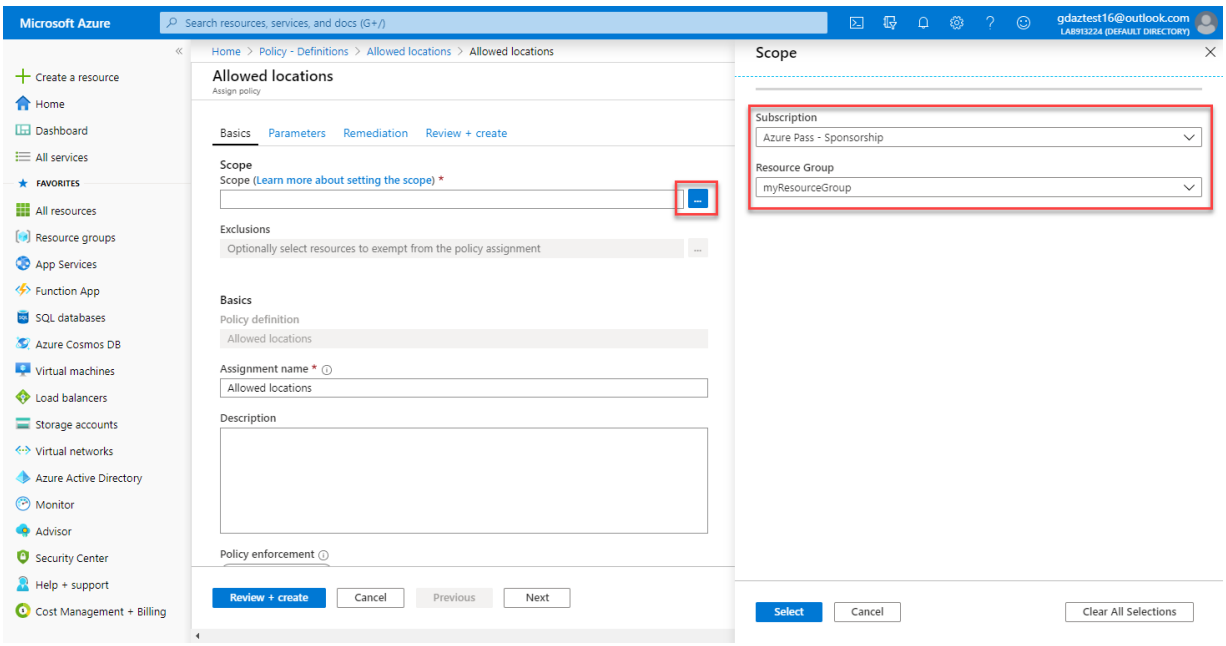
Note: Policy definitions take an array of locations as parameters. A policy rule is an 'if-then' statement. The 'if' clause checks to see if the resource location is included in the parameterized list, and if not the 'then' clause denies the resource creation.



7. Click **Assign**.



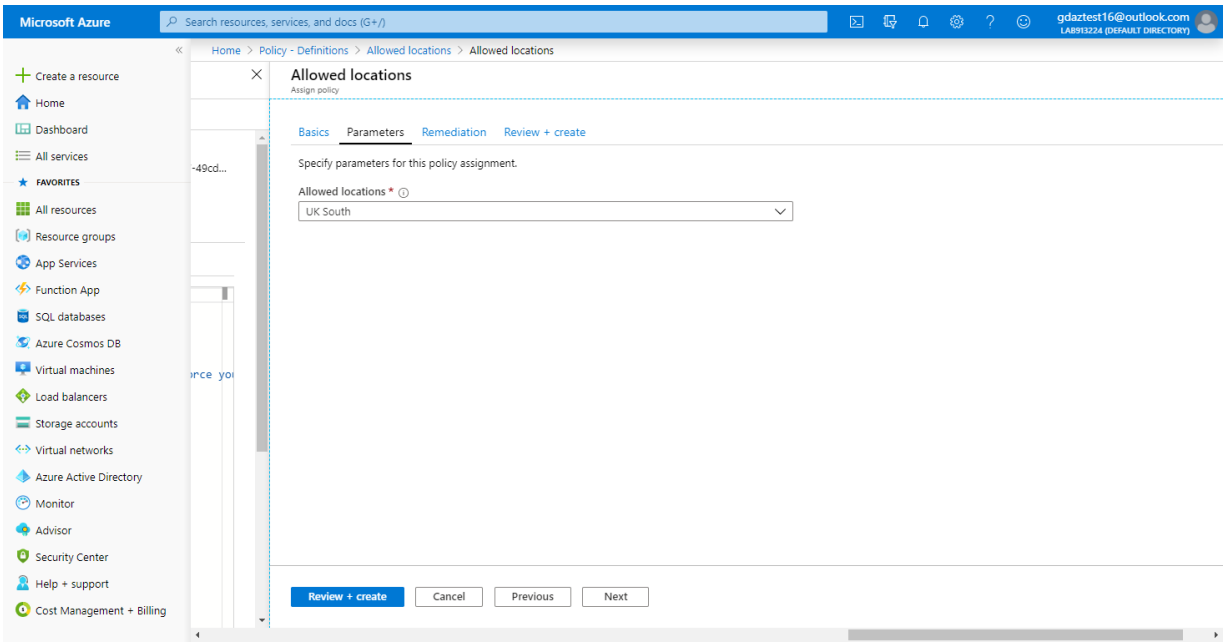
8. Click the Elipsis (...) button and assign the policy to your Subscription and the **myResourceGroup** resource group then click **Select**.



9. Complete the remainder of the policy assignment **Basics** tab with the following settings:

   - Exclusions: **Leave blank**
   - Assignment name: **Allow UK South for myResourceGroup**
   - Description: **Allow resources to be created in UK South Only for myResourceGroup**
   - Policy enforcement: **Enabled**
   - Assigned by: **Your name**

10. Click **Next** to proceed to the **Parameters** tab. In this tab you can provide values for parameters that are specified in the policy defintion. Select **UK South** as the allowed location.



11. Click **Review + create**, followed by **Create** to create the policy assignment.

12. You will see a notification that the assignment was successful, and that the assignment will take around 30 minutes to complete.

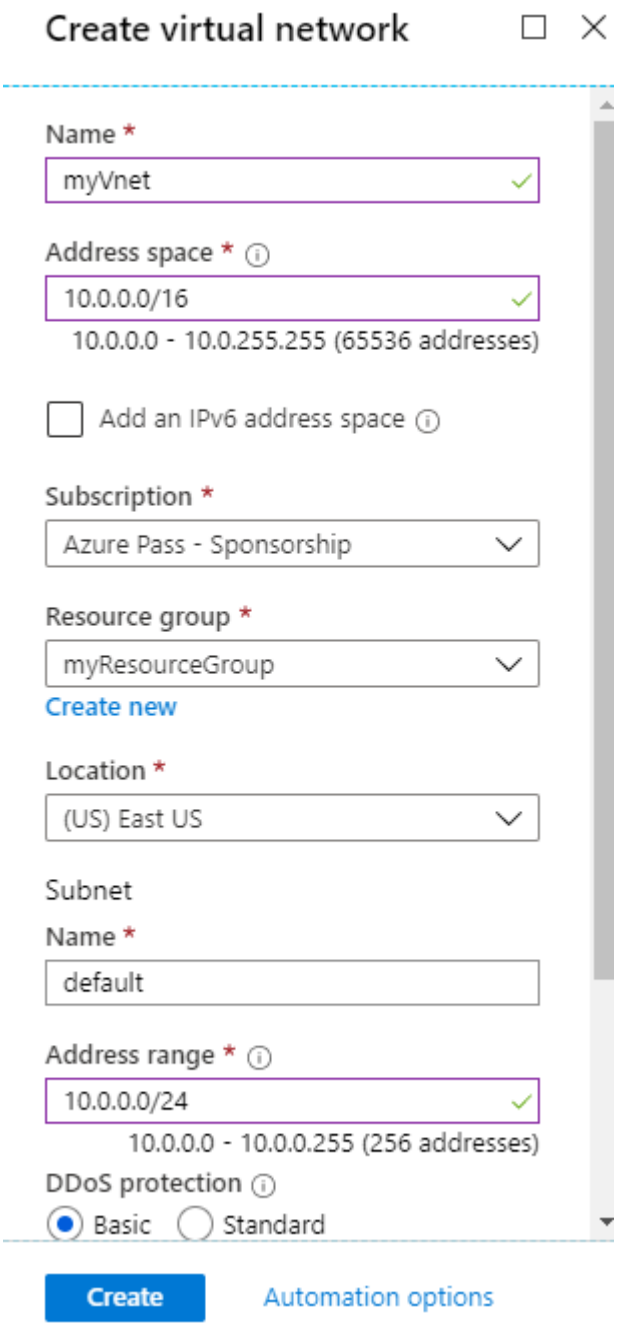
    Note: The reason the Azure policy assignment takes up to 30 minutes to be assigned is that is has to replicate globally although in the real world it generally only takes 2 - 3 minutes to be implemented. If the next task fails, simply wait a few minutes and attempt the steps again.

17.1.2 Task 2: Verify the Azure Policy Assignment

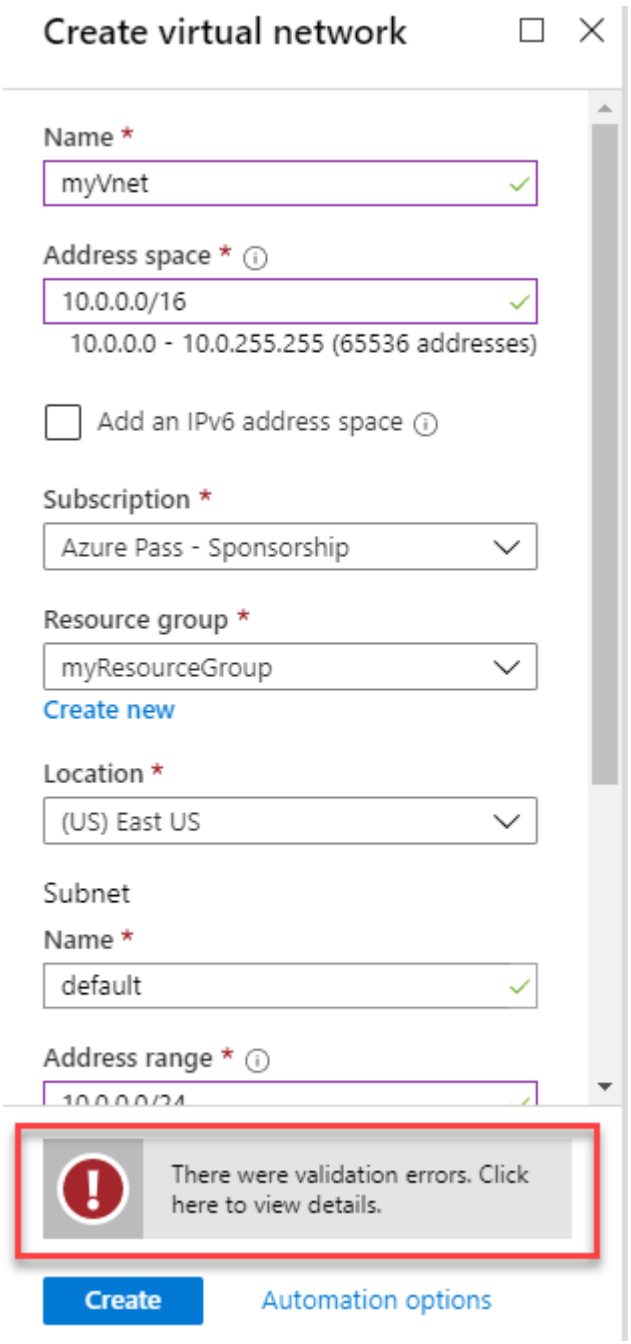
In this task, you will verify that the policy assignment created in the previous task is effective by attempting to create a virtual network in both a permitted region and a different region.

1. Click **Virtual Networks** on the Hub menu.
2. On the **Virtual Networks** blade, click **+ Add**
3. First, you will try to create a virtual network in East US. Since this is not an allowed location, the request should be blocked. Complete in the **Create virtual network** blade as follows:
   - Name: **myVnet**
   - Address space: **10.0.0.0/16**
   - Resource group: **myResourceGroup**
   - Location: **East US**
   - Address range: **10.0.0.0/24**

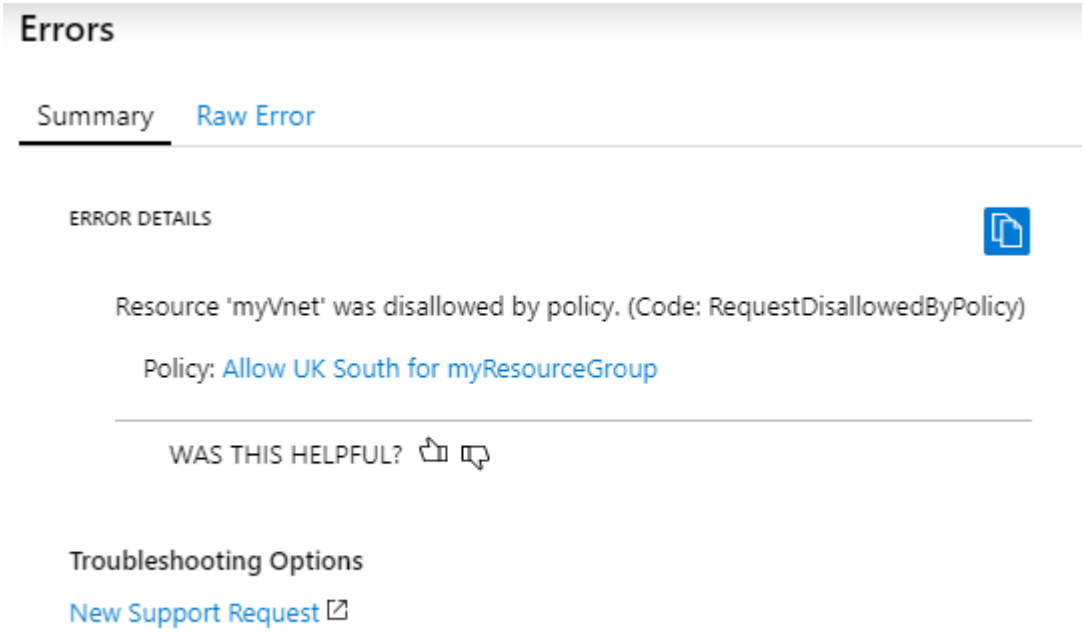
   Leave the other settings at their default values and click **Review + create**.



4. Once you click create you will see a validation error. Click the error to open the error details.



5. You will see the error states you are disallowed by policy.



6. Return back to the Basics tab and change the resource location to **UK South** as this is the location permitted by the policy. Click **Review + create** again and verify that the operation is successful.

Summary

In this exercise, you learned to use Azure policy by browsing the built-in policy definitions and creating a policy assignment.

Results: You have now completed this lab.

18 Module 1: Lab 8: Protecting Azure Resources with Resource Manager Locks

Scenario

Resource Manager Locks provide a way for administrators to lock down Azure resources to prevent deletion or changing of a resource. These locks sit outside of the Role Based Access Controls (RBAC) hierarchy and when applied will place the restriction on the resource for all users. These are very useful when you have an important resource in your subscription which users should not be able to delete or change and can help prevent accidental and malicious changes or deletion.

There are two types of resource locks that can be applied:

- **CanNotDelete** - This prevents anyone from deleting a resource whilst the lock is in place, however they may make changes to it.
- **ReadOnly** - As the name suggests, it makes the resource read only, so no changes can be made and it cannot be deleted. Resource locks can be applied to subscriptions, resource groups or individual resources as required. When you lock Subscription, all resources in that subscription (including ones added later) inherit the same lock. Once applied, these locks impact all users regardless of their roles. If it becomes necessary to delete or change a resource with a lock in place, then the lock will need to be removed before this can occur.

Permissions

Permission to set and remove locks requires access to one of the following RBAC permissions:

- `Microsoft.Authorization/*`
- `Microsoft.Authorization/locks/*`

By default, these actions are only available on the Owner and User Access Administrator built in RBAC Roles, however you can add them to custom roles as required. As mentioned, users with these roles are still subject

to the locks, but obviously they can remove them if required. Creation and deletion of a lock is tracked in the Azure Activity log.

19 Lab 7: Protecting Azure Resources with Resource Manager Locks

Locks can be created both at the time of creation of a resource inside an ARM template, or later using the portal or PowerShell.

19.1 Exercise 1: Creating Locks

The best way to ensure that locks are in place and protecting your resources is to create them at run time and configure them in your ARM templates. Locks are top level ARM resources; they do not sit underneath the resource being locked. They refer to the resource being locked, so this must exist first.

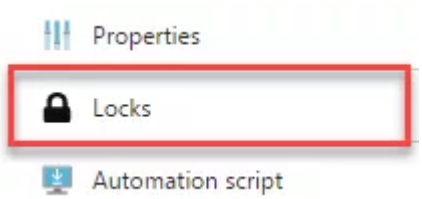
19.1.1 Task 1: Adding a Lock (Portal)

1. Open the Cloud Shell (PowerShell) and run the following commands to create a Resource Group and Storage Account. *(Change XXXXXX in the command to something unique)*

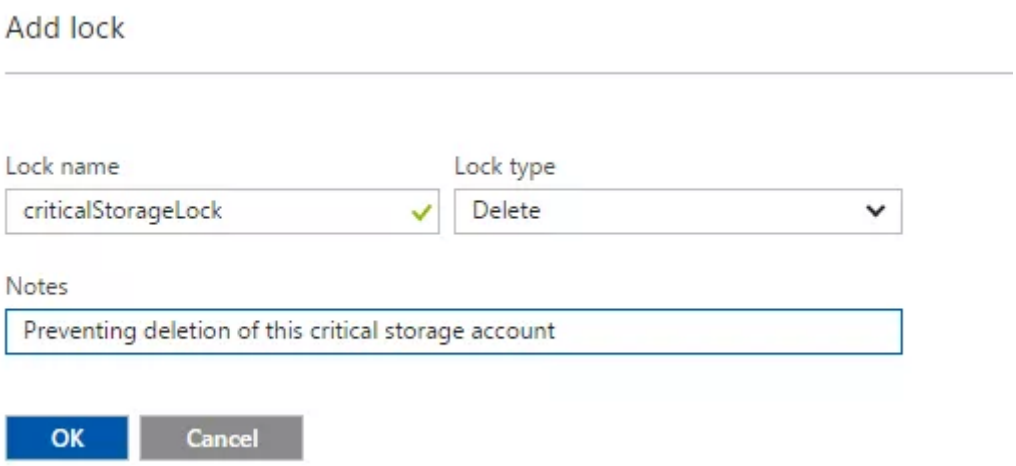
   ```
   New-AzResourceGroup -Name LockRG -Location EastUS
   ```

   ```
   New-AzStorageAccount -ResourceGroupName LockRG -Name XXXXXX -Location  EastUS -SkuName Standard_LR
   ```

2. Locate the Storage Account and select it. In the main blade, click the "Locks" icon



3. Click **Add**
4. Give the lock a name and description, then select the type, deletion or read only.



5. Click **OK** to save the lock. The resource is now protected.
6. Remove the lock by simply going back into **Locks**, select the lock and then go to delete.

19.1.2 Task 2: Adding a Lock (PowerShell)

1. Open the Cloud Shell (PowerShell) and run the following commands to create a Lock on the Storage Account. *(Change XXXXXX in the command to the name of your Storage Account)*

   ```
   Connect-AzureAD
   ```

```
New-AzResourceLock -LockLevel CanNotDelete -LockName criticalStorageLock -ResourceName XXXXXX -Res
```

2. To remove a lock use the following command. *(Change XXXXXX in the command to the name of your Storage Account)*

   ```
   Remove-AzResourceLock -LockName criticalStorageLock -ResourceName  XXXXX -ResourceGroupName LockRG
   ```

   If prompted to confirm, enter **Y** and press **Enter**

By using Resource Logs you can put in place an extra line of defense against accidental or malicious changing and/or deletion of your most important resources. It's not perfect, as your administrators can still remove these locks, but doing so requires a conscious effort, as the only purpose for removing a lock is to circumvent it. As these locks sit outside of RBAC you can apply them and be sure that they are impacting all your users, regardless of what roles or custom permissions you may have granted them.

Results: You have now completed this lab.

20 Module 1: Lab 9: Transferring Subscriptions

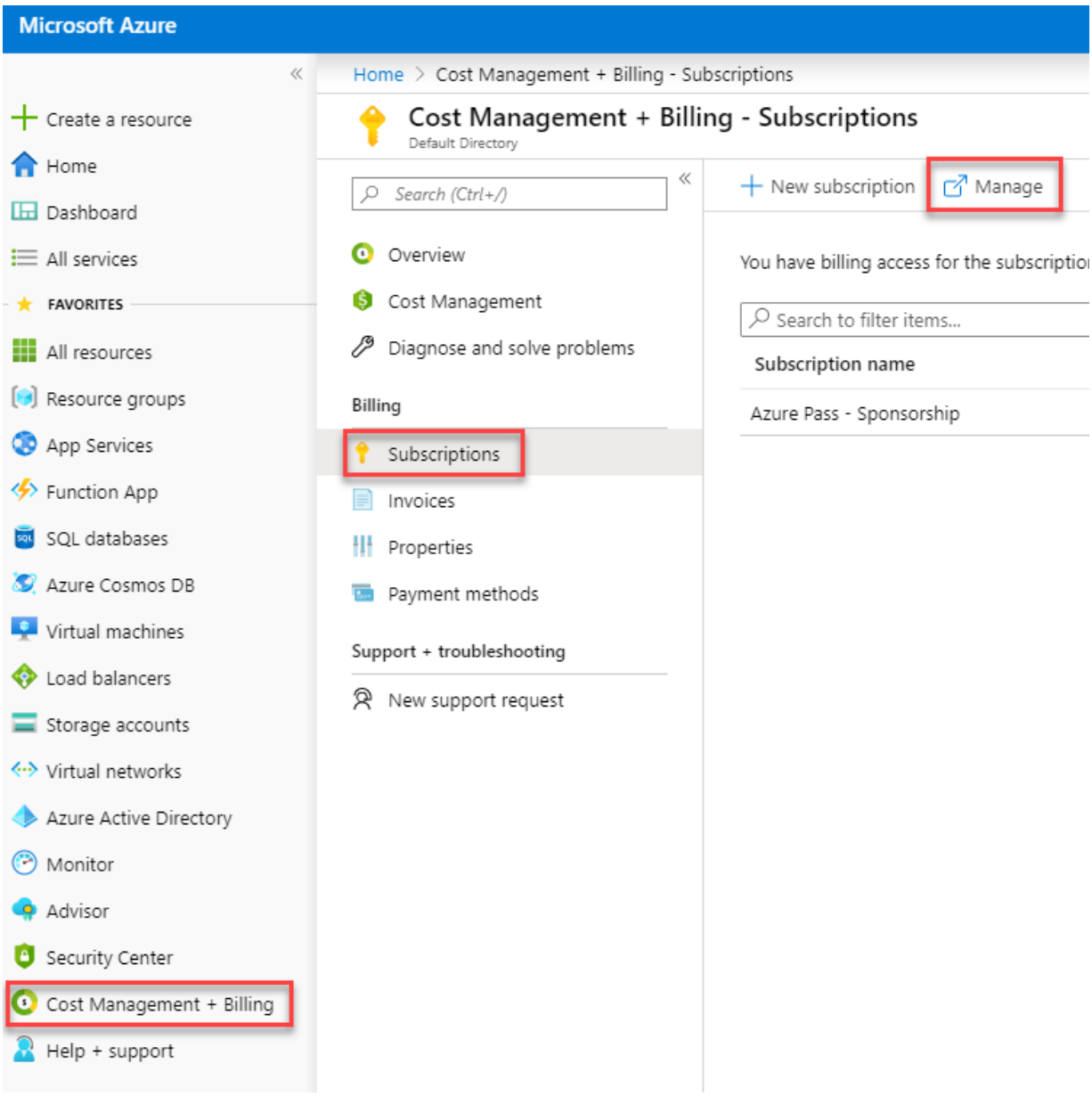
Scenario

Occasionally, a need arises for transferring a subscription from an owner to an Azure AD tenant. In order to transfer a subscription from an Azure AD owner to another subscription, you need access to another subscription. If you do not have access to multiple subscriptions at this time, just review the process outlined below.

20.1 Exercise 1: Transfer Azure subscriptions between Azure AD tenants

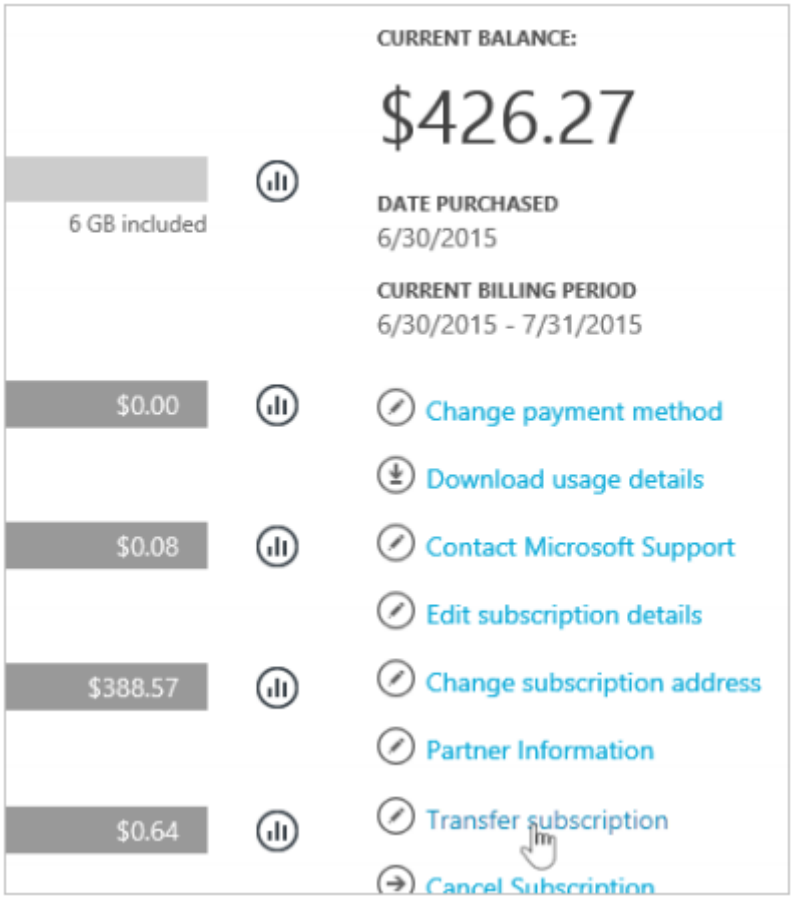
20.1.1 Task 1: To transfer the ownership of an Azure subscription

1. Sign in at the Azure Portal as the account admin.
2. Navigate to **Cost Management + Billing** > **Subscriptions** and click **Manage**.



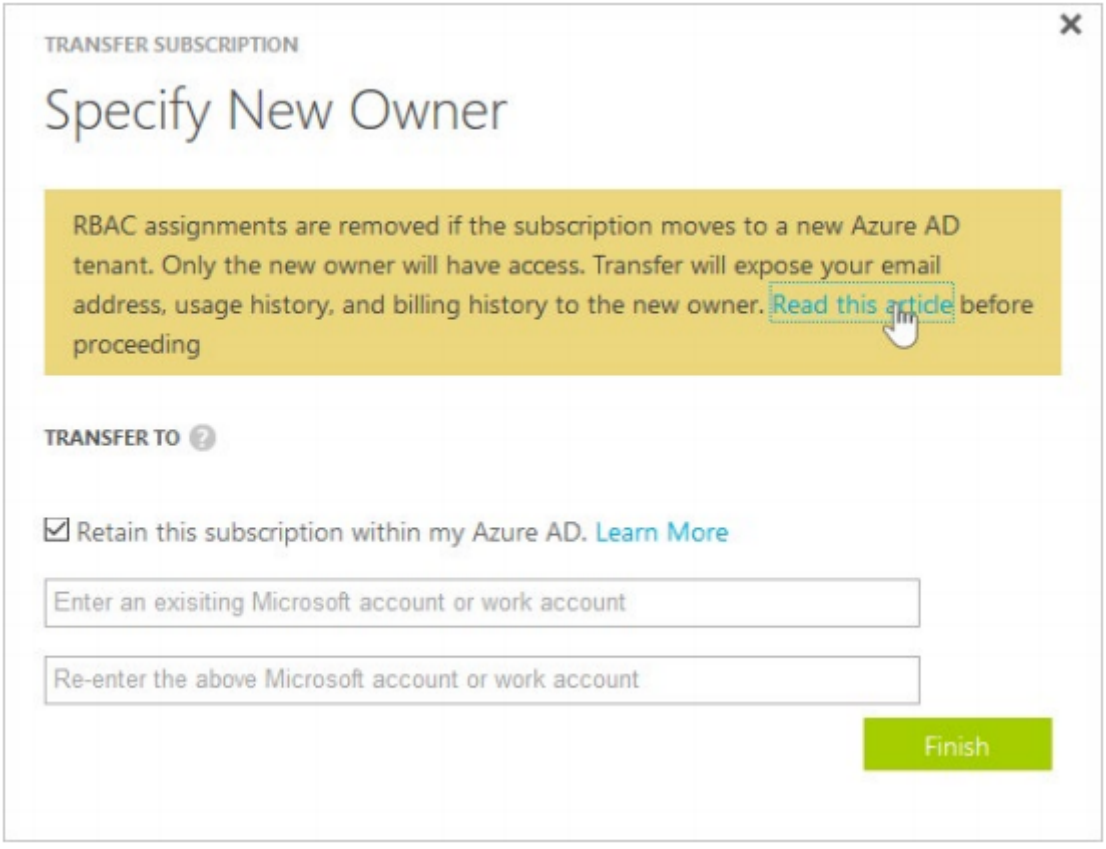
3. Select your subscription.
4. Verify that your subscription is eligible for self-serve transfer by checking the Offer and Offer ID against the supported offers list.

   Note: At the time of writing, Azure Pass - Sponsorship subscriptions are not eligible for transfer. Follow this guide through for reference.
5. Select **Transfer subscription**.

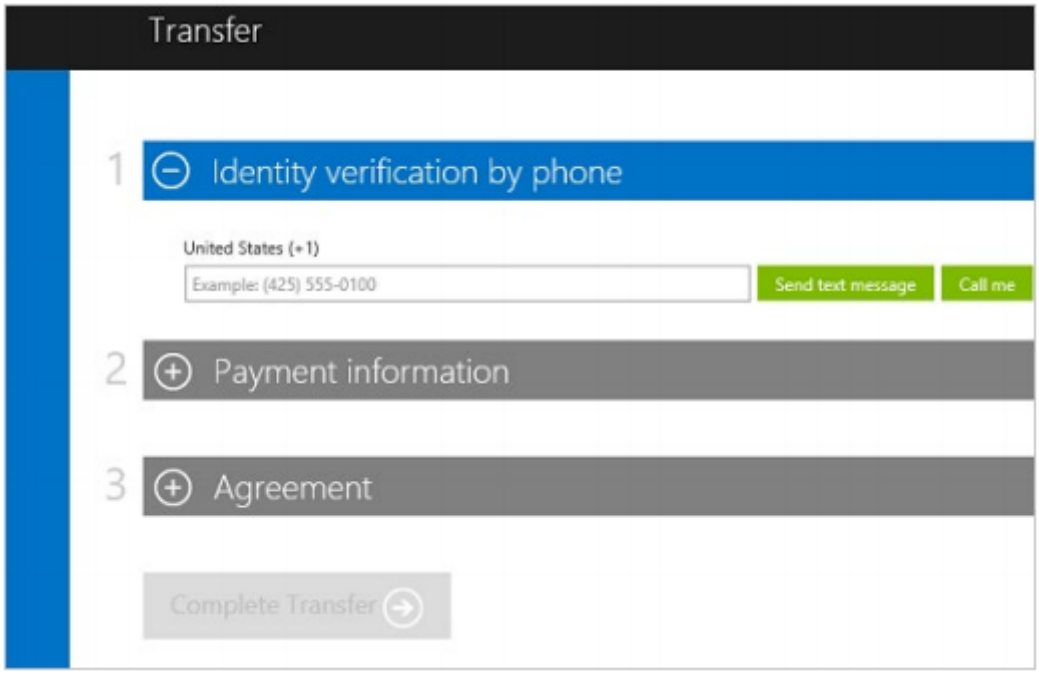


6. Specify the recipient.

   Note: If you transfer a subscription to a new Azure AD tenant, all role assignments in RBAC will be permanently deleted from the source tenant and not migrated to the target tenant.



7. The recipient automatically gets an email with an acceptance link.
8. The recipient selects the link and follows the instructions, including entering their payment information



9. Azure completes the subscription transfer.

At this point, billing ownership of the Azure subscription would be transfered to the new subscription.

Results: You have now completed this module.

21 Module 2: Lab 2 - Function Apps

Azure Function Apps uses the Azure App Service infrastructure. This topic shows you how to create a function app in the Azure portal. A function app is the container that hosts the execution of individual functions. When you create a function app in the App Service hosting plan, your function app can use all the features of App Service.

21.1 Exercise 1: Create a Function and Trigger

21.1.1 Task 1: Lab Setup

1. In your browser, navigate to the following URL to open the ARM template:

   `https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2F`

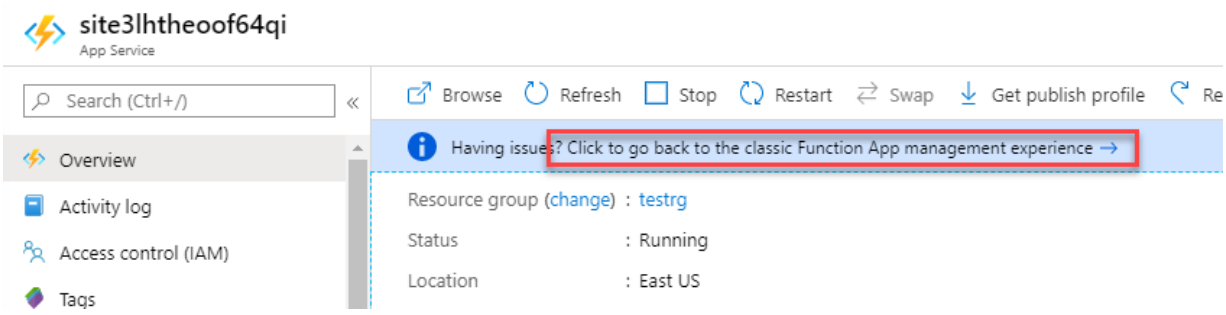
2. Click **Create new** under the Resource Group section.
3. Enter **myResourceGroup** as the name and click **OK**.
4. Select the check box at the bottom of the blade to agree to the terms.
5. Select **Purchase**

21.1.2 Task 2: Add a HTTP trigger to your function app

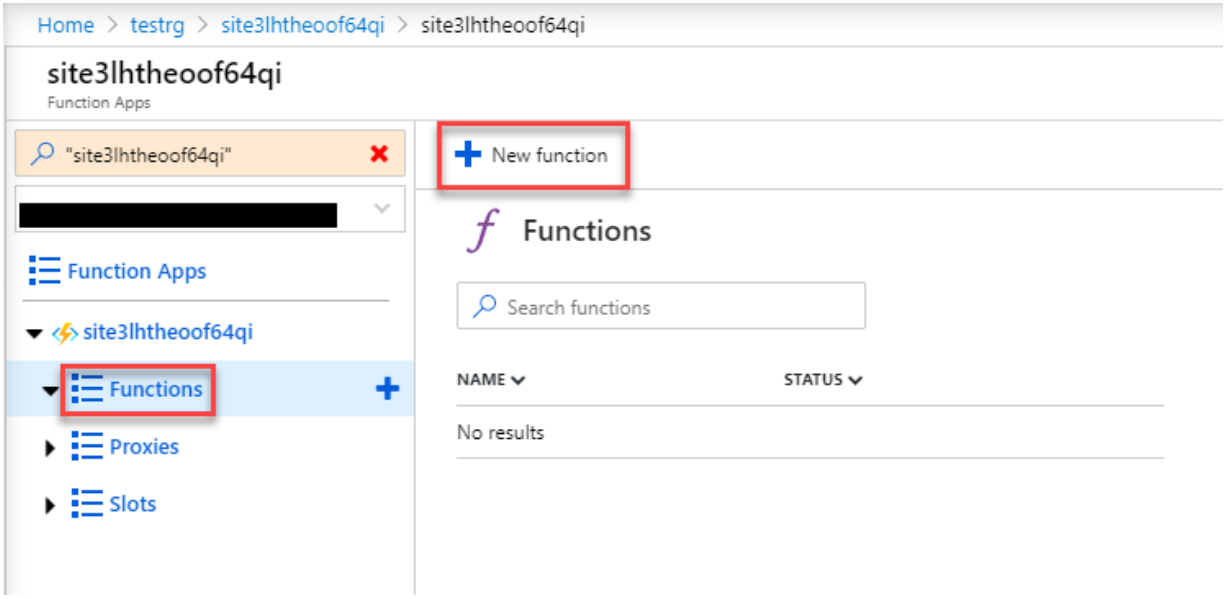
1. Select **Resource Groups**
2. Select the resource group you created in the lab setup
3. Select the function app service that has been created in the resource group

   Note: There are no functions currently assigned to the function app

4. Select **Click to go back to the classic Function App management experience**.



5. Click **Functions**.
6. Click + **New Function**.



7. In the top right click the slide button for **Experimental Language support**,

   Note: New languages have now been added to the triggers
8. Select **HTTP trigger**.
9. Change the language to **PowerShell**.
10. Leave the name as the default,
11. Make sure **Authorization** Level is set to **Function**,
12. Click **Create**.

You have now created template PS1 HTTP trigger. If the template code does not appear, refresh the page

21.1.3 Task 3: Test a REST call to the HTTP trigger

1. Note the name of the function app (**next to the function app icon**).
2. Under the **HTTP trigger** function click **Manage**.
3. Under function keys select **copy** under actions on the default function key and paste this in a notepad file
4. Navigate to the following URL and copy the PowerShell code

   `https://raw.githubusercontent.com/godeploy/AZ500/master/AZ500%20Mod2%20Lab%202/RESTgetHTTPtrigger.`
5. Open a **PowerShell ISE** window and and paste the PowerShell code from the previous **URL**. *(If the Script pane is not open, select View/Show Script Pane.)*
6. Populate the variable `$functionappname = ""` with the name of your function app
7. Populate the variable `$functionkey = ""` with the long function key copied from the portal in the earlier step

8. Run the powershell script (F5)
9. In the results in the **ISE** you should see the following output

```
This is result of  a normal GET operation calling your HTTP trigger
Hello

This is result of  a normal GET operation calling your HTTP trigger with an extra parameter passed
Hello World!

This is result of a normal PUT operation calling your HTTP trigger that feeds a hash table convert
Hello Max Power
```

Results: You have successfully built a HTTP trigger function app and communicated to it using REST based commands.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Congratulations: You have now completed this lab.

22 Module 2: Lab 3 - Create a Kubernetes Cluster

Azure Kubernetes Service (AKS) is a managed Kubernetes service that lets you quickly deploy and manage clusters. In this quickstart, you deploy an AKS cluster using the Azure CLI. A multi-container application that includes a web front end and a Redis instance is run in the cluster. You then see how to monitor the health of the cluster and pods that run your application.

22.1 Exercise 1: Create an AKS environment

22.1.1 Task 1: Prepare the environment and Create a Resource Group.

1. Open a browser and navigate to the **Azure Portal** `https://portal.azure.com`
2. Click **Cloudshell** icon.

3. Select **Azure CLI BASH** if required and create a storage account.
4. Run the following command in the **Cloud Shell** to create a new **Resource Group.**

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

   Note: If you receive an error regarding the `Microsoft.Network` resource provider not being registered then run the following command and rerun the command in Step 4. Otherwise continue to Task 2.

   ```
   az provider register --namespace 'Microsoft.Network'
   ```

22.1.2 Task 2: Create the AKS Cluster in CLI

1. Run the following command in the **CloudShell**.

   ```
   az aks create  --resource-group myAKSResourceGroup --name myAKSCluster --node-count 1 --enable-add
   ```

2. After a few minutes, the command completes and returns **JSON-formatted** information about the cluster.

22.1.3 Task 3: Connect to the cluster

To manage a Kubernetes cluster, you use kubectl, the Kubernetes command-line client. If you use Azure Cloud Shell, `kubectl` is already installed.

1. Open **Azure Cloud Shell** in Bash mode.
2. To configure `kubectl` to connect to your **Kubernetes cluster**, use the az-aks-get-credentials command. This command downloads credentials and configures the **Kubernetes CLI** to use them.

```
az aks get-credentials --resource-group myAKSResourceGroup --name myAKSCluster
```

3. To verify the connection to your cluster, use the kubectl-get command to return a list of the cluster nodes.

```
kubectl get nodes
```

4. The following example output shows the single node created in the previous steps. Make sure that the status of the node is *Ready*:

```
NAME                       STATUS   ROLES   AGE     VERSION
aks-nodepool1-31718369-0   Ready    agent   6m44s   v1.9.11
```

22.1.4 Task 4: Run the application

A Kubernetes manifest file defines a desired state for the cluster, such as what container images to run. In this lab, a manifest is used to create all objects needed to run the Azure Vote application. This manifest includes two kubernetes-deployment - one for the sample Azure Vote Python applications, and the other for a Redis instance. Two kubernetes-service are also created - an internal service for the Redis instance, and an external service to access the Azure Vote application from the internet. The manifest file has been created and saved to the Godeploy Github page for this lab. The file is azure-vote.yaml and can be found at **https://raw.githubusercontent.com/MicrosoftLearning/AZ-500-Azure-Security/master/Allfiles/Labs/Mod2_Lab**

1. Run the following command in the cloud shell, this will directly pull the yaml file needed from GitHub to deploy the AKS application

```
kubectl apply -f https://raw.githubusercontent.com/MicrosoftLearning/AZ-500-Azure-Security/master/
```

2. The following example output shows the **Deployments and Services** created successfully:

```
deployment "azure-vote-back" created
service "azure-vote-back" created
deployment "azure-vote-front" created
service "azure-vote-front" created
```

Note: When the application runs, a Kubernetes service exposes the application front end to the internet. This process can take a few minutes to complete.

22.1.5 Task 5: Test the application

1. To monitor progress, use the kubectl-get command with the `--watch` argument.

```
kubectl get service azure-vote-front --watch
```

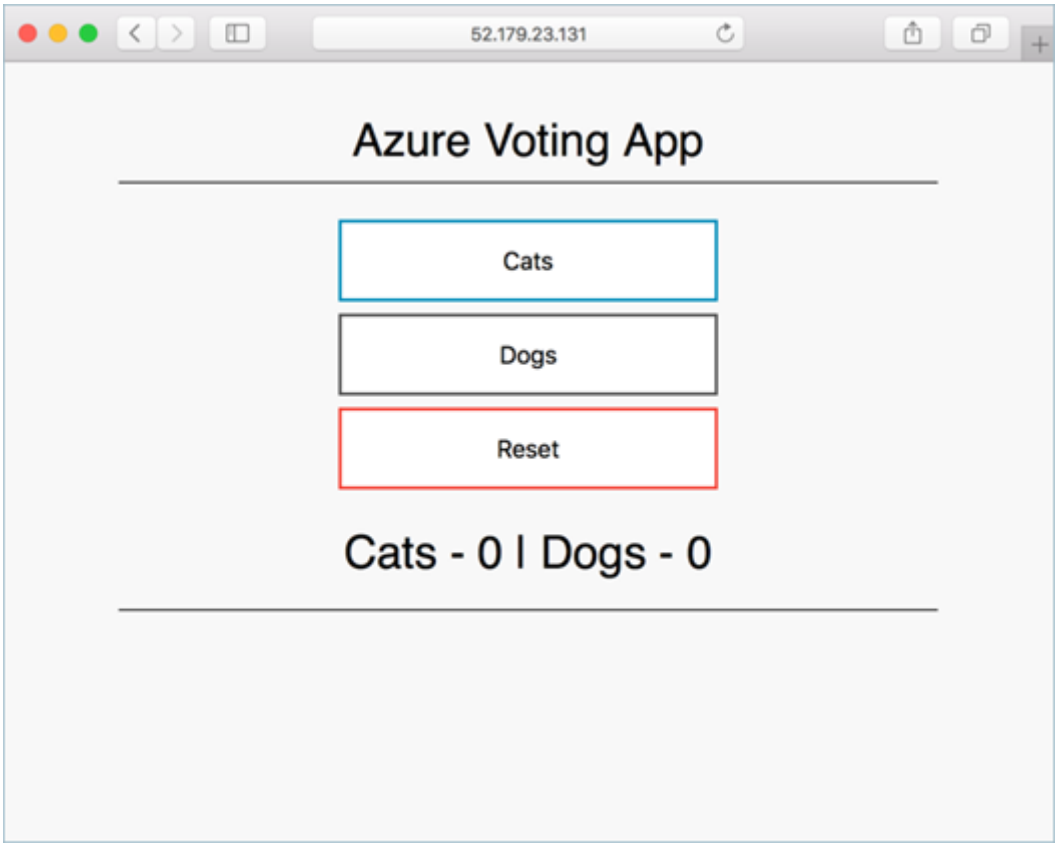
2. Initially the *EXTERNAL-IP* for the *azure-vote-front* service is shown as *pending*.

```
NAME               TYPE           CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
azure-vote-front   LoadBalancer   10.0.37.27   <pending>     80:30572/TCP     6s
```

3. When the *EXTERNAL-IP* address changes from *pending* to an actual public IP address, use `CTRL-C` to stop the `kubectl` watch process. The following example output shows a valid public IP address assigned to the service:

```
azure-vote-front   LoadBalancer   10.0.37.27   52.179.23.131   80:30572/    TCP   2m
```

4. To see the Azure Vote app in action, open a web browser to the external IP address of your service as shown in the result of the previous command.



22.1.6 Task 6: Monitor health and logs

When the AKS cluster was created, Azure Monitor for containers was enabled to capture health metrics for both the cluster nodes and pods. These health metrics are available in the Azure portal.

To see current status, uptime, and resource usage for the Azure Vote pods, complete the following steps:

1. Open a web browser to the Azure portal.
2. Select your resource group, such as *myAKSResourceGroup*, then select your AKS cluster, such as *myAKSCluster*.
3. Under **Monitoring** on the left-hand side, choose **Insights**
4. Across the top, choose to + **Add Filter**
5. Select *Namespace* as the property, then choose `<All but kube-system>`
6. Choose to view the **Containers**.

   The *azure-vote-back* and *azure-vote-front* containers are displayed
7. Click the view live data button then switch to the running application (Cats/Dogs) to click the vote buttons then switch back to view the live data under the Logs window.

22.1.7 Task 7: Delete the cluster

When the cluster is no longer needed, use the **`az group delete`** command to remove the resource group, container service, and all related resources.

1. Run the following command in the **Cloud Shell** in Bash mode to delete the Resource Group.

   ```
   az group delete --name myAKSResourceGroup --yes --no-wait
   ```

   Note: It may take some time to delete the Resource Group. The `--no-wait` option runs the command in the background.

Results: You have now completed this lab.

23 Module 2: Lab 4 - Create a VNet

Scenario

In this module, you'll will be introduced to Azure virtual networks. What are virtual networks and how are they organized? How do you create and configure virtual networks with templates, PowerShell, CLI, or the Azure portal? What is the difference between public, private, static, and dynamic IP addressing? How are system routes, routing tables, and routing algorithms used? Lessons include:

- Introducing Virtual Networks
- Creating Azure Virtual Networks
- Review of IP Addressing

23.1 Exercise 1: Create a virtual network using the Azure portal

Scenario

A virtual network is the fundamental building block for your private network in Azure. It enables Azure resources, like virtual machines (VMs), to securely communicate with each other and with the internet. In this lab, you will learn how to create a virtual network using the Azure portal. Then, you can deploy two VMs into the virtual network, securely communicate between the two VMs, and connect to the VMs from the internet.

23.1.1 Task 1: Create a virtual network

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

Setting	Value
Subscription	Select your subscription.
Resource group	Select **Create new**, enter *myResourceGroup*, then select **OK**.
Name	Enter *myVirtualNetwork*.
Location	Select **East US**.

Select the IP Addresses tab and enter the following values:

Setting	Value
Address space	Enter *10.1.0.0/16*.
Subnet - Name	Enter *myVirtualSubnet*.
Subnet - Address range	Enter *10.1.0.0/24*.

3. Leave the rest as default and select **Review + create**, then click **Create**.

23.1.2 Task 2: Create virtual machines

Create two VMs in the virtual network:

1. On the upper-left side of the screen, select **Create a resource** > **Compute** > **Virtual machine**.
2. In **Create a virtual machine - Basics**, enter or select this information:

Setting	Value
PROJECT DETAILS	
Subscription	Select your subscription.
Resource group	Select **myResourceGroup**. You created this in the previous section.
INSTANCE DETAILS	
Virtual machine name	Enter *myVm1*.
Region	Select **East US**.
Availability options	Leave the default **No infrastructure redundancy required**.
Image	Select **Windows Server 2019 Datacenter**.
Size	Select **Standard DS1 v2**.

Setting	Value
ADMINISTRATOR ACCOUNT	
Username	Enter a username of your choosing.
Password	Pa55w.rd1234
Confirm Password	Reenter password.
INBOUND PORT RULES	
Public inbound ports	Select **Allow selected ports**.
Select inbound ports	Select **HTTP** and **RDP**.
SAVE MONEY	
Already have a Windows license?	Leave the default **No**.

3. Select **Next : Disks**.
4. In **Create a virtual machine - Disks**, leave the defaults and select **Next : Networking**.
5. In **Create a virtual machine - Networking**, select this information:

Setting	Value
Virtual network	Leave the default **myVirtualNetwork**.
Subnet	Leave the default **myVirtualSubnet (10.1.0.0/24)**.
Public IP	Leave the default **(new) myVm1-ip**.

6. Select **Next : Management**.
7. In **Create a virtual machine - Management**, for **Diagnostics storage account**, select **Create New**.
8. In **Create storage account**, enter or select this information:

Setting	Value
Name	Enter *myvmstorageaccount*. If this name is taken, create a unique name.
Account kind	Leave the default **Storage (general purpose v1)**.
Performance	Leave the default **Standard**.
Replication	Leave the default **Locally-redundant storage (LRS)**.

9. Select **OK**
10. Select **Review + create**. You're taken to the **Review + create** page where Azure validates your configuration.
11. When you see the **Validation passed** message, select **Create**.

23.1.3 Task 3: Create the second VM

1. Complete steps 1 and 9 from above.

   Note: In step 2, for the **Virtual machine name**, enter *myVm2*. In step 7, for **Diagnosis storage account**, make sure you select **myvmstorageaccount**.

2. Select **Review + create**. You're taken to the **Review + create** page and Azure validates your configuration.
3. When you see the **Validation passed** message, select **Create**.

23.1.4 Task 4: Connect to a VM from the internet

After you've created *myVm1*, connect to the internet.

1. In the portal's search bar, enter *myVm1*.
2. Select the **Connect** button.

3. Select **RDP**, then **Download RDP File**. Azure creates a Remote Desktop Protocol (*.rdp*) file and downloads it to your computer.
4. Open the downloaded *.rdp* file.
    1. If prompted, select **Connect**.
    2. Enter the username and password you specified when creating the VM.

    Note: You may need to select **More choices** > **Use a different account**, to specify the credentials you entered when you created the VM.
5. Select **OK**.
6. You may receive a certificate warning during the sign in process. If you receive a certificate warning, select **Yes** or **Continue**.

23.1.5 Task 5: Communicate between VMs

1. In the Remote Desktop of *myVm1*, open PowerShell.
2. Enter `ping myVm2`

    You'll receive a message similar to this:

    ```
    Pinging myVm2.0v0zze1s0uiedpvtxz5z0r0cxg.bx.internal.clouda
    Request timed out.
    Request timed out.
    Request timed out.
    Request timed out.

    Ping statistics for 10.1.0.5:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
    ```

    The `ping` fails because `ping` uses the Internet Control Message Protocol (ICMP). By default, ICMP isn't allowed through the Windows firewall.
3. To allow *myVm2* to ping *myVm1* in a later step, enter this command:

    ```
    New-NetFirewallRule -DisplayName "Allow ICMPv4-In" -Protocol ICMPv4
    ```

    This command allows ICMP inbound through the Windows firewall:
4. Close the remote desktop connection to *myVm1*.
5. Complete the steps in **Connect to a VM from the internet** task again, but connect to *myVm2*.
6. From a command prompt, enter `ping myvm1`.

    You'll get back something like this message:

    ```
    Pinging myVm1.0v0zze1s0uiedpvtxz5z0r0cxg.bx.internal.cloudapp.net [10.1.0.4] with 32 bytes of data
    Reply from 10.1.0.4: bytes=32 time=1ms TTL=128
    Reply from 10.1.0.4: bytes=32 time<1ms TTL=128
    Reply from 10.1.0.4: bytes=32 time<1ms TTL=128
    Reply from 10.1.0.4: bytes=32 time<1ms TTL=128

    Ping statistics for 10.1.0.4:
        Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms
    ```

    You receive replies from *myVm1* because you allowed ICMP through the Windows firewall on the *myVm1* VM in step 3.
7. Close the remote desktop connection to *myVm2*.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this lab.

24 Module 2: Lab 5 - NSGs

You can filter network traffic inbound to and outbound from a virtual network subnet with a network security group. Network security groups contain security rules that filter network traffic by IP address, port, and protocol. Security rules are applied to resources deployed in a subnet. In this tutorial, you learn how to:

- Create a network security group and security rules
- Create a virtual network and associate a network security group to a subnet
- Deploy virtual machines (VM) into a subnet
- Test traffic filters

24.1 Exercise 1: Filter network traffic with a network security group using the Azure portal

24.1.1 Task 1: Create a virtual network

1. Select **+ Create a resource** on the upper left corner of the Azure portal.
2. Select **Networking**, and then select **Virtual network**.
3. Enter, or select, the following information, accept the defaults for the remaining settings, and then select **Review + create**, then click **Create**:

Setting	Value
Name	myVirtualNetwork
Subscription	Select your subscription.
Resource group	Select **Create new** and enter *myResourceGroup*.
Location	Select **East US**.

Select the IP Addresses tab and enter the following values:

Setting	Value
Address space	10.0.0.0/16
Subnet- Name	Change the default subnet name to **mySubnet** and click **Save**
Subnet - Address range	10.0.0.0/24

24.1.2 Task 2: Create application security groups

An application security group enables you to group together servers with similar functions, such as web servers.

1. Select **+ Create a resource** on the upper left corner of the Azure portal.
2. In the **Search the Marketplace** box, enter *Application security group*. When **Application security group** appears in the search results, select it, select **Application security group** again under **Everything**, and then select **Create**.
3. Enter, or select, the following information, and then select **Create**:

Setting	Value
Name	myAsgWebServers
Subscription	Select your subscription.
Resource group	Select **Use existing** and then select **myResourceGroup**.
Location	East US

4. Complete step 3 again, specifying the following values:

Setting	Value
Name	myAsgMgmtServers
Subscription	Select your subscription.
Resource group	Select **Use existing** and then select **myResourceGroup**.

Setting	Value
Location	East US

24.1.3 Task 3: Create a network security group

1. Select **+ Create a resource** on the upper left corner of the Azure portal.
2. Select **Networking**, and then select **Network security group**.
3. Enter, or select, the following information, and then select **Create**:

Setting	Value
Name	myNsg
Subscription	Select your subscription.
Resource group	Select **Use existing** and then select *myResourceGroup*.
Location	East US

24.1.4 Task 4: Associate network security group to subnet

1. In the *Search resources, services, and docs* box at the top of the portal, begin typing *myNsg*. When **myNsg** appears in the search results, select it.
2. Under **SETTINGS**, select **Subnets** and then select **+ Associate**.
3. Under **Associate subnet**, select **Virtual network** and then select **myVirtualNetwork**. Select **Subnet**, select **mySubnet**, and then select **OK**.

24.1.5 Task 5: Create security rules

1. Under **SETTINGS**, select **Inbound security rules** and then select **+ Add**.
2. Create a security rule that allows ports 80 and 443 to the **myAsgWebServers** application security group. Under **Add inbound security rule**, enter, or select the following values, accept the remaining defaults, and then select **Add**:

Setting	Value
Destination	Select **Application security group**, and then select **myAsgWebServers** for **Application sec**
Destination port ranges	Enter 80,443
Protocol	Select TCP
Name	Allow-Web-All

3. Complete step 2 again, using the following values:

Setting	Value
Destination	Select **Application security group**, and then select **myAsgMgmtServers** for **Application s**
Destination port ranges	Enter 3389
Protocol	Select TCP
Priority	Enter 110
Name	Allow-RDP-All

In this tutorial, RDP (port 3389) is exposed to the internet for the VM that is assigned to the *myAsgMgmtServers* application security group. For production environments, instead of exposing port 3389 to the internet, it's recommended that you connect to Azure resources that you want to manage using a VPN or private network connection.

24.1.6 Task 6: Create virtual machines

1. Select **+ Create a resource** found on the upper left corner of the Azure portal.

2. Select **Compute**, and then select **Windows Server 2016 Datacenter**.
3. Enter, or select, the following information, and accept the defaults for the remaining settings:

Setting	Value
Subscription	Select your subscription.
Resource group	Select **Use existing** and select **myResourceGroup**.
Name	myVmWeb
Location	Select **East US**.
User name	Enter a user name of your choosing.
Password	Pa55w.rd1234

4. Select a size for the VM and then select **Select**.
5. Under **Networking**, select the following values, and accept the remaining defaults:

Setting	Value
Virtual network	Select **myVirtualNetwork**.
NIC network security group	Select **None**.

6. Under **Management**, select **Off** for **Boot diagnostics**.
7. Select **Review + Create** at the bottom left corner, select **Create** to start VM deployment.

24.1.7 Task 7: Create the second VM

Complete above steps 1-7 again, but in step 3, name the VM *myVmMgmt*. The VM takes a few minutes to deploy. Do not continue to the next step until the VM is deployed.

24.1.8 Task 8: Associate network interfaces to an ASG

When the portal created the VMs, it created a network interface for each VM and attached the network interface to the VM. Add the network interface for each VM to one of the application security groups you created previously:

1. In the *Search resources, services, and docs* box at the top of the portal, begin typing *myVmWeb*. When the **myVmWeb** VM appears in the search results, select it.
2. Under **SETTINGS**, select **Networking**. Select **Application security groups**, then **Configure the application security groups**, then select **myAsgWebServers** for **Application security groups**, and then select **Save**.
3. Complete steps 1 and 2 again, searching for the **myVmMgmt** VM and selecting the **myAsgMgmtServers** ASG.

24.1.9 Task 9: Test traffic filters

1. Connect to the *myVmMgmt* VM. Enter *myVmMgmt* in the search box at the top of the portal. When **myVmMgmt** appears in the search results, select it. Select the **Connect** button.
2. Select **RDP**, then **Download RDP file**.
3. Open the downloaded rdp file and select **Connect**. Enter the user name and password you specified when creating the VM. You may need to select **More choices**, then **Use a different account**, to specify the credentials you entered when you created the VM.
4. Select **OK**.
5. You may receive a certificate warning during the sign-in process. If you receive the warning, select **Yes** or **Continue** to proceed with the connection.

   The connection succeeds because port 3389 is allowed inbound from the internet to the *myAsgMgmtServers* application security group that the network interface attached to the *myVmMgmt* VM is in.

6. Connect to the *myVmWeb* VM from the *myVmMgmt* VM by entering the following command in a PowerShell session:

   ```
   mstsc /v:myVmWeb
   ```

   You are able to connect to the myVmWeb VM from the myVmMgmt VM because VMs in the same virtual network can communicate with each other over any port, by default. You can't, however, create a remote desktop connection to the *myVmWeb* VM from the internet because the security rule for the *myAsgWebServers* doesn't allow port 3389 inbound from the internet, and inbound traffic from the Internet is denied to all resources, by default.

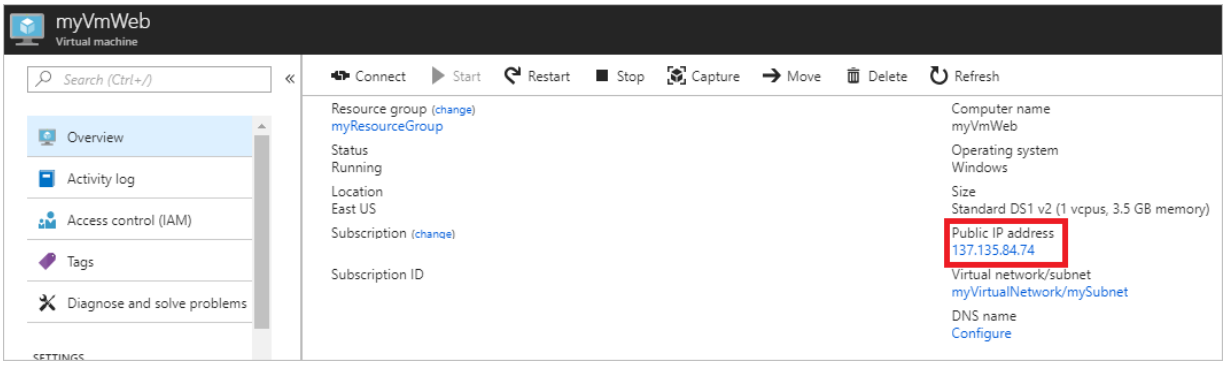
7. To install Microsoft IIS on the *myVmWeb* VM, enter the following command from a PowerShell session on the *myVmWeb* VM:

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

8. After the IIS installation is complete, disconnect from the *myVmWeb* VM, which leaves you in the *myVmMgmt* VM remote desktop connection.

9. Disconnect from the *myVmMgmt* VM.

10. In the *Search resources, services, and docs* box at the top of the Azure portal, begin typing *myVmWeb* from your computer. When **myVmWeb** appears in the search results, select it. Note the **Public IP address** for your VM. The address shown in the following picture is 137.135.84.74, but your address is different:



11. To confirm that you can access the *myVmWeb* web server from the internet, open an internet browser on your computer and browse to `http://<public-ip-address-from-previous-step>`. You see the IIS welcome screen because port 80 is allowed inbound from the internet to the *myAsgWebServers* application security group that the network interface attached to the *myVmWeb* VM is in.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this lab.

25 Module 2: Lab 6 - NVA

Scenario

Azure routes traffic between all subnets within a virtual network, by default. You can create your own routes to override Azure's default routing. The ability to create custom routes is helpful if, for example, you want to route traffic between subnets through a network virtual appliance (NVA). In this lab, you learn how to:

- Create a route table
- Create a route
- Create a virtual network with multiple subnets
- Associate a route table to a subnet
- Create an NVA that routes traffic
- Deploy virtual machines (VM) into different subnets
- Route traffic from one subnet to another through an NVA

25.1 Exercise 1: Route network traffic with a route table using the Azure portal

25.1.1 Task 1: Create a route table

1. On the upper-left side of the screen, select **Create a resource** > **Networking** > **Route table**, alternatively search for Route table in the Azure portal.
2. In **Create route table**, enter or select this information:

Setting	Value
Name	Enter *myRouteTablePublic*.
Subscription	Select your subscription.
Resource group	Select **Create new**, enter *myResourceGroup*, and select *OK*.
Location	Leave the default **East US**.
Virtual network gateway route propagation	Leave the default **Enabled**.

3. Select **Create**.

25.1.2 Task 2: Create a route

1. In the portal's search bar, enter *myRouteTablePublic*.
2. When **myRouteTablePublic** appears in the search results, select it.
3. In **myRouteTablePublic** under **Settings**, select **Routes** > **+ Add**.
4. In **Add route**, enter or select this information:

Setting	Value
Route name	Enter *ToPrivateSubnet*.
Address prefix	Enter *10.0.1.0/24*.
Next hop type	Select **Virtual appliance**.
Next hop address	Enter *10.0.2.4*.

5. Select **OK**.

25.1.3 Task 3: Associate a route table to a subnet

Before you can associate a route table to a subnet, you have to create a virtual network and subnet.

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

Setting	Value
Subscription	Select your subscription.
Resource group	Select ***Select existing*** > **myResourceGroup**.
Name	Enter *myVirtualNetwork*.
Location	Leave the default **East US**.

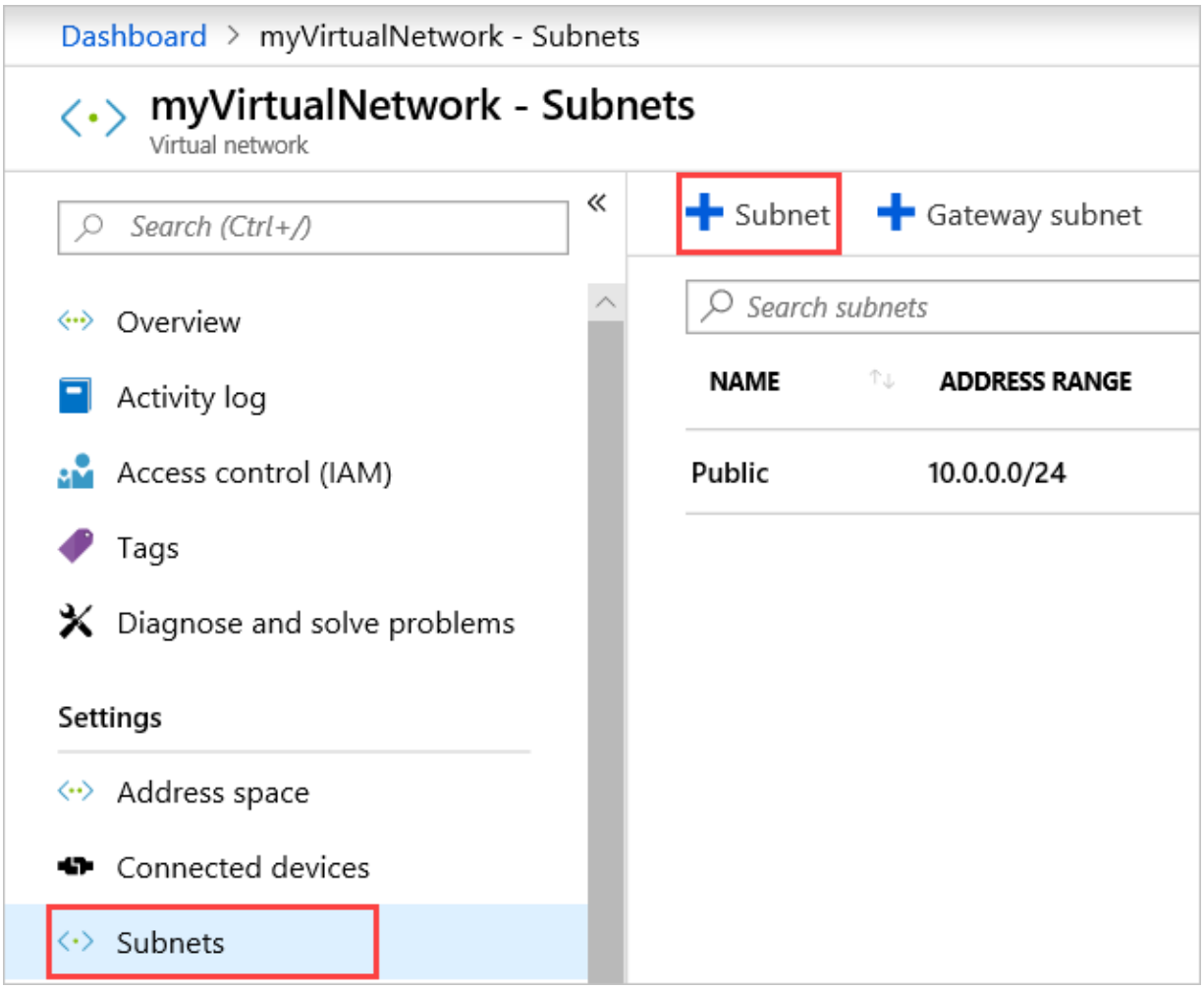
Select the IP Addresses tab and enter the following values:

Setting	Value
Address space	Enter *10.0.0.0/16*.
Subnet - Name	Select the existing default subnet to change the name to Public then select Save.
Subnet - Address range	Enter *10.0.0.0/24*.

3. Leave the rest of the defaults and select **Review + create**, then click **Create**.

25.1.4 Task 4: Add subnets to the virtual network

1. In the portal's search bar, enter *myVirtualNetwork*.
2. When **myVirtualNetwork** appears in the search results, select it.
3. In **myVirtualNetwork**, under **Settings**, select **Subnets** > + **Subnet**.



4. In **Add subnet**, enter this information:

Setting	Value
Name	Enter *Private*.
Address space	Enter *10.0.1.0/24*.

5. Leave the rest of the defaults and select **OK**.
6. Select + **Subnet** again. This time, enter this information:

Setting	Value
Name	Enter *DMZ*.
Address space	Enter *10.0.2.0/24*.

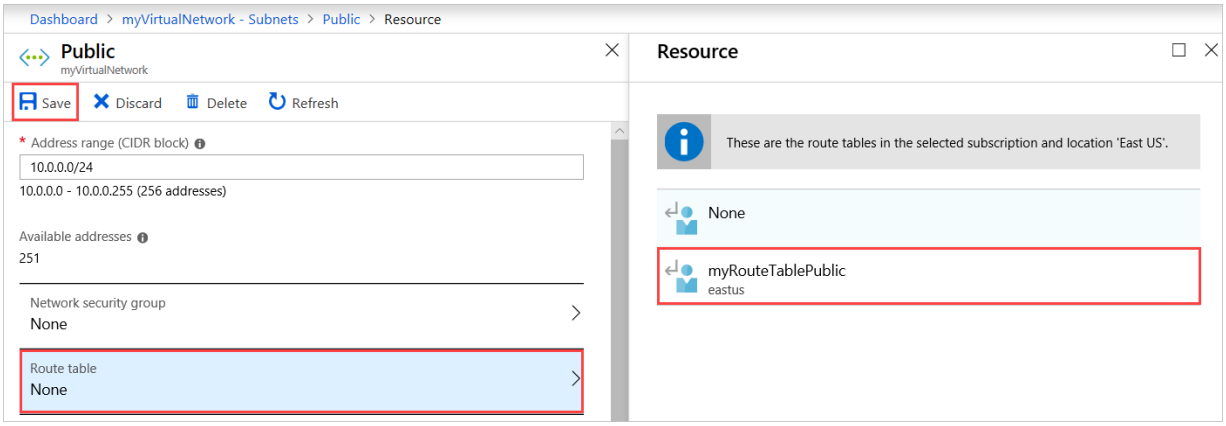
7. Like the last time, leave the rest of the defaults and select **OK**.

   Azure shows the three subnets: **Public**, **Private**, and **DMZ**.

25.1.5 Task 5: Associate myRouteTablePublic to your Public subnet

1. Select **Public**.

2. In **Public**, select **Route table** > **MyRouteTablePublic** > **Save**.



25.1.6 Task 6: Create an NVA

NVAs are VMs that help with network functions like routing and firewall optimization. You can select a different operating system if you want. This tutorial assumes you're using **Windows Server 2016 Datacenter**.

1. On the upper-left side of the screen, select **Create a resource** > **Compute** > **Virtual Machine**.
2. In **Create a virtual machine - Basics**, enter or select this information:

Setting	Value
PROJECT DETAILS	
Subscription	Select your subscription.
Resource group	Select **myResourceGroup**
INSTANCE DETAILS	
Virtual machine name	Enter *myVmNva*
Region	Select **East US**.
Availability options	Leave the default **No infrastructure redundancy required**.
Image	Select **Windows Server 2016 Datacenter**.
Size	Select **Standard DS1 v2**.
ADMINISTRATOR ACCOUNT	
Username	Enter a user name of your choosing.
Password	Pa55w.rd1234
Confirm Password	Reenter password.
INBOUND PORT RULES	
Public inbound ports	Select **None**.
SAVE MONEY	
Already have a Windows license?	Leave the default **No**.

3. Select **Next : Disks**.
4. In **Create a virtual machine - Disks**, select the settings that are right for your needs.
5. Select **Next : Networking**.
6. In **Create a virtual machine - Networking**, select this information:

Setting	Value
Virtual network	Leave the default **myVirtualNetwork**.
Subnet	Select **DMZ (10.0.2.0/24)**.
Public IP	Select **None**. You don't need a public IP address. The VM won't connect over the internet.

7. Leave the rest of the defaults and select **Next : Management**.
8. In **Create a virtual machine - Management**, for **Diagnostics storage account**, select **Create New**.

9. In **Create storage account**, enter or select this information:

Setting	Value
Name	Enter *a unique storage account name.*
Account kind	Leave the default **Storage (general purpose v1)**.
Performance	Leave the default **Standard**.
Replication	Leave the default **Locally-redundant storage (LRS)**.

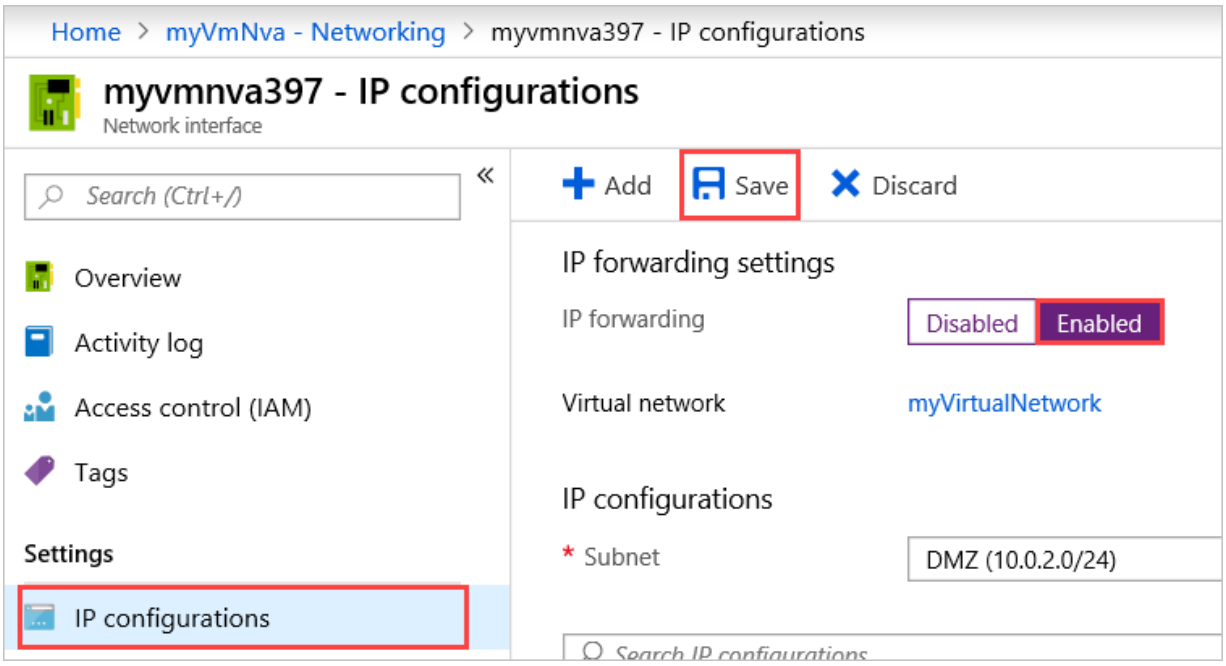
10. Select **OK**
11. Select **Review + create**. You're taken to the **Review + create** page and Azure validates your configuration.
12. When you see that **Validation passed**, select **Create**.

    The VM takes a few minutes to create. Don't keep going until Azure finishes creating the VM. The **Your deployment is underway** page will show you deployment details.
13. When your VM is ready, select **Go to resource**.

25.1.7 Task 7: Turn on IP forwarding

Turn on IP forwarding for *myVmNva*. When Azure sends network traffic to *myVmNva*, if the traffic is destined for a different IP address, IP forwarding will send the traffic to the correct location.

1. On **myVmNva**, under **Settings**, select **Networking**.
2. Select **myvmnva123**. That's the network interface Azure created for your VM. It will have a string of numbers to make it unique for you.
3. Under **Settings**, select **IP configurations**.
4. On **myvmnva123 - IP configurations**, for **IP forwarding**, select **Enabled** and then select **Save**.



25.1.8 Task 8: Create public and private virtual machines

Create a public VM and a private VM in the virtual network. Later, you'll use them to see that Azure routes the *Public* subnet traffic to the *Private* subnet through the NVA.

1. Complete steps 1-12 of the Create an NVA task. Use most of the same settings. These values are the ones that have to be different:

Setting	Value
PUBLIC VM	
BASICS	
Virtual machine name	Enter *myVmPublic*.
NETWORKING	
Subnet	Select **Public (10.0.0.0/24)**.
Public IP address	Accept the default.
Public inbound ports	Select **Allow selected ports**.
Select inbound ports	Select **HTTP** and **RDP**. *If the Networking tab does not allow the selected ports to be chang*
MANAGEMENT	
Diagnostics storage account	Leave the default.
PRIVATE VM	
BASICS	
Virtual machine name	Enter *myVmPrivate*.
NETWORKING	
Subnet	Select **Private (10.0.1.0/24)**.
Public IP address	Accept the default.
Public inbound ports	Select **Allow selected ports**.
Select inbound ports	Select **HTTP** and **RDP**.
MANAGEMENT	
Diagnostics storage account	Leave the default.

You can create the *myVmPrivate* VM while Azure creates the *myVmPublic* VM. Don't continue with the rest of the steps until Azure finishes creating both VMs.

25.1.9 Task 9: Route traffic through an NVA

1. Sign in to myVmPrivate over remote desktop
2. In the portal's search bar, enter *myVmPrivate*.
3. When the **myVmPrivate** VM appears in the search results, select it.
4. Select **Connect** then select **RDP** to create a remote desktop connection to the *myVmPrivate* VM.
5. In **Connect to virtual machine**, select **Download RDP File**. Azure creates a Remote Desktop Protocol (*.rdp*) file and downloads it to your computer.
6. Open the downloaded *.rdp* file.
    1. If prompted, select **Connect**.
    2. Enter the user name and password you specified when creating the Private VM.
    3. You may need to select **More choices** > **Use a different account**, to use the Private VM credentials.
7. Select **OK**.

    You may receive a certificate warning during the sign in process.
8. Select **Yes** to connect to the VM.

25.1.10 Task 10: Enable ICMP through the Windows firewall

In a later step, you'll use the trace route tool to test routing. Trace route uses the Internet Control Message Protocol (ICMP), which the Windows Firewall denies by default. Enable ICMP through the Windows firewall.

1. In the Remote Desktop of *myVmPrivate*, open PowerShell.
2. Enter this command:

    ```
    New-NetFirewallRule -DisplayName "Allow ICMPv4-In" -Protocol ICMPv4
    ```

    You're using trace route to test routing in this tutorial. For production environments, we don't recommend allowing ICMP through the Windows Firewall.

25.1.11 Task 11: Turn on IP forwarding within myVmNva

You turned on IP forwardingfor the VM's network interface using Azure. The VM's operating system also has to forward network traffic. Turn on IP forwarding for *myVmNva* VM's operating system with these commands.

1. From a command prompt on the *myVmPrivate* VM, open a remote desktop to the *myVmNva* VM:

   ```
   mstsc /v:myvmnva
   ```

2. From PowerShell on the *myVmNva*, enter this command to turn on IP forwarding:

   ```
   Set-ItemProperty -Path HKLM:\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters -Name IpEnableRout
   ```

3. Restart the *myVmNva* VM. From the taskbar, select **Start button** > **Power button**, **Other (Planned)** > **Continue**.

   That also disconnects the remote desktop session.

4. After the *myVmNva* VM restarts, create a remote desktop session to the *myVmPublic* VM. While still connected to the *myVmPrivate* VM, open a command prompt and run this command:

   ```
   mstsc /v:myVmPublic
   ```

5. In the Remote Desktop of *myVmPublic*, open PowerShell.

6. Enable ICMP through the Windows firewall by entering this command:

   ```
   New-NetFirewallRule -DisplayName "Allow ICMPv4-In" -Protocol ICMPv4
   ```

25.1.12 Task 12: Test the routing of network traffic

First, let's test routing of network traffic from the *myVmPublic* VM to the *myVmPrivate* VM.

1. From PowerShell on the *myVmPublic* VM, enter this command:

   ```
   tracert myVmPrivate
   ```

   The response is similar to this example:

   ```
   Tracing route to myVmPrivate.vpgub4nqnocezhjgurw44dnxrc.bx.internal.cloudapp.net [10.0.1.4]
   over a maximum of 30 hops:

   1    <1 ms     *        1 ms  10.0.2.4
   2     1 ms     1 ms     1 ms  10.0.1.4

   Trace complete.
   ```

   You can see the first hop is to 10.0.2.4. It's NVA's private IP address. The second hop is to the private IP address of the *myVmPrivate* VM: 10.0.1.4. Earlier, you added the route to the *myRouteTablePublic* route table and associated it to the *Public* subnet. As a result, Azure sent the traffic through the NVA and not directly to the *Private* subnet.

2. Close the remote desktop session to the *myVmPublic* VM, which leaves you still connected to the *myVmPrivate* VM.

3. From a command prompt on the *myVmPrivate* VM, enter this command:

   ```
   tracert myVmPublic
   ```

   It tests the routing of network traffic from the *myVmPrivate* VM to the *myVmPublic* VM. The response is similar to this example:

   ```
   Tracing route to myVmPublic.vpgub4nqnocezhjgurw44dnxrc.bx.internal.cloudapp.net [10.0.0.4]
   over a maximum of 30 hops:

   1     1 ms     1 ms     1 ms  10.0.0.4

   Trace complete.
   ```

   You can see Azure routes traffic directly from the *myVmPrivate* VM to the *myVmPublic* VM. By default, Azure routes traffic directly between subnets.

4. Close the remote desktop session to the *myVmPrivate* VM.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results : You have now completed this lab.

26 Module 2: Lab 7: Service Endpoints

Virtual network service endpoints enable you to limit network access to some Azure service resources to a virtual network subnet. You can also remove internet access to the resources. Service endpoints provide direct connection from your virtual network to supported Azure services, allowing you to use your virtual network's private address space to access the Azure services. Traffic destined to Azure resources through service endpoints always stays on the Microsoft Azure backbone network. In this tutorial, you learn how to:

- Create a virtual network with one subnet
- Add a subnet and enable a service endpoint
- Create an Azure resource and allow network access to it from only a subnet
- Deploy a virtual machine (VM) to each subnet
- Confirm access to a resource from a subnet
- Confirm access is denied to a resource from a subnet and the internet

26.1 Exercise 1: Restrict network access to PaaS resources with virtual network service endpoints using the Azure portal

26.1.1 Task 1: Create a virtual network

1. Select **+ Create a resource** on the upper left corner of the Azure portal.
2. Select **Networking**, and then select **Virtual network**.
3. Enter, or select, the following information, and then select **Review + create**, then click **Create**:

Setting	Value
Subscription	Select your subscription
Resource group	Select **Create new** and enter *myResourceGroup*.
Name	myVirtualNetwork
Location	Select **East US**

Select the IP Addresses tab and enter the following values:

Setting	Value
Address space	10.0.0.0/16
Subnet Name	Public
Subnet Address range	10.0.0.0/24

Select the Security tab and enter the following values:

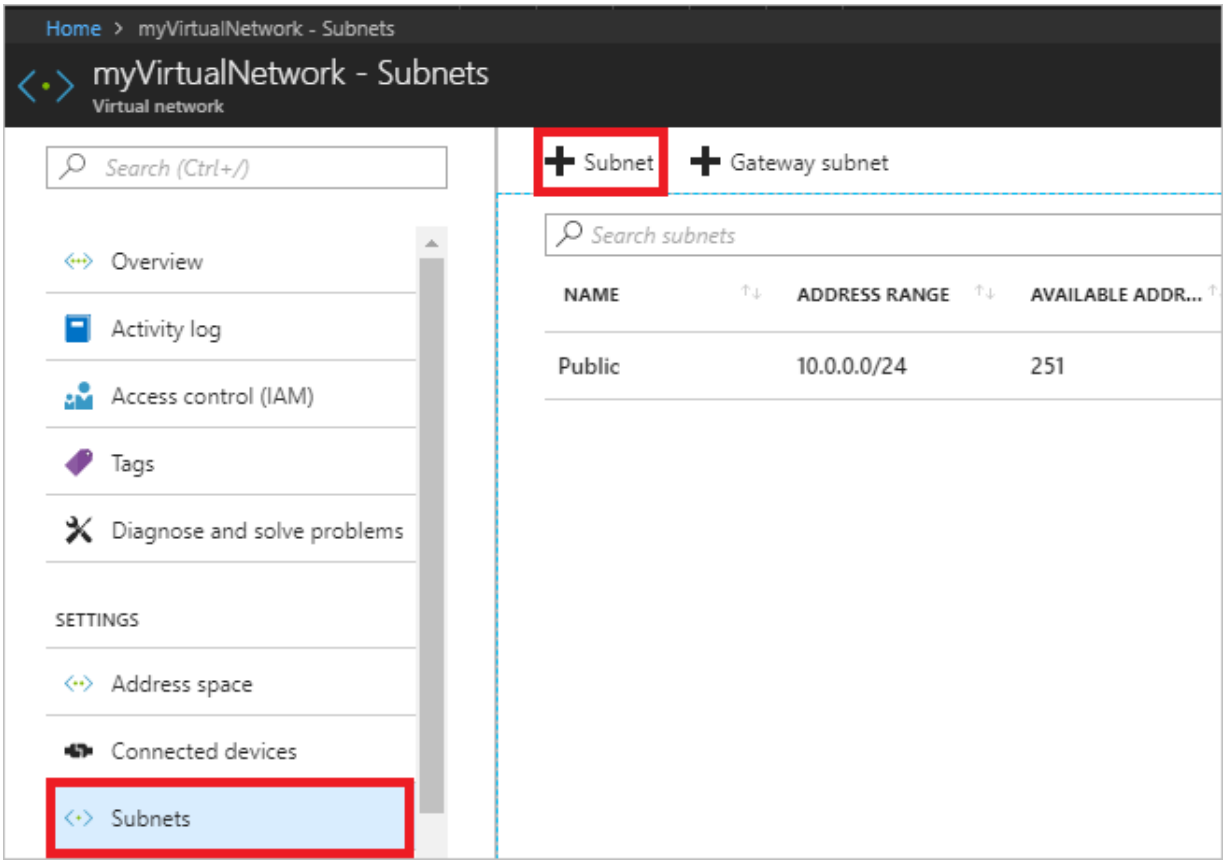
Setting	Value
DDoS protection	Basic
Firewall	Disabled

26.1.2 Task 2: Enable a service endpoint

Service endpoints are enabled per service, per subnet. Create a subnet and enable a service endpoint for the subnet.

1. In the **Search resources, services, and docs** box at the top of the portal, enter *myVirtualNetwork*. When **myVirtualNetwork** appears in the search results, select it.

2. Add a subnet to the virtual network. Under **SETTINGS**, select **Subnets**, and then select + **Subnet**, as shown in the following picture:



3. Under **Add subnet**, select or enter the following information, and then select **OK**:

Setting	Value
Name	Private
Address range	10.0.1.0/24
Service endpoints	Select **Microsoft.Storage** under **Services**

26.1.3 Task 3: Restrict network access for a subnet

By default, all VMs in a subnet can communicate with all resources. You can limit communication to and from all resources in a subnet by creating a network security group and associating it to the subnet.

1. Select + **Create a resource** on the upper left corner of the Azure portal.
2. Select **Networking**, and then select **Network security group**.
3. Under **Create a network security group**, enter, or select, the following information, and then select **Review + create**, then click **Create**:

Setting	Value
Name	myNsgPrivate
Subscription	Select your subscription
Resource group	Select **Use existing** and select *myResourceGroup*.
Location	Select **East US**

4. After the network security group is created, enter *myNsgPrivate* in the **Search resources, services, and docs** box at the top of the portal. When **myNsgPrivate** appears in the search results, select it.
5. Under **SETTINGS**, select **Outbound security rules**.
6. Select + **Add**.

7. Create a rule that allows outbound communication to the Azure Storage service. Enter, or select, the following information, and then select **Add**:

Setting	Value
Source	Select **VirtualNetwork**
Source port ranges	*
Destination	Select **Service Tag**
Destination service tag	Select **Storage**
Destination port ranges	*
Protocol	Any
Action	Allow
Priority	100
Name	Allow-Storage-All

8. Create another outbound security rule that denies communication to the internet. This rule overrides a default rule in all network security groups that allows outbound internet communication. Complete steps 5-7 again, using the following values:

Setting	Value
Source	Select **VirtualNetwork**
Source port ranges	*
Destination	Select **Service Tag**
Destination service tag	Select **Internet**
Destination port ranges	*
Protocol	Any
Action	Deny
Priority	110
Name	Deny-Internet-All

9. Under **SETTINGS**, select **Inbound security rules**.
10. Select **+ Add**.
11. Create an inbound security rule that allows Remote Desktop Protocol (RDP) traffic to the subnet from anywhere. The rule overrides a default security rule that denies all inbound traffic from the internet. Remote desktop connections are allowed to the subnet so that connectivity can be tested in a later step. Under **SETTINGS**, select **Inbound security rules**, select **+Add**, enter the following values, and then select **Add**:

Setting	Value
Source	Any
Source port ranges	*
Destination	Select **VirtualNetwork**
Destination port ranges	3389
Protocol	Any
Action	Allow
Priority	120
Name	Allow-RDP-All

12. Under **SETTINGS**, select **Subnets**.
13. Select **+ Associate**
14. Under **Associate subnet**, select **Virtual network** and then select **myVirtualNetwork** under **Choose a virtual network**.
15. Under **Choose subnet**, select **Private**, and then select **OK**.

26.1.4 Task 4: Restrict network access to a resource

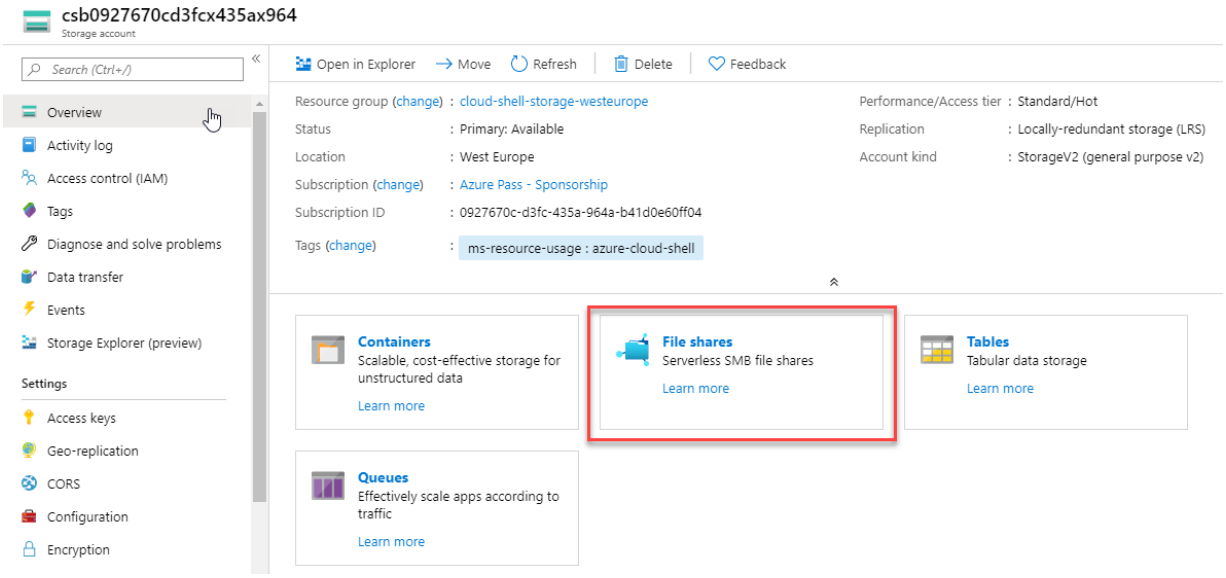
The steps necessary to restrict network access to resources created through Azure services enabled for service endpoints varies across services. See the documentation for individual services for specific steps for each service. The remainder of this tutorial includes steps to restrict network access for an Azure Storage account, as an example.

1. Select + **Create a resource** on the upper left corner of the Azure portal.
2. Select **Storage**, and then select **Storage account - blob, file, table, queue**.
3. Enter, or select, the following information, accept the remaining defaults, and then select **Review + create**, then click **Create**:

Setting	Value
Name	Enter a name that is unique across all Azure locations, between 3-24 characters in length, using only num
Account kind	StorageV2 (general purpose v2)
Location	Select **East US**
Replication	Locally-redundant storage (LRS)
Subscription	Select your subscription
Resource group	Select **Use existing** and select *myResourceGroup*.

26.1.5 Task 5: Create a file share in the storage account

1. After the storage account is created, enter the name of the storage account in the **Search resources, services, and docs** box, at the top of the portal. When the name of your storage account appears in the search results, select it.
2. Select **File shares**, as shown in the following picture:



3. Select + **File share**.
4. Enter *my-file-share* under **Name**, and then select **Create**.
5. Close the **File service** box.

26.1.6 Task 6: Restrict network access to a subnet

By default, storage accounts accept network connections from clients in any network, including the internet. Deny network access from the internet, and all other subnets in all virtual networks, except for the *Private* subnet in the *myVirtualNetwork* virtual network.

1. Under **SETTINGS** for the storage account, select **Firewalls and virtual networks**.
2. Select **Selected networks**.
3. Select **+Add existing virtual network**.

4. Under **Add networks**, select the following values, and then select **Add**:

Setting	Value
Subscription	Select your subscription.
Virtual networks	Select **myVirtualNetwork**, under **Virtual networks**
Subnets	Select **Private**, under **Subnets**

5. Select **Save**.
6. Close the **Firewalls and virtual networks** box.
7. Under **SETTINGS** for the storage account, select **Access keys**.
8. Note the **Key** value, as you'll have to manually enter it in a later step when mapping the file share to a drive letter in a VM.

26.1.7 Task 7: Create virtual machines

To test network access to a storage account, deploy a VM to each subnet.

1. Select **+ Create a resource** found on the upper left corner of the Azure portal.
2. Select **Compute**, and then select **Virtual Machine**.
3. Enter, or select, the following information and then select **OK**:

Setting	Value
Name	myVmPublic
User name	Enter a user name of your choosing.
Password	Pa55w.rd1234
Subscription	Select your subscription.
Resource group	Select **Use existing** and select **myResourceGroup**.
Location	Select **East US**.
Image	Select **Windows Server 2019 Datacenter**.

4. Select a size for the virtual machine and then select **Select**.
5. On the **Networking** tab select **myVirtualNetwork**. Then select **Subnet**, and select the **Public** subnet.
6. Under **Network Security Group**, select **Basic** and allow port 3389.
7. Click **Review + create**.
8. On the **Summary** page, select **Create** to start the virtual machine deployment. The VM takes a few minutes to deploy, but you can continue to the next step while the VM is creating.
9. Complete steps 1-8 again, but in step 3, name the virtual machine *myVmPrivate* and in step 5, select the **Private** subnet.

The VM takes a few minutes to deploy. Do not continue to the next step until it finishes creating and its settings open in the portal.

26.1.8 Task 8: Confirm access to storage account

1. Once the myVmPrivate VM finishes creating, open the blade for the VM by selecting **Go to resource**. Select the **Connect** button, then select RDP.
2. After selecting the **Connect** button and RDP, select the Download RDP File button. A Remote Desktop Protocol (.rdp) file is created and downloaded to your computer.
3. Open the downloaded rdp file. If prompted, select **Connect**. Enter the user name and password you specified when creating the VM. You may need to select **More choices**, then **Use a different account**, to specify the credentials you entered when you created the VM.
4. Select **OK**.

5. You may receive a certificate warning during the sign-in process. If you receive the warning, select **Yes** or **Continue** to proceed with the connection.

6. On the *myVmPrivate* VM, map the Azure file share to drive Z using PowerShell. Before running the commands that follow, replace `<storage-account-key>` and `<storage-account-name>` with values you supplied and retrieved in the **Create a storage account** task.

```
$acctKey = ConvertTo-SecureString -String "<storage-account-key>" -AsPlainText -Force
$credential = New-Object System.Management.Automation.PSCredential -ArgumentList "Azure\<storage-a
New-PSDrive -Name Z -PSProvider FileSystem -Root "\\<storage-account-name>.file.core.windows.net\m
```

   The Azure file share successfully mapped to the Z drive.

7. Confirm that the VM has no outbound connectivity to the internet from a command prompt:

```
ping bing.com
```

   You receive no replies because the network security group associated to the *Private* subnet does not allow outbound access to the internet.

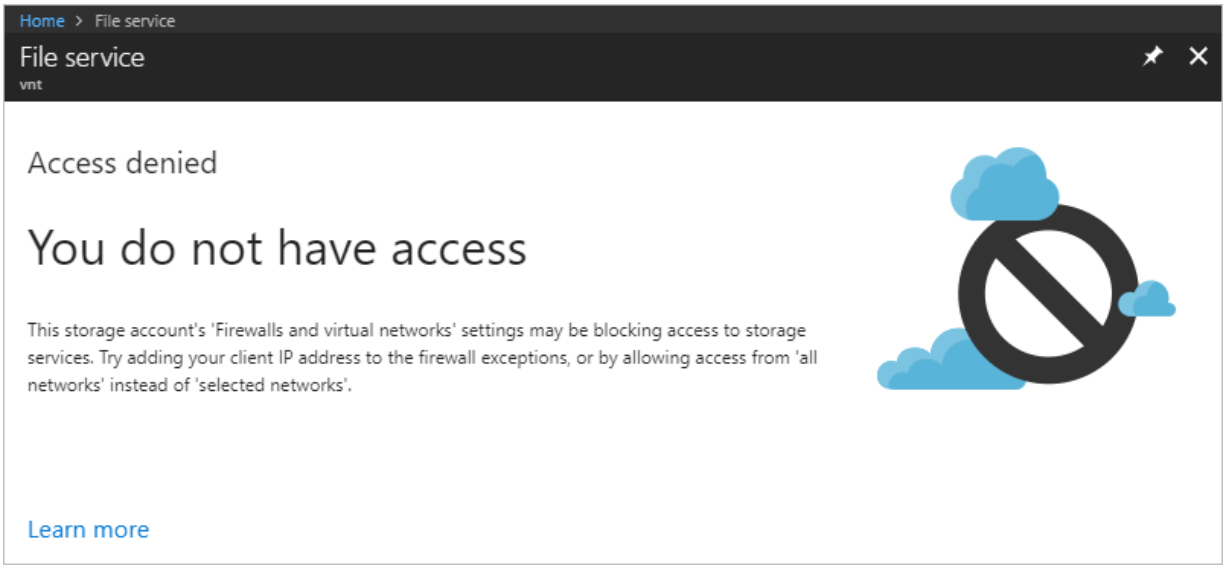
8. Close the remote desktop session to the *myVmPrivate* VM.

26.1.9 Task 9: Confirm access is denied to storage account

1. Enter *myVmPublic* In the **Search resources, services, and docs** box at the top of the portal.
2. When **myVmPublic** appears in the search results, select it.
3. Complete steps 1-6 in the Confirm access to storage account task for the *myVmPublic* VM.

   After a short wait, you receive a `New-PSDrive : Access is denied` error. Access is denied because the *myVmPublic* VM is deployed in the *Public* subnet. The *Public* subnet does not have a service endpoint enabled for Azure Storage. The storage account only allows network access from the *Private* subnet, not the *Public* subnet.

4. Close the remote desktop session to the *myVmPublic* VM.
5. From your computer, browse to the Azure portal.
6. Enter the name of the storage account you created in the **Search resources, services, and docs** box. When the name of your storage account appears in the search results, select it.
7. Select **File shares** then select *my-file-share*.
8. You receive the error shown in the following screenshot:



   Access is denied, because your computer is not in the *Private* subnet of the *MyVirtualNetwork* virtual network.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results : You have now completed this lab.

27 Module 2: Lab 8 - VNet Peering

Scenario

You can connect virtual networks to each other with virtual network peering. These virtual networks can be in the same region or different regions (also known as Global VNet peering). Once virtual networks are peered, resources in both virtual networks are able to communicate with each other, with the same latency and bandwidth as if the resources were in the same virtual network. In this tutorial, you learn how to:

- Create two virtual networks
- Connect two virtual networks with a virtual network peering
- Deploy a virtual machine (VM) into each virtual network
- Communicate between VMs

27.0.1 Exercise 1: Create Virtual Networks and implement Peering.

27.0.2 Task 1: Create virtual networks

1. Select **+ Create a resource** on the upper, left corner of the Azure portal.
2. Select **Networking**, and then select **Virtual network**.
3. Enter, or select, the following information, accept the defaults for the remaining settings, and then select Click **Review + create**, then click **Create**:

Setting	Value
Name	myVirtualNetwork1
Subscription	Select your subscription.
Resource group	Select **Create new** and enter *myResourceGroup*.
Location	Select **East US**.

Select the IP Addresses tab and enter the following values:

Setting	Value
Address space	10.0.0.0/16
Subnet Name	Subnet1
Subnet Address range	10.0.0.0/24

4. Complete steps 1-3 again, with the following changes:

Setting	Value
Name	myVirtualNetwork2
Address space	10.1.0.0/16
Resource group	Select **Use existing** and then select **myResourceGroup**.
Subnet Address range	10.1.0.0/24

27.0.3 Task 2: Peer virtual networks

1. In the Search box at the top of the Azure portal, begin typing *MyVirtualNetwork1*. When **myVirtualNetwork1** appears in the search results, select it.
2. Select **Peerings**, under **SETTINGS**, and then select **+ Add**.
3. Enter, or select, the following information, accept the defaults for the remaining settings, and then select

OK.

Setting	Value
Name	myVirtualNetwork1-myVirtualNetwork2
Subscription	Select your subscription.
Virtual network	myVirtualNetwork2 - To select the *myVirtualNetwork2* virtual network, select **Virtual network**, then s
Name	myVirtualNetwork2-myVirtualNetwork1

The **PEERING STATUS** is *Initiated*.

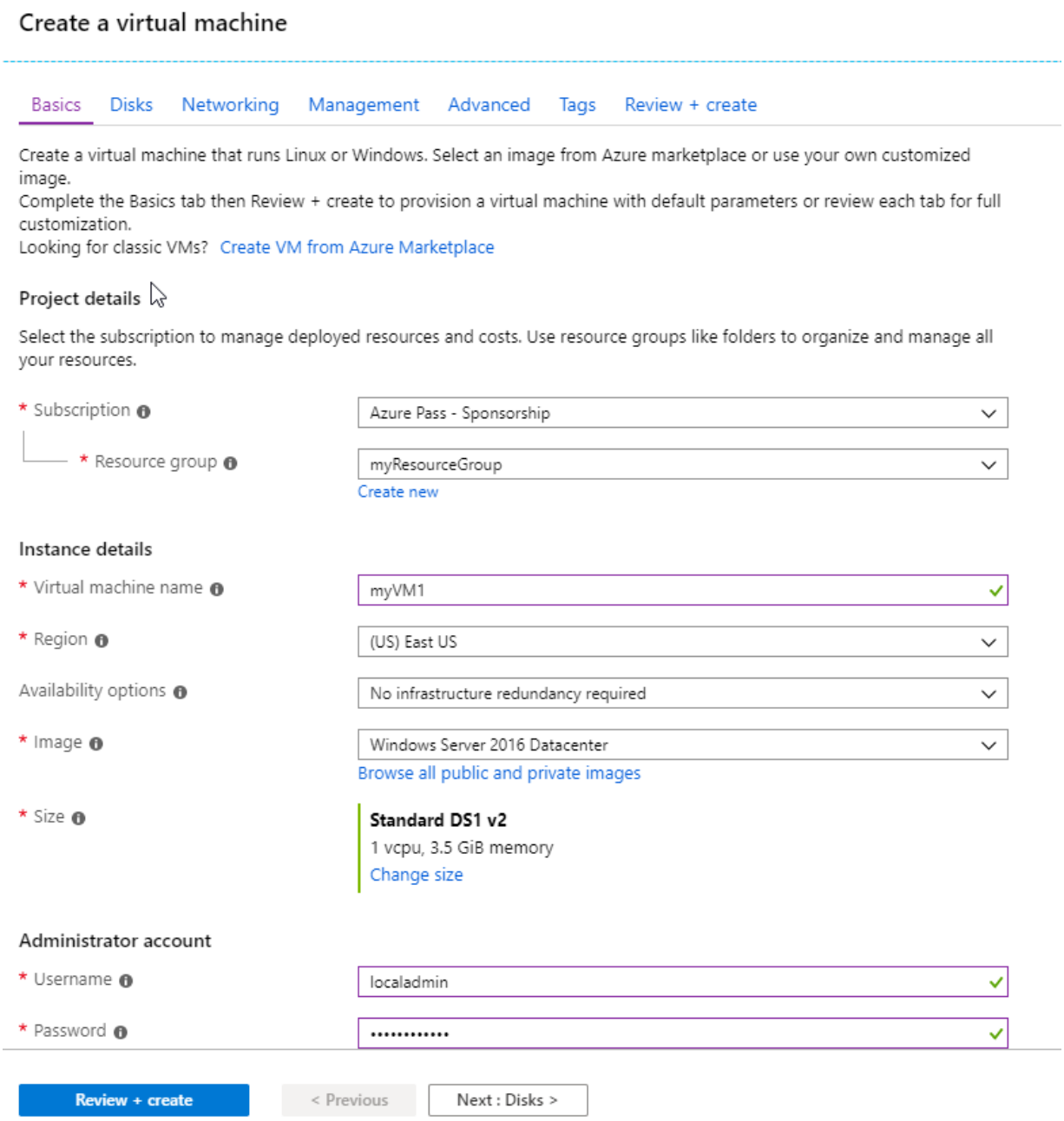
If you don't see the status, refresh your browser.

The **PEERING STATUS** is *Connected*. Azure also changed the peering status for the *myVirtualNetwork2-myVirtualNetwork1* peering from *Initiated* to *Connected*. Virtual network peering is not fully established until the peering status for both virtual networks is *Connected*.

27.0.4 Task 3: Create virtual machines

1. Select **+ Create a resource** on the upper, left corner of the Azure portal.
2. Select **Compute** > **Virtual Machine**. When creating the VM, select **Windows Server 2016 Datacenter** as the operating system.
3. Enter, or select, the following information for **Basics**, accept the defaults for the remaining settings, and then select **Create**:

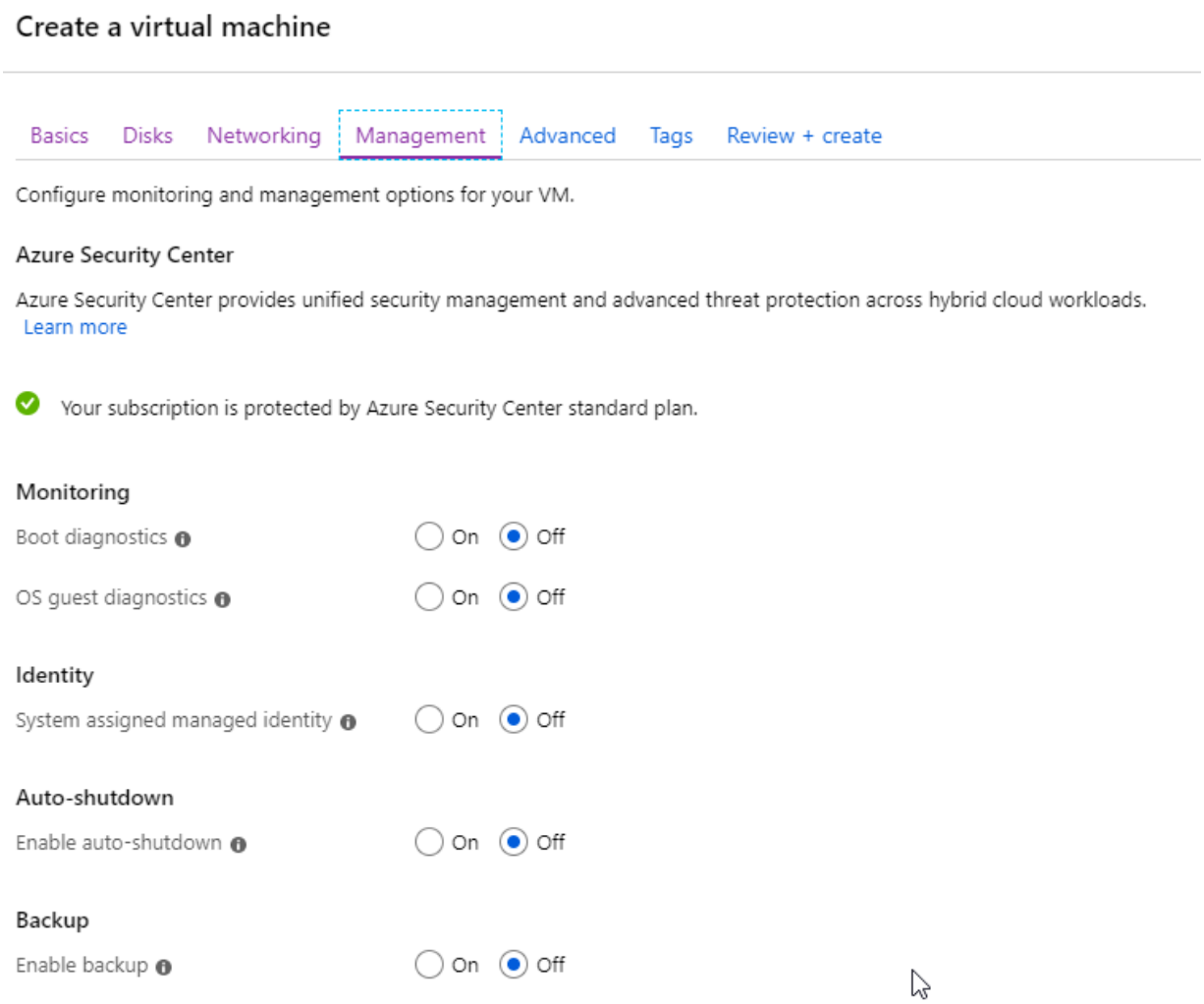
Setting	Value
Resource group	Select **myResourceGroup**.
Name	myVM1
Region	East US
User name	localadmin
Password	Pa55w.rd1234



4. Select the Networking Tab:

Setting	Value
Virtual network	myVirtualNetwork1 - If it's not already selected, select **Virtual network** and then select **myVirtu**
Subnet	Subnet1 - If it's not already selected, select **Subnet** and then select **Subnet1** under **Choose subn**
Public inbound ports	Select **Allow selected ports**
Select inbound ports	**RDP**

5. Select the Management Tab and turn all the radio buttons to **Off**.



6. Select **Review + create** and click **Create**.
7. Complete the above steps again, with the following changes (The VMs take a few minutes to create. Do not continue with the remaining steps until both VMs are created.):

Setting	Value
Name	myVM2
Virtual network	myVirtualNetwork2

27.0.5 Task 4: Communicate between VMs

1. In the *Search* box at the top of the portal, begin typing *myVM1*. When **myVM1** appears in the search results, select it.
2. Create a remote desktop connection to the *myVm1* VM by selecting **Connect**, then selecting **RDP**, then selecting the **Download RDP File** button.
3. To connect to the VM, open the downloaded RDP file. If prompted, select **Connect**.
4. Enter the user name and password you specified when creating the VM (you may need to select **More choices**, then **Use a different account**, to specify the credentials you entered when you created the VM), then select **OK**.
5. You may receive a certificate warning during the sign-in process. Select **Yes** to proceed with the connection.
6. In a later step, ping is used to communicate with the *myVm2* VM from the *myVm1* VM. Ping uses the Internet Control Message Protocol (ICMP), which is denied through the Windows Firewall, by default.

On the *myVm1* VM, enable ICMP through the Windows firewall, so that you can ping this VM from *myVm2* in a later step, using PowerShell:

```
New-NetFirewallRule -DisplayName "Allow ICMPv4-In" -Protocol ICMPv4
```

Though ping is used to communicate between VMs in this tutorial, allowing ICMP through the Windows Firewall for production deployments is not recommended.

7. To connect to the *myVm2* VM, enter the following command from a command prompt on the *myVm1* VM: *If prompted, enter the credentials. Being able to connect this verifies you can use the peering connection to myVM2 using RDP on the internal network.*

   ```
   mstsc /v:10.1.0.4
   ```

8. Since you enabled ping on *myVm1*, you can now ping it by IP address: *This verifies that the established peer is functioning as expected.*

   ```
   ping 10.0.0.4
   ```

9. Disconnect your RDP sessions to both *myVM1* and *myVM2*.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results : You have now completed this lab.

28 Lab 9: Azure DNS

Scenario

In this module, you will learn about DNS basics and specifically implementing Azure DNS. In the DNS Basics lesson you will review DNS domains, zones, record types, and resolution methods. In the Azure DNS lesson, we will cover delegation, metrics, alerts, and DNS hosting schemes.

Objectives

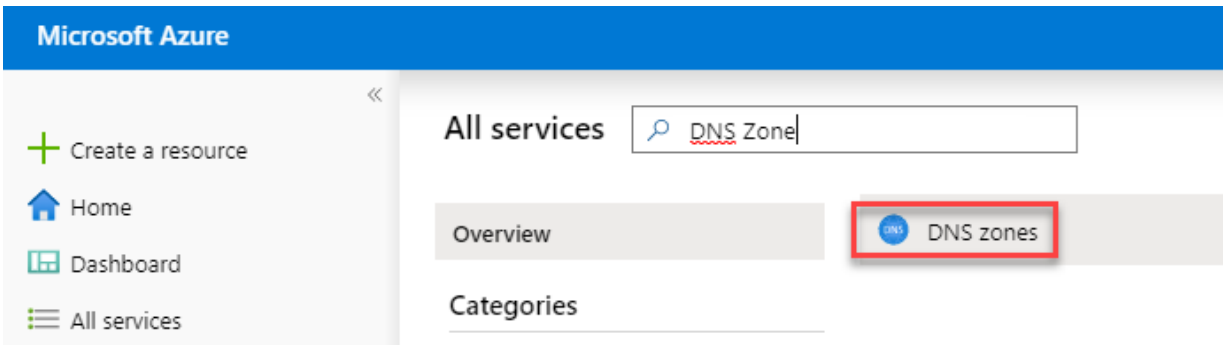
Lessons include:

- Azure DNS Basics
- Implementing Azure DNS

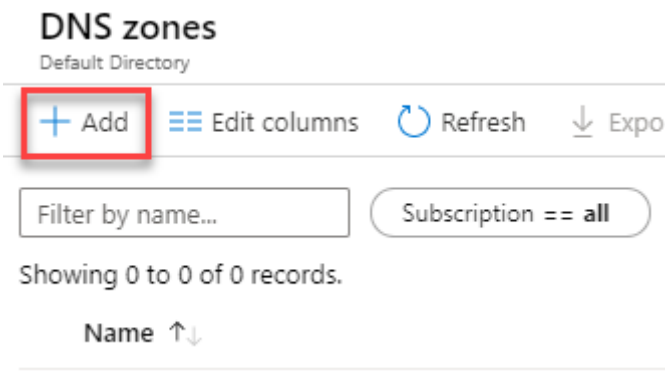
28.1 Exercise 1: DNS Zones

28.1.1 Task 1: Create a DNS zone

1. Sign in to the Azure Portal.
2. On the Hub menu, click **All services** and search for and select **DNS zones**.

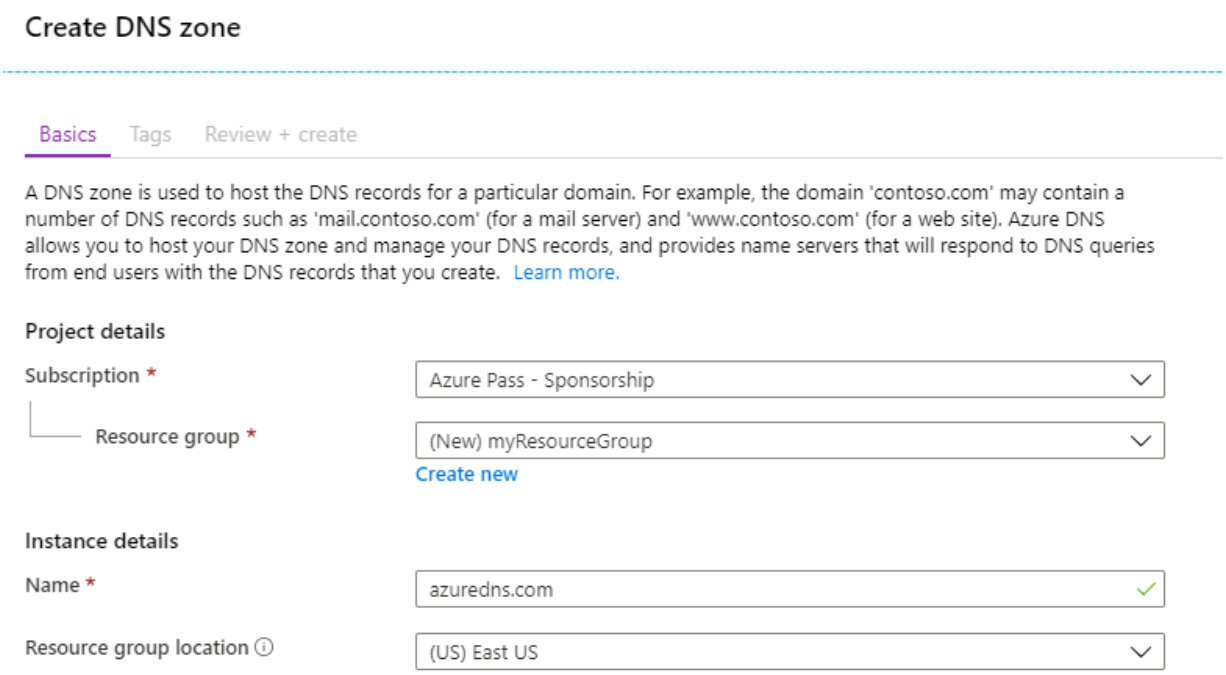


3. Click + **Add**.



4. On the **Create DNS zone** blade enter the following values, then click **Review + create** and then click **Create**:

Setting	Value	Details
Subscription	***Your subscription***	Select a subscription to create the DNS zone in.
Resource group	Create new: ***myResourceGroup***	Create a resource group. The resource group name must be unique
Name	**_see details_**	The name of the DNS zone (yours must be unique)
Location	East US	



28.2 Exercise 2: Manage DNS records and record sets by using the Azure portal

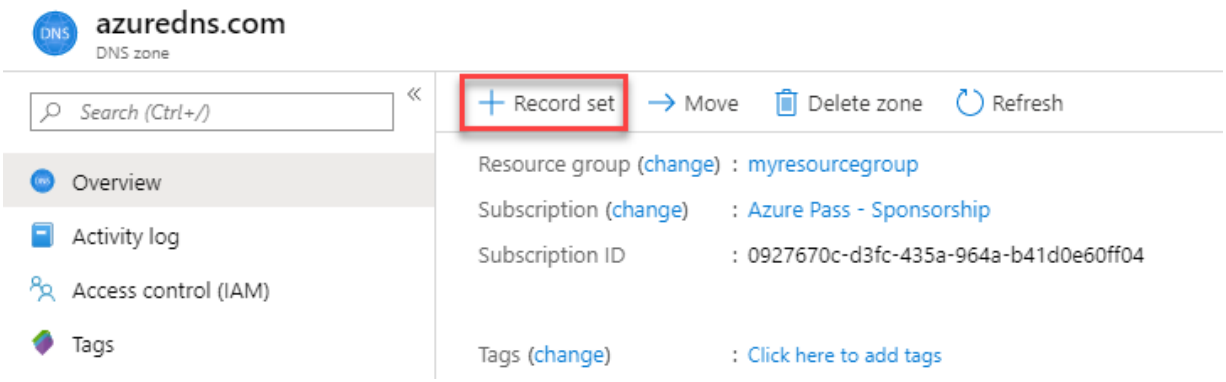
This exercise shows you how to manage record sets and records for your DNS zone by using the Azure portal.

28.2.1 Task 1: Add a new record to a record set

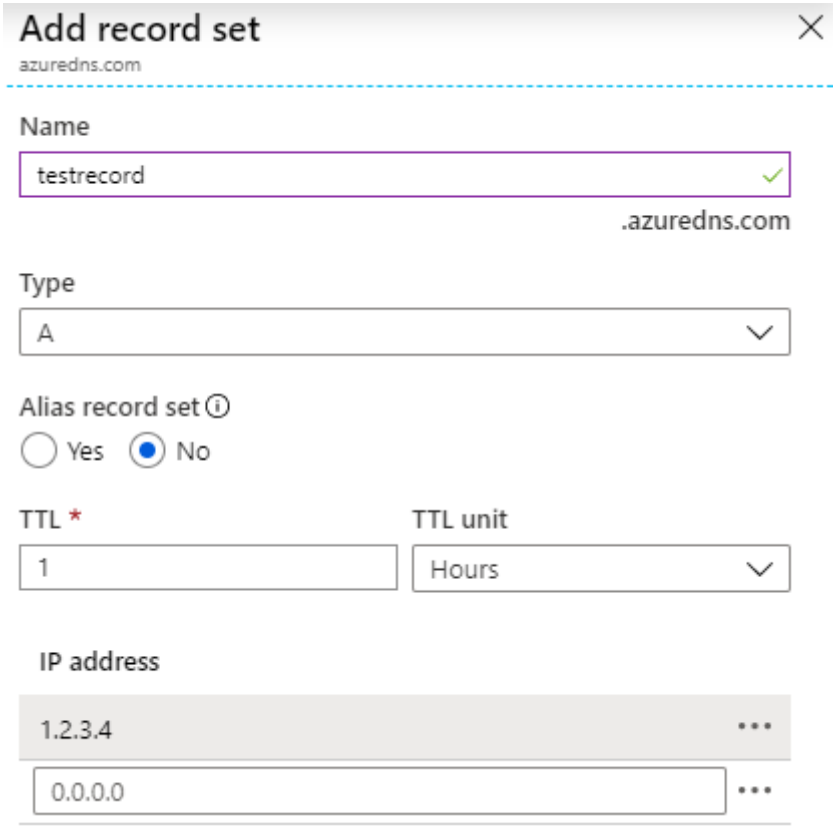
1. In the Azure Portal, navigate to **All resources** and select your DNS zone you created in the previous task.

   Note: Each DNS zone is its own resource, and information such as number of record-sets and name servers are viewable from this view.

2. Click **+ Record Set**.

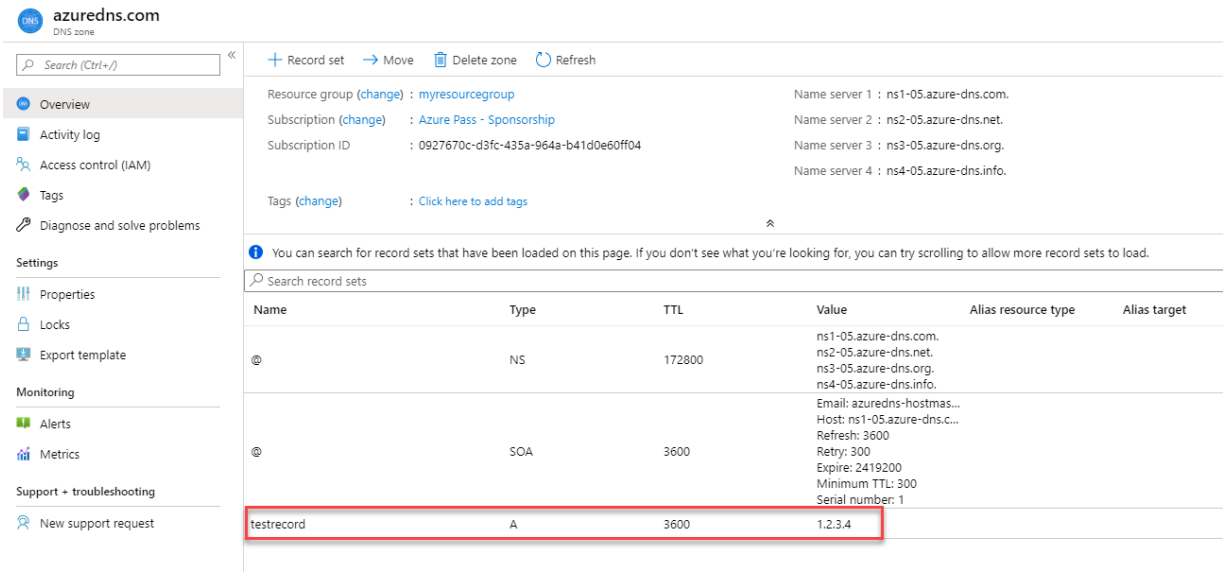


3. Enter **testrecord** for the name and **1.2.3.4** as the IP address and click **OK**.

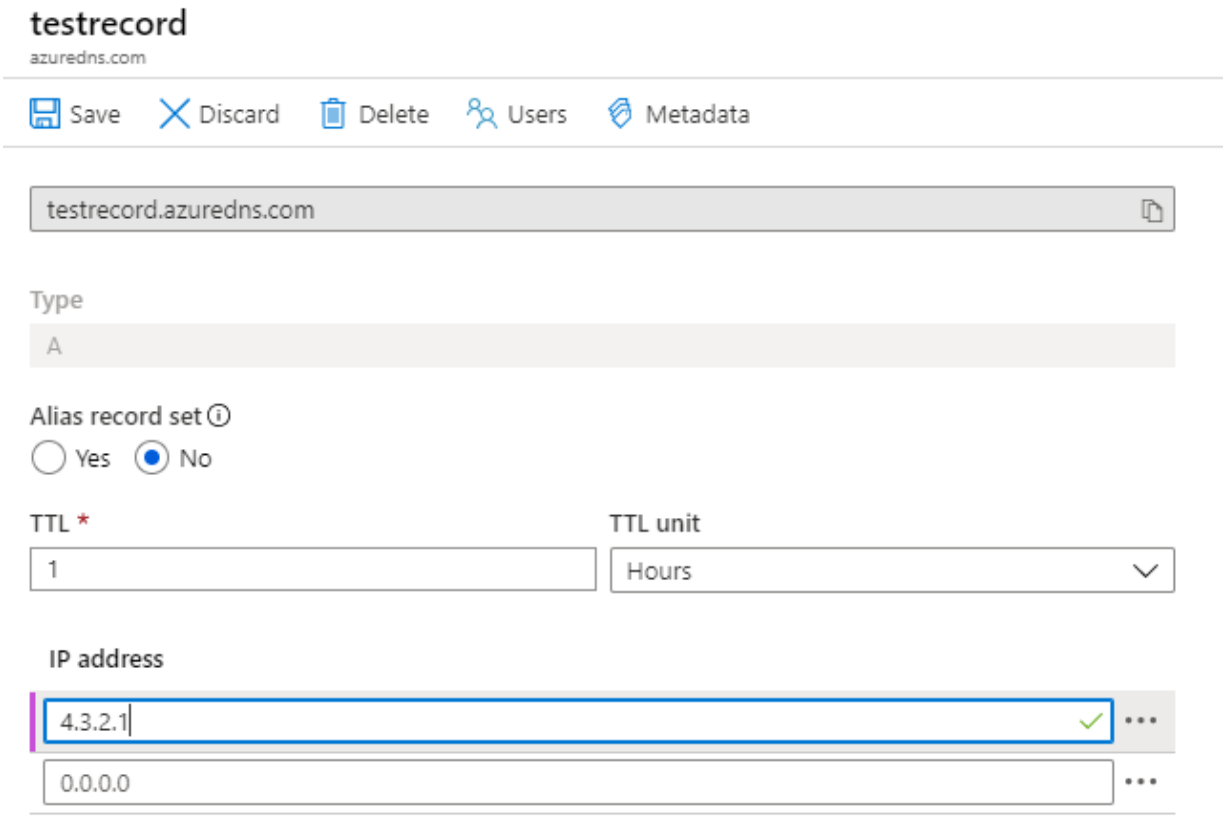


28.2.2 Task 2: Update a record

1. In the Overview blade for your DNS zone, select the testrecord you created.



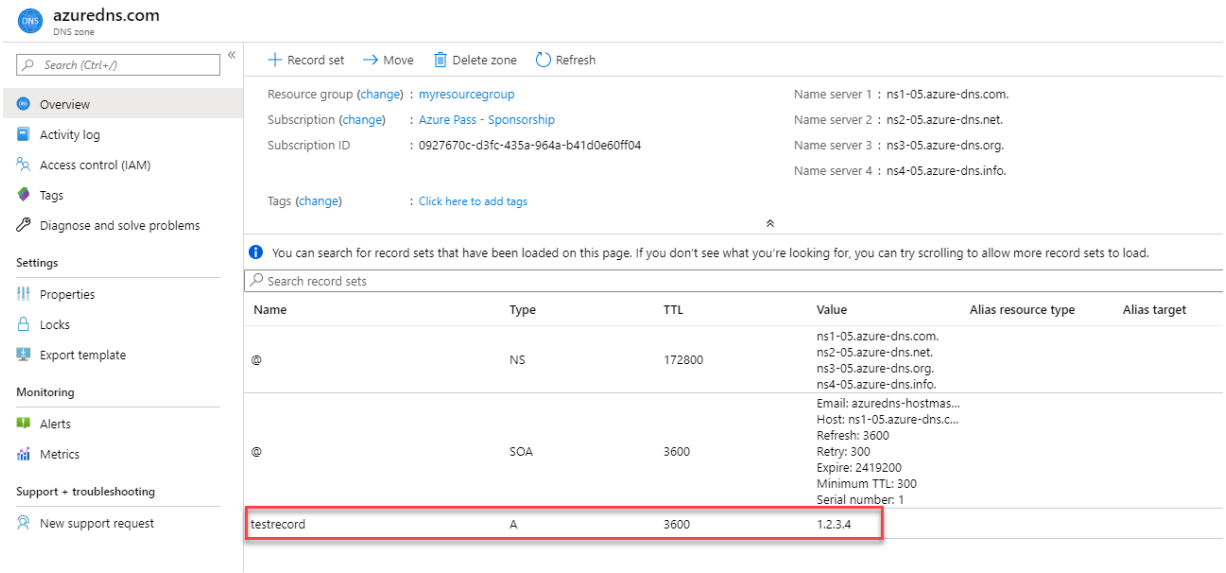
2. Under IP Address add the test address of **4.3.2.1** and click **Save**.



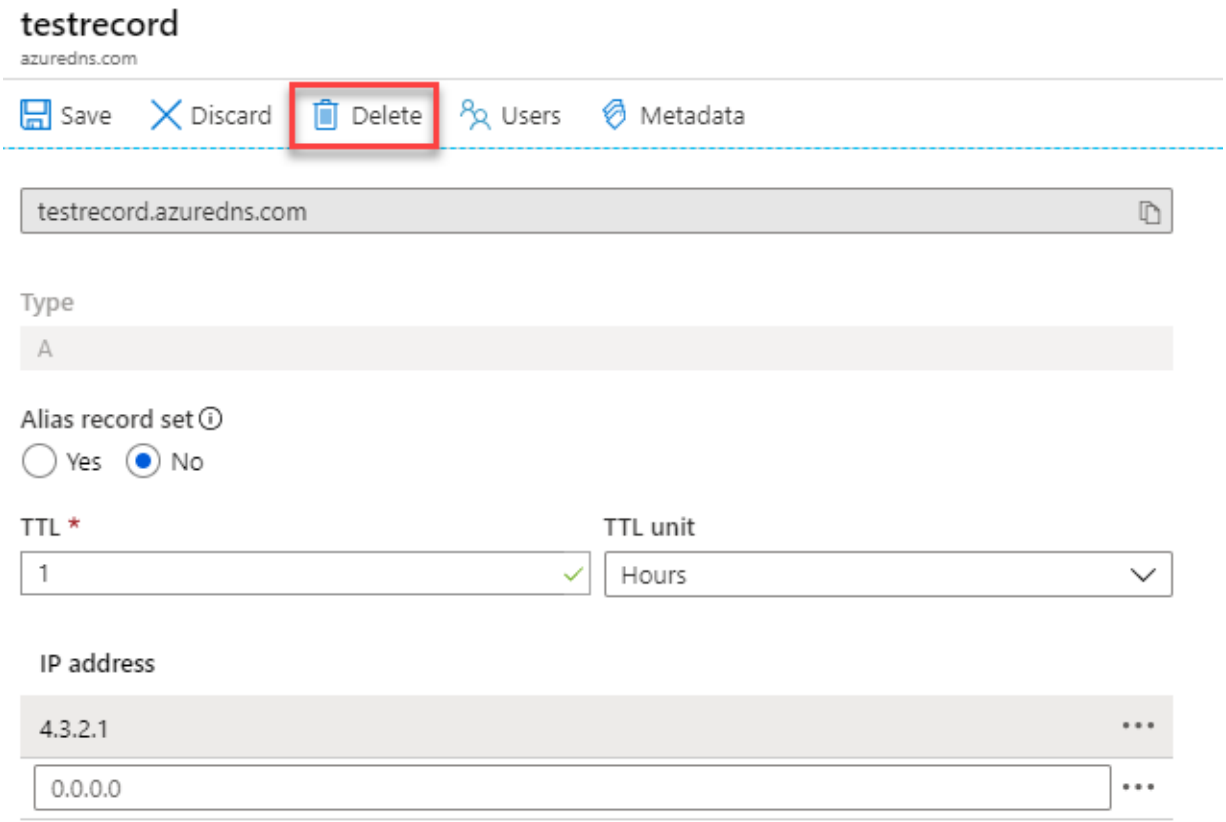
28.2.3 Task 3: Remove a record from a record set

You can use the Azure portal to remove records from a record set. Note that removing the last record from a record set does not delete the record set.

1. In the Overview pane for your DNS zone, select the testrecord you created.



2. Select **Delete** and click **Yes** when prompted.



Work with NS and SOA records

NS and SOA records that are automatically created are managed differently from other record types.

Modify SOA records

You cannot add or remove records from the automatically created SOA record set at the zone apex (name = "@"). However, you can modify any of the parameters within the SOA record (except "Host") and the record set TTL.

Modify NS records at the zone apex

The NS record set at the zone apex is automatically created with each DNS zone. It contains the names of the Azure DNS name servers assigned to the zone.

You can add additional name servers to this NS record set, to support co-hosting domains with more than one

DNS provider. You can also modify the TTL and metadata for this record set. However, you cannot remove or modify the pre-populated Azure DNS name servers.

Note that this applies only to the NS record set at the zone apex. Other NS record sets in your zone (as used to delegate child zones) can be modified without constraint.

Delete SOA or NS record sets

You cannot delete the SOA and NS record sets at the zone apex (name = "@") that are created automatically when the zone is created. They are deleted automatically when you delete the zone.

You are then prompted to confirm you are wanting to delete the DNS zone. Deleting a DNS zone also deletes all records that are contained in the zone.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

---

29 Module 2: Lab 10 - Load Balancer

Scenario

In this module, you will learn about three ways to distribute network traffic: Azure Load Balancer, Azure Traffic Manager, and Azure Application Gateway. The Azure Load Balancer delivers high availability and network performance to your applications. The Azure Traffic Manager allows you to control the distribution of user traffic to your service endpoints. The Azure Application Gateway is a web traffic load balancer that enables you to manage traffic to your web applications.

Lessons include:

- Azure Load Balancer
- Azure Traffic Manager
- Azure Application Gateway

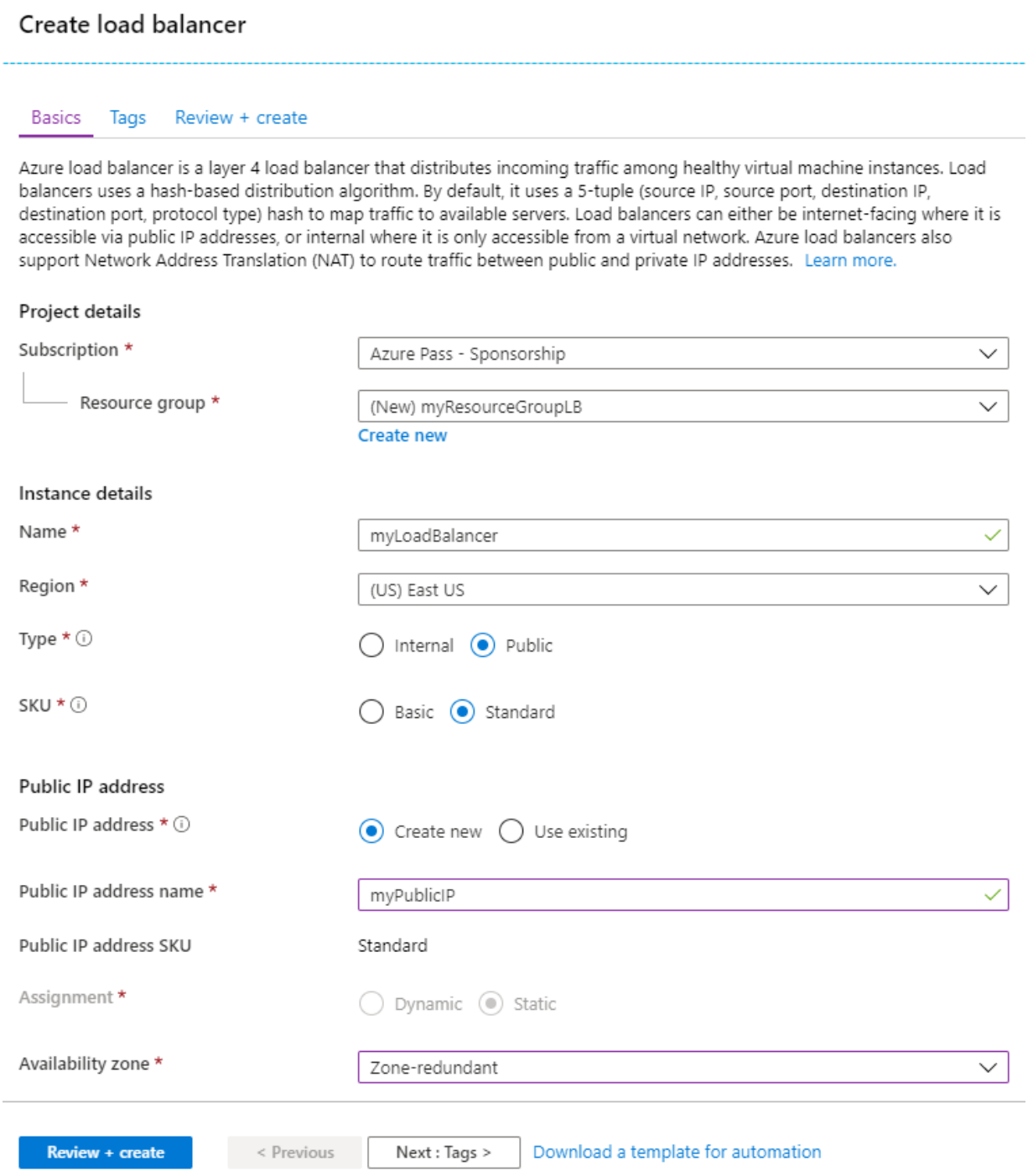
29.1 Exercise 1: Distributing Network Traffic using a Standard Load Balancer

In this section, you create a public load balancer that helps load balance virtual machines. Standard Load Balancer only supports a Standard Public IP address. When you create a Standard Load Balancer, and you must also create a new Standard Public IP address that is configured as the frontend (named as *LoadBalancerFrontend* by default) for the Standard Load Balancer.

29.1.1 Task 1: Create a public load balancer

1. On the top left-hand side of the screen, click **Create a resource** > **Networking** > **Load Balancer**.
2. In the **Create load balancer** page, enter or select the following information, accept the defaults for the remaining settings, and then select **Review + create**:

Setting	Value
Subscription	Select your subscription.
Resource group	Select **Create new**, and then type myResourceGroupLB
Name	*myLoadBalancer*
Region	Select **East US**.
Type	Public
SKU	Standard
Public IP address	Select **Create new** and type *myPublicIP* in the name box.
Availability zone	**Zone-redundant**



3. On the Validation screen click **Create**.

29.1.2 Task 2: Create a virtual network

1. On the top left-hand side of the screen click **+ Create a resource** > **Networking** > **Virtual network** and enter these values for the virtual network:

   - **myVnet** - for the name of the virtual network.
   - **myResourceGroupLB** - for the name of the existing resource group

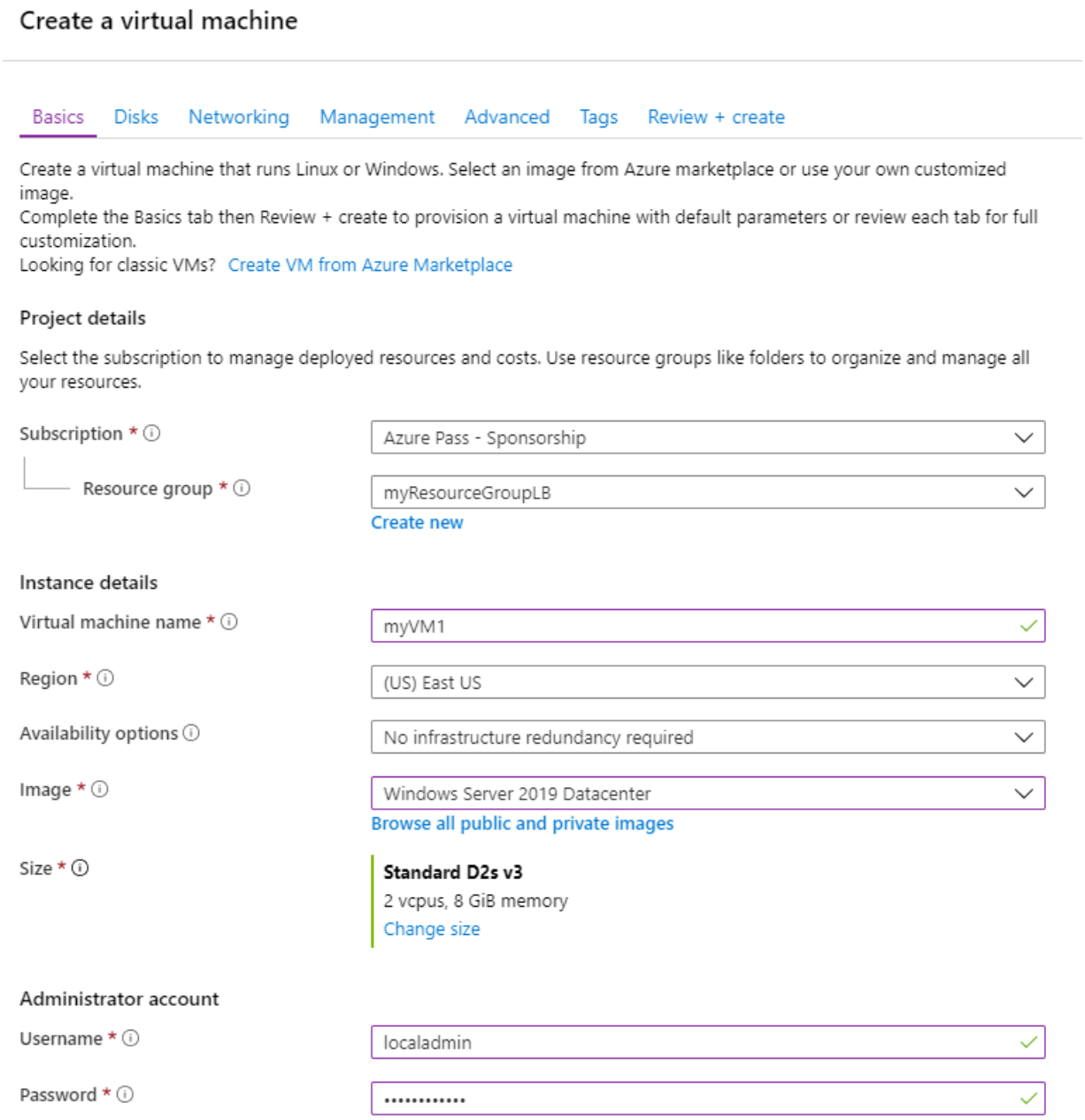
   Select the IP Addresses tab and enter the following values:

   - **10.0.0.0/16** - for the Address space
   - **myBackendSubnet** - for the subnet name.
   - **10.0.0.0/24** - for the Subnet Address range

2. Click **Review + create**, then click **Create** to create the virtual network.

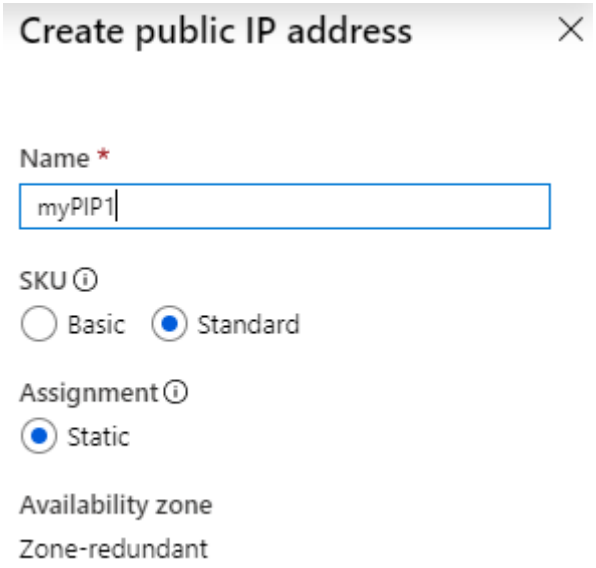
29.1.3 Task 3: Create virtual machines

1. On the top left-hand side of the screen, click **Create a resource** > **Compute** > **Virtual Machine** and enter these values for the virtual machine:
   - **myResourceGroupLB** - for **Resource group**, select *myResourceGroupLB* from the drop down menu.
   - **myVM1** - for the name of the virtual machine.
   - **Image** - Windows Server 2019 Datacenter.
   - **localadmin** - for the **Username**
   - **Pa55w.rd1234** - for the **Password**
   - **HTTP (80) & RDP (3389)** - for the inbound port rules.



2. Click the Networking Tab and under Public IP click **Create new**. Name the IP Address **myPIP1** and click the **Standard SKU** then click **OK**.

   Note: If you do not select the Standard SKU here you will have problems later in the lab.



3. Select the **Management** Tab and ensure all radio buttons are **No** or **Off**.
4. Click **Review + create** then click **Create**.
5. Repeat the steps above to create a second VM, called ***myVM2*** using ***myPIP2*** for the new Public IP address.

29.1.4 Task 4: Install IIS

1. Click **All resources** in the left-hand menu, and then from the resources list click **myVM1** that is located in the *myResourceGroupLB* resource group.
2. On the **Overview** page, click **Connect** to RDP into the VM.
3. Log into the VM with username *localadmin*.
4. Open PowerShell and run the following command to install IIS.

   ```
   Install-WindowsFeature Web-Server
   ```

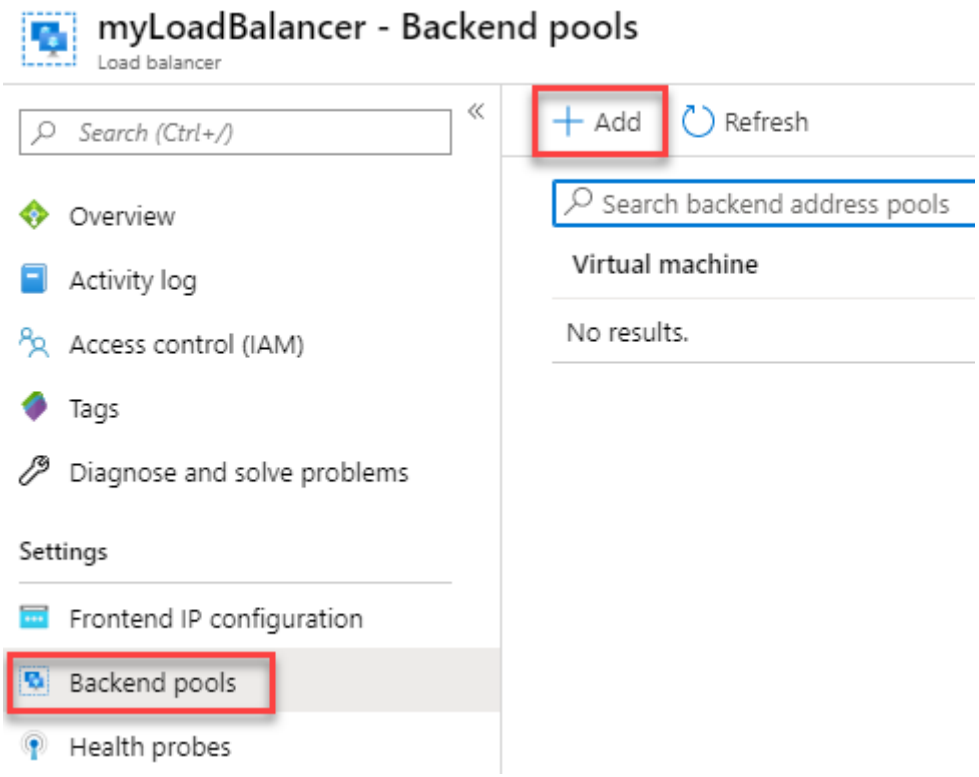
5. Repeat steps 1 to 4 for the virtual machine *myVM2*.

29.1.5 Task 5: Create load balancer resources

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

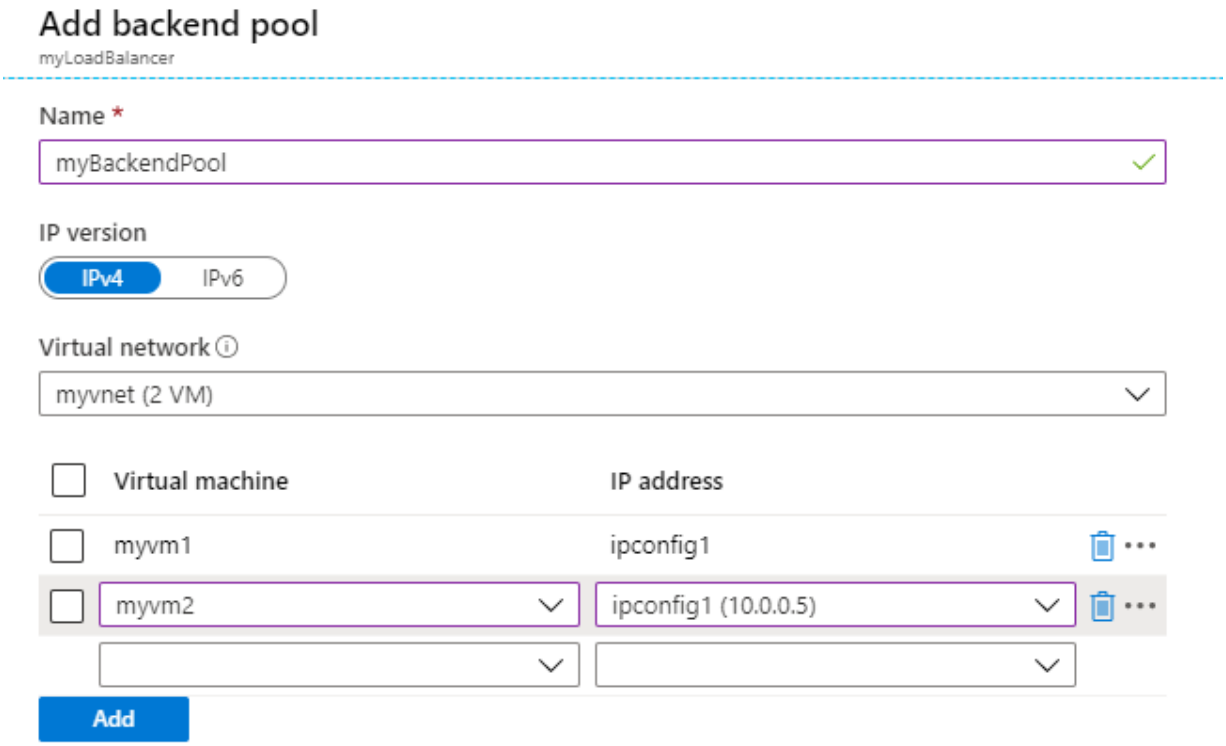
To distribute traffic to the VMs, a backend address pool contains the IP addresses of the virtual (NICs) connected to the load balancer. Create the backend address pool *myBackendPool* to include *VM1* and *VM2*.

1. Click **All resources** in the left-hand menu, and then click **myLoadBalancer** from the resources list.
2. Under **Settings**, click **Backend pools**, then click **Add**.

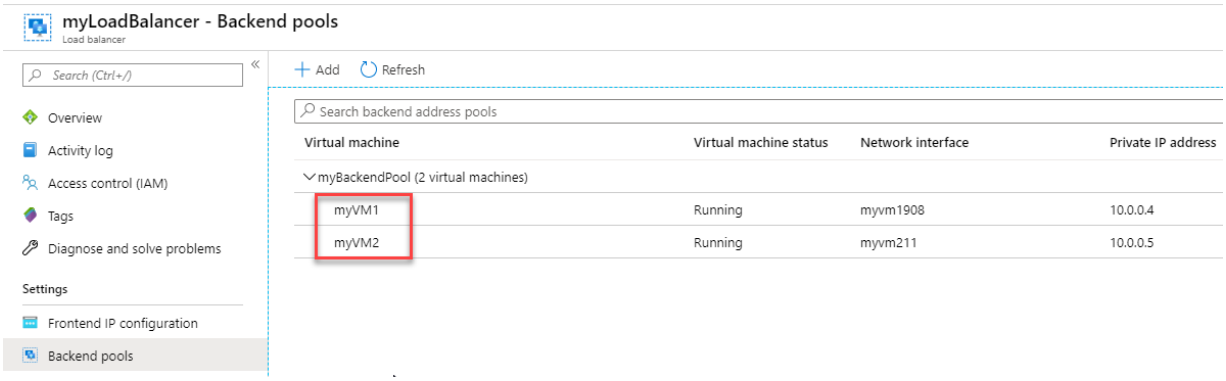


3. On the **Add a backend pool** page, do the following:

- For name, type *myBackendPool*, as the name for your backend pool.
- For **Virtual network**, select *myVNet*.
- Add *myVM1* and *my VM2* under **Virtual Machine** along with their corresponding IP addresses, and then select **Add**.



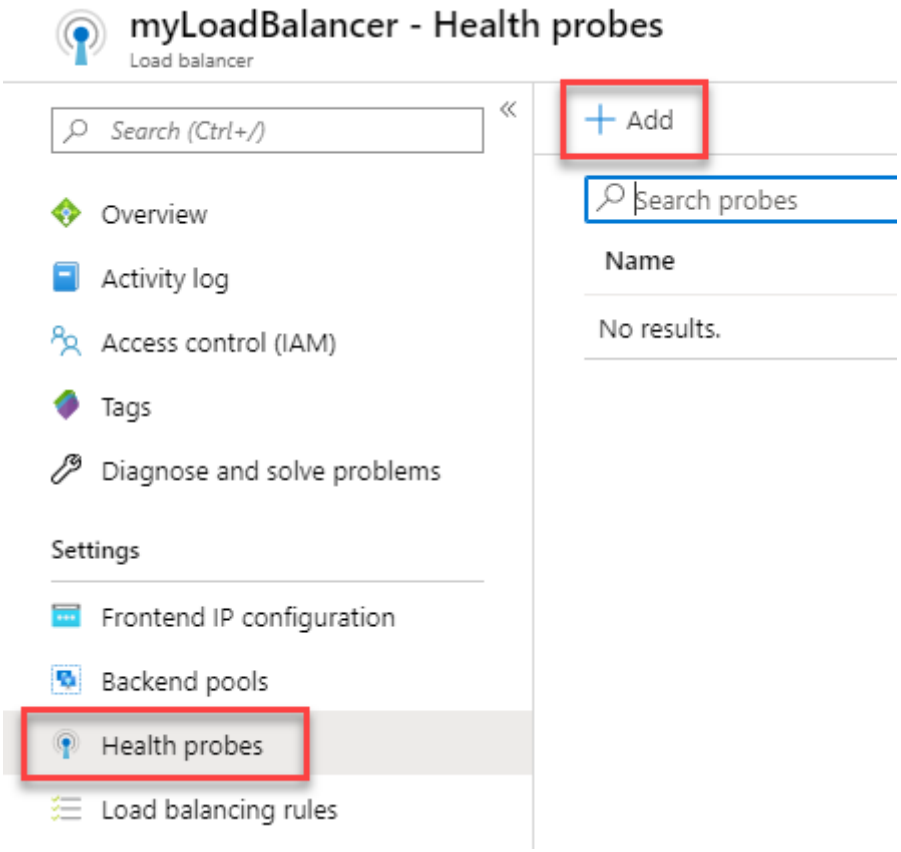
3. Check to make sure your load balancer backend pool setting displays both the VMs **VM1** and **VM2**.



29.1.6 Task 6: Create a health probe

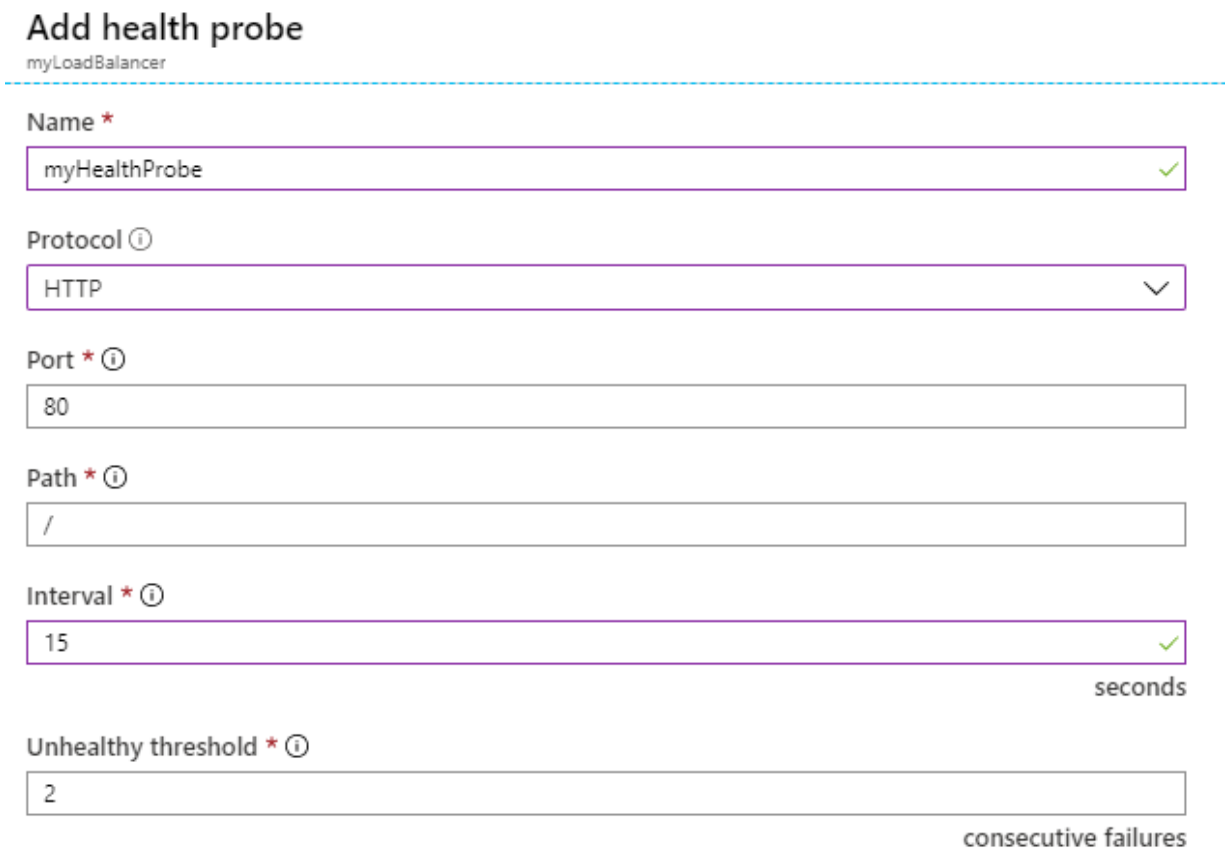
To allow the load balancer to monitor the status of your app, you use a health probe. The health probe dynamically adds or removes VMs from the load balancer rotation based on their response to health checks. Create a health probe *myHealthProbe* to monitor the health of the VMs.

1. On the Load Balancer blade, under **Settings**, click **Health probes**, then click **Add**.



2. Use these values to create the health probe:

   - *myHealthProbe* - for the name of the health probe.
   - **HTTP** - for the protocol type.
   - *80* - for the port number.
   - */* - for the URI path.
   - *15* - for number of **Interval** in seconds between probe attempts.
   - *2* - for number of **Unhealthy threshold** or consecutive probe failures that must occur before a VM is considered unhealthy.

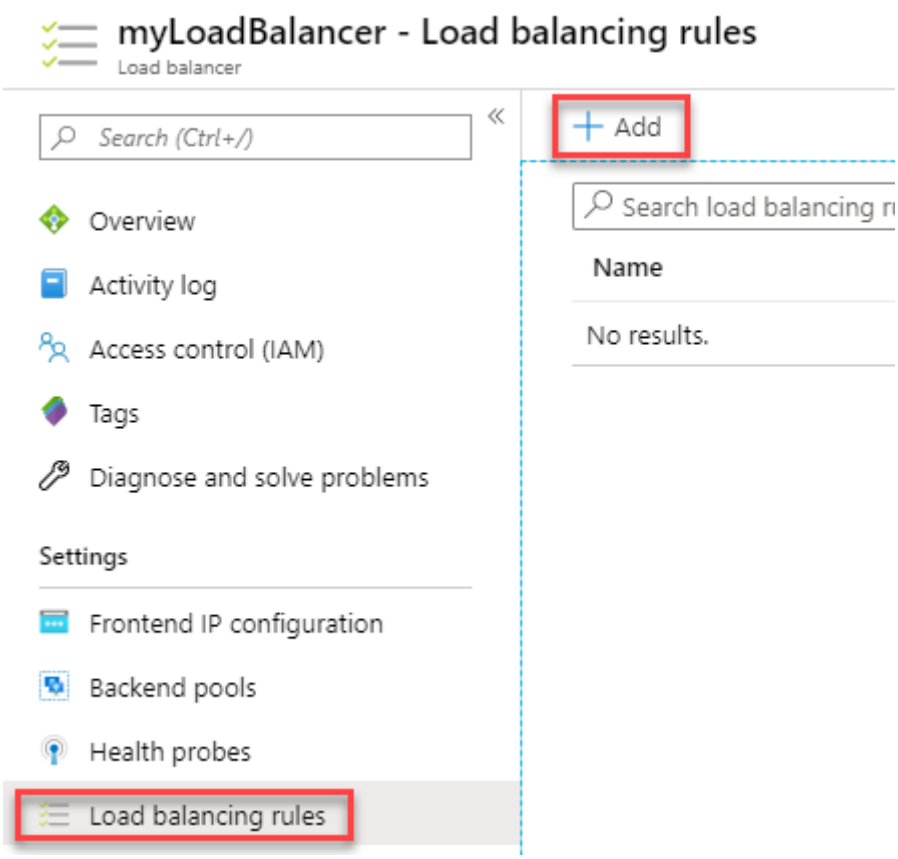


3. Click **OK**.

29.1.7 Task 7: Create a load balancer rule

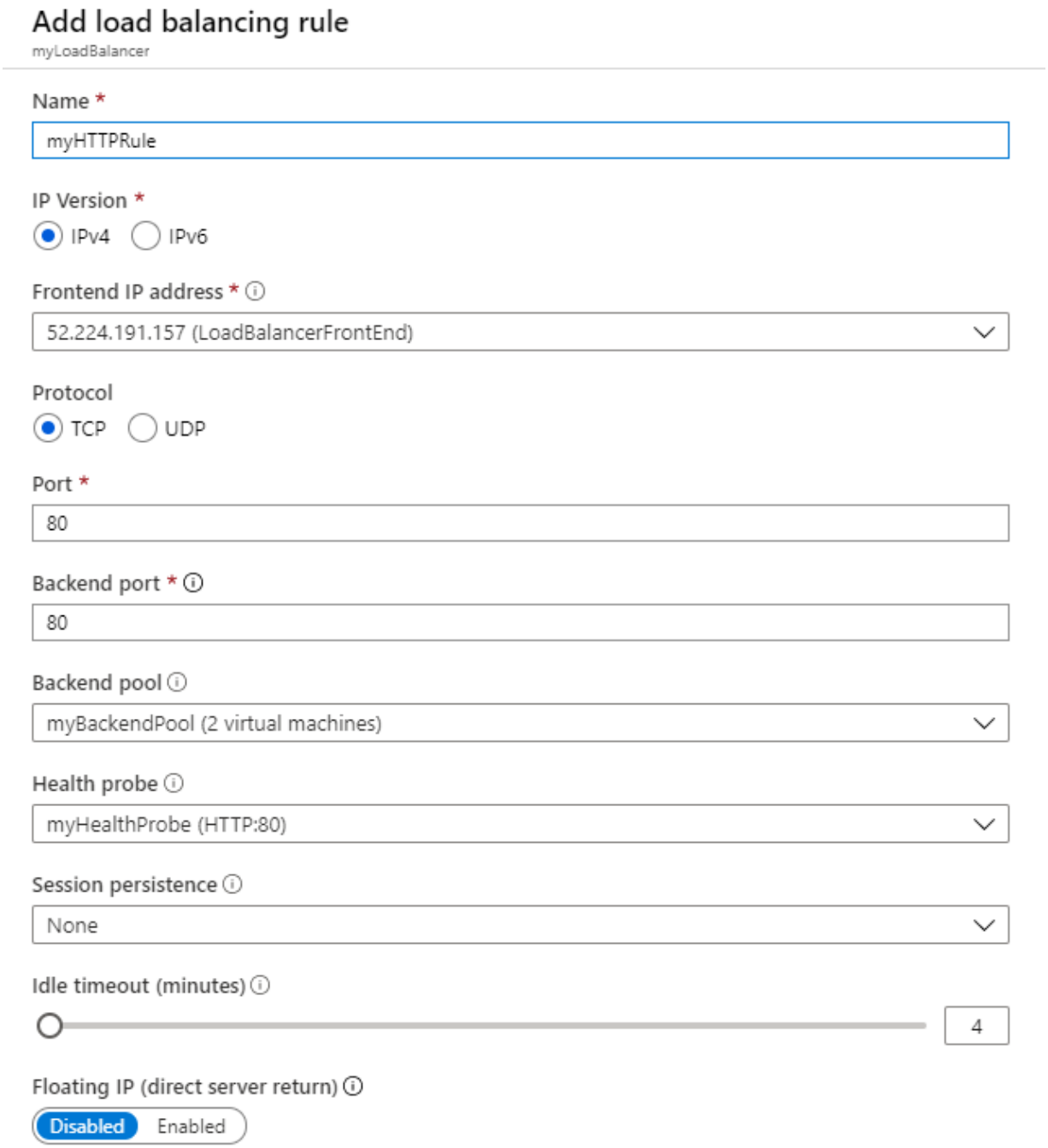
A load balancer rule is used to define how traffic is distributed to the VMs. You define the frontend IP configuration for the incoming traffic and the backend IP pool to receive the traffic, along with the required source and destination port. Create a load balancer rule *myLoadBalancerRuleWeb* for 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. On the Load Balancer blade, under **Settings**, click **Load balancing rules**, then click **Add**.



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

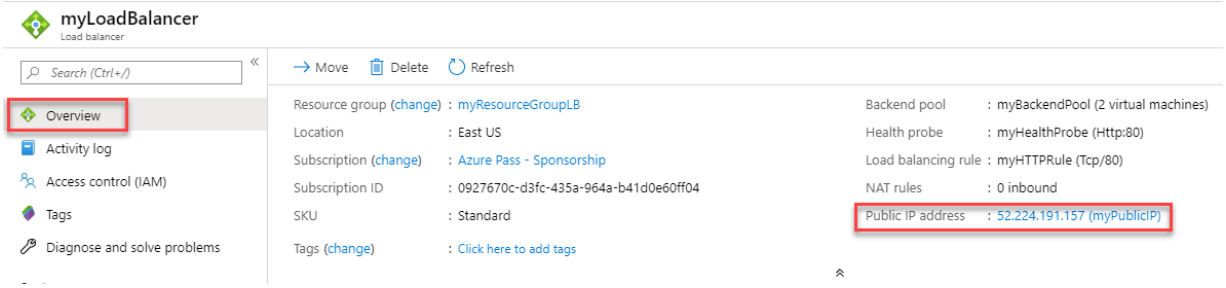
   - *myHTTPRule* - for the name of the load balancing rule.
   - **TCP** - for the protocol type.
   - *80* - for the port number.
   - *80* - for the backend port.
   - *myBackendPool* - for the name of the backend pool.
   - *myHealthProbe* - for the name of the health probe.



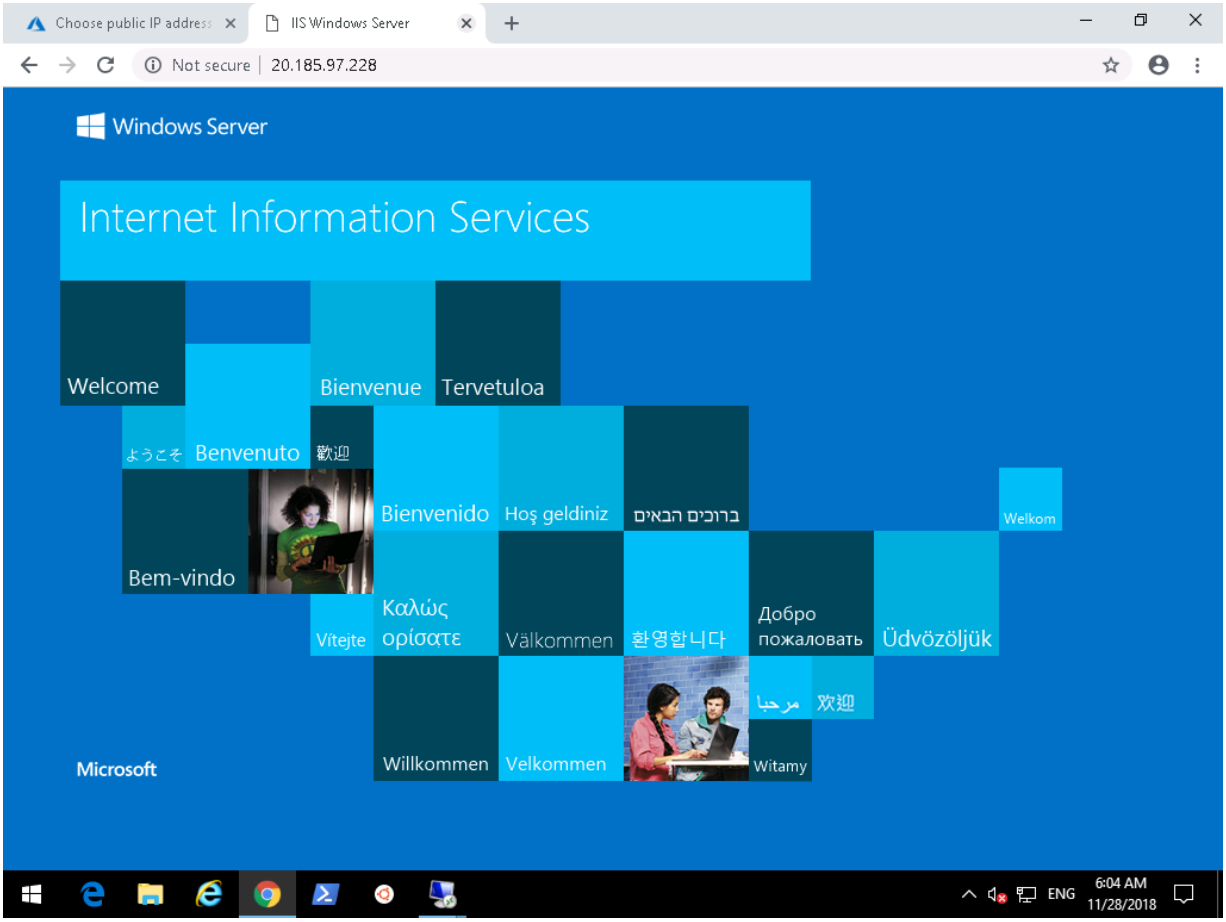
3. Click **OK**.

29.1.8 Task 8: Test the load balancer

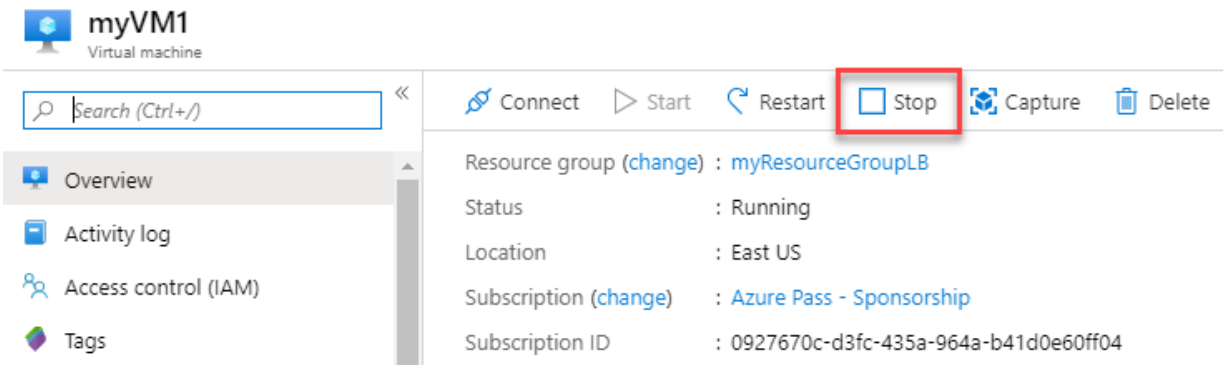
1. Find the public IP address for the Load Balancer on the **Overview** screen.



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.



3. Notice the IIS default page loads.
4. In the Azure Portal click on **Virtual Machines** in the hub menu. Select myVM1 and in the **Overview** blade click **Stop** and confirm **Yes**.



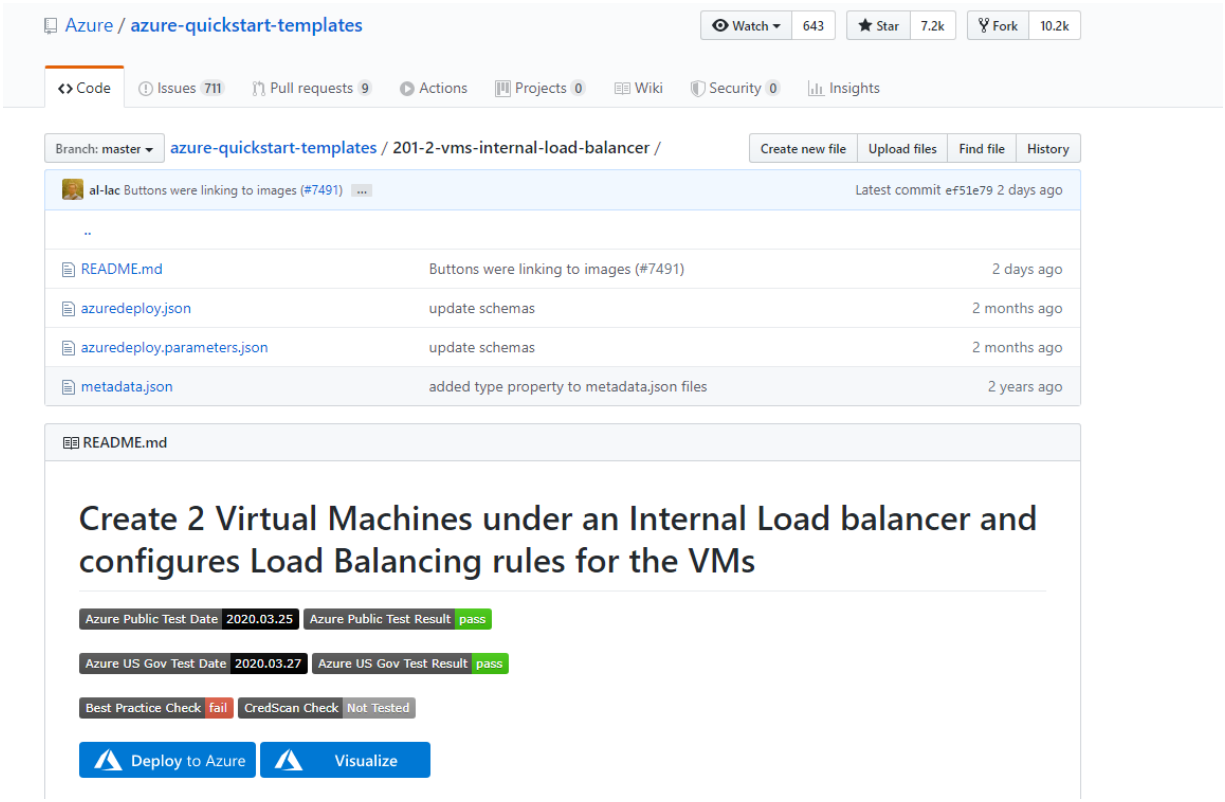
5. Wait until the myVM1 Virtual Machine has stopped then go back to the browser tab with the load lanancer public IP and click refresh to confirm myVM2 is continuing to service the requests and the load balancer is functioning as expected.

29.2 Exercise 2: Load Balancer ARM Deployments

29.2.1 Task 1: Deploy an ARM template

This template allows you to create 2 Virtual Machines under a Load balancer and configure a load balancing rule on Port 80. This template also deploys a Storage Account, Virtual Network, Public IP address, Availability Set and Network Interfaces. In this template, we use the resource loops capability to create the network interfaces and virtual machines

1. In a new tab in your browser, navigate to the following URL `https://aka.gd/2E2MAjh`
2. Click **Deploy to Azure**



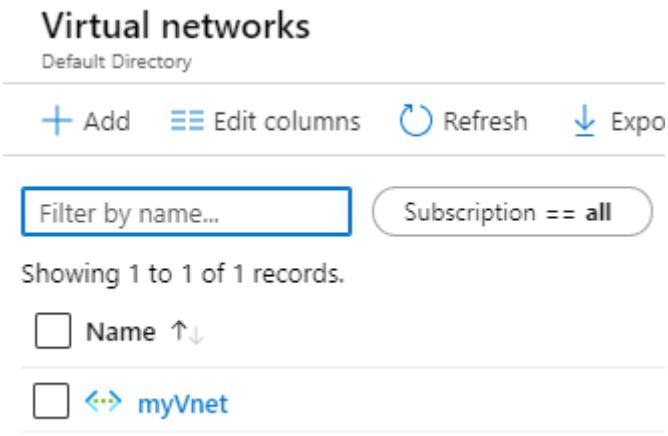
3. On the template blade that opens, enter the following details:
   - Resource group: **myResourceGroupLB**
   - Admin Username: **localadmin**
   - Admin Password: **Pa55w.rd1234**
4. Click **I agree....** and click **Purchase**.

30 Exercise 3: Deploying Application Gateways

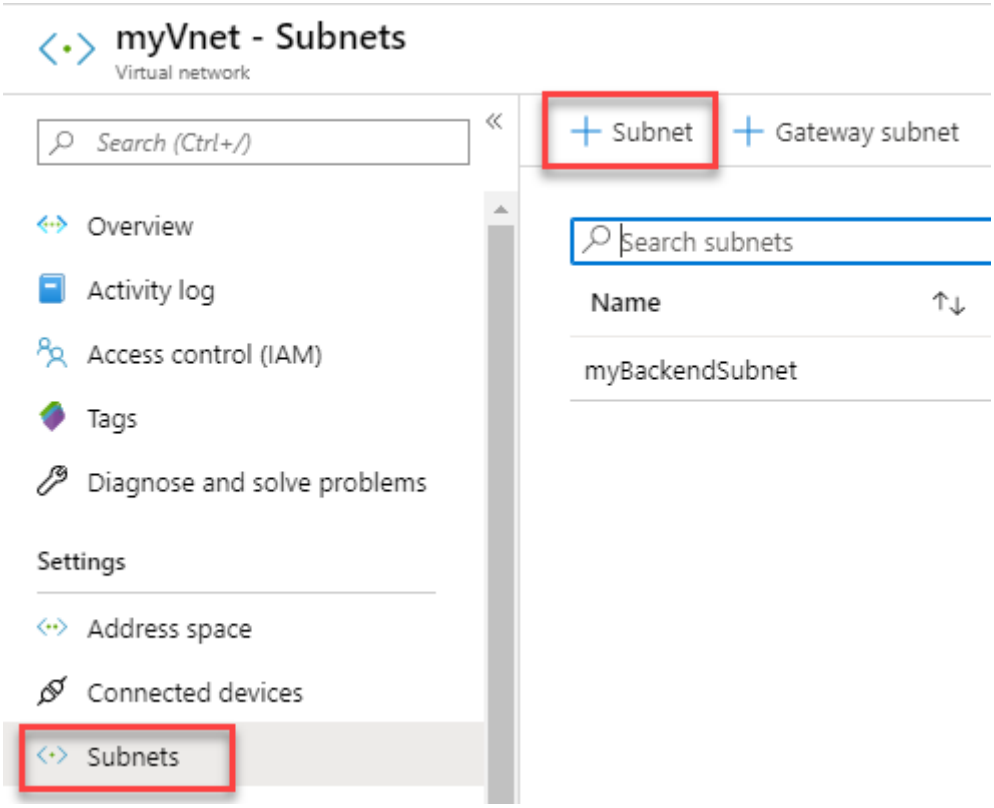
30.0.1 Task 1: Create an application gateway

A virtual network is needed for communication between the resources that you create. Two subnets are created in this example: one for the application gateway, and the other for the backend servers. You can create a virtual network at the same time that you create the application gateway.

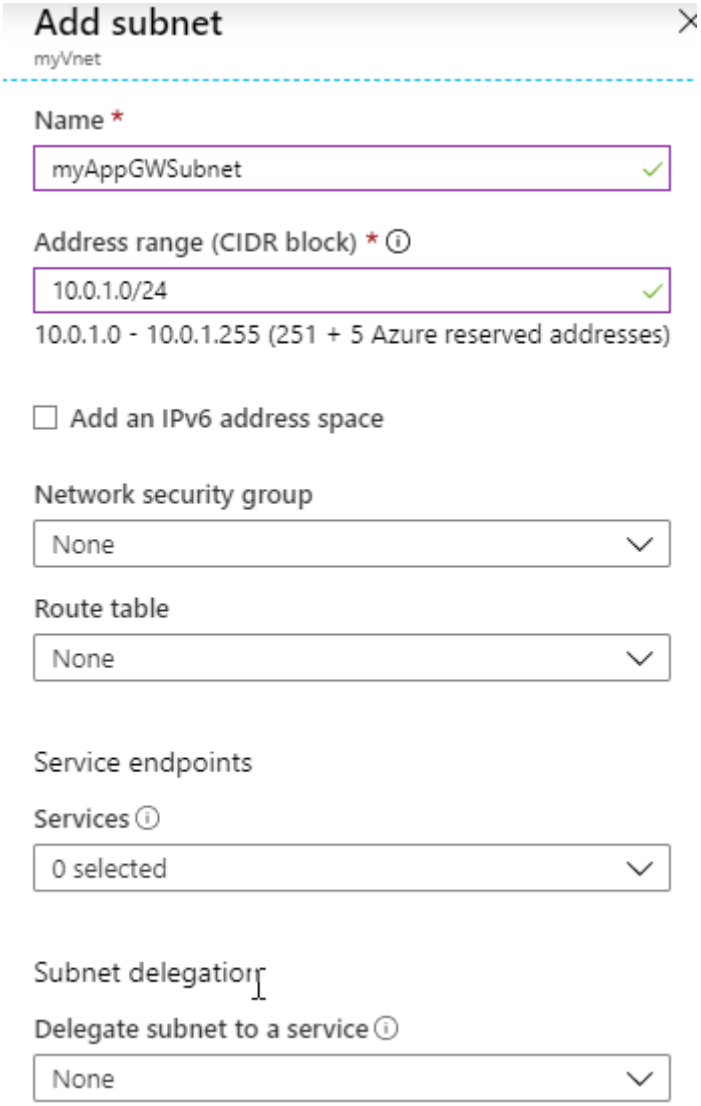
1. First you need to create a subnet for the Application Gateway to reside in. Click **Virtual networks** on hub menu and select **myVNet**.



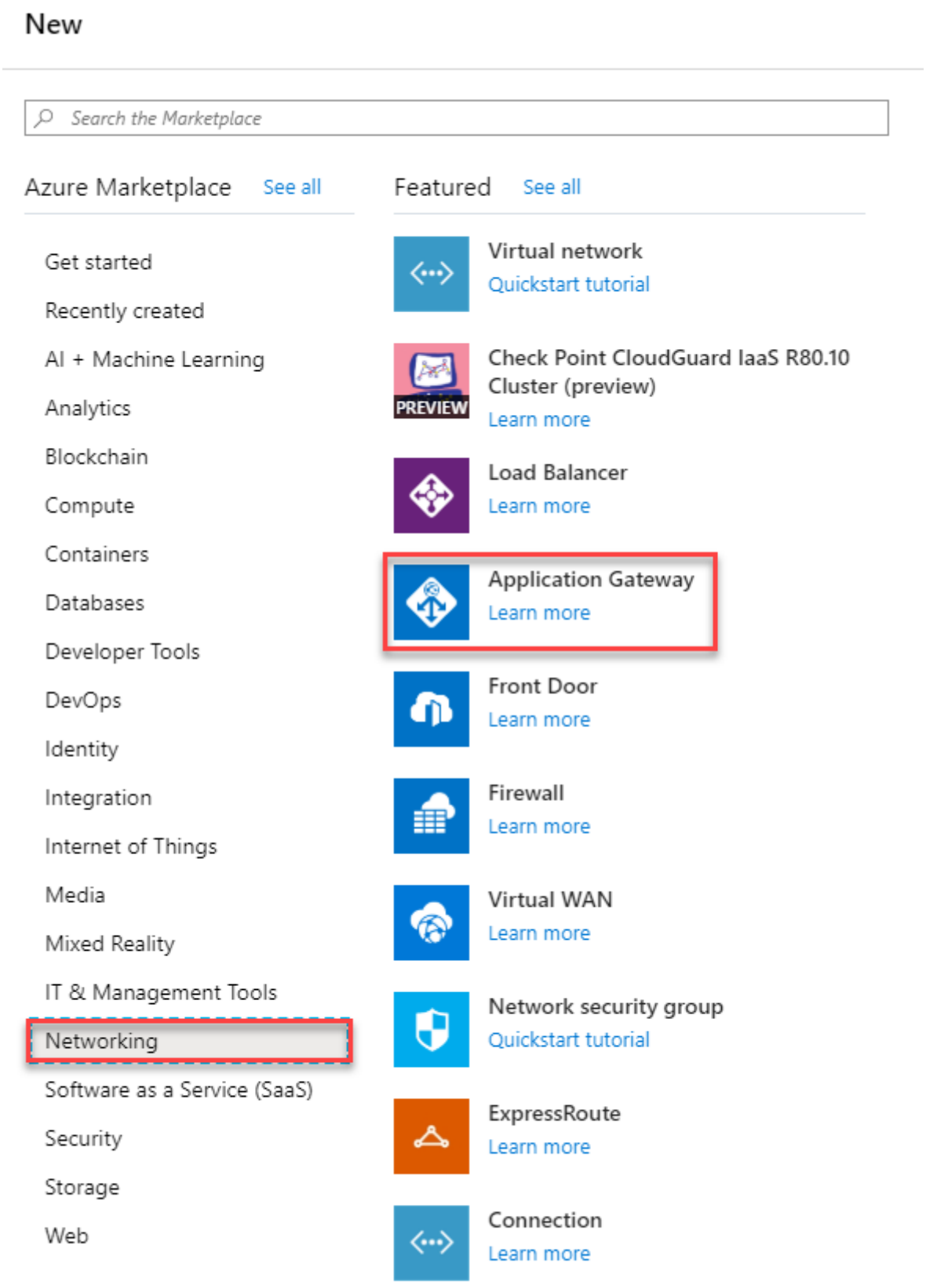
2. Click **Subnets** and click + **Subnet**.



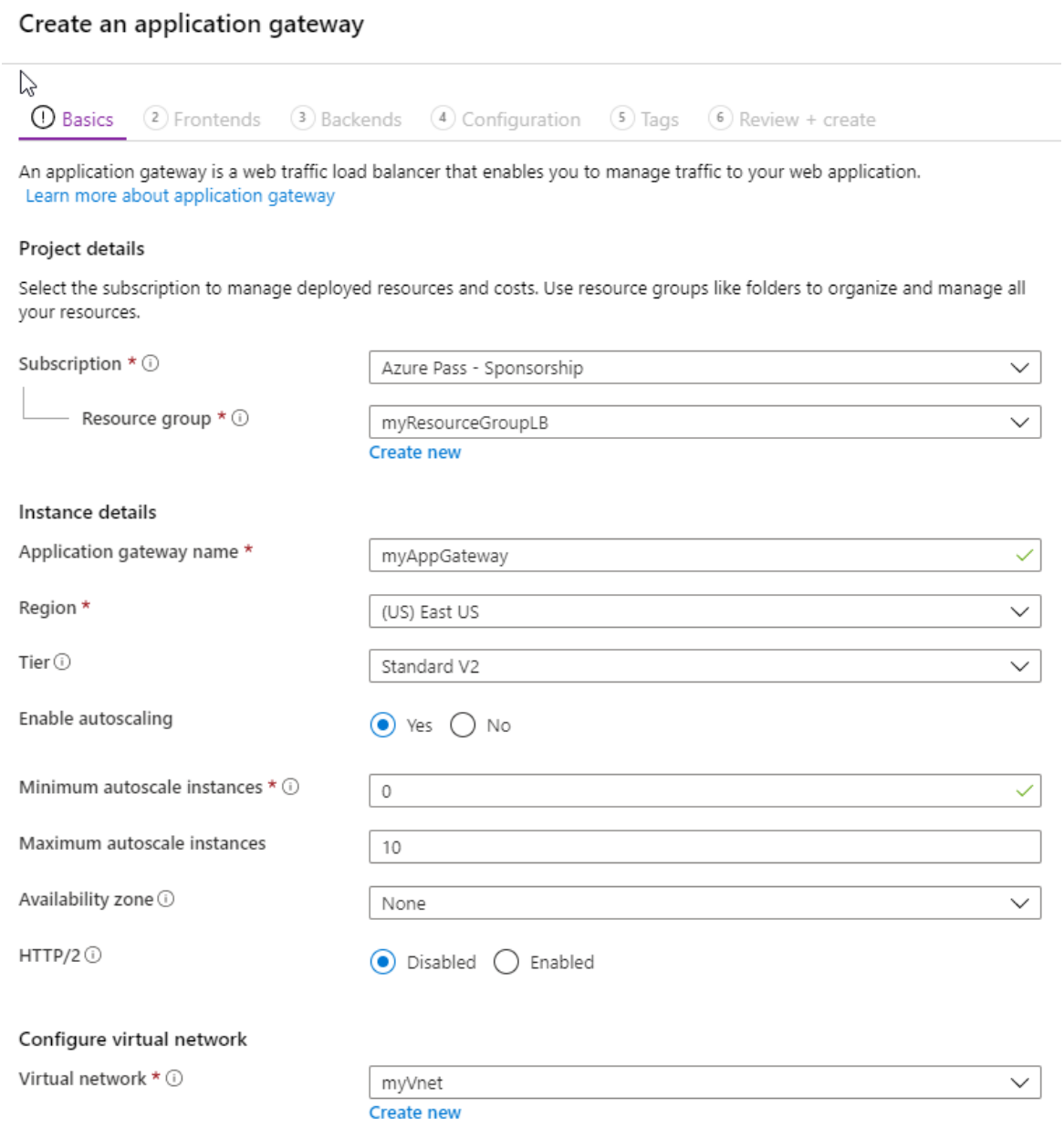
3. Enter **myAppGWSubnet** as the name and click **OK**.



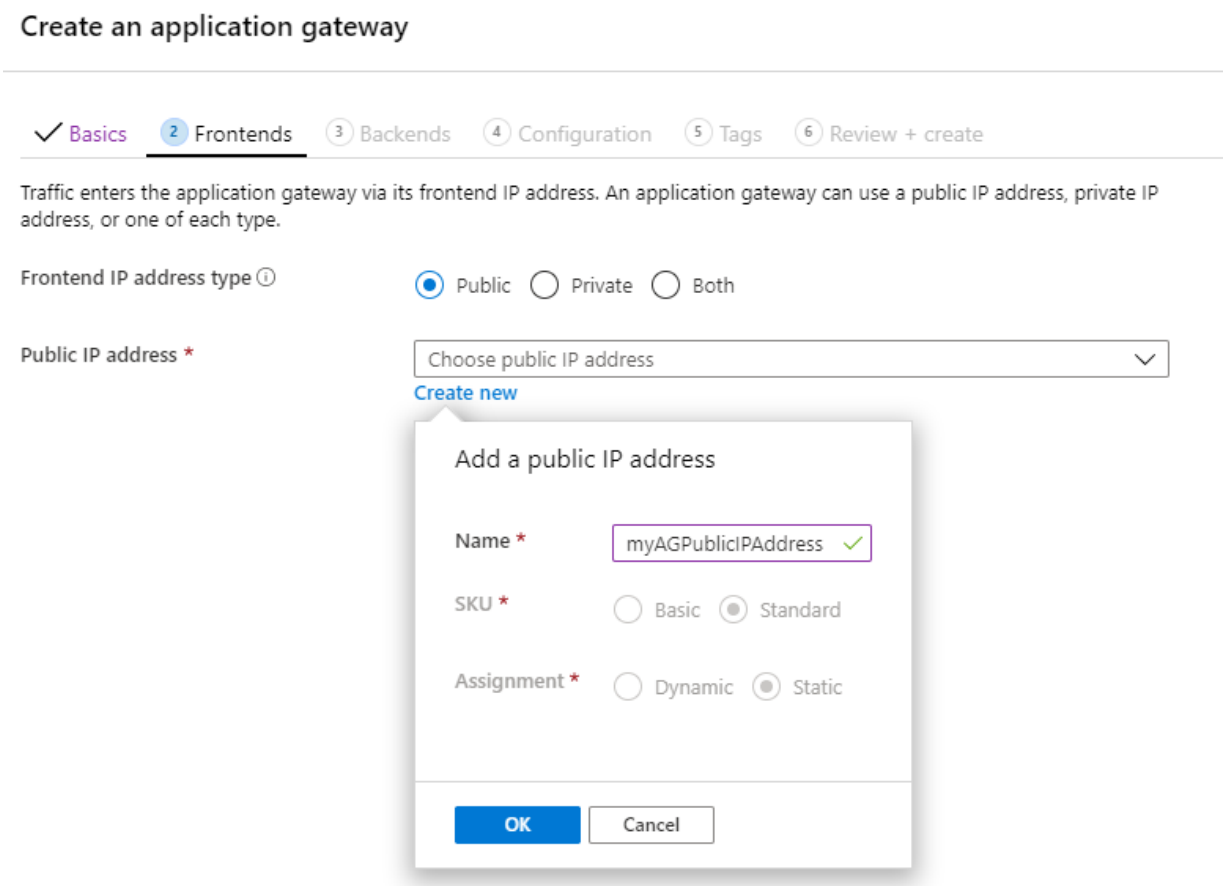
4. Click **Create a resource** found on the upper left-hand corner of the Azure portal.
5. Click **Networking** and then click **Application Gateway** in the Featured list.



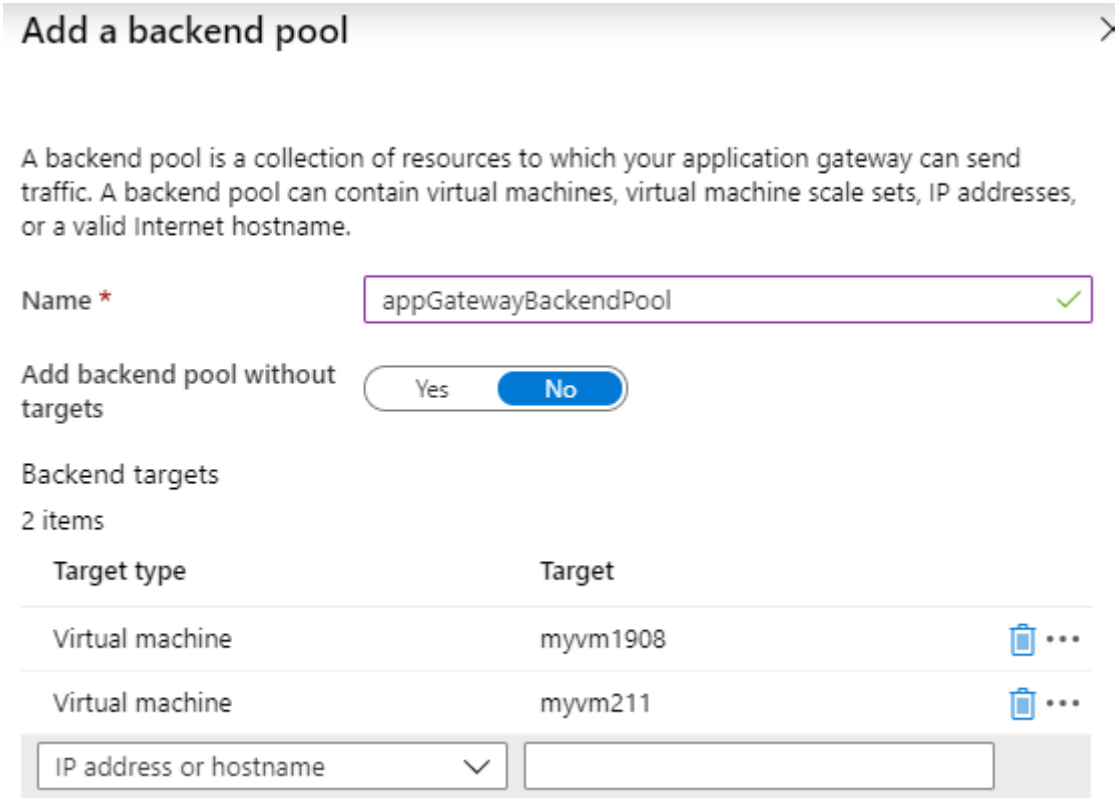
6. Enter these values for the application gateway basics blade then click **Next**:
   - *myAppGateway* - for the name of the application gateway.
   - *myResourceGroupLB* - select the already existing Resource Group.
   - *myVnet* - select the already existing Virtual network.



7. Under **Frontend configuration** blade, ensure **IP address type** is set to **public**, and under **Public IP address**, click **Create new**. Type ***myAGPublicIPAddress*** for the public IP address name and then click **OK**.

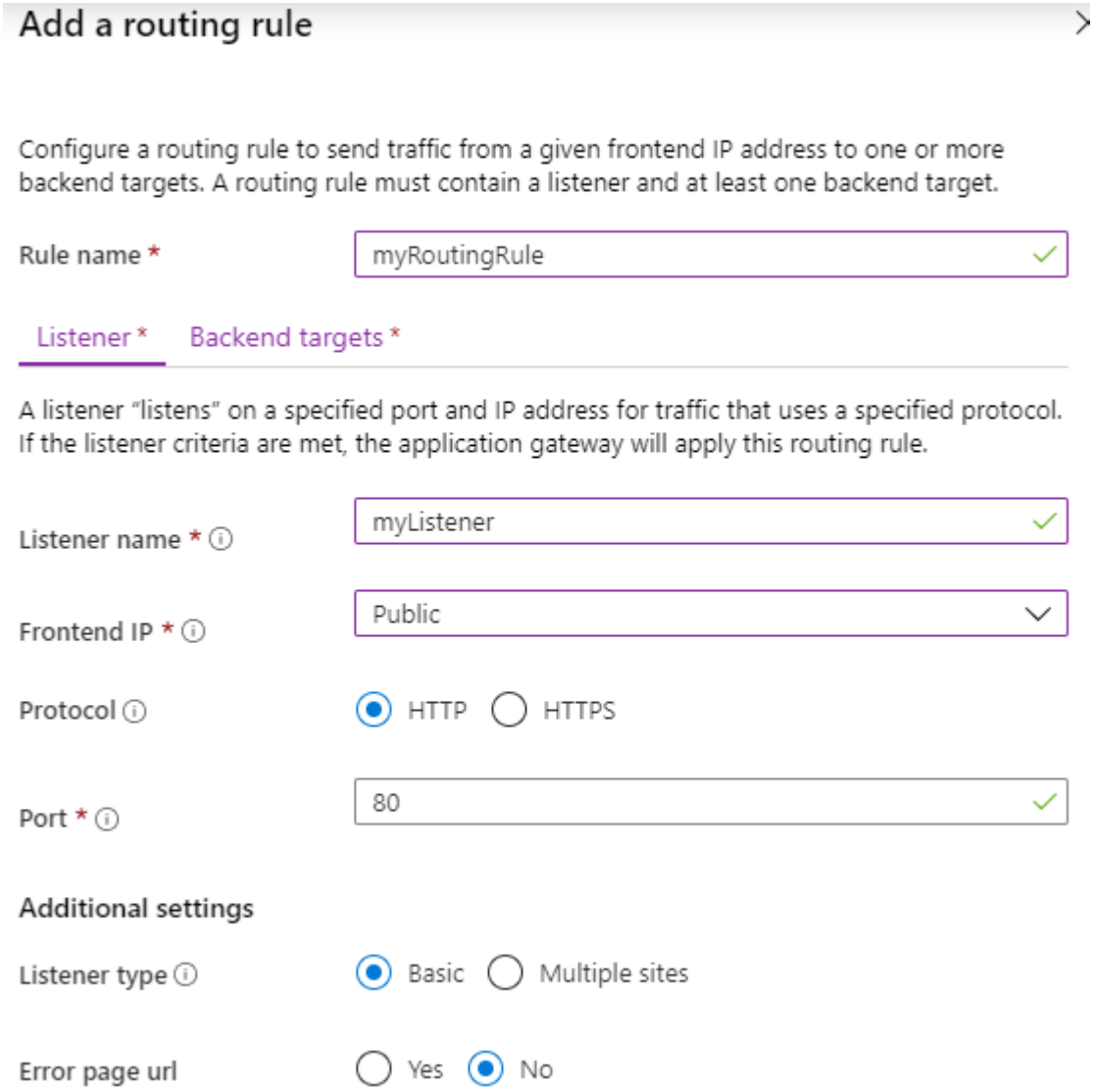


8. Click **Next**.
9. Select **+Add a backend pool**.
10. Enter the name **appGatewayBackendPool**. Under backend targets select **Virtual Machine** and add myVM1 and myVM2 virtual machines and their associated network interfaces then click **Add**.

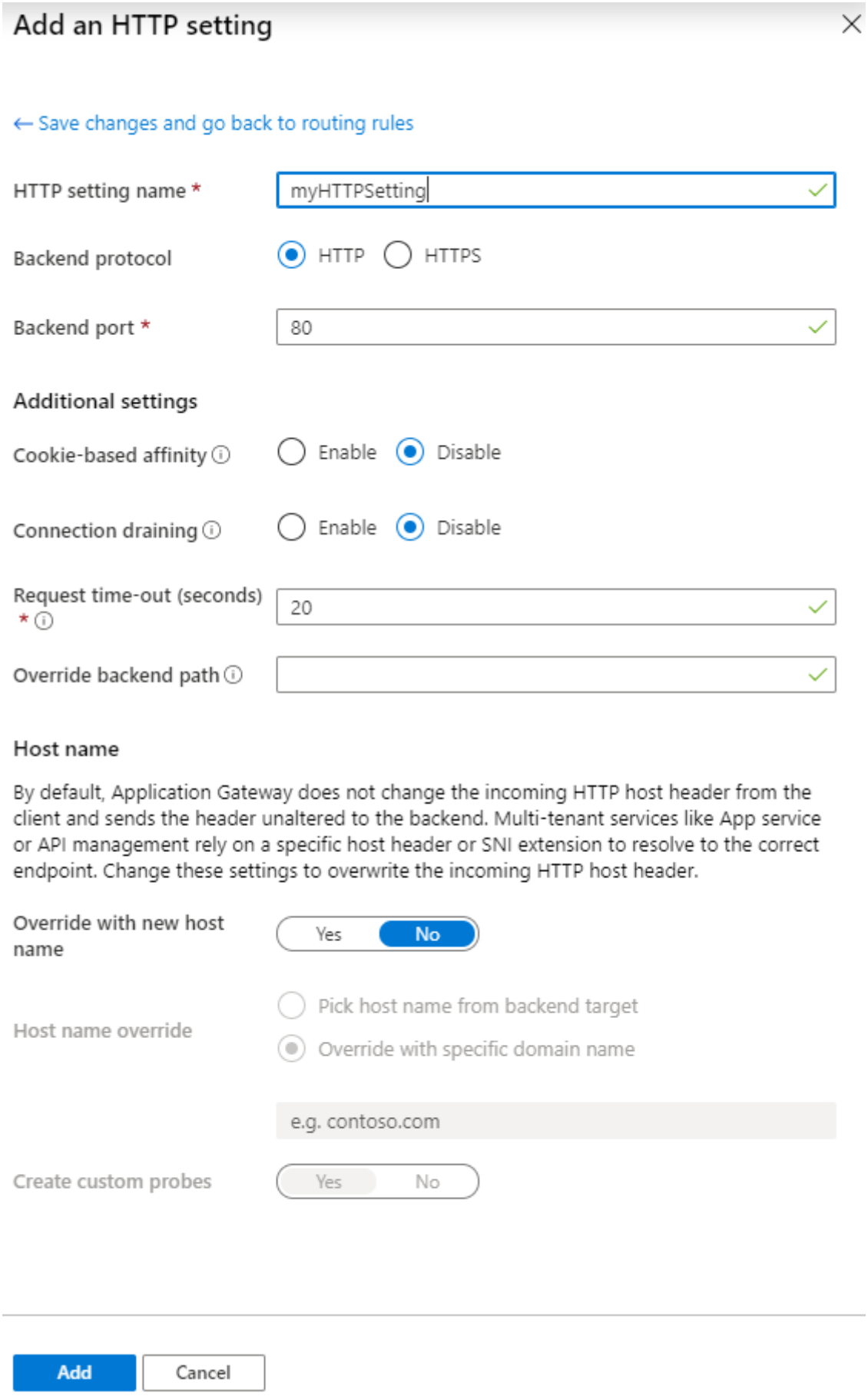


11. Click **Next**.
12. On the **Configuration** tab, you'll connect the frontend and backend pool you created using a routing rule.
13. Select **Add a rule** in the **Routing rules** column.
14. In the **Add a routing rule** window that opens, enter *myRoutingRule* for the **Rule name**.
15. A routing rule requires a listener. On the **Listener** tab within the **Add a routing rule** window, enter the following values for the listener:
    - **Listener name**: Enter *myListener* for the name of the listener.
    - **Frontend IP**: Select **Public** to choose the public IP you created for the frontend.

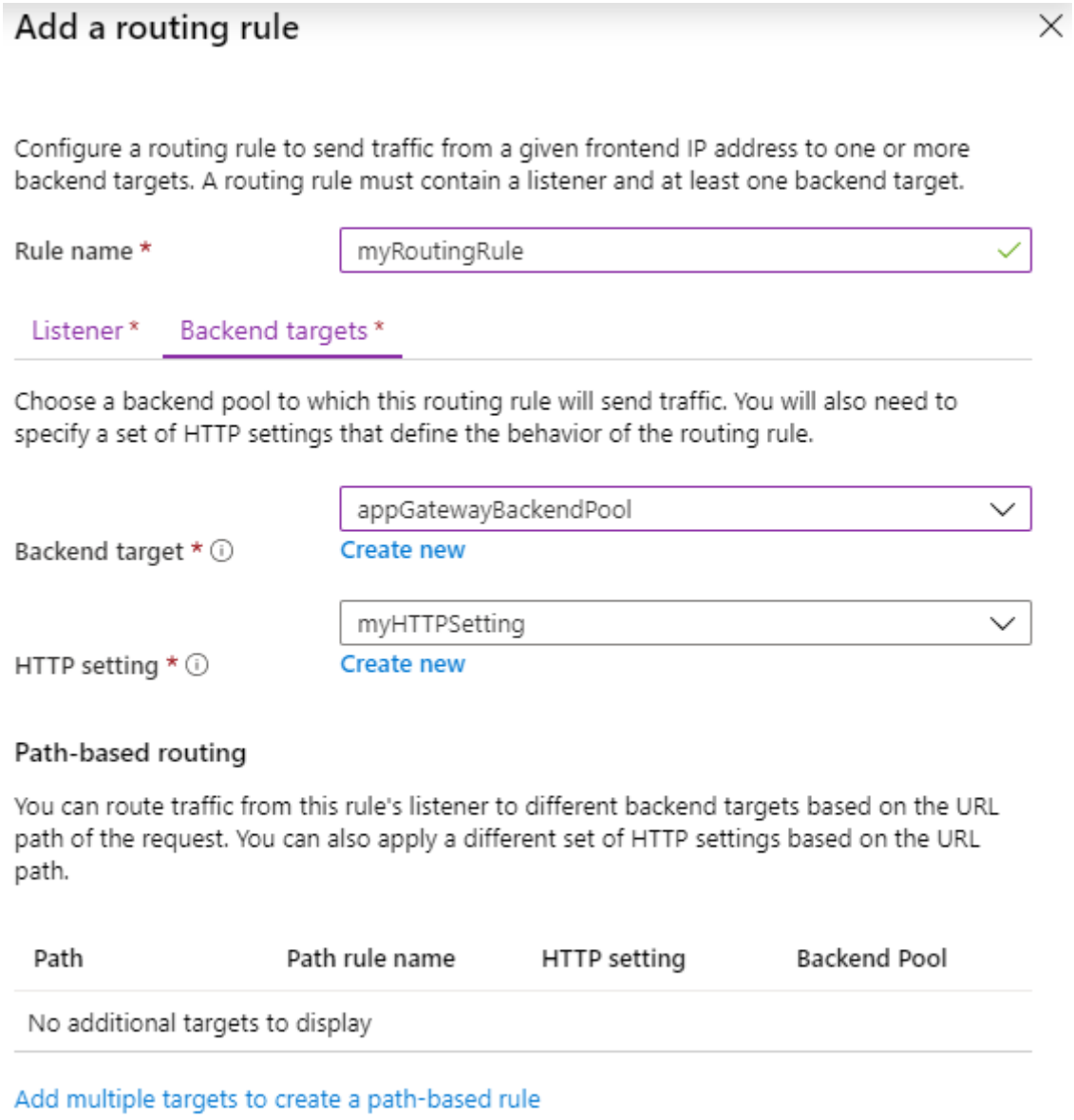
      Accept the default values for the other settings on the **Listener** tab, then select the **Backend targets** tab to configure the rest of the routing rule.



16. On the **Backend targets** tab, select **appGatewayBackendPool** for the **Backend target**.
17. For the **HTTP setting**, select **Add new** to create a new HTTP setting. The HTTP setting will determine the behavior of the routing rule. In the **Add an HTTP setting** window that opens, enter *myHTTPSetting* for the **HTTP setting name**. Accept the default values for the other settings in the **Add an HTTP setting** window, then select **Add** to return to the **Add a routing rule** window.



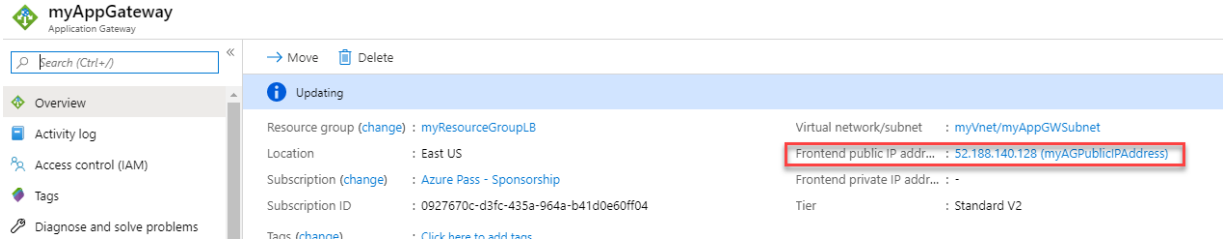
18. On the **Add a routing rule** window, select **Add** to save the routing rule and return to the **Configuration** tab.



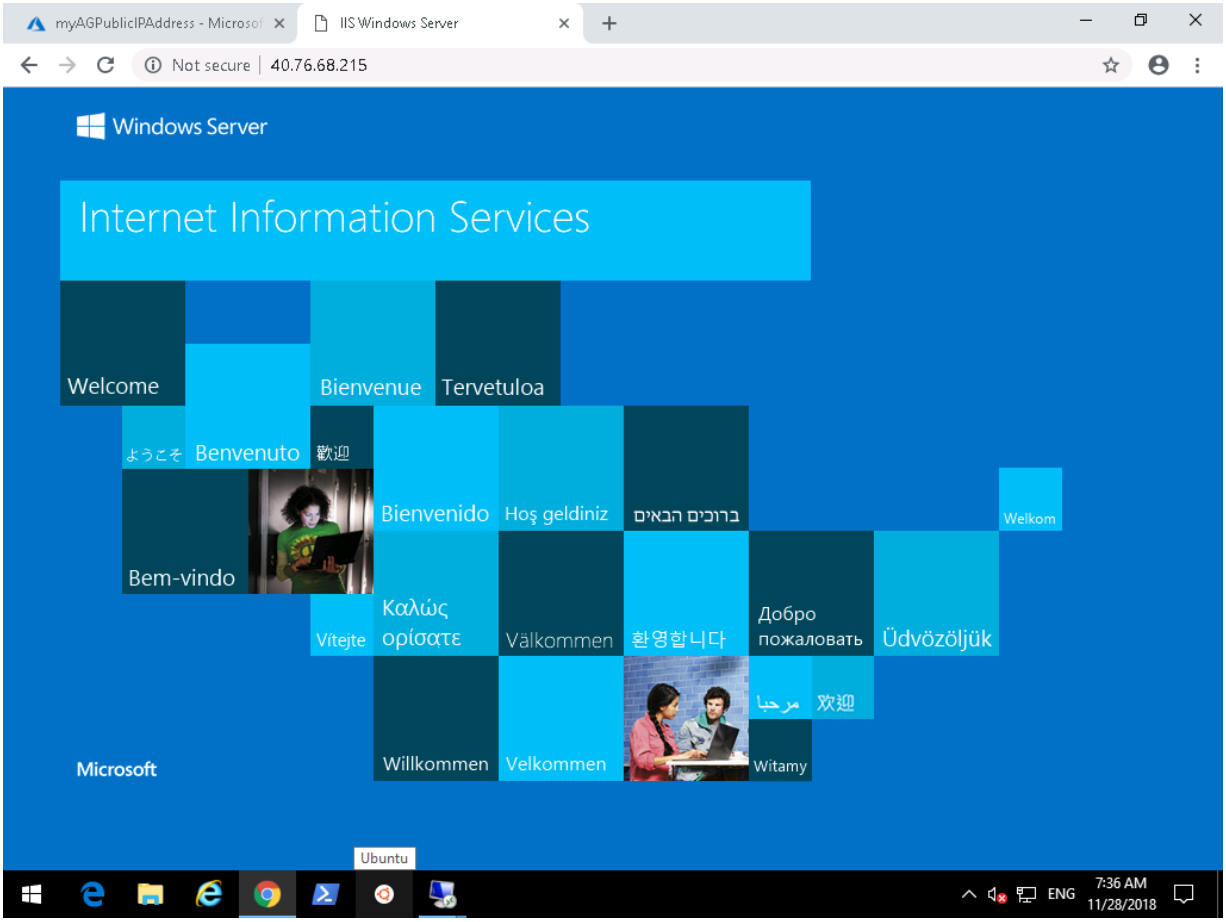
19. Select **Add**
20. Select **Next: Tags** and then **Next: Review + create**, then select **Create**.

30.0.2 Task 2: Test the application gateway

1. Find the public IP address for the application gateway on the Overview screen. Click **All resources** and then click **myAGPublicIPAddress**.



2. Copy the public IP address, and then paste it into the address bar of your browser.



3. To verify, go to **Network Watcher**, choose **Topology** then choose **myResourceGroupLB** to see the overall network diagram.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this lab.

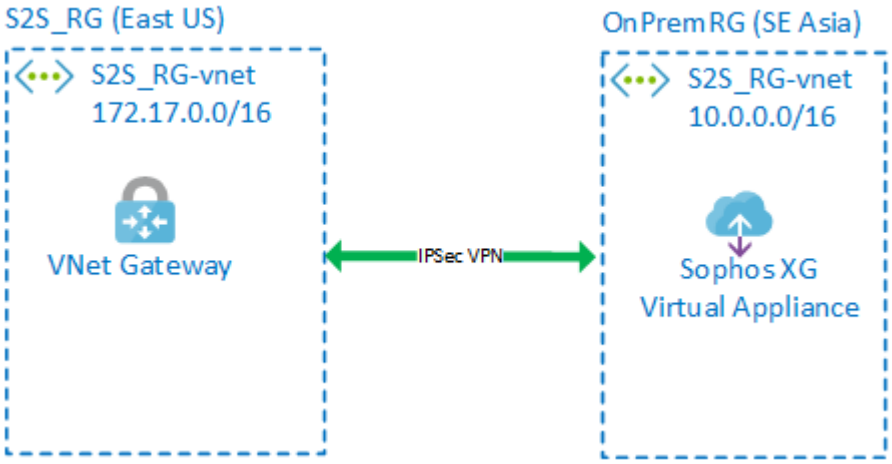
---

31 Module 2: Lab 11: On-Prem to Azure Connections - VPN Gateways and Tunnelling

31.1 Exercise 1: Deploy Virtual Appliances and Gateways for intersite connectivity.

31.1.1 Task 1: Deploy a Virtual Appliance.

In this task you will create a Sophos XG Virtual Appliance which will emulate an on-premises device. The layout of this is depicted in the diagaram below



1. In your browser, navigate to the following URL to open the ARM template:

   `https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2F`

2. Login to the portal if required.
3. On the Custom deployment enter or select the following details:

Setting	Value
Resource Group	*Create New* **OnPremRG**
Location	**Southeast Asia**
Admin Password	**Pa55w.rd1234**
Public IP DNS	*Enter a unique name*
Storage Name	*Enter a unique name*

4. Scroll to the bottom of the blade and click the check box next to I agree to the terms and conditions..... and click **Purchase**.

31.1.2 Task 2: Create a Resource Group and VNet.

In this task you will create a Virtual Machine and a Virtual Network inside a new Resource group which will be used to connect to your emulated On-Prem environment.

1. Login to your Azure Portal `https://portal.azure.com`
2. Click **Create a resource** > **Networking** > **Virtual Network**
3. Change the values in the **Create virtual network** blade change the values to be the same as the output below:

   - **Name** S2S_RG-vnet
   - **Resource group** Create New: S2S_RG
   - **Location**: East US

   Click the IP Addresses tab and enter the following values:

   - **Address space** 172.17.0.0/16
   - **Subnet address range** 172.17.0.0/24

4. Click **Create**. **Note:** You can continue to the next task without having to wait for the deployment to complete.

31.1.3 Task 3: Create a Gateway Subnet and a Virtual network Gateway.

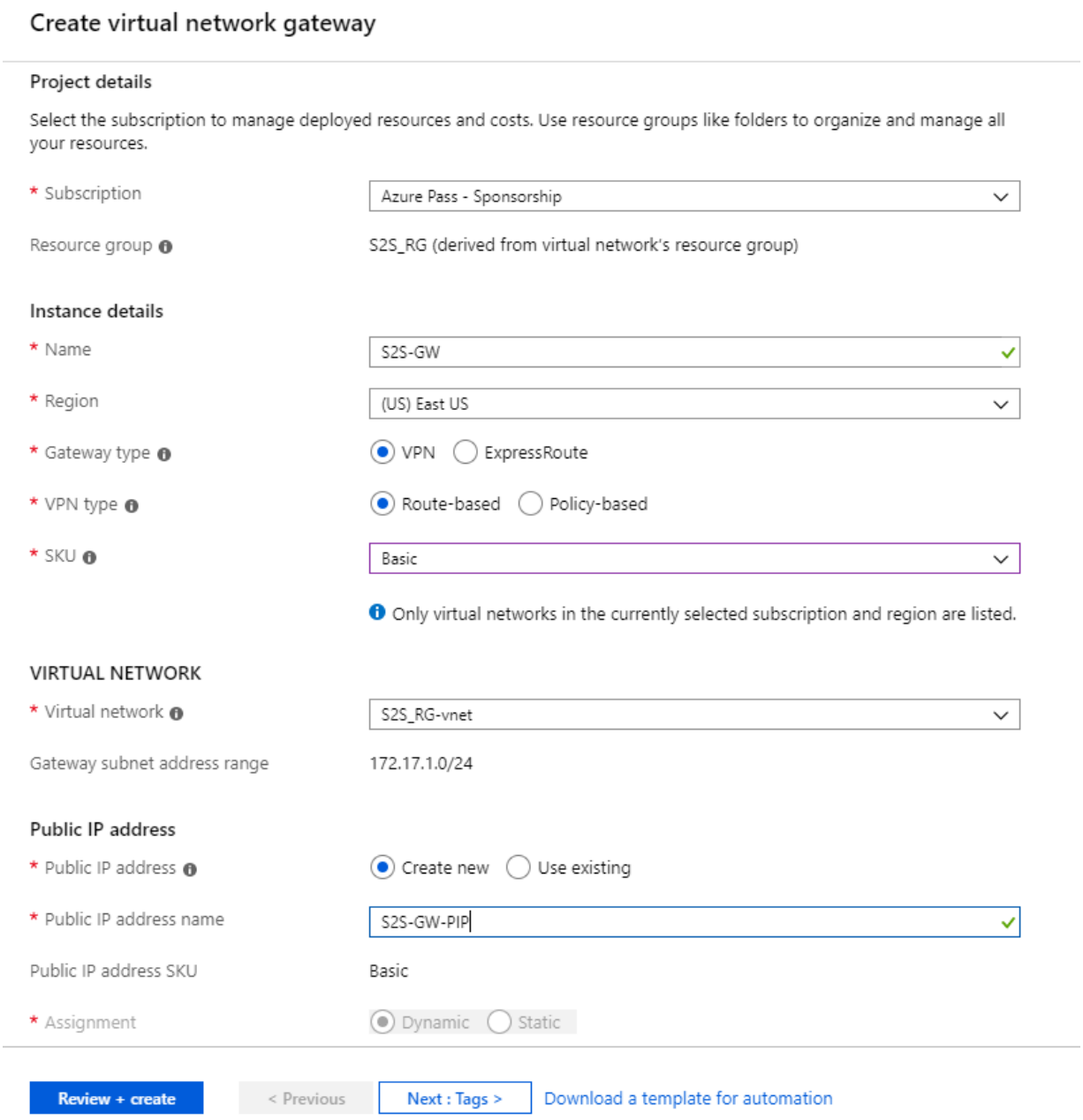
In this task you will Create a Gateway Subnet and a Virtual network Gateway which will enable you to create a connection between On-Prem and your Azure VNet.

1. In the Azure Portal click **Resource Groups** on the Hub Menu.
2. Click the **S2S_RG** resource group that has been created for you.

3. In the S2S_RG Resource Group blade click the **S2S_RG-vnet**.
4. On the **S2S_RG-vnet** menu click **Subnets**.
5. Click **+ Gateway subnet**.

   Note: You need to create a Gateway subnet for the Gateway machines to reside in. All the routing is done by the Azure Software Defined Networking.
6. Leave the default options on the **Add subnet** blade and click **OK**.
7. Click **+ Create a resource**.
8. Search for Virtual Network Gateway and select **Virtual network gateway**.
9. Click **Create**.
10. On the **Create virtual network gateway** blade enter the following information:

    - **Name**: S2S-GW
    - **Name**: (US) East US
    - **Gateway type**: VPN
    - **VPN Type**: Route-based
    - **SKU**: Basic
    - **Virtual network**: Select the S2S_RG-vnet (this was created earlier when you deployed the VM)
    - **Public IP address**: (Create New) Name: S2S-GW-PIP

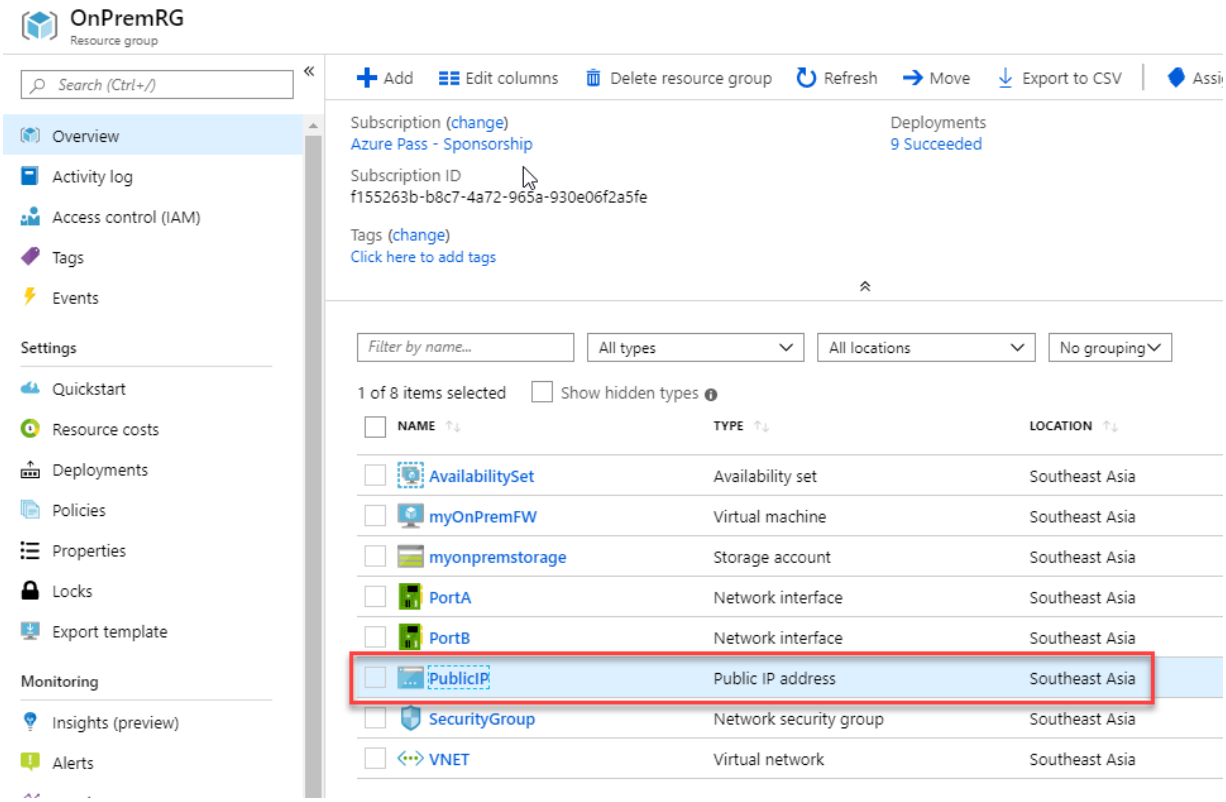


11. Click **Review + create** then on the summary screen click **Create**

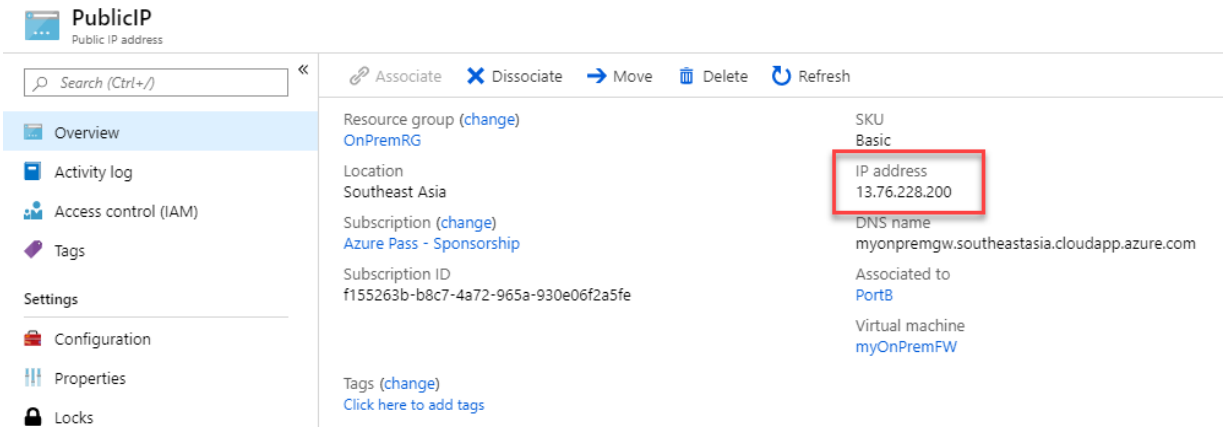
Note: The gateway may take up to 45 minutes to deploy, although in most cases it is much quicker. Monitor this by clicking on the Bell Icon. You can continue to the next task whilst the Gateway is deploying.

31.1.4 Task 4: Configure the Sophos virtual appliance.

1. On the Azure Portal Hub menu click **Resource Groups**.
2. Select the **OnPremRG** Resource Group.
3. Select the **PublicIP** Resource.

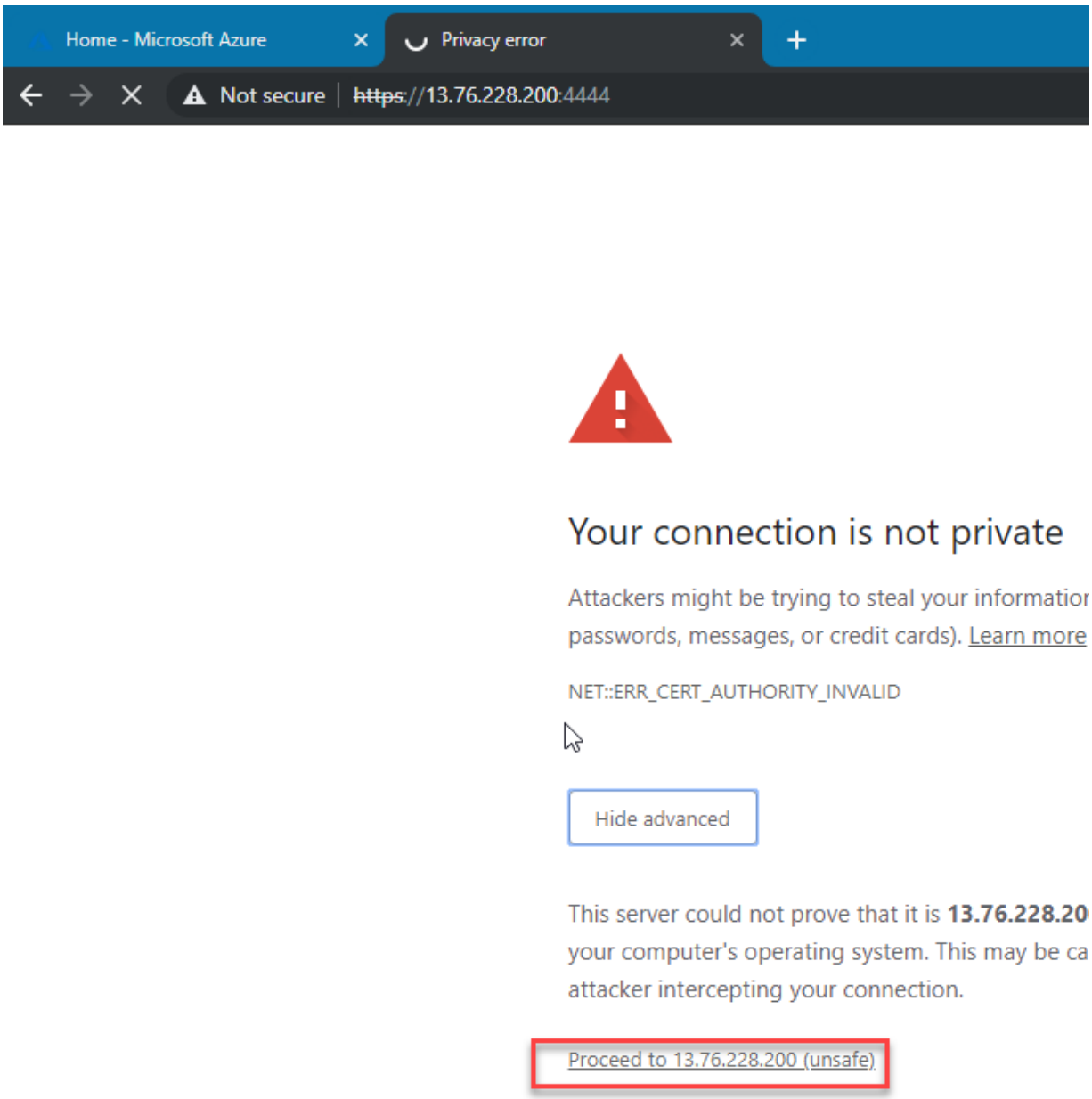


4. Make a note of the assigned Public IP address.

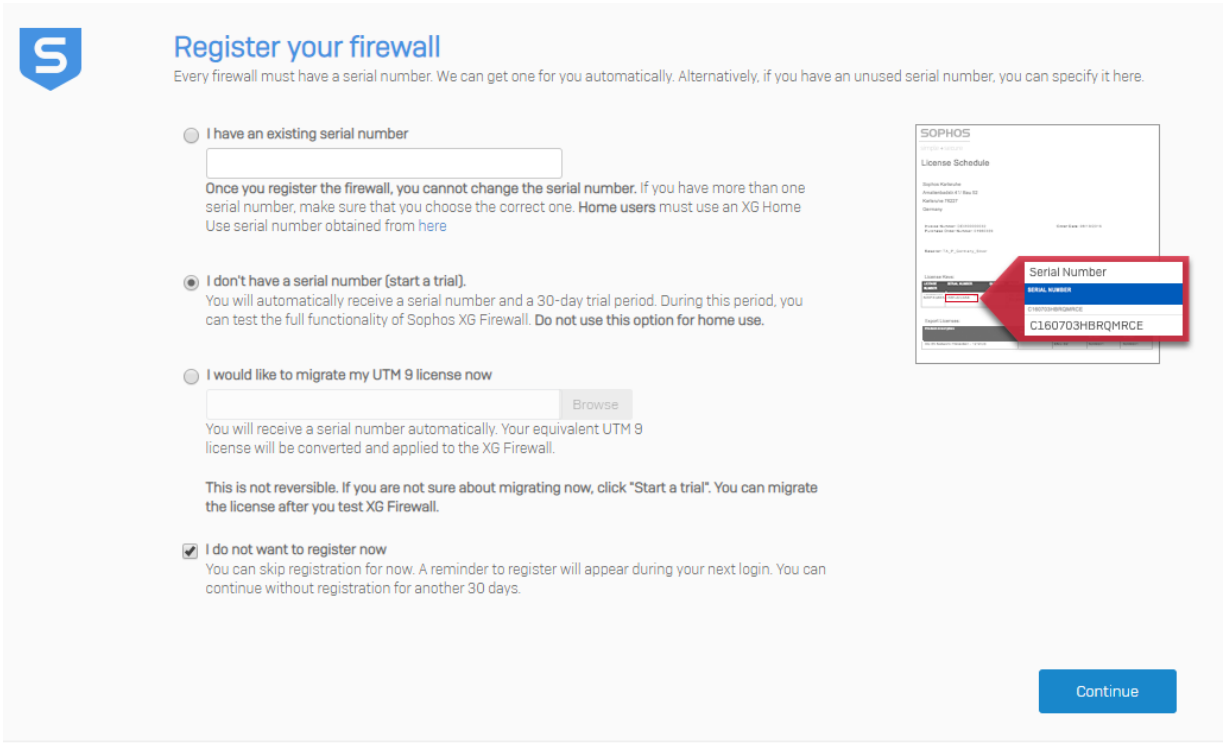


5. Open a new browser session and navigate to **`https://x.x.x.x:4444`** (where x.x.x.x is the public IP address you noted above).

6. Depending on your browser there may be different options to proceed with the connection. Select Details then Go to web page (not recommended)

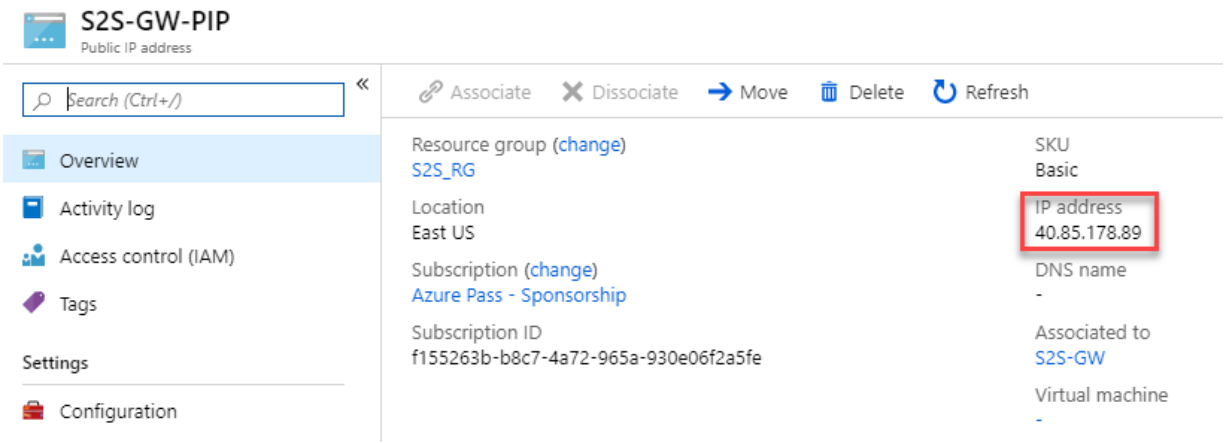


7. Log into the Firewall with the following credentials:
   - Admin
   - Pa55w.rd1234
8. Accept the licence agreement.
9. On the Register your firewall page click **I don't have a a serial number (start a trial)** and select **I do not want to register now** then click **Continue**.



10. On the Warning pop up click **Continue**.
11. Return back to the Azure Portal. Open the **S2S_RG** Resource Group and select the **S2S-GW-PIP** Public IP and make a note of it.

    Note: This is your Public IP you will connect your Sophos virtual appliance to via IPSec VPN. If the IP Address field is empty, continue to wait for the template deployment to complete, periodically checking the deployment status and refreshing the **S2S-GW-PIP** blade.



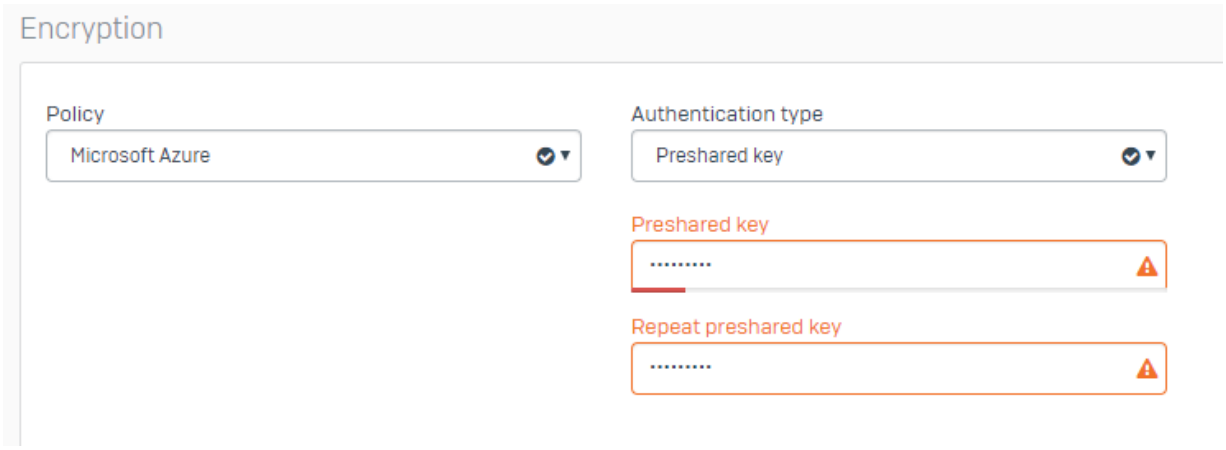
12. Return back to the Sophos Portal.
13. Go to **VPN** > **IPsec Connections**, select **Add** and configure the following settings: *If in doubt see the screenshots below*

    General Settings Section:

    - **Name**: On_Prem_to_Azure
    - **IP Version**: IPv4.
    - **Activate on Save:** Selected.
    - **Create firewall rule:** Selected.
    - **Description**: Site to Site connection from On Prem to Azure VNet.
    - **Connection Type**: Site-to-Site.
    - **Gateway Type**: Respond Only.

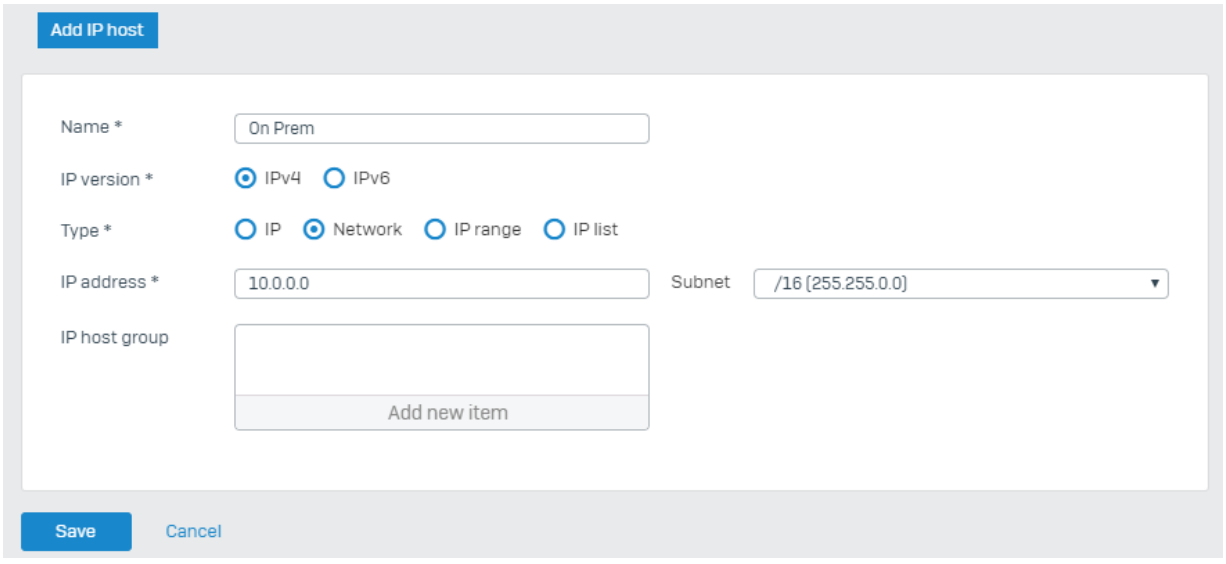
Encryption Section:

- **Policy**: Microsoft Azure.
- **Authentication Type**: Preshared Key.
- **Preshared Key**: 123456789
- **Repeat Preshared Key**: 123456789

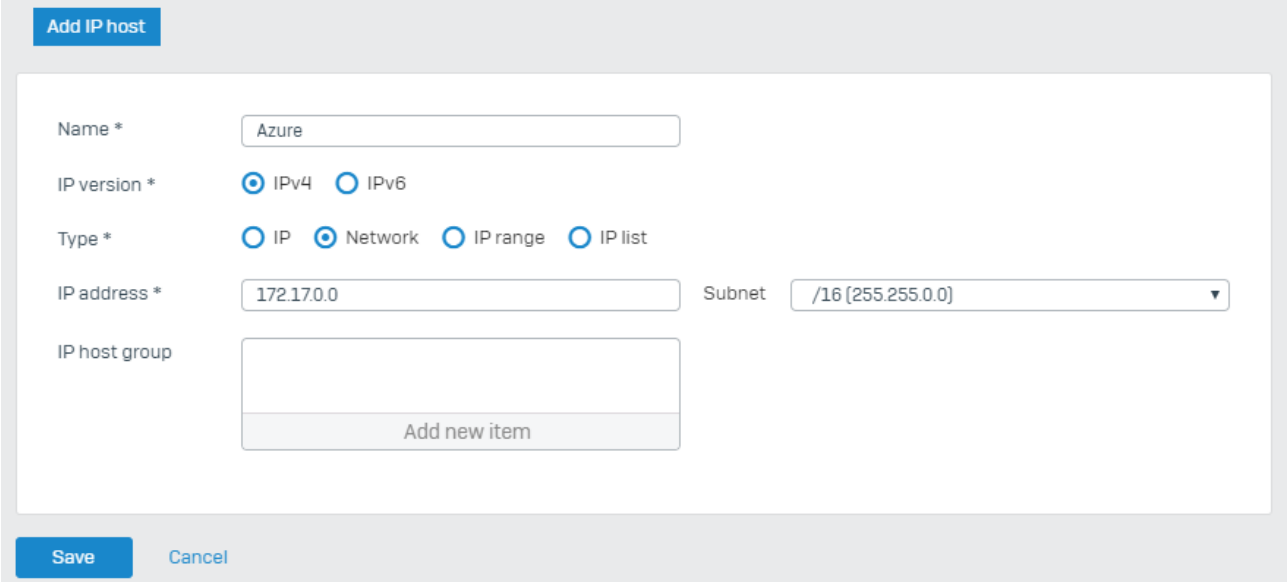


Gateway Settings Section:

- **Listening Interface**: Select the default.
- **Gateway Address**: Input the public IP of the Azure VPN gateway noted earlier.
- **Local ID**: IP Address.
- **Remote ID**: IP Address.
- **Local ID**: Enter the public IP of the on-premises Sophos XG Firewall.
- **Remote ID**: Input the public IP of the Azure VPN gateway that you noted earlier.
- **Local Subnet**: Enter the local subnet of 10.0.0.0 /16 (255.255.0.0)



- **Remote Subnet**: Enter the remote subnet 172.17.0.0 /16 (255.255.0.0)



1. **Advanced**: leave the default settings.
2. Upon clicking **Save**, the IPsec connection is activated.

Note: Do not click on the button under the **Connection** column as it will override the configuration settings set on the IPsec connection (**Gateway type: Respond only**). This is to avoid issues since Azure must initiate the tunnel.

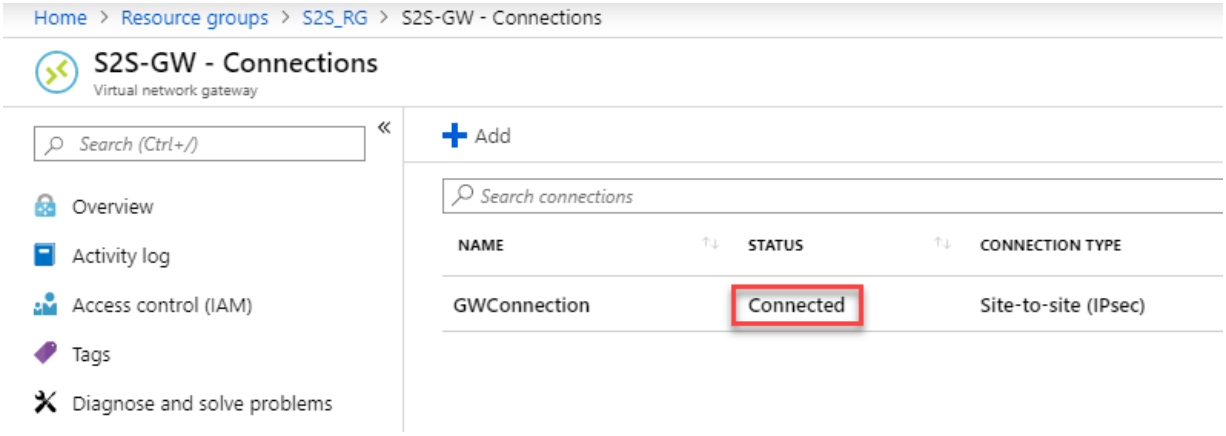
31.1.5 Task 5: Creating Azure connection.

In this task you will create a connection on your Azure Gateway to the On-Prem firewall and establish the connection.

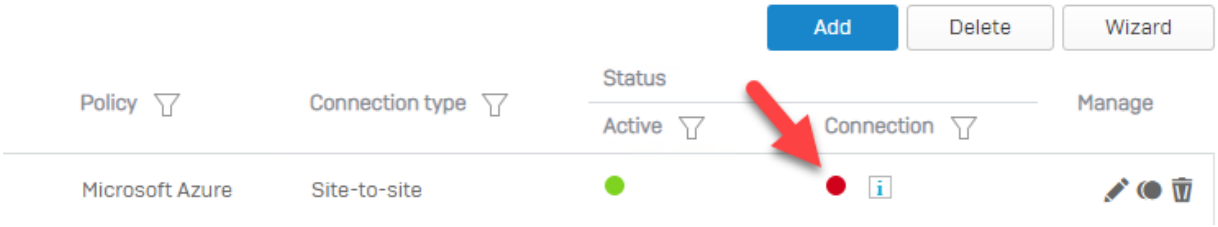
1. Click on **Resource Groups** on the **Hub Menu**.
2. Select the **S2S_RG** Resource Group.
3. Select your **S2S-GW** Gateway.
4. Click **Connections** from the S2S-GW menu.
5. Click **Add**.
6. Enter the following information in the **Add connection** blade.
   - **Name:** GWConnection
   - **Connection type:** Site-to-site (IPSec)
   - **Virtual Network Gateway:** S2S-GW
7. Click the **Local network gateway**
8. Click **Create new**.
9. Enter the following information in the **Create local network gateway** blade:
   - **Name:** OnPremGW
   - **IP address:** *Enter your IP address of your Sophos on prem firewall you recorded earlier*
   - **Address space:** 10.0.0.0/16 *(Note: This is the IP range of your On-Prem servers)*
10. Click **OK**.
11. In the **Shared key (PSK)** box enter `123456789` then click **OK**.

    Note: This key is just for this lab. In the real world you would use something with greater complexity.
12. Refresh the page and the connection should be established.

    Note: It may take 30 seconds to establish the connection. If the connection still fails to connect, return to the Sophos Portal and click



Note: If the connection still fails to connect, return to the Sophos Portal and click the red/orange icon to force the connection to be established.



WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this lab.

32 Module 2: Lab 12 - Azure Firewall

Scenario

Controlling outbound network access is an important part of an overall network security plan. For example, you may want to limit access to web sites, or the outbound IP addresses and ports that can be accessed.

One way you can control outbound network access from an Azure subnet is with Azure Firewall. With Azure Firewall, you can configure:

- Application rules that define fully qualified domain names (FQDNs) that can be accessed from a subnet.
- Network rules that define source address, protocol, destination port, and destination address.

Network traffic is subjected to the configured firewall rules when you route your network traffic to the firewall as the subnet default gateway.

32.1 Exercise 1: Deploy an Azure Firewall

32.1.1 Task 1: Lab Setup

1. In your browser, navigate to the following URL to open the ARM template:

   `https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2F`

2. Click **Create new** under the Resource Group and use the resource group name of **Test-FW-RG**
3. Select the loction of **East US**
4. Leave all the other fields as the pre-populated defaults
5. Select the I agree.... check box and click **Purchase** and wait for the deployment to complete.

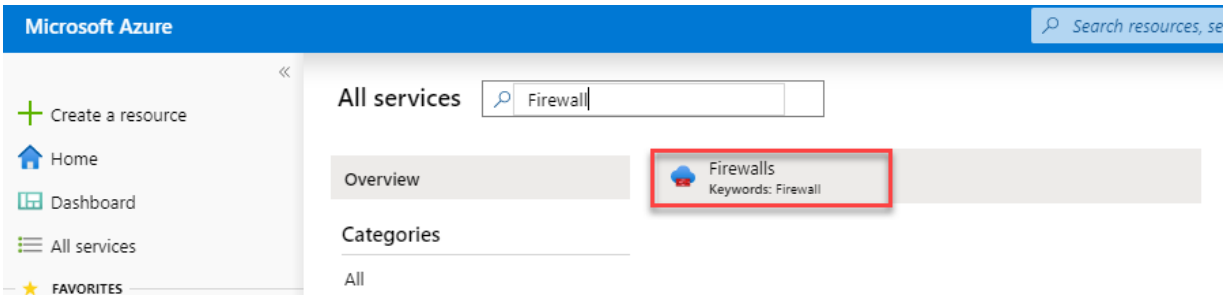
   This lab setup template will set up the following resources for the lab

Name	Type	Location
azureFirewalls-ip	Public IP address	East US
Firewall-route	Route table	East US
Srv-Jump	Virtual machine	East US
Srv-Jump_OsDisk	Disk	East US
srv-jump121	Network interface	East US
Srv-Jump-nsg	Network security group	East US
Srv-Jump-PIP	Public IP address	East US
Srv-Work	Virtual machine	East US
Srv-Work_OsDisk_1	Disk	East US
srv-work267	Network interface	East US
Srv-Work-nsg	Network security group	East US
Test-FW-VN	Virtual network	East US

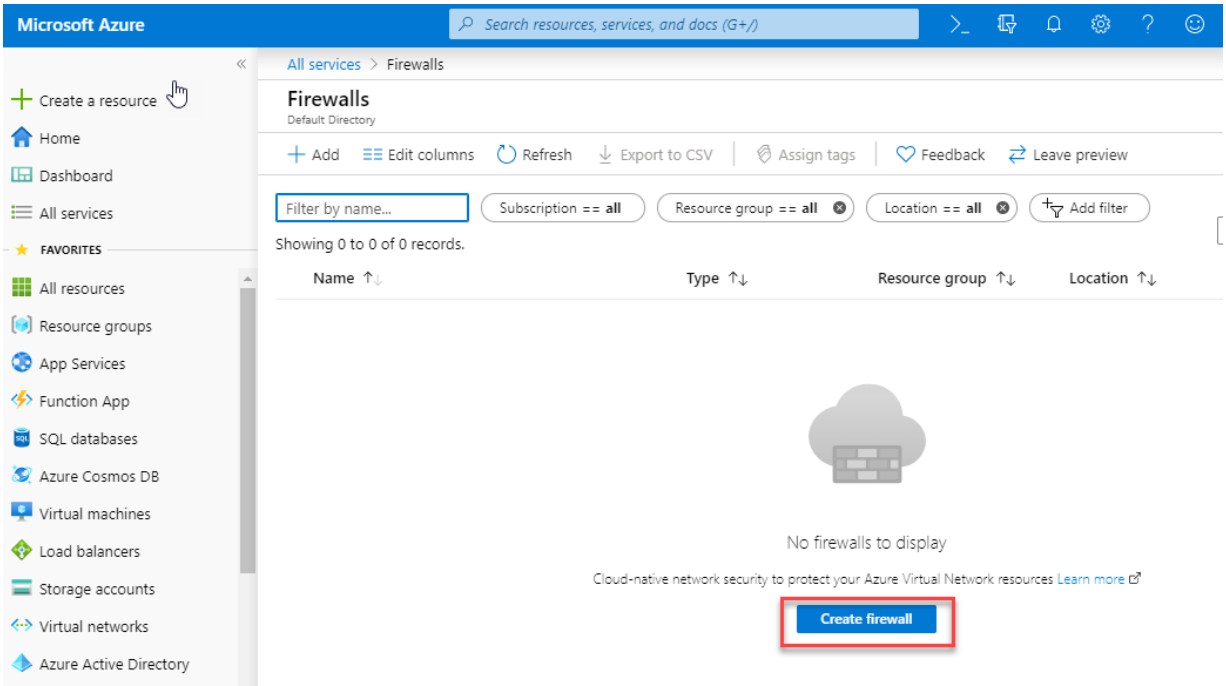
32.1.2 Task 2: Deploy the firewall

In this task you will deploy the Azure firewall into the VNet.

1. In the Azure portal, click **All services** and search for and select **Firewalls**.



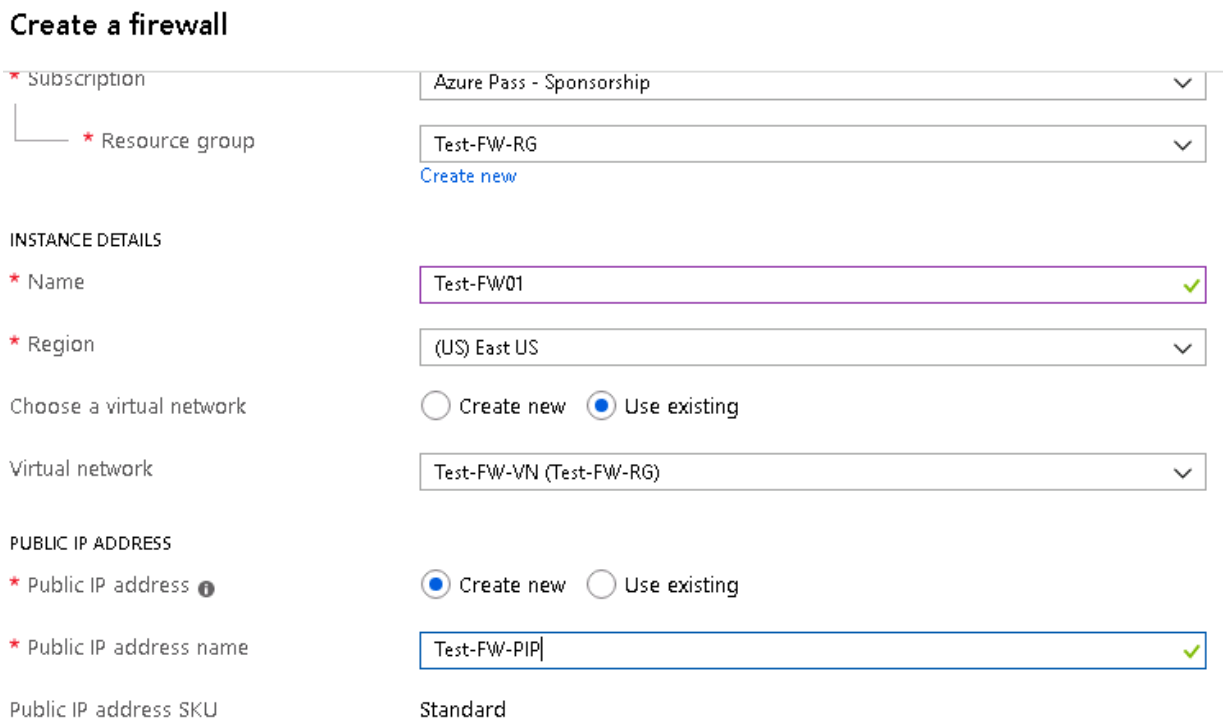
2. On the **Firewalls** blade click **Create firewall**.



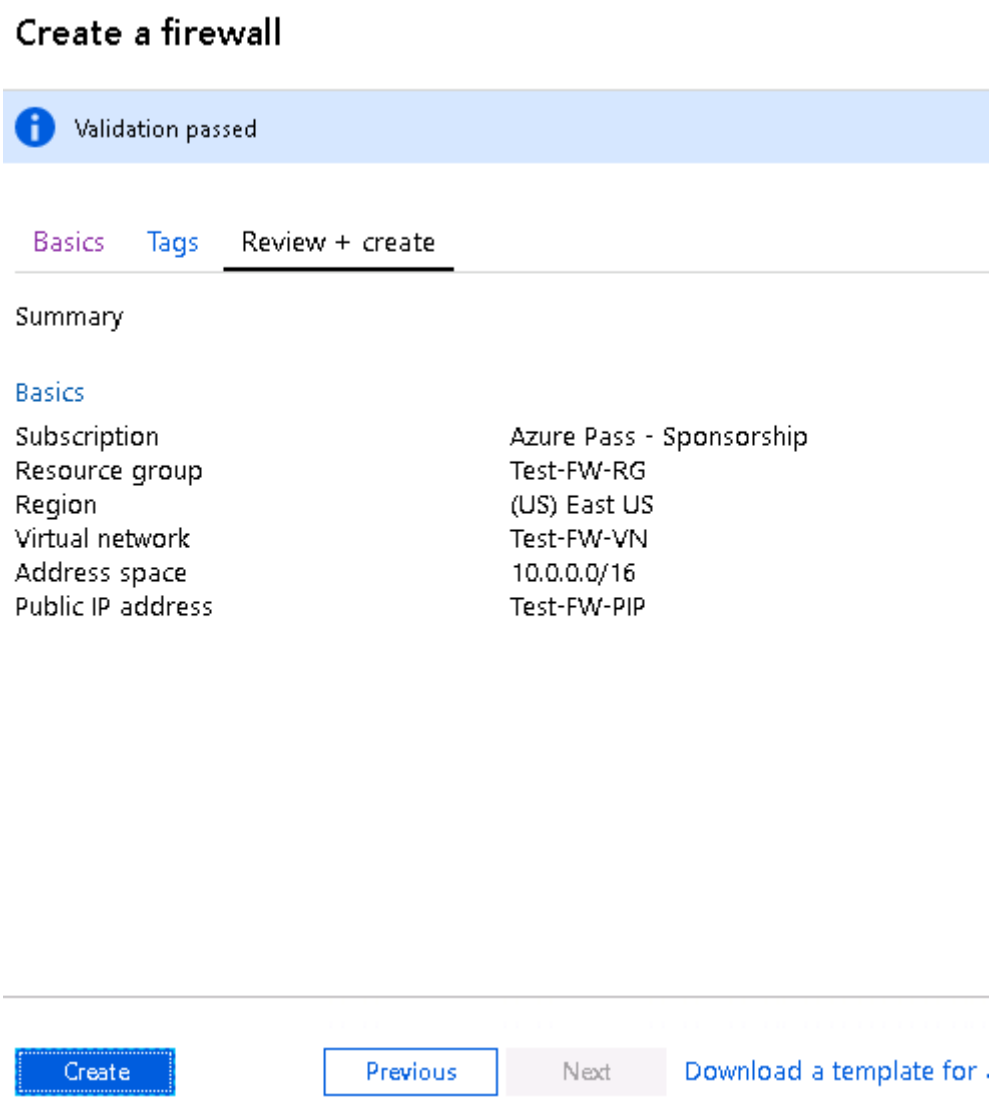
3. On the **Create a Firewall** blade, use the following table to configure the firewall:

Setting	Value
Subscription	*your subscription*
Resource group	**Use existing**: Test-FW-RG
Name	Test-FW01
Location	East US

Setting	Value
Choose a virtual network	**Use existing**: Test-FW-VN
Public IP address	**Add new**. **TEST-FW-PIP** The Public IP address must be the Standard SKU type.

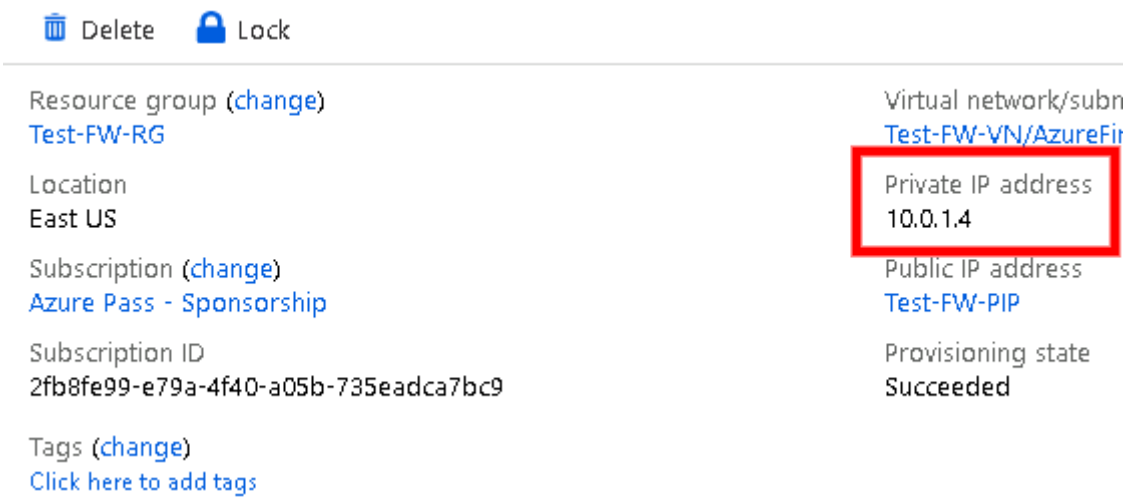


4. Click **Review + create**.
5. Review the summary, and then click **Create** to create the firewall.



This will take a few minutes to deploy.

6. After the deployment completes, go to the **Test-FW-RG** resource group, and click the **Test-FW01** firewall.
7. Make a note of the **Private IP** address. You'll use it later when you create the default route.



32.1.3 Task 3: Create a default route

For the **Workload-SN** subnet, configure the outbound default route to go through the firewall.

1. From the Azure portal home page, click **All services**.
2. Under **Networking**, click **Route tables**.
3. Click **Add**.
4. For **Name**, type **Firewall-route**.
5. For **Subscription**, select your subscription.
6. For **Resource group**, select **Use existing**, and select **Test-FW-RG**.
7. For **Location**, select **East US**.
8. Click **Create**.
9. Click **Refresh**, and then click the **Firewall-route** route table.
10. Click **Subnets** > **Associate**.
11. Click **Virtual network** > **Test-FW-VN**.
12. For **Subnet**, click **Workload-SN**. Make sure that you select only the **Workload-SN** subnet for this route, otherwise your firewall won't work correctly.
13. Click **OK**.
14. Click **Routes** > **Add**.
15. For **Route name**, type **FW-DG**.
16. For **Address prefix**, type **0.0.0.0/0**
17. For **Next hop type**, select **Virtual appliance**.

    Azure Firewall is actually a managed service, but virtual appliance works in this situation.
18. For **Next hop address**, type the private IP address for the firewall that you noted previously.
19. Click **OK**.

32.1.4 Task 4: Configure an application rule

In this task you will create an application rule that allows outbound access to `msn.com`.

1. Open the **Test-FW-RG** resource group and click the **Test-FW01** firewall.
2. On the **Test-FW01** page, under **Settings** section, click **Rules**.
3. Click the **Application rule collection** tab.
4. Click **Add application rule collection**.
5. For **Name**, type **App-Coll01**.
6. For **Priority**, type **200**.
7. For **Action**, select **Allow**.
8. Under **Rules**, **Target FQDNs**, for **Name**, type **AllowGH**.
9. For **Source Addresses**, type **10.0.2.0/24**.
10. For **Protocol:port**, type **http, https**.
11. For **Target FQDNS**, type **msn.com**
12. Click **Add**.

Azure Firewall includes a built-in rule collection for infrastructure FQDNs that are allowed by default. These FQDNs are specific for the platform and can't be used for other purposes.

32.1.5 Task 5: Configure a network rule

In this task you will create a network rule that allows outbound access to two IP addresses on port 53 (DNS).

1. Click the **Network rule collection** tab.
2. Click **Add network rule collection**.
3. For **Name**, type **Net-Coll01**.
4. For **Priority**, type **200**.
5. For **Action**, select **Allow**.
6. Under **Rules** in the **IP Addresses** section, for **Name**, type **AllowDNS**.
7. For **Protocol**, select **UDP**.
8. For **Source Addresses**, type **10.0.2.0/24**.
9. For Destination address, type **209.244.0.3,209.244.0.4**
10. For **Destination Ports**, type **53**.
11. Click **Add**.

32.1.6 Task 6: Change the primary and secondary DNS address for the Srv-Work network interface

For testing purposes in this tutorial, you configure the primary and secondary DNS addresses. This isn't a general Azure Firewall requirement.

1. From the Azure portal, open the **Test-FW-RG** resource group.
2. Click the network interface for the **Srv-Work** virtual machine.
    1. Under **Settings**, click **Networking**.
    2. Select the NIC
3. Under **DNS servers**, click **Custom**.
4. Type **209.244.0.3** in the **Add DNS server** text box, and **209.244.0.4** in the next text box.
5. Click **Save** and wait until it has successfully saved..
6. Restart the **Srv-Work** virtual machine.

32.1.7 Task 7: Test the firewall

In this task you will test the firewall to confirm that it works as expected.

1. From the Azure portal, review the network settings for the **Srv-Work** virtual machine and note the private IP address.
2. Connect to the **Srv-Jump** virtual machine using RDP, and from there open a remote desktop connection to the **Srv-Work** private IP address.
    - **Username**: localadmin
    - **Password**: Pa55w.rd1234
3. Open Internet Explorer and browse to `https://www.msn.com`
4. Click **OK** > **Close** on the security alerts.

    You should see the MSN home page.
5. Browse to `https://msn.com`
    - You should be blocked by the firewall.
    - So now you've verified that the firewall rules are working:
        - You can browse to the one allowed FQDN, but not to any others.
        - You can resolve DNS names using the configured external DNS server.
6. To troubleshoot, go to **Network Watcher**, choose **Topology** then choose **TEST-FW-RG** to see the overall network diagram.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this lab.

33 Module 2 - Implement Platform Protection

33.1 Lab 13 - Secure Admin Access

SSH is an encrypted connection protocol that allows secure sign-ins over unsecured connections. SSH is the default connection protocol for Linux VMs hosted in Azure. Although SSH itself provides an encrypted connection, using passwords with SSH connections still leaves the VM vulnerable to brute-force attacks or guessing of passwords. A more secure and preferred method of connecting to a VM using SSH is by using a public-private key pair, also known as SSH keys.

- The public key is placed on your Linux VM, or any other service that you wish to use with public-key cryptography.
- The private key on you local system is used by an SSH client to verify your identity when you connect to your Linux VM. Protect this private key. Do not share it.
- Depending on your organization's security policies, you can reuse a single public-private key pair to access multiple Azure VMs and services. You do not need a separate pair of keys for each VM or service you wish to access.

Your public key can be shared with anyone, but only you (or your local security infrastructure) should possess your private key.

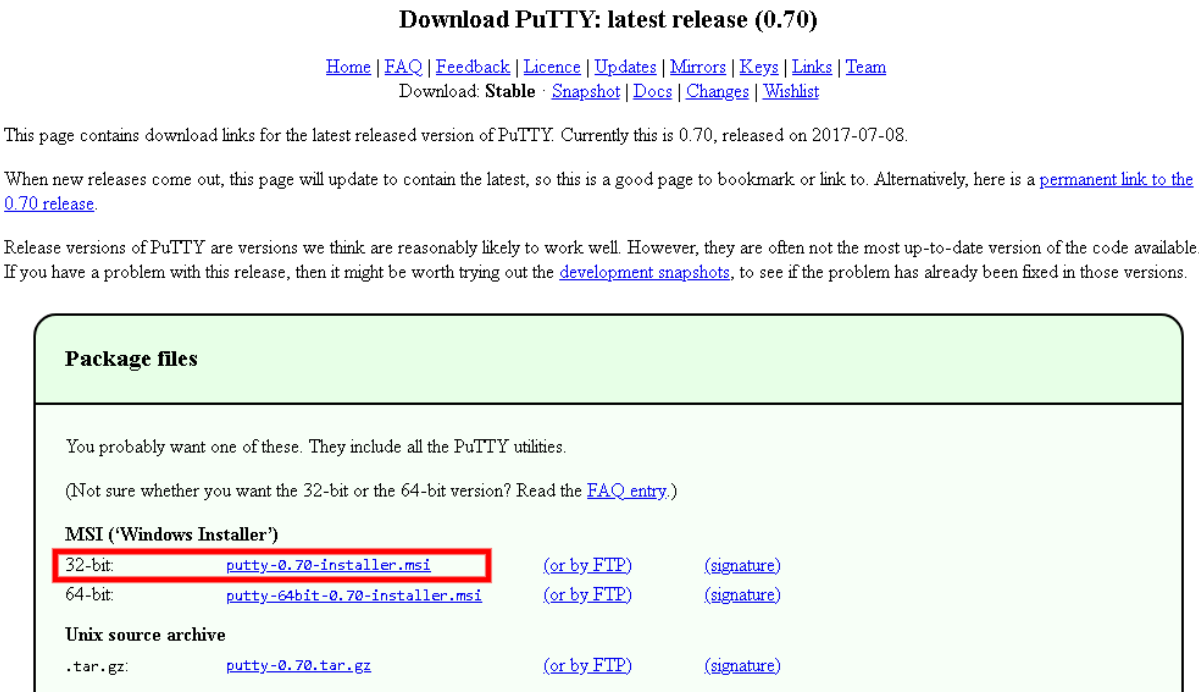
33.2 Exercise 1: Deploy and connect to an Azure VM securely.

33.2.1 Task 1: Create SSH keys with PuTTYgen

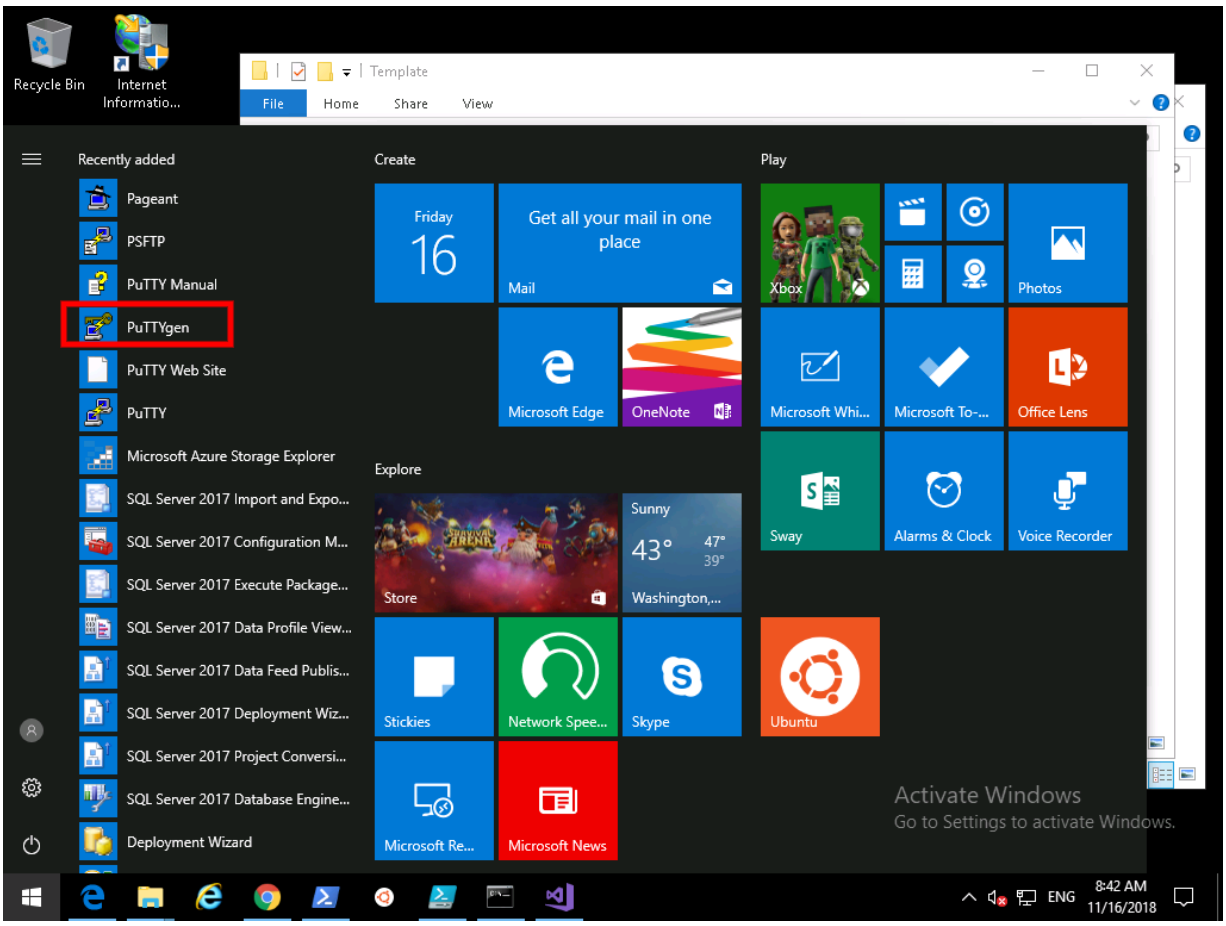
1. Open a browser and navigate to the following URL:

   `http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html`

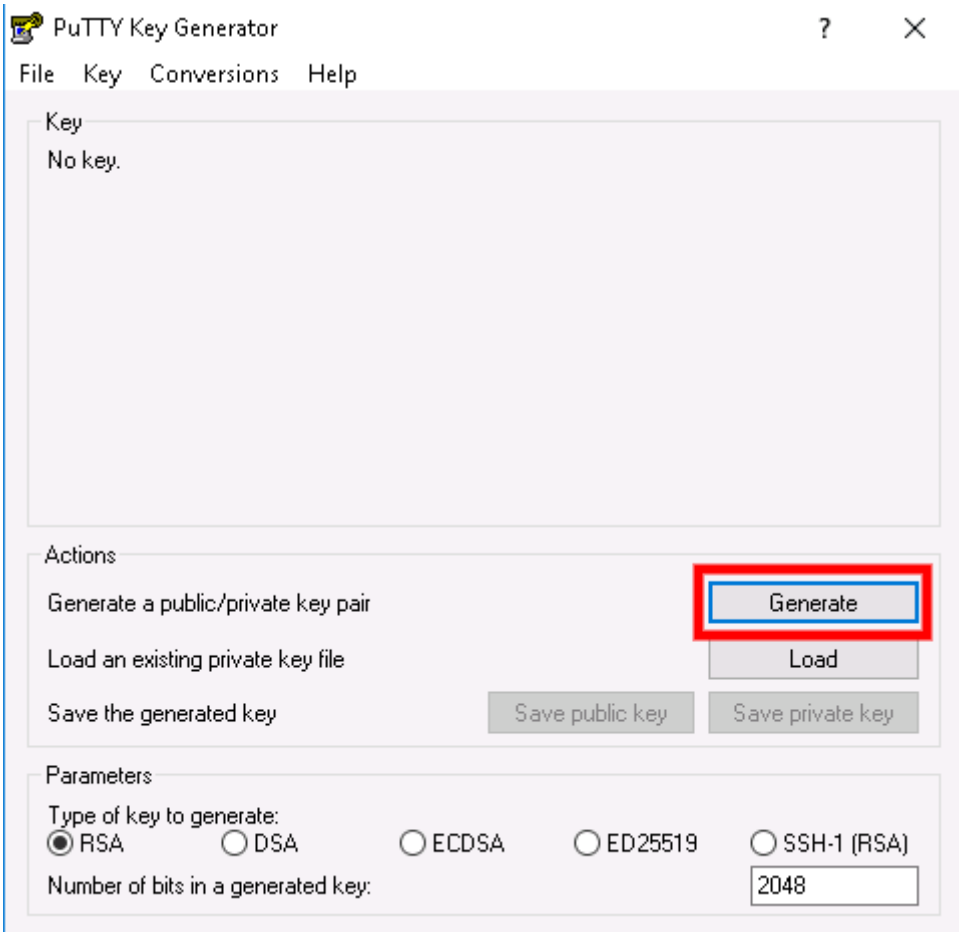
2. Download and install the **Putty Installer**.



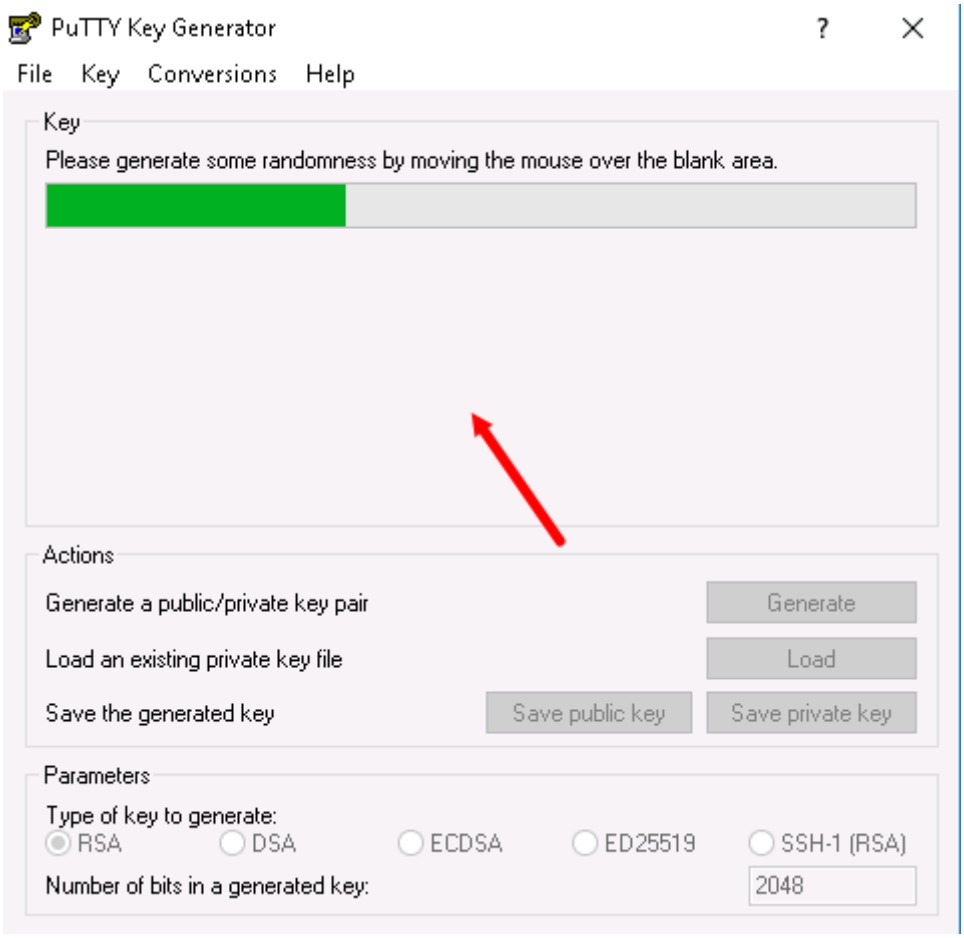
3. Click **Start** and navigate to **PuTTYgen**.



4. Click Generate. By default PuTTYgen generates a 2048-bit SSH-2 RSA key.



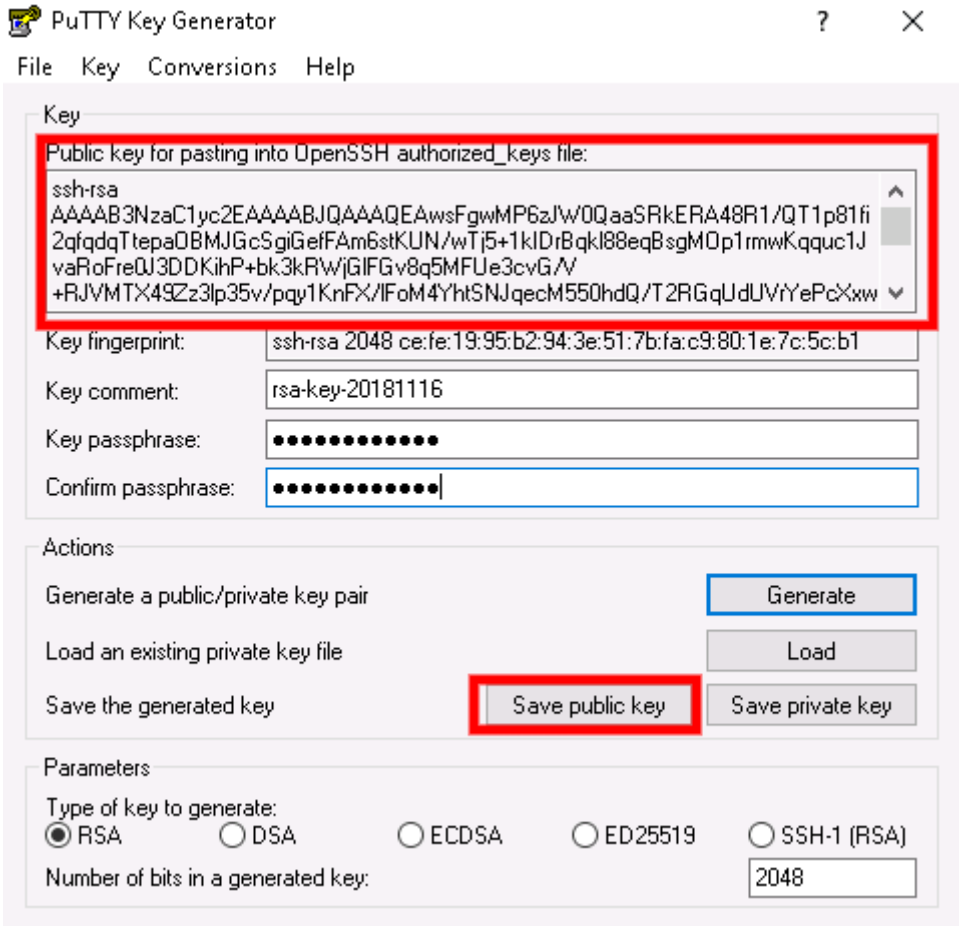
5. Move the mouse around in the blank area to provide randomness for the key.



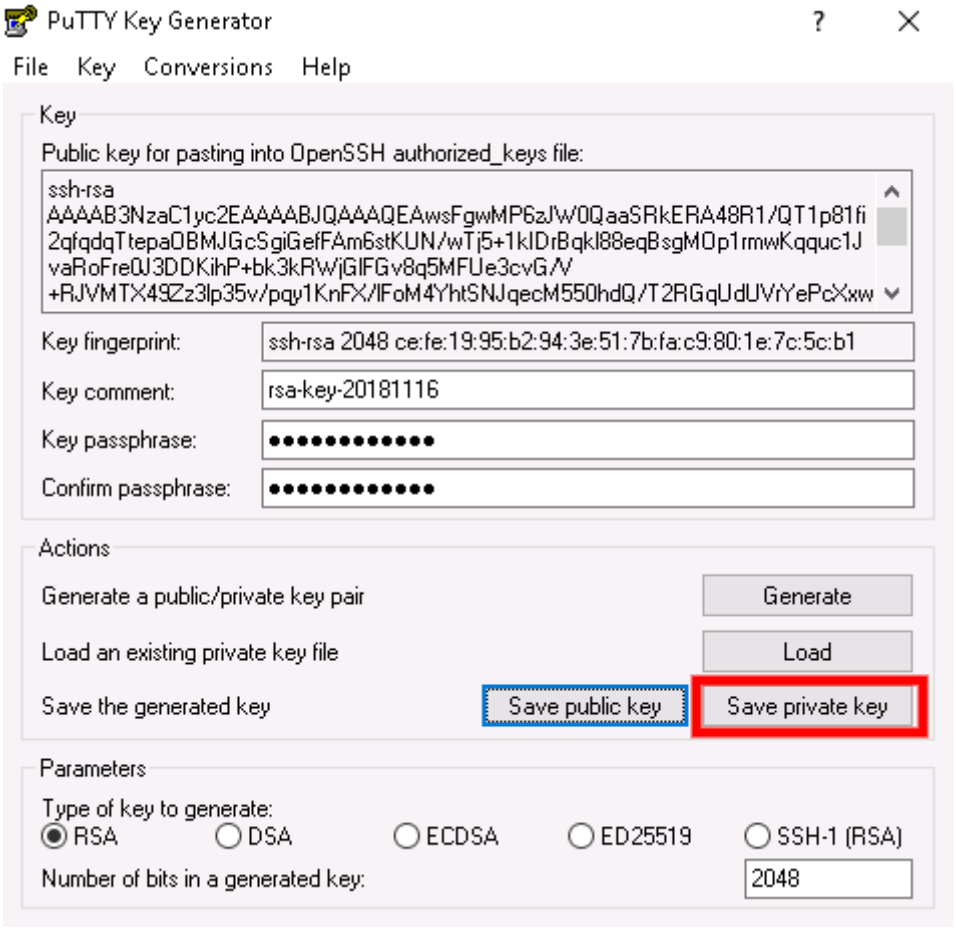
6. After the public key is generated, optionally enter and confirm a passphrase. You will be prompted for the passphrase when you authenticate to the VM with your private SSH key. Enter **Pa55w.rd1234** as the passphrase.

   Without a passphrase, if someone obtains your private key, they can log in to any VM or service that uses that key. We recommend you create a passphrase. However, if you forget the passphrase, there is no way to recover it.

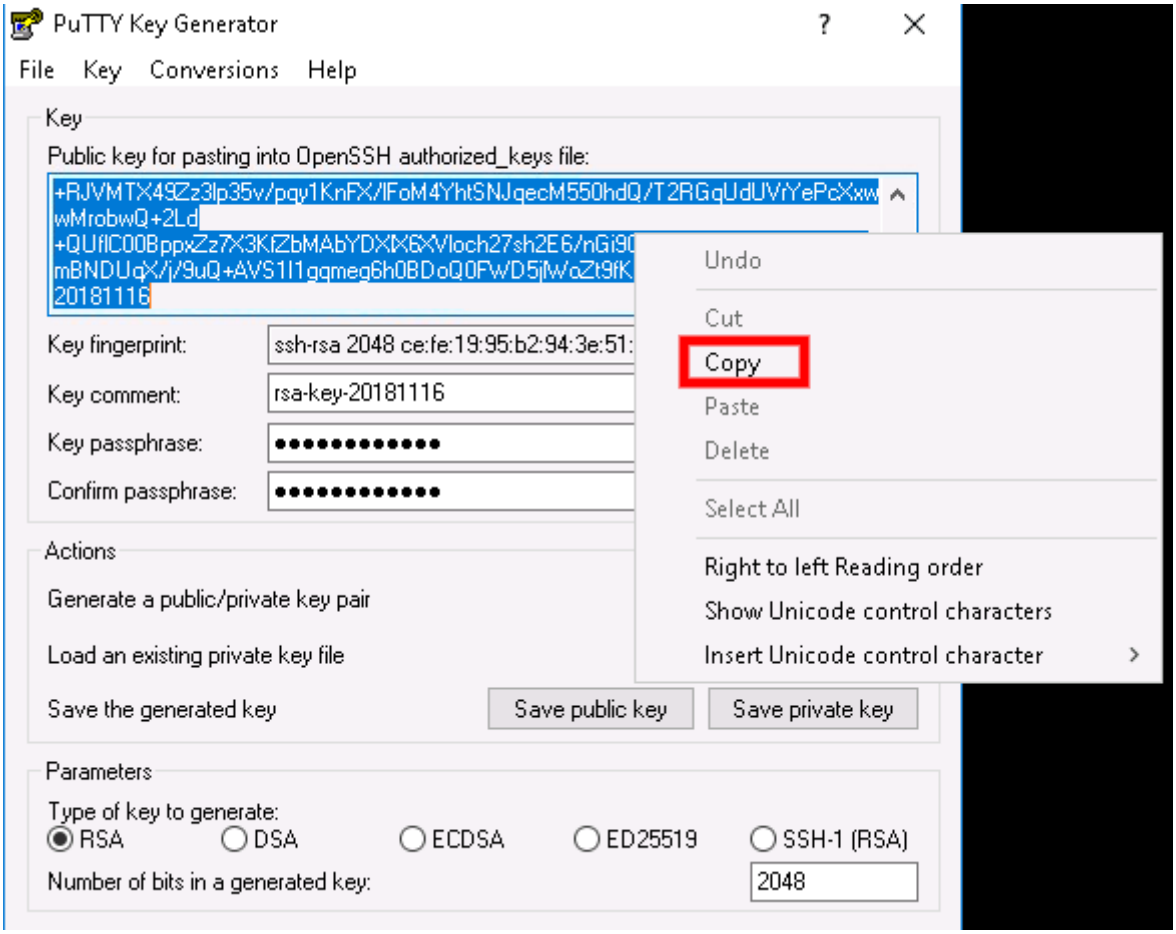
7. The public key is displayed at the top of the window. You can copy this entire public key and then paste it into the Azure portal or an Azure Resource Manager template when you create a Linux VM. Save the public key to a location on your machine and call the file **public**:



8. Save the private key to the same location but with the filename **private**.

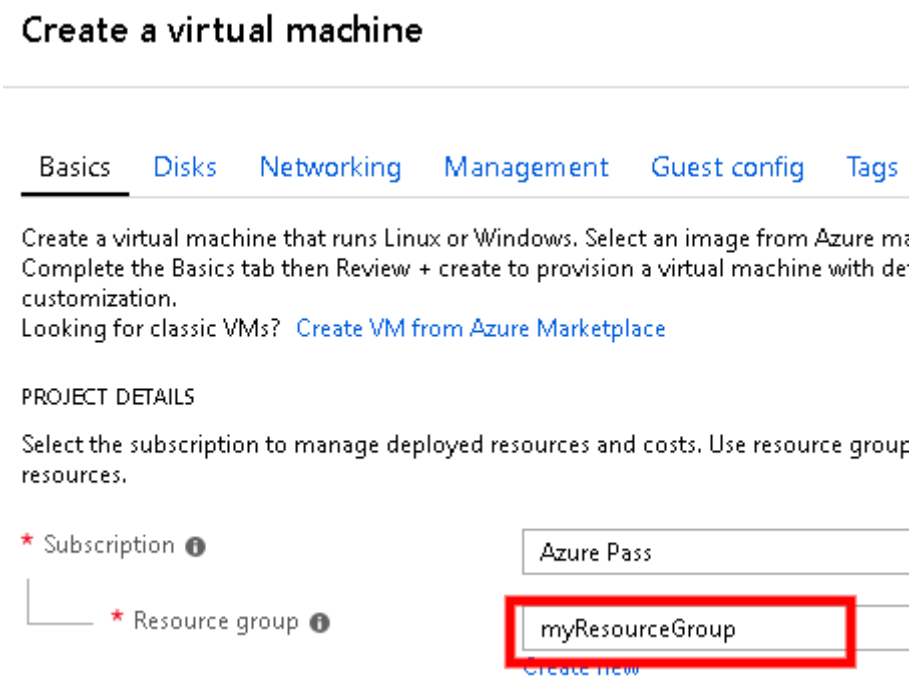


9. Highlight and copy the public key from the top window.

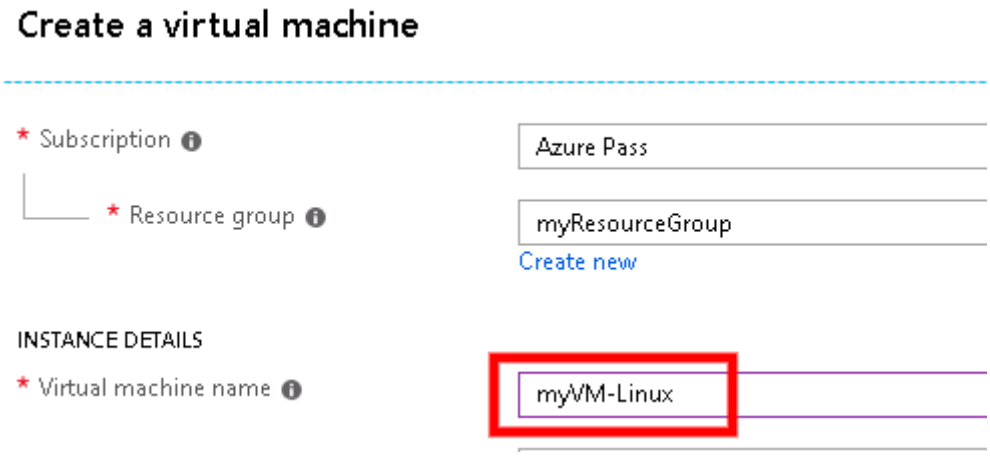


33.2.2 Task 2: Create a Linux virtual machine in the Azure portal

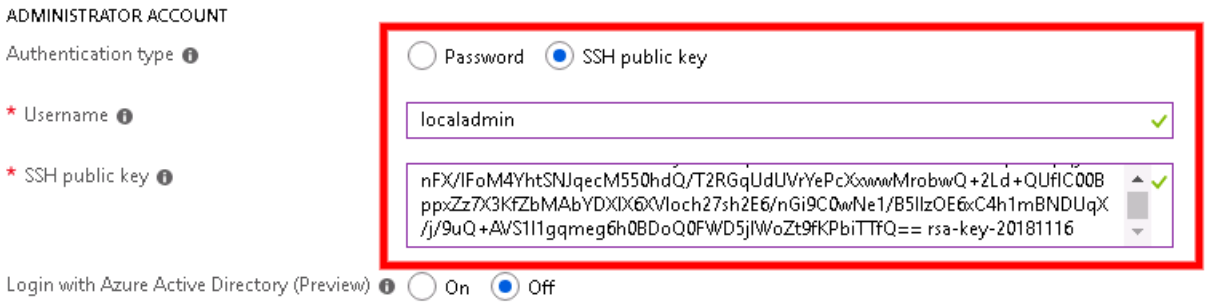
1. Navigate back to the **Azure Portal**.
2. Choose **Create a resource** in the upper left corner of the Azure portal.
3. In the search box above the list of Azure Marketplace resources, search for and select **Ubuntu Server 18.04 LTS** by Canonical, then choose **Create**.
4. In the **Basics** tab, under **Project details**, make sure the correct subscription is selected and then choose the **Resource group** *myResourceGroup*.



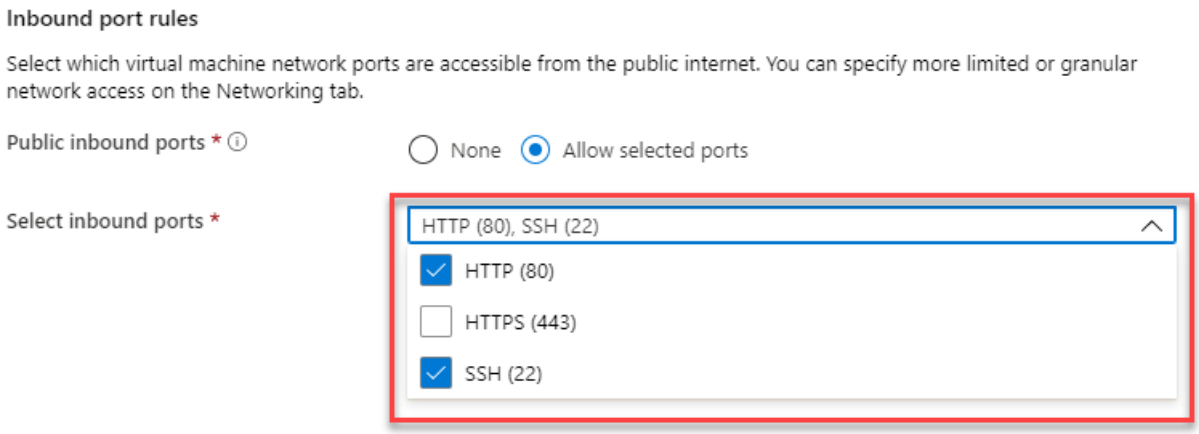
5. Under **Instance details**, type *myVM-Linux* for the **Virtual machine name** and choose *East US* for your **Region**. Leave the other defaults.



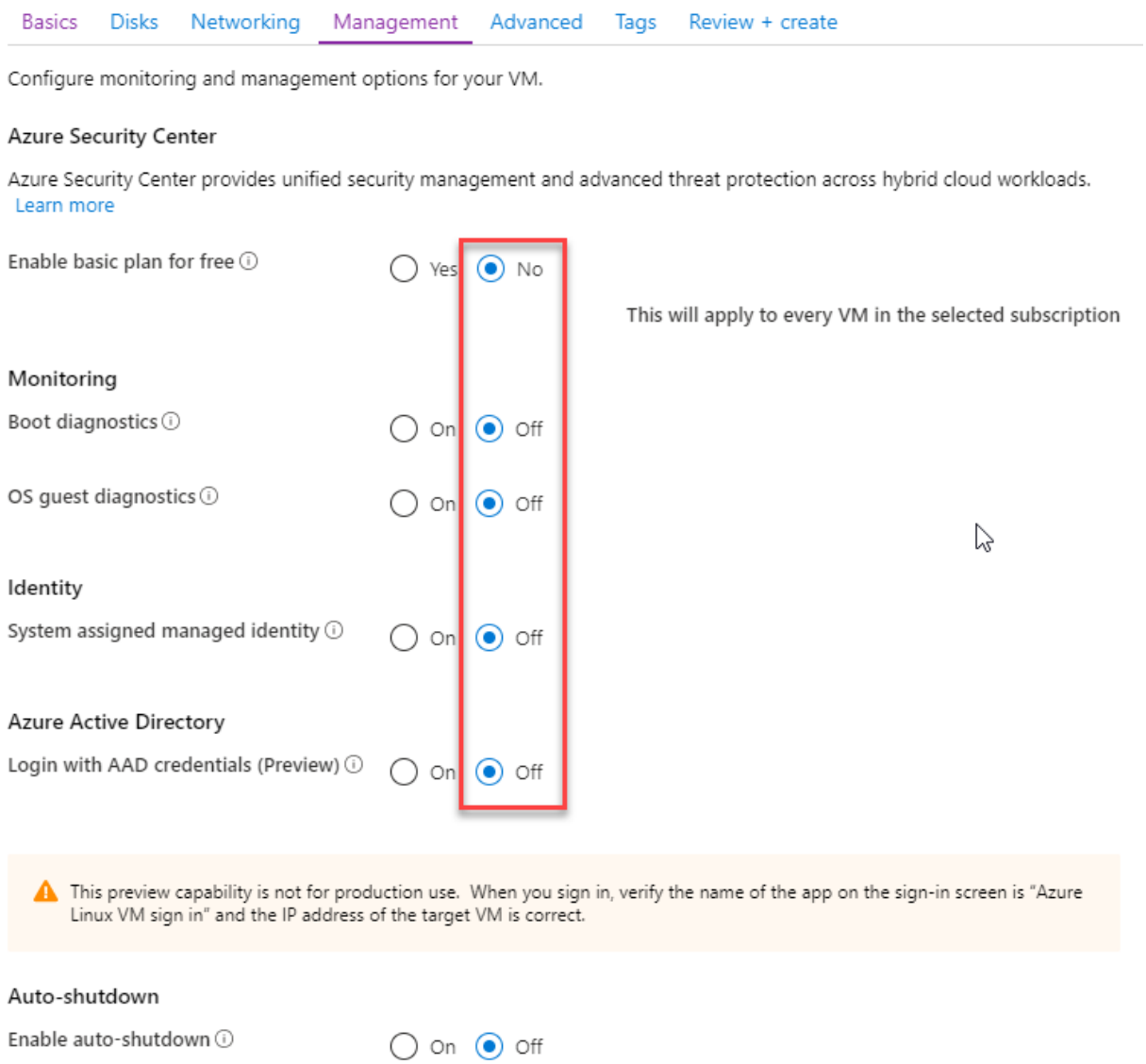
6. Under **Administrator account**, select **SSH public key**, type the user name **localadmin**, then paste your public key into the text box. Remove any leading or trailing white space in your public key.



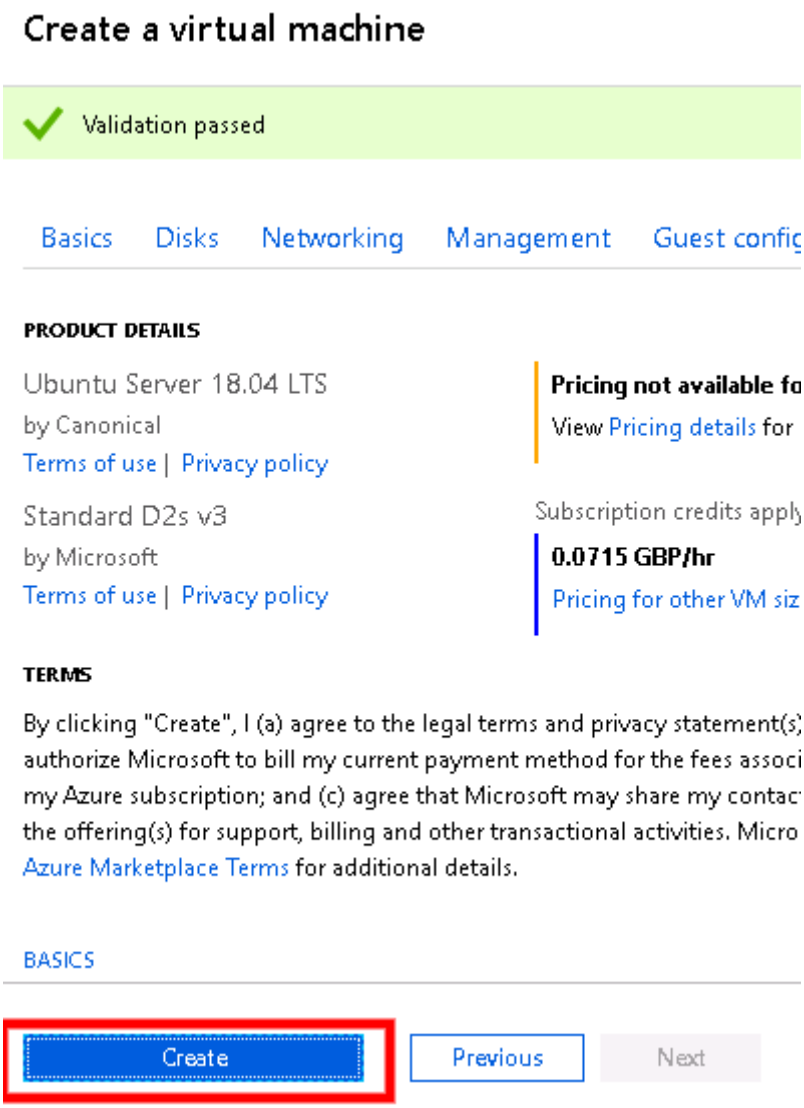
7. Under **Inbound port rules** > **Public inbound ports**, choose **Allow selected ports** and then select **SSH (22)** and **HTTP (80)** from the drop-down.



8. Click the **Management** tab and select **No** or **Off** for all options.



9. Leave the remaining defaults and then select the **Review + create** button at the bottom of the page.
10. On the **Create a virtual machine** page, you can see the details about the VM you are about to create. When you are ready, select **Create**.



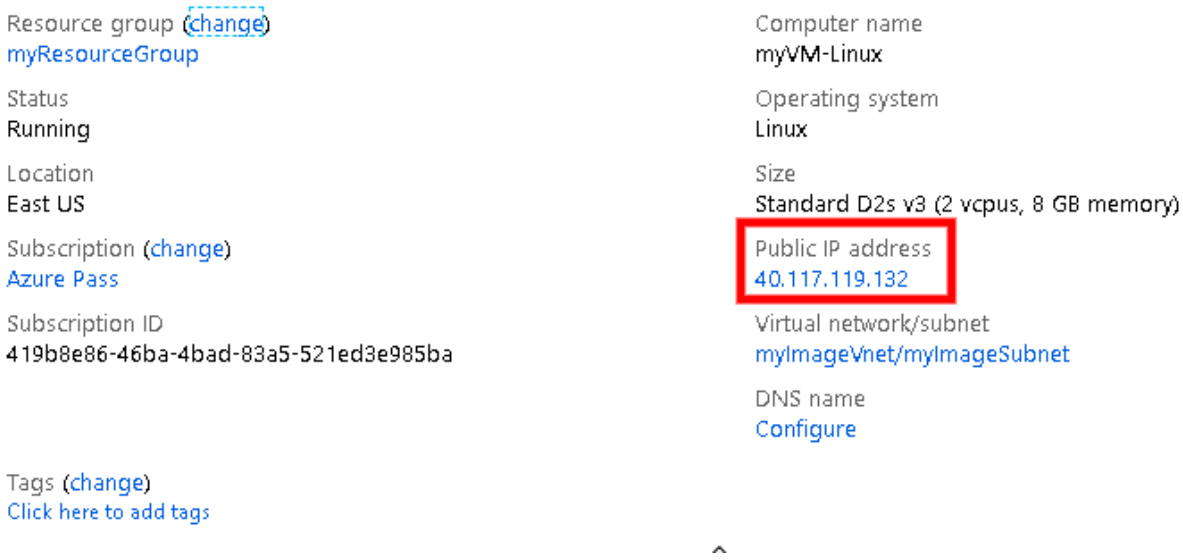
It will take a few minutes for your VM to be deployed. When the deployment is finished, move on to the next section.

33.2.3 Task 3: Connect to your VM

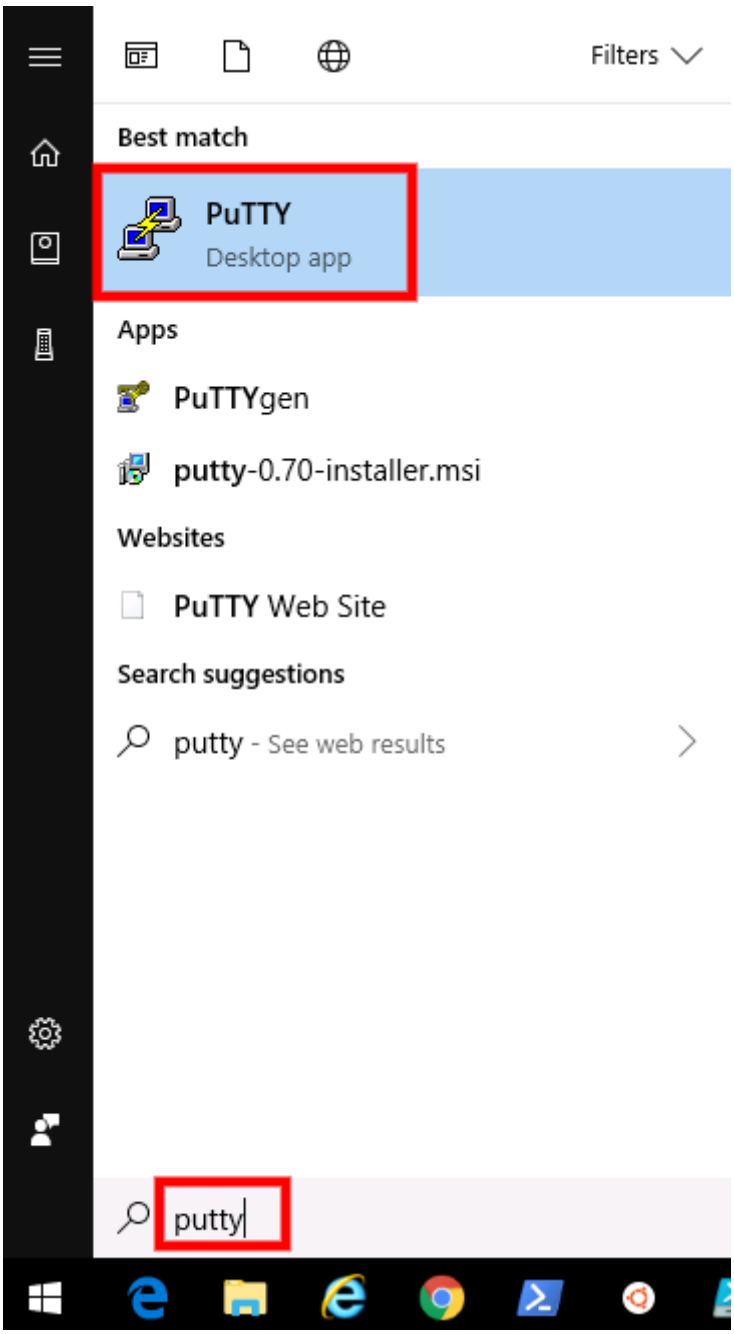
One way to make an SSH connection to your Linux VM from Windows is to use an SSH client. This is the preferred method if you have an SSH client installed on your Windows system, or if you use the SSH tools in Bash in Azure Cloud Shell. If you prefer a GUI-based tool, you can connect with PuTTY. In this task you will use PuTTY.

1. In the **Azure Portal Hub Menu** click **Virtual Machines** then select your **myVM-Linux** machine.
2. In the Overview blade, note down or copy the **Public IP Address** of your virtual machine *If this does not display, refresh your browser.*

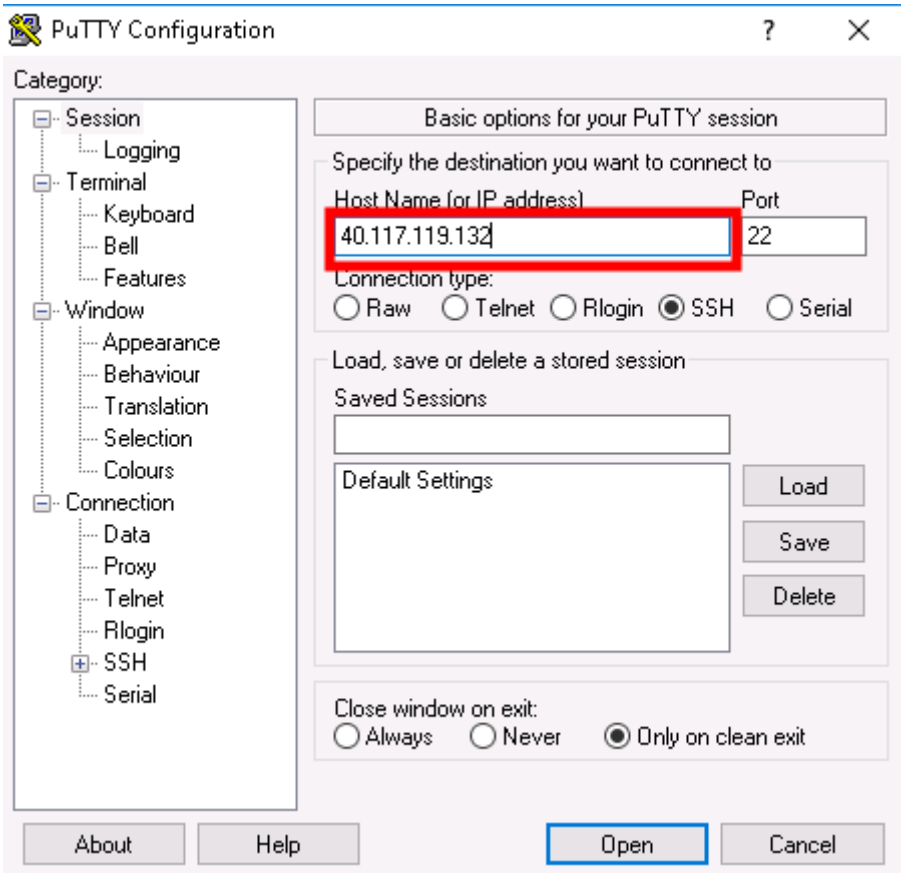
   Note: Your public IP will be different to what is shown in the screenshot.



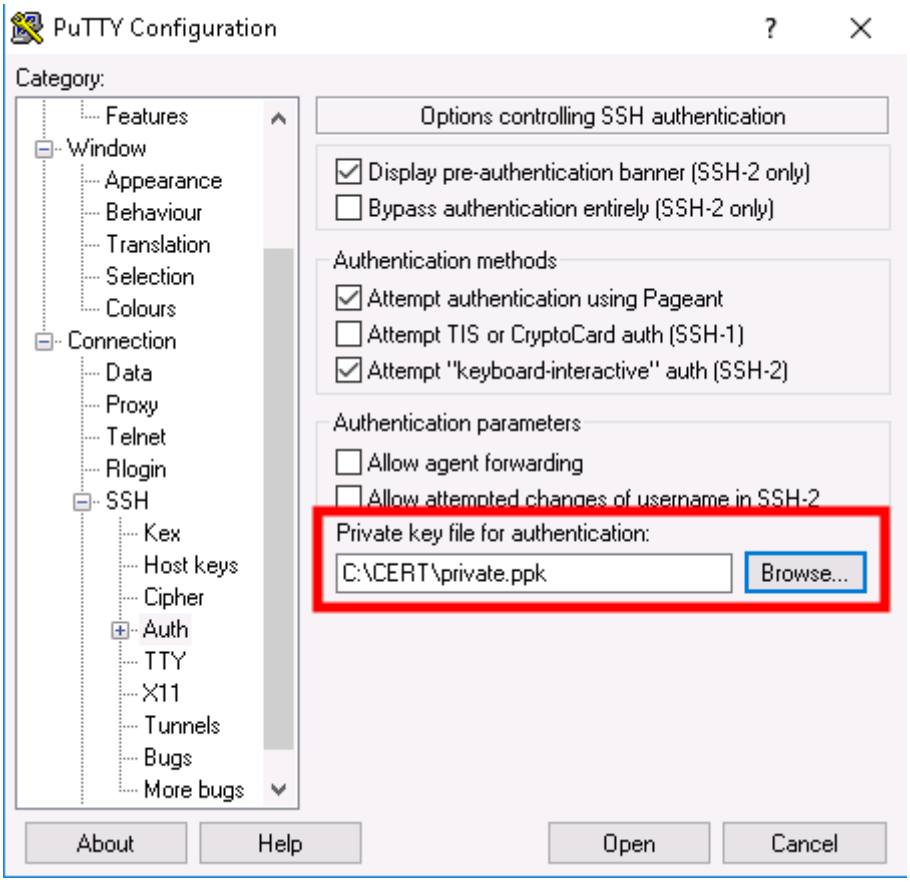
3. Start **PuTTy** by clicking the start menu and searching for PuTTY.



4. Type in or paste in your Public IP Address of your Linux Azure Linux VM:



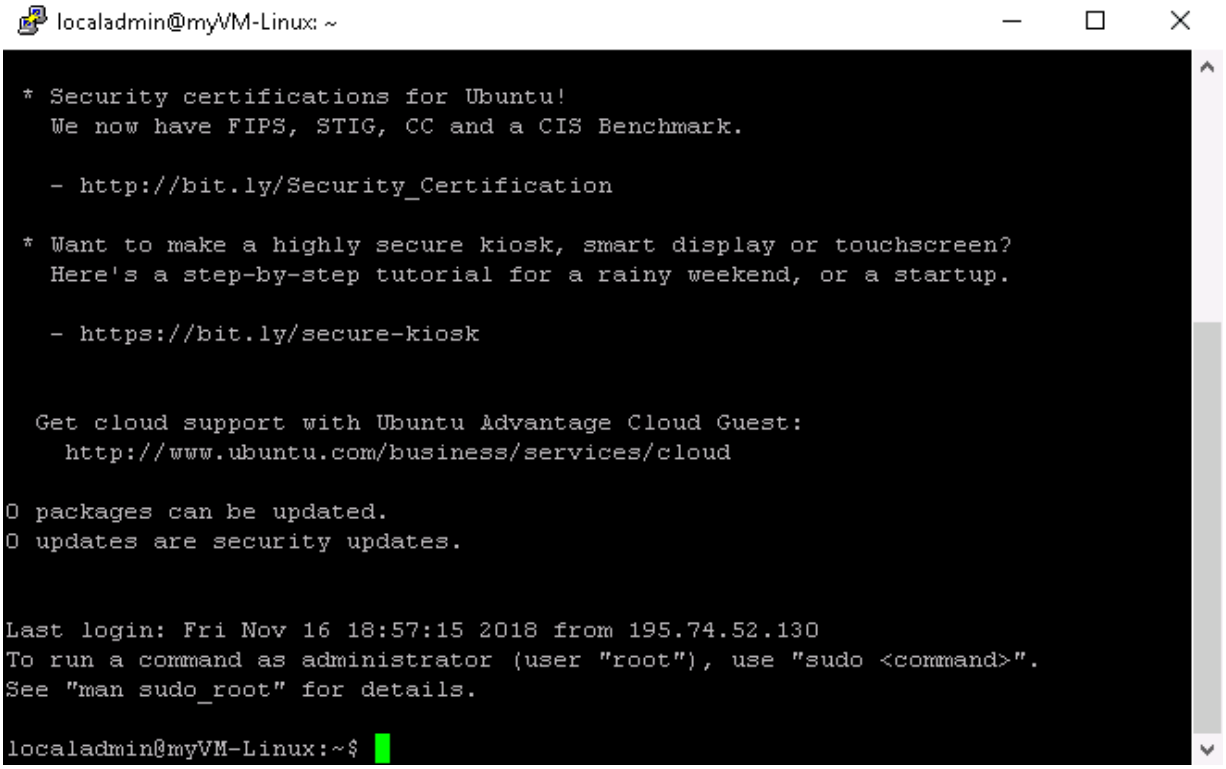
5. Select the **Connection** > **SSH** > **Auth** category. Browse to and select your PuTTY private key (.ppk file):



6. Click **Open** to connect to your VM.
7. Click **Yes** to continue on the pop up.

8. On the `login as` screen enter **localadmin** and press **Enter** then enter the password Pa55w.rd1234 and press **Enter**. ***Note:*** *As you type the password the curser will not move*

9. You are now logged into the Linux VM hosted in Azure.



WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this Lab.

34 Module 2: Lab 14 - Azure Bastion

Scenario

The Azure Bastion service is a new fully platform-managed PaaS service that you provision inside your virtual network. It provides secure and seamless RDP/SSH connectivity to your virtual machines directly in the Azure portal over SSL. When you connect via Azure Bastion, your virtual machines do not need a public IP address.

Bastion provides secure RDP and SSH connectivity to all VMs in the virtual network in which it is provisioned. Using Azure Bastion protects your virtual machines from exposing RDP/SSH ports to outside world while still providing secure access using RDP/SSH. With Azure Bastion, you connect to the virtual machine directly from the Azure portal. You don't need an additional client, agent, or piece of software.

34.1 Exercise 1: Implement Azure Bastion

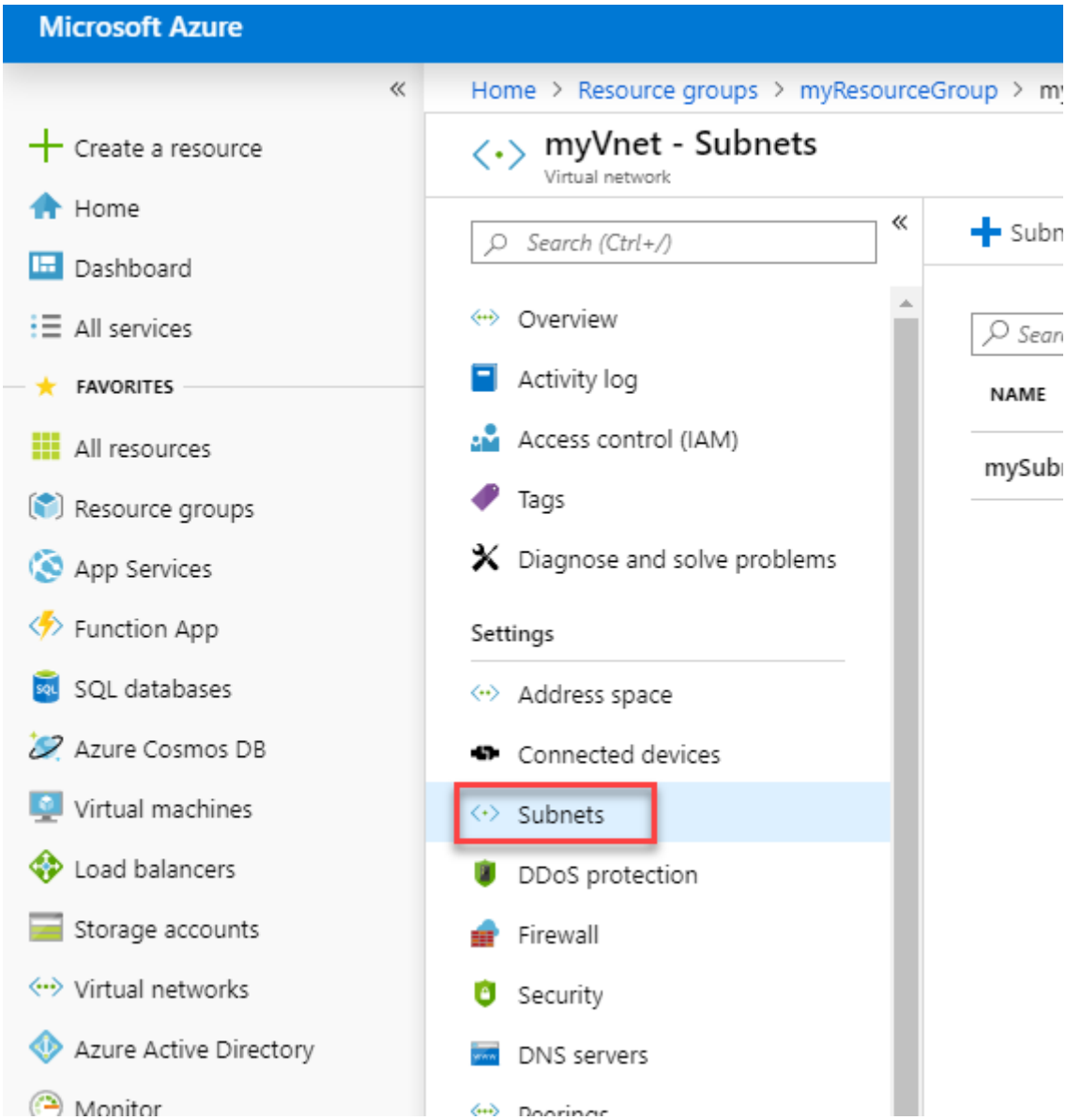
34.1.1 Task 1: Enable Azure Bastion on your subscription

1. Open a browser and navigate to `https://portal.azure.com`
2. Open the **Cloud Shell** in PowerShell mode and create storage if required.
3. Run the following 2 commands to create a Resource Group and Virtual Machine and VNet to test the Azure Bastion service.

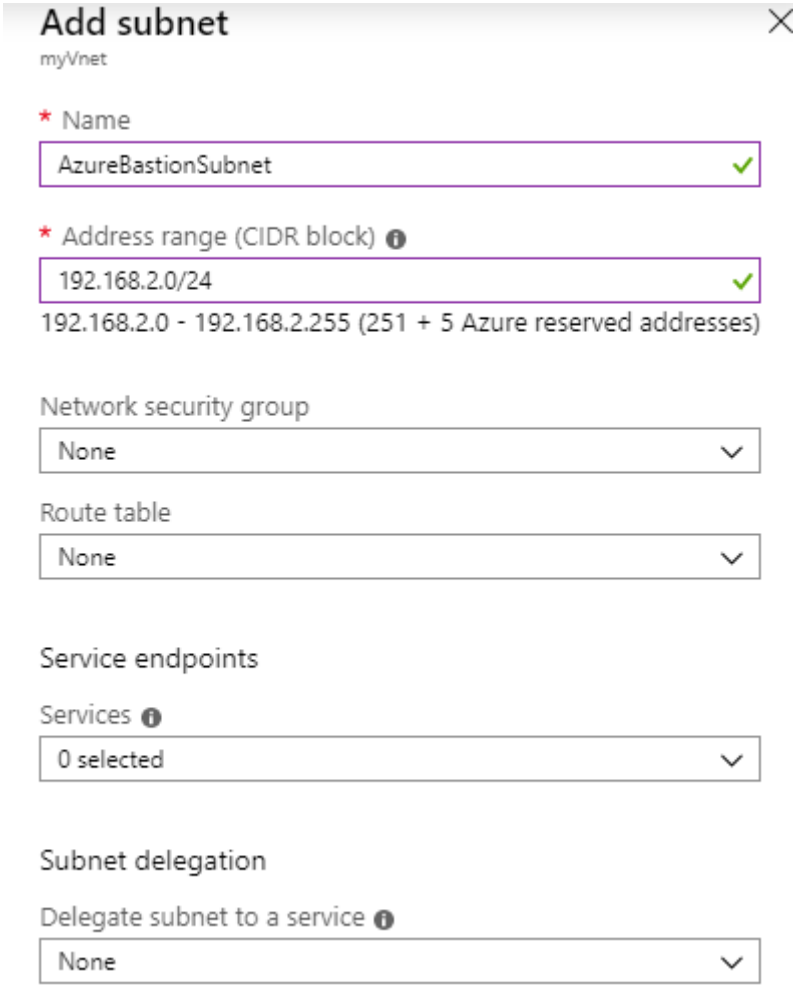
```
New-AzResourceGroup -Name myResourceGroup -Location "East US"
New-AzVm -ResourceGroupName "myResourceGroup" -Name "myVM" -Location "East  US" -VirtualNetworkNam
```

Note: The VM is being created without a Public IP Address.

4. When prompted enter **LocalAdmin** and **Pa55w.rd1234** for the credentials.
5. In the Azure Portal navigate to your **myVnet** and click **Subnets**.

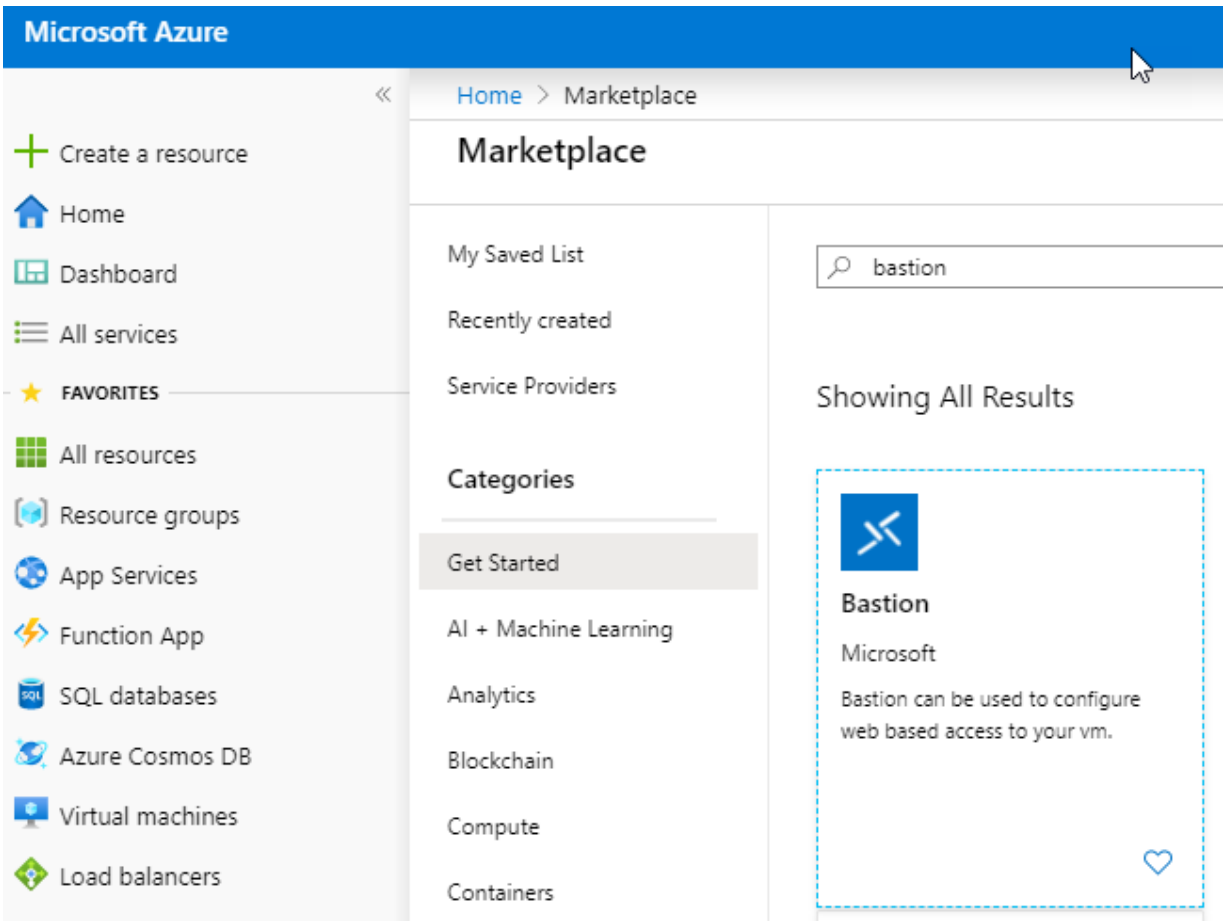


6. Click + **Subnet** and create a subnet with the following details:
   - Name: **AzureBastionSubnet** *(Note this is case sensitive)*
   - Address Range: **192.168.2.0/24**
7. Click **OK**.

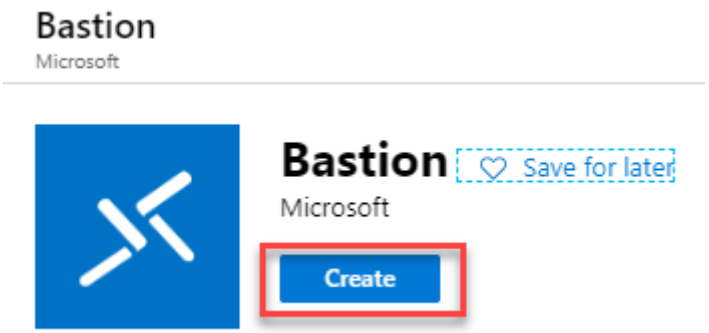


34.1.2 Task 2: Create a bastion host

1. From the home page in the **Azure portal** click **+ Create a resource**.
2. On the **New** page, in the *Search the Marketplace* field, type **Bastion**, then click **Enter** to get to the search results.
3. From the results, click **Bastion**.



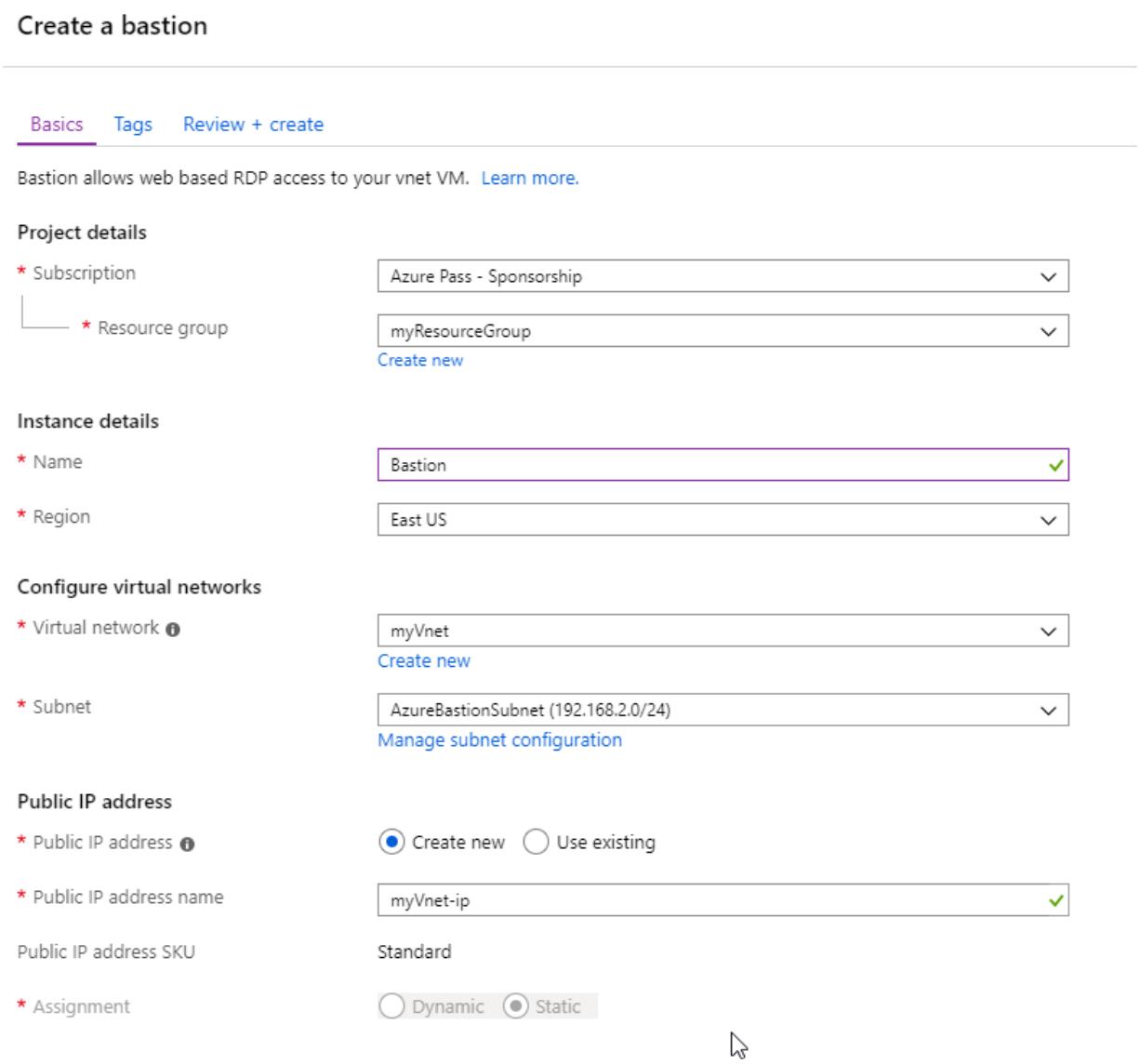
4. On the **Bastion** page, click **Create** to open the **Create a bastion** page.



5. On the **Create a bastion** page, configure a new Bastion resource. Specify the configuration settings below.

   - **Subscription**: Select your Subscription
   - **Resource Group**: myResourceGroup
   - **Name**: Bastion
   - **Region**: East US
   - **Virtual network**: **myVnet**
   - **Subnet**: **AzureBastionSubnet**
   - **Public IP address**: The public IP of the Bastion resource on which RDP/SSH will be accessed (over port 443). Create a new public IP, or use an existing one. The public IP address must be in the same region as the Bastion resource you are creating.
   - **Public IP address name**: Leave as default
   - **Public IP address SKU**: Prepopulated by default to **Standard**. Azure Bastion uses/supports only the Standard Public IP SKU.
   - **Assignment**: Prepopulated by default to **Static**.

1. When you have finished specifying the settings, click **Review + Create**. This validates the values. Once validation passes, you can begin the creation process.

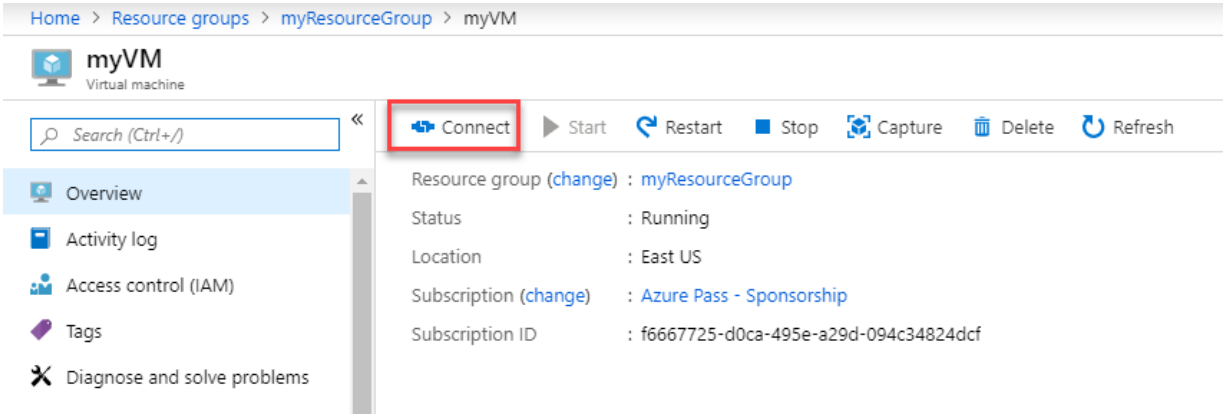


2. On the Create a bastion page, click **Create**.
3. You will see a message letting you know that your deployment is underway. Status will display on this page as the resources are created. It takes about 5 mins for the Bastion resource to be created and deployed.

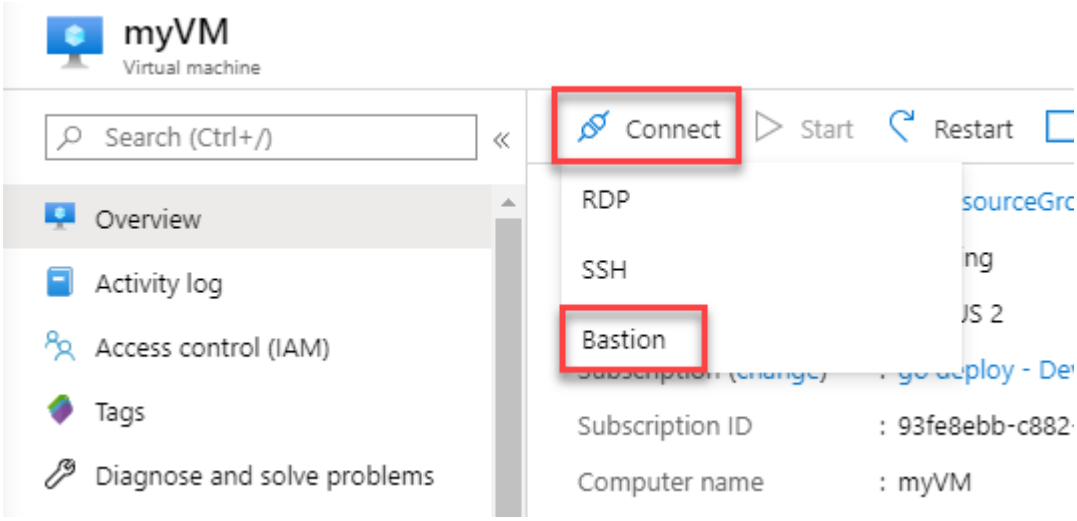
34.1.3 Task 3: Connect to a VM using a bastion host

If you create a bastion host in the portal by using an existing VM, various settings will automatically default corresponding to your virtual machine and/or virtual network.

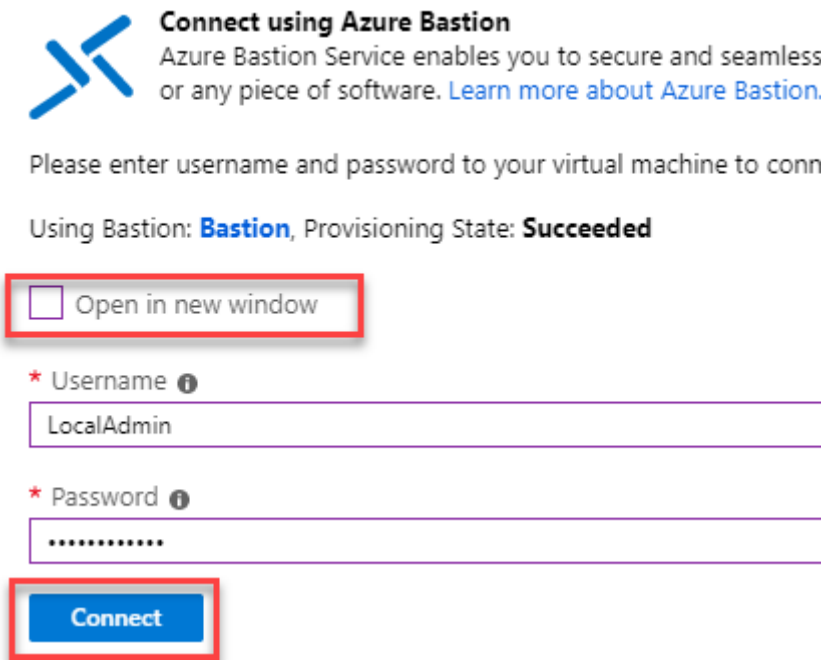
1. In the **Azure portal**, navigate to your virtual machine, then click **Connect**.



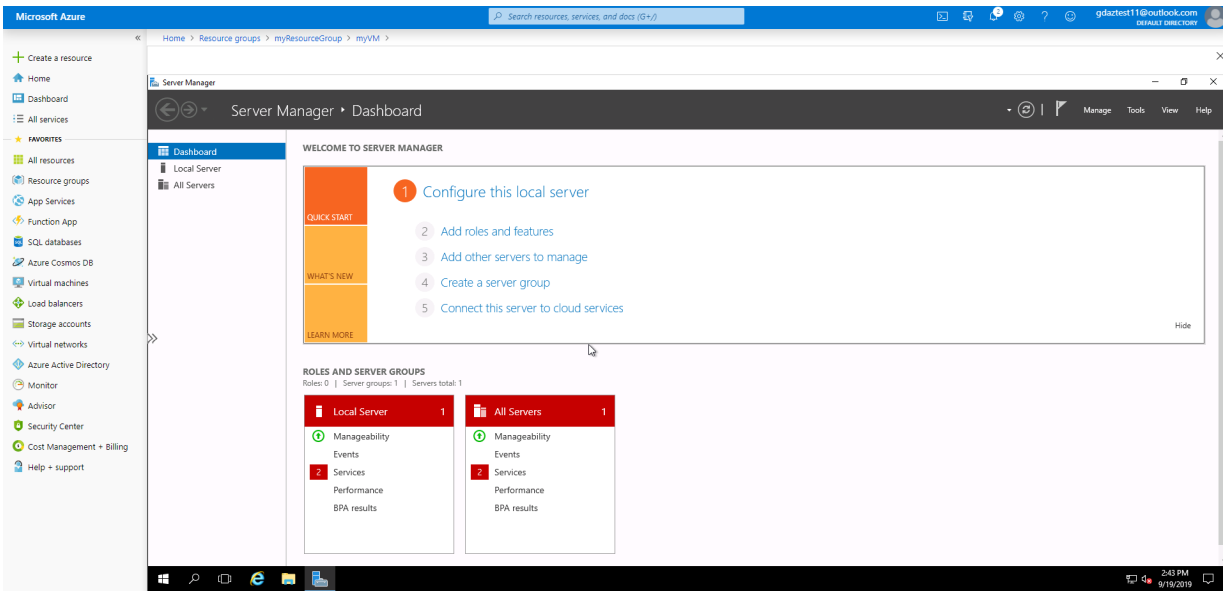
2. On the dropdown, click **Bastion**.



3. De-select Open in new window and then enter **LocalAdmin** and **Pa55w.rd1234** for the credentials and click **Connect**.



4. You should now be connected to your VM.



WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this lab.

35 Module 2: Lab 15 - Manage Azure DDoS Protection Standard

In this lab you will learn how to enable and disable distributed denial of service (DDoS) protection, and use telemetry to mitigate a DDoS attack with Azure DDoS Protection Standard. DDoS Protection Standard protects Azure resources such as virtual machines, load balancers, and application gateways that have an Azure public IP address assigned to it.

35.1 Exercise 1: Implement DDoS protection in Azure.

35.1.1 Task 1: Create a DDoS protection plan

A DDoS protection plan defines a set of virtual networks that have DDoS protection standard enabled, across subscriptions. You can configure one DDoS protection plan for your organization and link virtual networks from multiple subscriptions to the same plan. The DDoS Protection Plan itself is also associated with a subscription, that you select during the creation of the plan. The DDoS Protection Plan works across regions and subscriptions. Example -you can create the plan in Region East-US and link to subscription #1 in your tenant. The same plan can be linked to virtual networks from other subscriptions in different regions, across your tenant. The subscription the plan is associated to incurs the monthly recurring bill for the plan, as well as overage charges, in case the number of protected public IP addresses exceed 100.

Creation of more than one plan is not required for most organizations. A plan cannot be moved between subscriptions. If you want to change the subscription a plan is in, you have to delete the existing plan and create a new one.

1. Select **Create a resource** in the upper left corner of the Azure portal.
2. Search for *DDoS*. When **DDos protection plan** appears in the search results, select it.
3. Select **Create**.
4. Enter or select your own values, or enter, or select the following example values, and then select **Create**:

Setting	Value
Name	myDdosProtectionPlan
Subscription	Select your subscription.
Resource group	Select **Create new** and enter *myResourceGroup*
Location	East US

35.1.2 Task 2: Enable DDoS for a new virtual network

1. Select **Create a resource** in the upper left corner of the Azure portal.
2. Select **Networking**, and then select **Virtual network**.
3. Enter or select your own values, of enter or select the following example values, accept the remaining defaults, and then select **Review + create**, then click **Create**:

Setting	Value
Name	myVirtualNetwork
Subscription	Select your subscription.
Resource group	Select **Use existing**, and then select **myResourceGroup**
Location	East US
DDos protection	Select the **Security Tab**. Select **Standard** and then under **DDoS protection**, select **myDdosProtec**

   You cannot move a virtual network to another resource group or subscription when DDoS Standard is enabled for the virtual network. If you need to move a virtual network with DDoS Standard enabled, disable DDoS Standard first, move the virtual network, and then enable DDoS standard. After the move, the auto-tuned policy thresholds for all the protected public IP addresses in the virtual network are reset.

4. Click **All Services** and search for and select **Public IP addresses**.
5. Click **+ Add** and create an **IPv4** Public IP address in your **myResourceGroup** Resource Group.

35.1.3 Task 3: Disable DDoS for a virtual network

1. Enter the name of the virtual network you want to disable DDoS protection standard for in the **Search resources, services, and docs box** at the top of the portal. When the name of the virtual network appears in the search results, select it.
2. Select **DDoS protection**, under **SETTINGS**.
3. Select **Basic** under **DDoS protection plan** and then select **Save**.

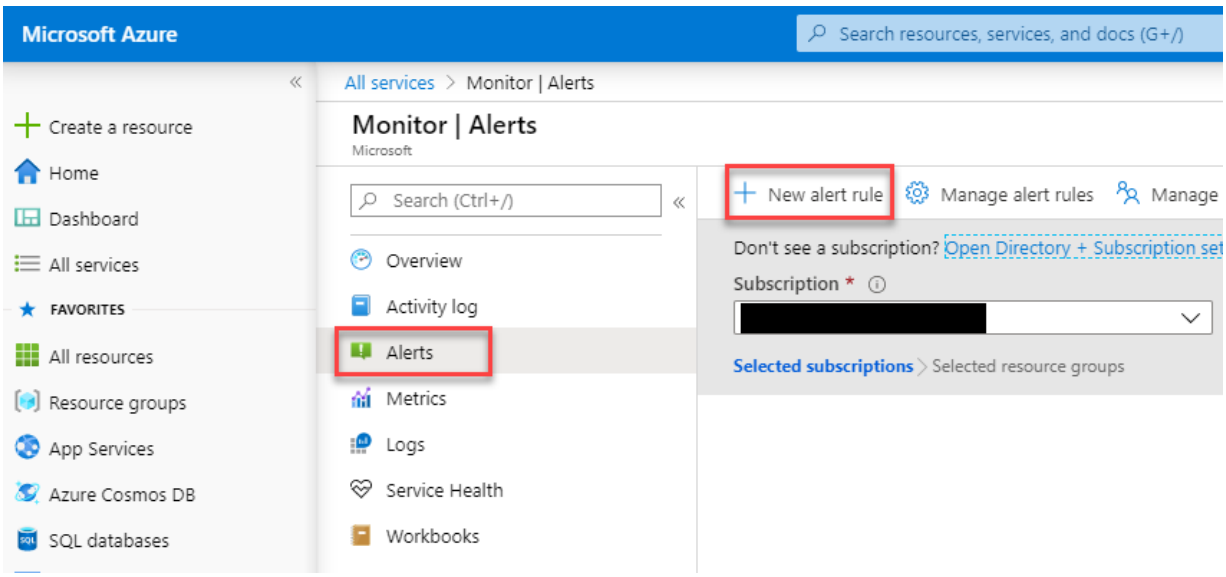
35.1.4 Task 4: Work with DDoS protection plans

1. Select **All services** on the top, left of the portal.
2. Enter *DDoS* in the **Filter** box. When **DDoS protection plans** appear in the results, select it.
3. Select the protection plan you want to view from the list.
4. All virtual networks associated to the plan are listed.
5. If you want to delete a plan, you must first dissociate all virtual networks from it.

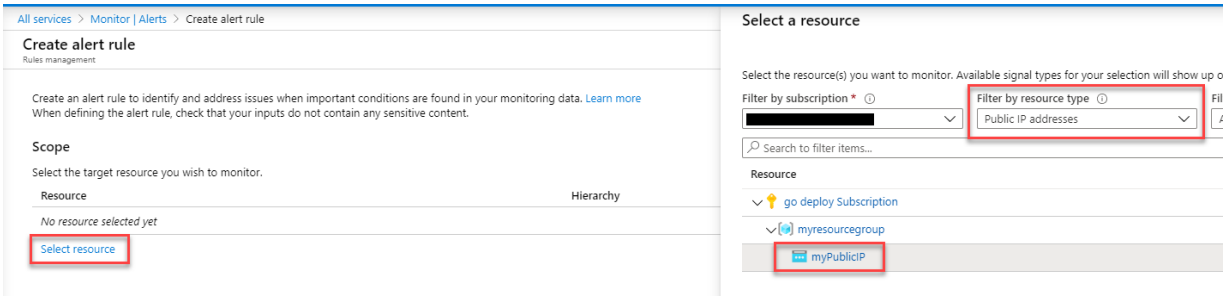
35.1.5 Task 5: Configure alerts for DDoS protection metrics

You can select any of the available DDoS protection metrics to alert you when there's an active mitigation during an attack, using the Azure Monitor alert configuration. When the conditions are met, the address specified receives an alert email:

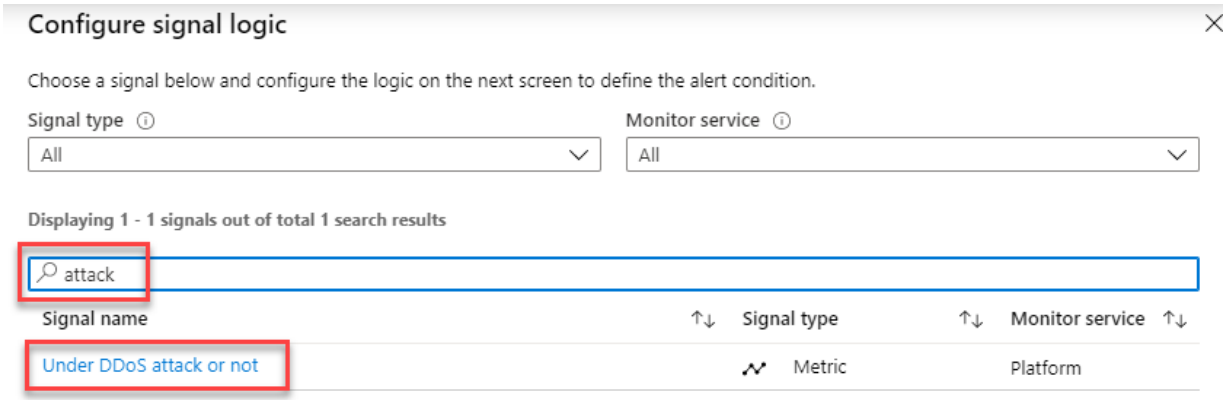
1. Select **All services** on the top, left of the portal.
2. Enter *Monitor* in the **Filter** box. When **Monitor** appears in the results, select it.
3. Select **Alerts**.
4. On the **Monitor** blade click **Alerts** then click **+ New alert rule**.



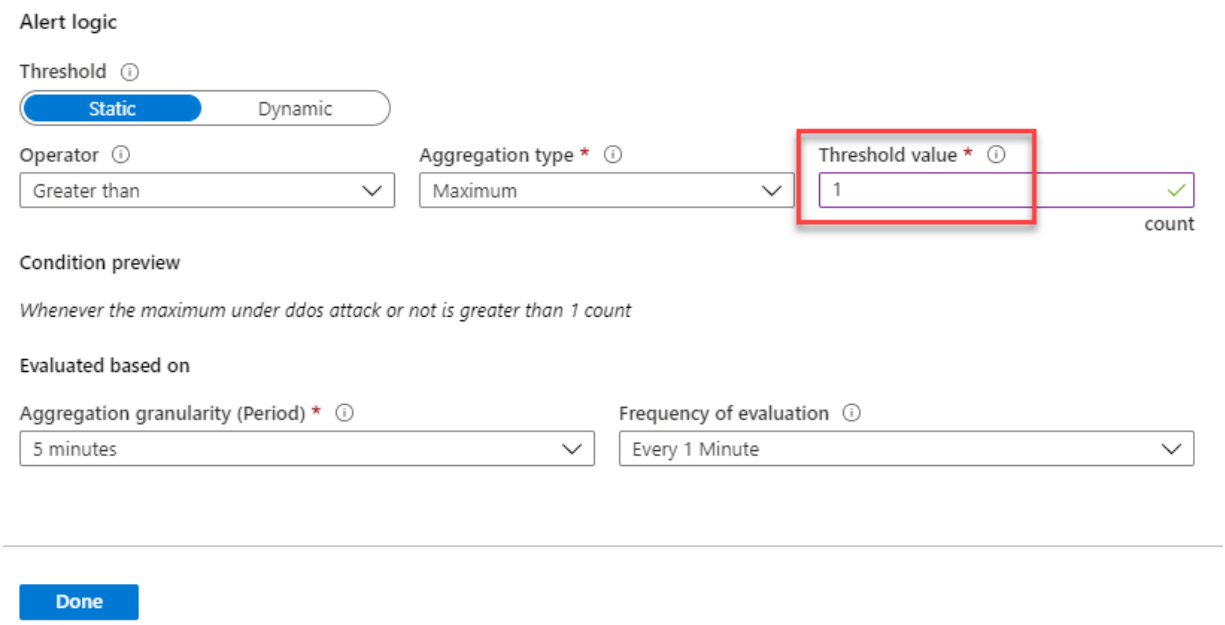
5. Click **Select resource**.
6. Select your Subscription and then in the Filter by resource type drop down, select **Public IP addresses** then select **myPublicIP** (or the name of your public ip address you created earlier) then click **Done**.



7. Click **Select condition**.
8. Search for ***attack*** and select **Under DDos attack or not**.



9. Scroll down the **Configure signal logic** blade and in the **Threashold value** enter **1** - **1** means you are under attack. **0** means you are not under attack. Click **Done**.

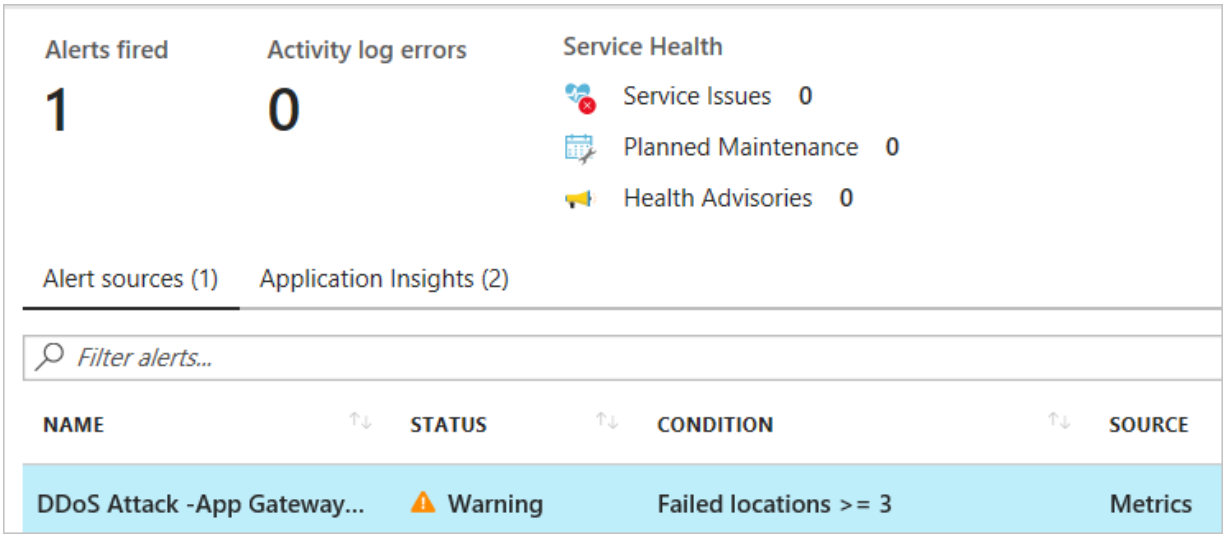


10. On the **Create alert rule** blade, click **Select action group** then click **+ Create action group.** and enter the following details.

Setting	Value
Action group name	**DDOS AG**
Short Name	**ddosag**
Subscription	Select your subscription
Resource Group	Select **myResourceGroup**
Action Name	DDOS Alert Email
Action Type	Select **Email/SMS message/Push/Voice** then select email and enter a valid email address and cli

11. Back on the Add action group blade click **OK**.
12. Give the Alert a name and click **Create alert rule**.

    Within a few minutes of an attack detection, you would receive an email from Azure Monitor metrics that looks similar to the following screenshot:



35.1.6 Task 6: Use DDoS protection telemetry

Telemetry for an attack is provided through Azure Monitor in real time. The telemetry is available only for the duration that a public IP address is under mitigation. You don't see telemetry before or after an attack is

mitigated.

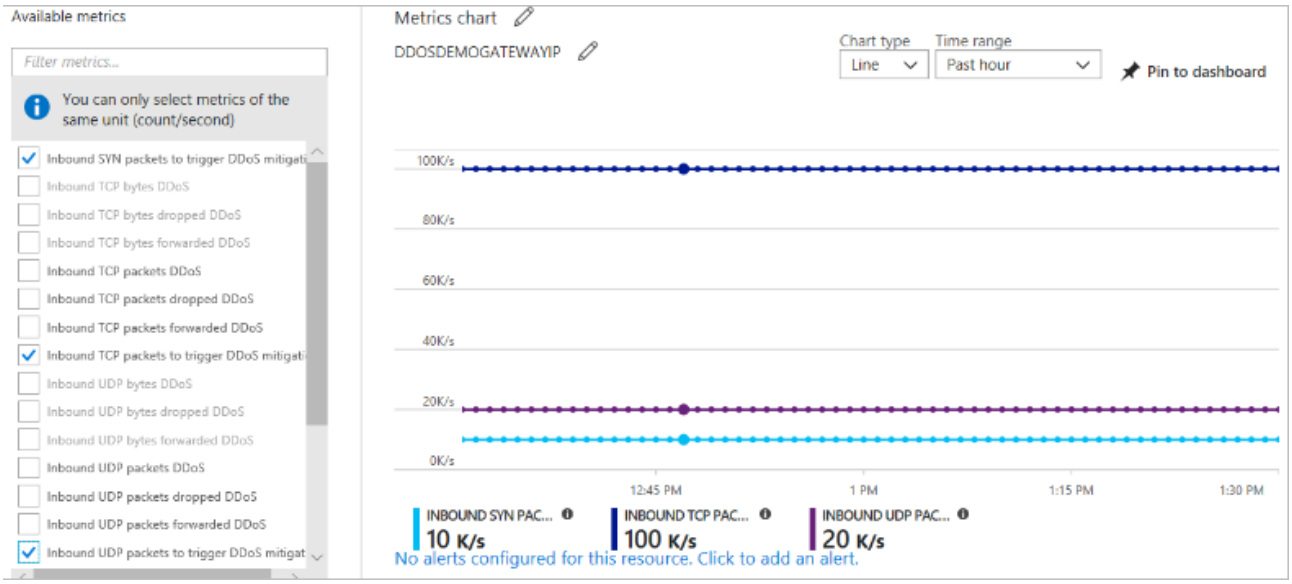
1. Select **All services** on the top, left of the portal.
2. Enter *Monitor* in the **Filter** box. When **Monitor** appears in the results, select it.
3. Select **Metrics**.
4. Select the **Subscription** and **Resource group** that contain the public IP address that you want telemetry for then click **Apply**.
5. A series of **Available Metrics** appear on the top of the screen. These metrics, when selected, are graphed in the **Azure Monitor Metrics Chart** on the overview screen.
6. Select the Public IP Address you created earlier and select the **Under DDoS attack or not** Metric and under **aggregation** select **Max**

The metric names present different packet types, and bytes vs. packets, with a basic construct of tag names on each metric as follows:

- **Dropped tag name** (for example, **Inbound Packets Dropped DDoS**): The number of packets dropped/scrubbed by the DDoS protection system.
- **Forwarded tag name** (for example **Inbound Packets Forwarded DDoS**): The number of packets forwarded by the DDoS system to the destination VIP - traffic that was not filtered.
- **No tag name** (for example **Inbound Packets DDoS**): The total number of packets that came into the scrubbing system - representing the sum of the packets dropped and forwarded.

35.1.7 Task 7: View DDoS mitigation policies

DDoS Protection Standard applies three auto-tuned mitigation policies (TCP SYN, TCP & UDP) for each public IP address of the protected resource, in the virtual network that has DDoS enabled. You can view the policy thresholds by selecting the **Inbound TCP packets to trigger DDoS mitigation** and **Inbound UDP packets to trigger DDoS mitigation** metrics with **aggregation** type as 'Max', as shown in the following picture:



Policy thresholds are auto-configured via Azure machine learning-based network traffic profiling. Only when the policy threshold is breached does DDoS mitigation occur for the IP address under attack.

35.1.8 Task 8: Configure DDoS attack mitigation reports

Attack mitigation reports uses the Netflow protocol data which is aggregated to provide detailed information about the attack on your resource. Anytime a public IP resource is under attack, the report generation will start as soon as the mitigation starts. There will be an incremental report generated every 5 mins and a post-mitigation report for the whole mitigation period. This is to ensure that in an event the DDoS attack continues for a longer duration of time, you will be able to view the most current snapshot of mitigation report every 5 minutes and a complete summary once the attack mitigation is over.

1. Select **All services** on the top, left of the portal.

2. Enter *Monitor* in the **Filter** box. When **Monitor** appears in the results, select it.
3. Under **SETTINGS**, select **Diagnostic Settings**.
4. Select the **Subscription** and **Resource group** that contain the public IP address you want to log.
5. Select **Public IP Address** for **Resource type**, then select the specific public IP address you want to log metrics for.
6. Select **+ Add diagnostic setting** to collect the DDoSMitigationReports log and give the Diagnostic setting the name **DDoSLog** and then select as many of the following options as you require then click **Save**:
    - **Archive to a storage account**: Data is written to an Azure Storage account.
    - **Stream to an event hub**: Allows a log receiver to pick up logs using an Azure Event Hub. Event hubs enable integration with Splunk or other SIEM systems.
    - **Send to Log Analytics**: Writes logs to the Azure Monitor service.

Both the incremental & post-attack mitigation reports include the following fields

- Attack vectors
- Traffic statistics
- Reason for dropped packets
- Protocols involved
- Top 10 source countries or regions
- Top 10 source ASNs

35.1.9 Task 9: Configure DDoS attack mitigation flow logs

Attack Mitigation Flow Logs allow you to review the dropped traffic, forwarded traffic and other interesting datapoints during an active DDoS attack in near-real time. You can ingest the constant stream of this data into your SIEM systems via event hub for near-real time monitoring, take potential actions and address the need of your defense operations.

1. Select **All services** on the top, left of the portal.
2. Enter *Monitor* in the **Filter** box. When **Monitor** appears in the results, select it.
3. Under **SETTINGS**, select **Diagnostic Settings**.
4. Select the **Subscription** and **Resource group** that contain the public IP address you want to log.
5. Select **Public IP Address** for **Resource type**, then select the specific public IP address you want to log metrics for.
6. Select **Turn on diagnostics to collect the DDoSMitigationFlowLogs log** and then select as many of the following options as you require:
    - **Archive to a storage account**: Data is written to an Azure Storage account. To learn more about this option.
    - **Stream to an event hub**: Allows a log receiver to pick up logs using an Azure Event Hub. Event hubs enable integration with Splunk or other SIEM systems.
    - **Send to Log Analytics**: Writes logs to the Azure Monitor service.
7. To view the flow logs data in Azure analytics dashboard.

Flow logs will have the following fields:

- Source IP
- Destination IP
- Source Port
- Destination port
- Protocol type
- Action taken during mitigation

35.1.10 Task 10: Validate DDoS detection (Optional and not part of the AZ-500 course)

Note: Only carry out this task if you feel comfortable to do so. This task does not form part of the course and is a stretch exercise for those students who wish to do so.

Microsoft has partnered with BreakingPoint Cloud to build an interface where you can generate traffic against DDoS Protection-enabled public IP addresses for simulations. The BreakPoint Cloud simulation allows you to:

- Validate how Microsoft Azure DDoS Protection protects your Azure resources from DDoS attacks
- Optimize your incident response process while under DDoS attack
- Document DDoS compliance
- Train your network security teams

Permissions

To work with DDoS protection plans, your account must be assigned to the network contributor role or to a role that is assigned the appropriate actions listed in the following table:

Action	Name
Microsoft.Network/ddosProtectionPlans/read	Read a DDoS protection plan
Microsoft.Network/ddosProtectionPlans/write	Create or update a DDoS protection plan
Microsoft.Network/ddosProtectionPlans/delete	Delete a DDoS protection plan
Microsoft.Network/ddosProtectionPlans/join/action	Join a DDoS protection plan

To enable DDoS protection for a virtual network, your account must also be assigned the appropriate actions for virtual networks.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this lab.

36 Module 2: Lab 16 - Antimalware for VMs

Scenario

Azure Security Center monitors the status of antimalware protection and reports this on the Endpoint protection issues blade. Security Center notes issues, such as detected threats and insufficient protection, that might make your VMs and computers vulnerable to malware threats. By using the information on Endpoint protection issues, you can make a plan to address any identified issues.

Security Center reports the following endpoint protection issues:

- Endpoint protection not installed on Azure VMs. A supported antimalware solution isn't installed on these Azure VMs.
- Endpoint protection not installed on non-Azure computers. A supported antimalware solution isn't installed on these non-Azure computers.
- Endpoint protection health issues:
- **Signature out of date**. An antimalware solution is installed on these VMs and computers, but the solution doesn't have the latest antimalware signatures.
  - **No real time protection**. An antimalware solution is installed on these VMs and computers, but it isn't configured for real-time protection. The service might be disabled, or Security Center might be unable to obtain the status because the solution isn't supported.
- **Not reporting**. An antimalware solution is installed but not reporting data.
- **Unknown**. An antimalware solution is installed, but either its status is unknown or it's reporting an unknown error.

36.1 Exercise 1: Deploy Antimalware for Azure VMs.

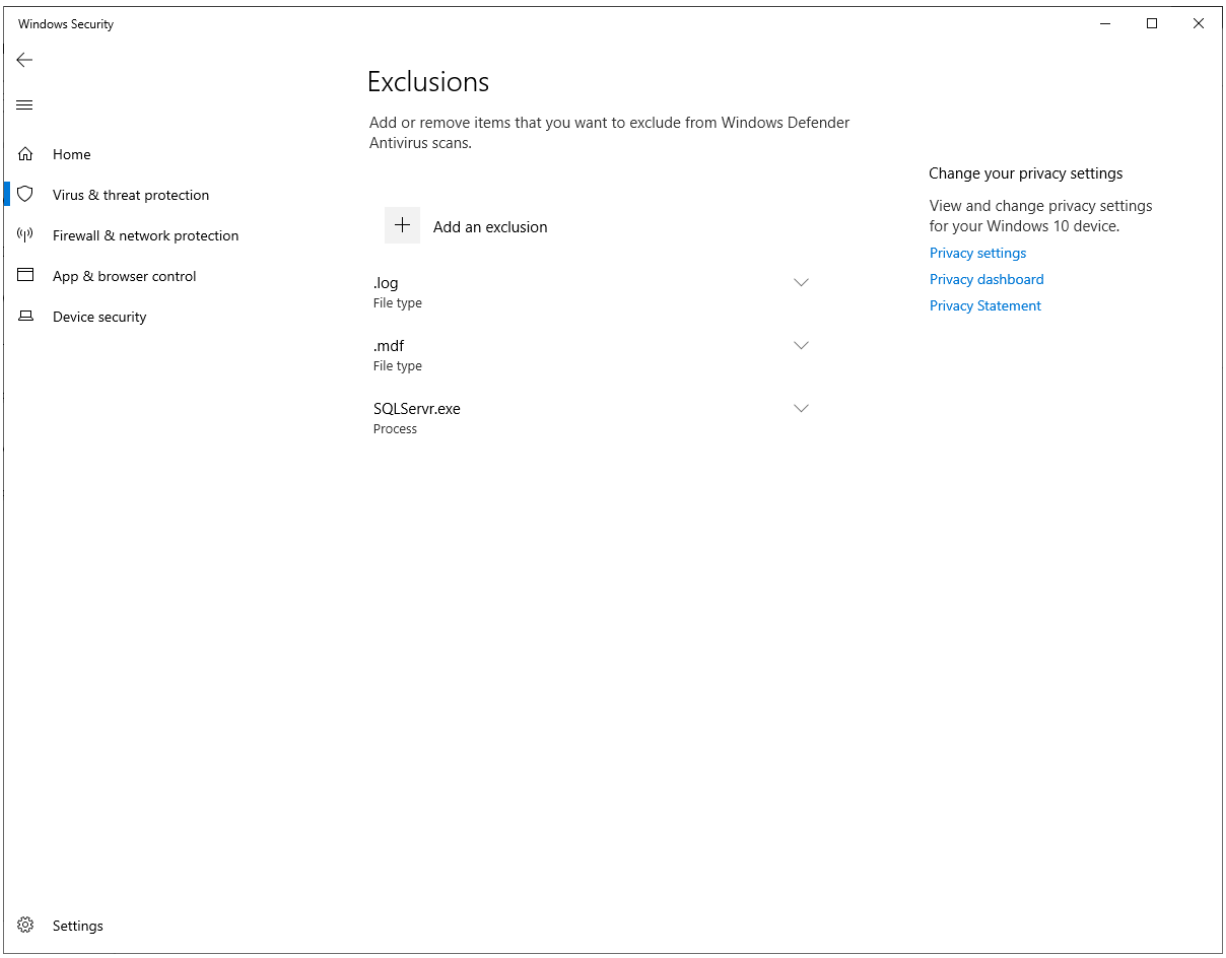
36.2 Task 1: Create an Azure Virtual Machine with the Antimalware extension

Enable and configure antimalware for VMs To enable and configure Microsoft Antimalware for Azure VMs by using the Azure portal while provisioning a VM, complete the following steps:

1. Sign in to the **Azure portal** at `https://portal.azure.com`
2. To create a new VM, navigate to **Virtual machines**, select **Add**, and then select **Windows Server 2019 Datacenter** under the image dropdown.
3. Enter the following details for the VM:

Option	Answer
Resource Group	Create new > **myResourceGroup**
VM Name	**myVM**
Region	**East US**
Image	**Windows Server 2019 Datacenter**
Username	**localadmin**
Password	**Pa55w.rd1234**
Public inbound ports	**RDP (3389)**

4. Click the **Management** Tab and ensure all the radio buttons are set to **Off**.
5. Click the **Advanced** Tab and click **Select an extension to install**.
6. Select the **Microsoft Antimalware** extension. *You may need to click Load More at the bottom of the list to see the extension.*
7. On the **Microsoft Antimalware** blade click **Create**.
8. In the Install extension section, you can configure files, locations, process exclusions, and other scan options.
9. In the Excluded file extensions enter **.mdf;.log**
10. In the Excluded processes enter **SQLServr.exe**
11. Select **OK**.
12. Back in the Settings section, select **Review + create**.
13. Click **Create** on the validation screen.
14. Once the VM has created log onto the VM via RDP.
15. Open the Settings app and select **Update & Security** > **Windows Security** > **Virus and Threat Protection** > **Manage Settings** > **Add or remove exclusions** to verify the deployment was successful.



WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this lab.

37 Module 2: Lab 17 - Manage Windows updates by using Azure Automation

Scenario

You can use the Update Management solution to manage updates and patches for your virtual machines. In this tutorial, you learn how to quickly assess the status of available updates, schedule installation of required updates, review deployment results, and create an alert to verify that updates apply successfully.

37.1 Exercise 1: Use Azure Automation to manage Windows Updates.

37.1.1 Task 1: Create a Resource Group

1. Click **Resource Groups** on the Azure hub menu.
2. Click **Add**.
3. Name the resource group **RunBooks**
4. Change the region to **East US**
5. Click **Review + create**, then click **Create**.

37.1.2 Task 2: Create Automation account

1. Click the **Create a resource** button found on the upper left-hand corner of Azure.

2. Select **IT & Management Tools**, and then select **Automation**.
3. Enter the following information.

Name	Resource Group	Location	Create Azure Run As account
MyAutomation	RunBooks	EastUS2	Yes

4. Click **Create**
5. When the deployment has completed, click **All Services**, select **Automation Accounts** and select the Automation Account you created.

37.1.3 Task 3: Create a VM for use

1. Click **Virtual Machines**
2. Click **Add**
3. Fill in the folowing details to create the VM

Resource Group	Virtual Machine Name	Region	Image	Username	Password
RunBooks	UpdateVM	EastUS	Windows Server 2016	localadmin	Pa55w.rd1234

4. Click **Review+Create**
5. Click **Create**

Note: Wait for the VM to deploy before moving on

37.1.4 Task 4: Enable Update Mangement

1. In the portal click Virtual machines
2. Select the VM you created in the previous steps
3. Under **Operations** click **Update management**
4. Click **Enable**

Validation is performed to determine whether Update Management is enabled for this VM. This validation includes checks for an Azure Log Analytics workspace and linked Automation account, and whether the Update Management solution is in the workspace.

A Log Analytics workspace is used to collect data that's generated by features and services like Update Management. The workspace provides a single location to review and analyze data from multiple sources.

The validation process also checks to see whether the VM is provisioned with the Microsoft Monitoring Agent (MMA) and Automation Hybrid Runbook Worker. This agent is used to communicate with Azure Automation and to obtain information about the update status. The agent requires port 443 to be open to communicate with the Azure Automation service and to download updates.

If any of the following prerequisites were found to be missing during onboarding, they're automatically added:

- **Log Analytics workspace**
- **An Automation account**
- **A Hybrid Runbook Worker (enabled on the VM)**

Note: Enabling the solution can take up to a few minutes. During this time, don't close the browser window. After the solution is enabled, information about missing updates on the VM flows to Azure Monitor logs. It can take between 30 minutes and 6 hours for the data to be available for analysis.

37.1.5 Task 5: View Update assesment

1. After the update management is enabled the **Update Management** pane will open

2. If updates are missing they will display here

   Note: If your VM has been created from the previous steps this is a template VM from Microsoft that will include all the latest Windows updates so you may not see any required updates for the VM

3. If there are updates available they will be listed with a link under the Inforamtion Link column that will allow you to view the details of that update and link to the Kb article of the update from Microsoft.

37.1.6 Task 6: Configure Alerts

1. Return to the resource group you created erlier
2. Select the **MyAutomation** automation account
3. Under **Monitoring** click **Alerts**
4. Click **New Alert Rule**
5. Click **Select condition**
6. Select **Total Update Deployment Runs** from the list
7. Click the Select * checkbox next to **Update Deployment Name and Status**
8. Under **Alert logic**, for **Threshold**, enter **1**
9. Click **Done**
10. Under **Alert Details**, fill in the **Alert Rule name** with **UpdateAlert**
11. Set the Severtiy to **Sev2**
12. Under **Action groups** click **Select action group** then click **Create action group**
13. Fill in the following details

Action Group name	Short Name	Resource Group	Action Name	Action Type
VM Updates Action group	VMup	Runbooks	Email	Email/SMS/Push/voice

14. Click **edit details** next to email and fill in **your email** that will be used for the alerts
15. Click **OK**
16. Click **OK** under **Create Action Group**
17. Verify your action group is listed
18. Click **Create alert rule**

37.1.7 Task 7: Schedule an Update Deployment

1. Return to your list of VMs
2. Select the VM you created earlier in the lab
3. Click **Update management**
4. Select **Schedule update deployment**
5. In new update deployment fill in the following settings
   - **Name** - ScheduledUpdates
   - **Schedule Settings** - Recurrence > reccuring
6. Click **OK** and then click **Create**
7. Click **deployment schedules** to view the list of active **deployment schedules**

37.1.8 Task 8: View results of an update deployment

Note: After the scheduled deployment starts, you can see the status for that deployment on the Update deployments tab under Update management. The status is In progress when the deployment is currently

running. When the deployment finishes, if it's successful, the status changes to Succeeded. When there are failures with one or more updates in the deployment, the status is partially failed.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed adding a scheduled Windows Update management with an alert system for a VM running Windows server in Azure

38 Module 2: Lab 18 - Custom Domains

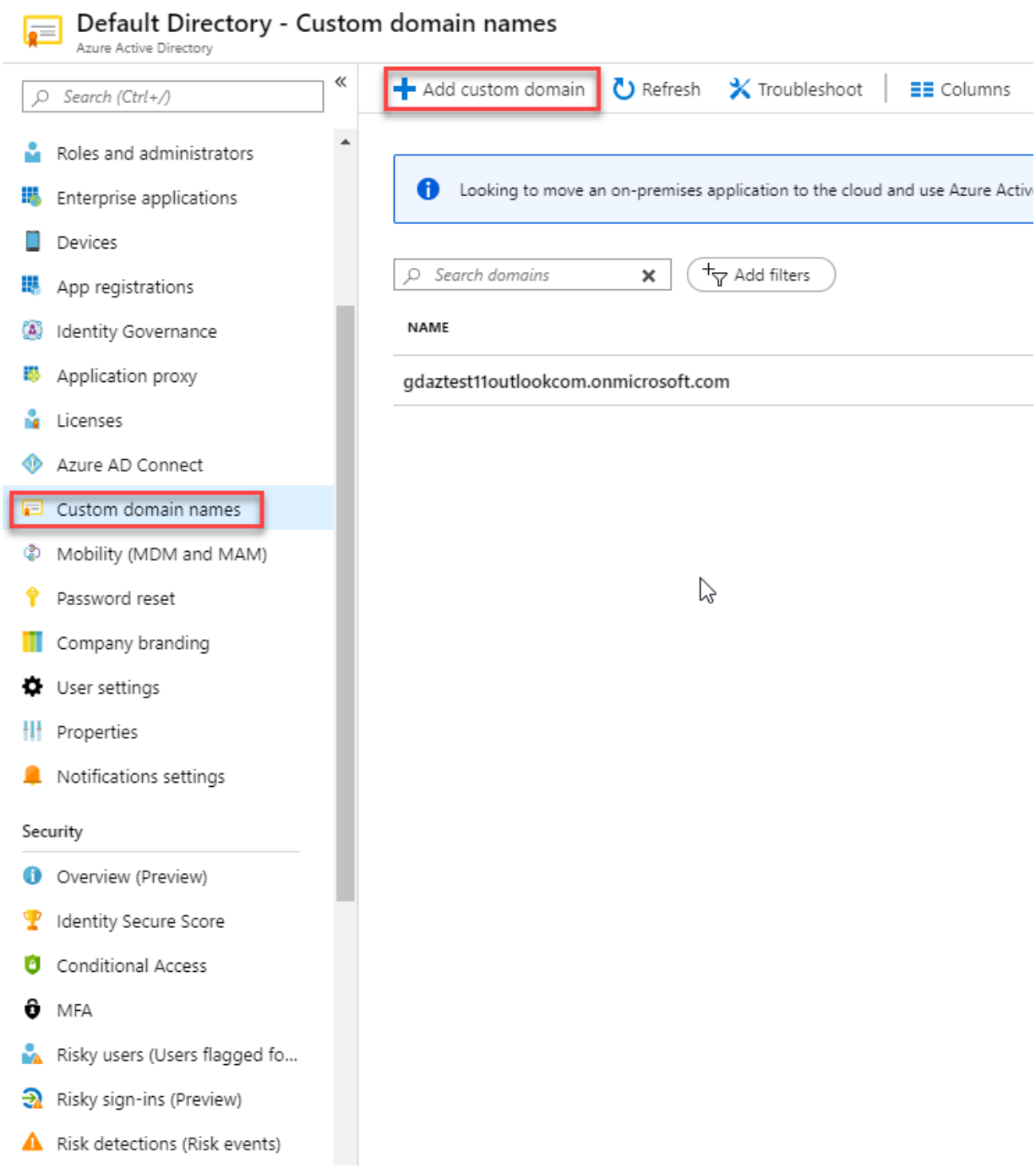
Scenario

Every new Azure AD tenant comes with an initial domain name, domainname.onmicrosoft.com. You can't change or delete the initial domain name, but you can add your organization's names to the list. Adding custom domain names helps you to create user names that are familiar to your users, such as alain@contoso.com.

38.0.1 Exercise 1: Add your custom domain name using the Azure Active Directory portal

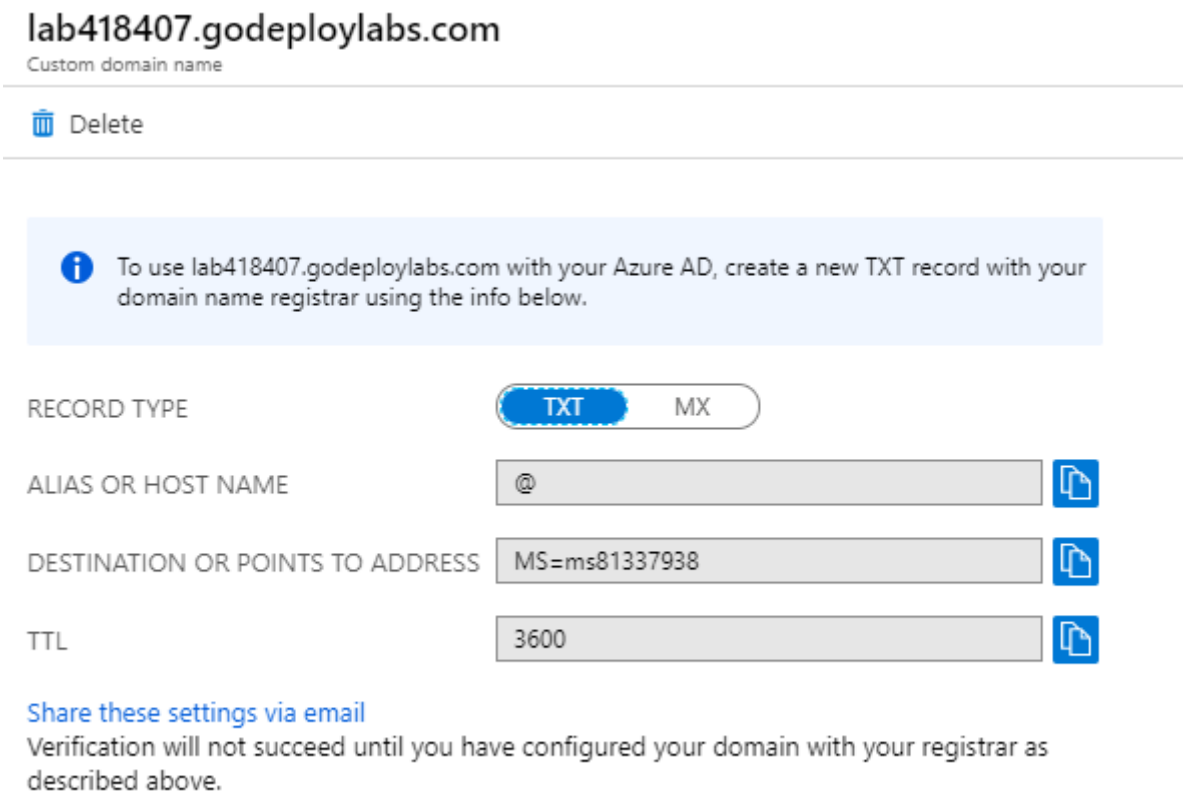
38.0.2 Task 1: Add your custom domain name to Azure AD

1. In the Azure Portal, select **Azure Active Directory**.
2. Select **Custom domain names**, and then select **Add custom domain**.



3. Navigate to the DNS tab in the Lab Environment to identify the unique custom domain name you have been allocated. This will be in the form of labxxxxxx.customdomainname.com.
4. In the Custom Domain name field type in your domain name you identified in the previous step and click **Add domain**.

   The unverified domain is added and the **labxxxxx.customdomainname.com** page appears showing you your DNS info.
5. Copy the DNS info from the **labxxxxx.customdomainname.com** page. For example, MS=ms64983159.



38.0.3 Task 2: Add your DNS information to the domain registrar

After you add your custom domain name to Azure AD, you must return to your domain registrar and add the Azure AD DNS information from your copied TXT file. Creating this TXT record for your domain "verifies" ownership of your domain name.

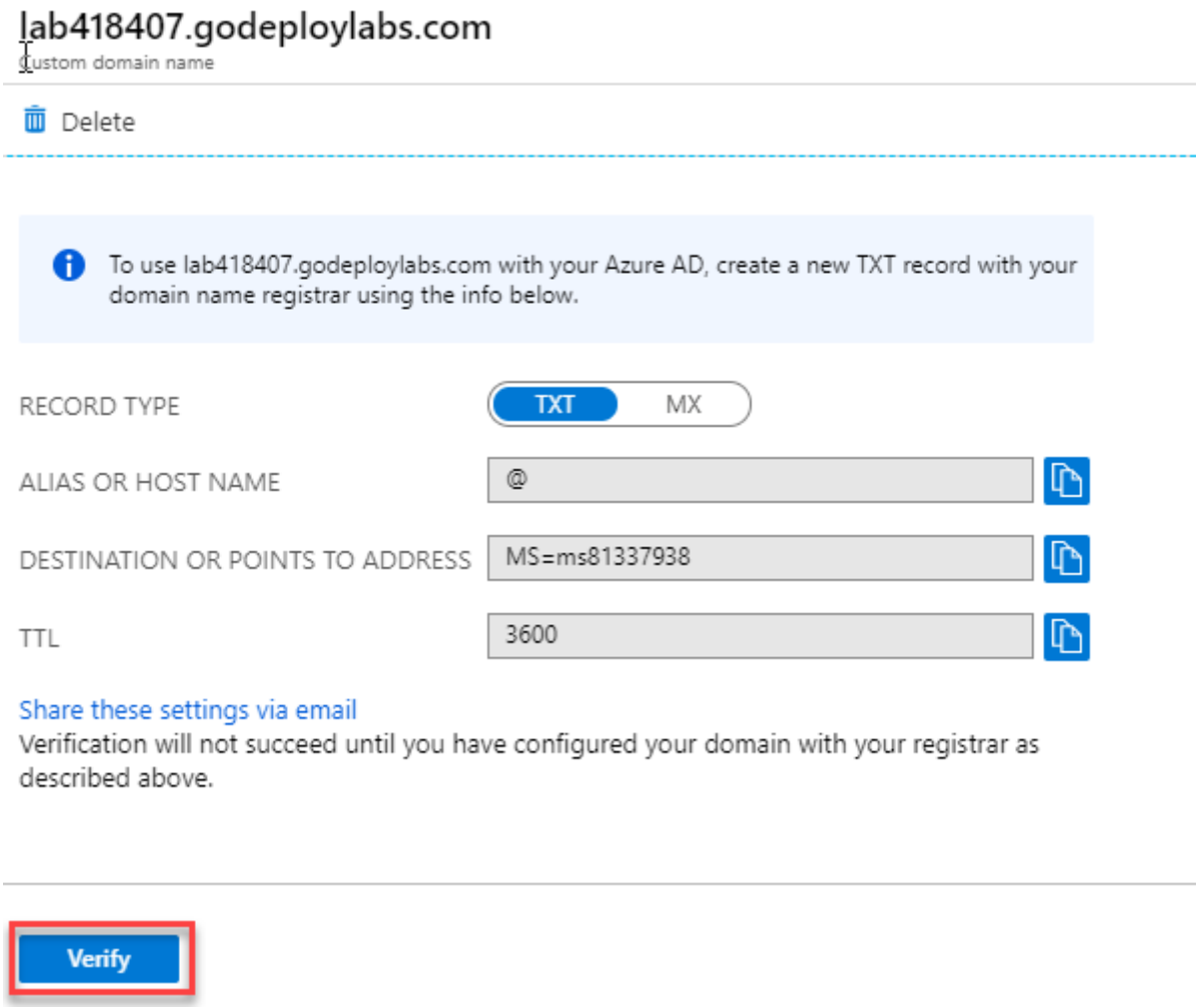
1. Go back to the DNS tab in the Lab environment and create a new TXT record for your domain based on your copied DNS information. You only need to enter the value record. Leave the name field blank and click **Save**.

Note: You can register as many domain names as you want. However, each domain gets its own TXT record from Azure AD. Be careful when entering your TXT file information at the domain registrar. If you enter the wrong, or duplicate information by mistake, you'll have to wait until the TTL times out (60 minutes) before you can try again.

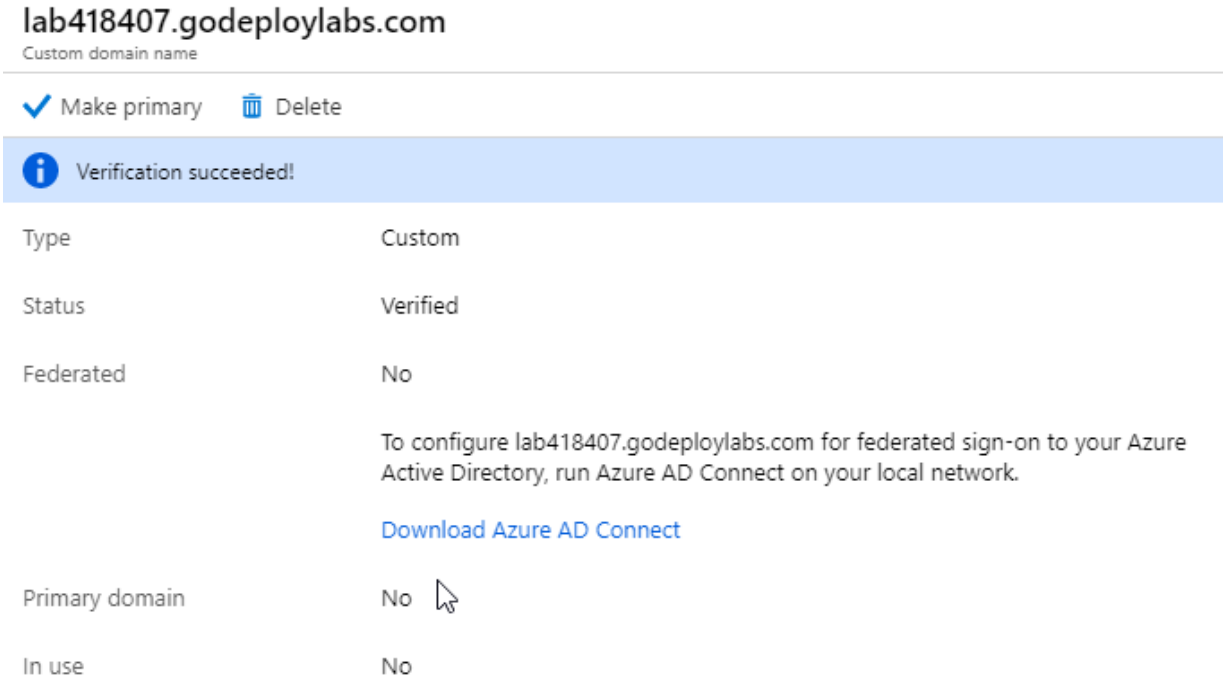
38.0.4 Task 2: Verify your custom domain name

After you register your custom domain name, you need to make sure it's valid in Azure AD. The propagation from your domain registrar to Azure AD can be instantaneous or it can take up to a few days, depending on your domain registrar.

1. Return back to the Azure Portal **labxxxxx.customdomainname.com** page and click **Verify**.



2. Your domain should now be verified.



3. After you've verified your custom domain name, you can delete your verification TXT record.

Results: You have now completed this lab.

39 Module 2: Lab 19 - Private DNS

Scenario

A DNS zone is used to host the DNS records for a particular domain. To start hosting your domain in Azure DNS, you need to create a DNS zone for that domain name. Each DNS record for your domain is then created inside this DNS zone. To publish a private DNS zone to your virtual network, you specify the list of virtual networks that are allowed to resolve records within the zone. These are called *linked* virtual networks. When autoregistration is enabled, Azure DNS also updates the zone records whenever a virtual machine is created, changes its' IP address, or is deleted.

In this lab, you learn how to:

- Create a private DNS zone
- Create a virtual network
- Link the virtual network
- Create test virtual machines
- Create an additional DNS record
- Test the private zone

39.1 Exercise 1: Create an Azure private DNS zone using the Azure portal

39.1.1 Task 1: Create a private DNS zone

The following example creates a DNS zone called **private.contoso.com** in a resource group called **MyAzureResourceGroup**.

A DNS zone contains the DNS entries for a domain. To start hosting your domain in Azure DNS, you create a DNS zone for that domain name.

1. On the portal search bar, type **private dns zones** in the search text box and press **Enter**.
2. Select **Private DNS zone**.
3. Select **Create private dns zone**.
4. On the **Create Private DNS zone** page, type or select the following values:
   - **Resource group**: Select **Create new**, enter *MyAzureResourceGroup*, and select **OK**. The resource group name must be unique within the Azure subscription.
   - **Name**: Type *private.contoso.com* for this example.
5. For **Resource group location** select **West Central US**.
6. Select **Review + Create**.
7. Select **Create**.

It may take a few minutes to create the zone.

39.1.2 Task 2: Create a virtual network

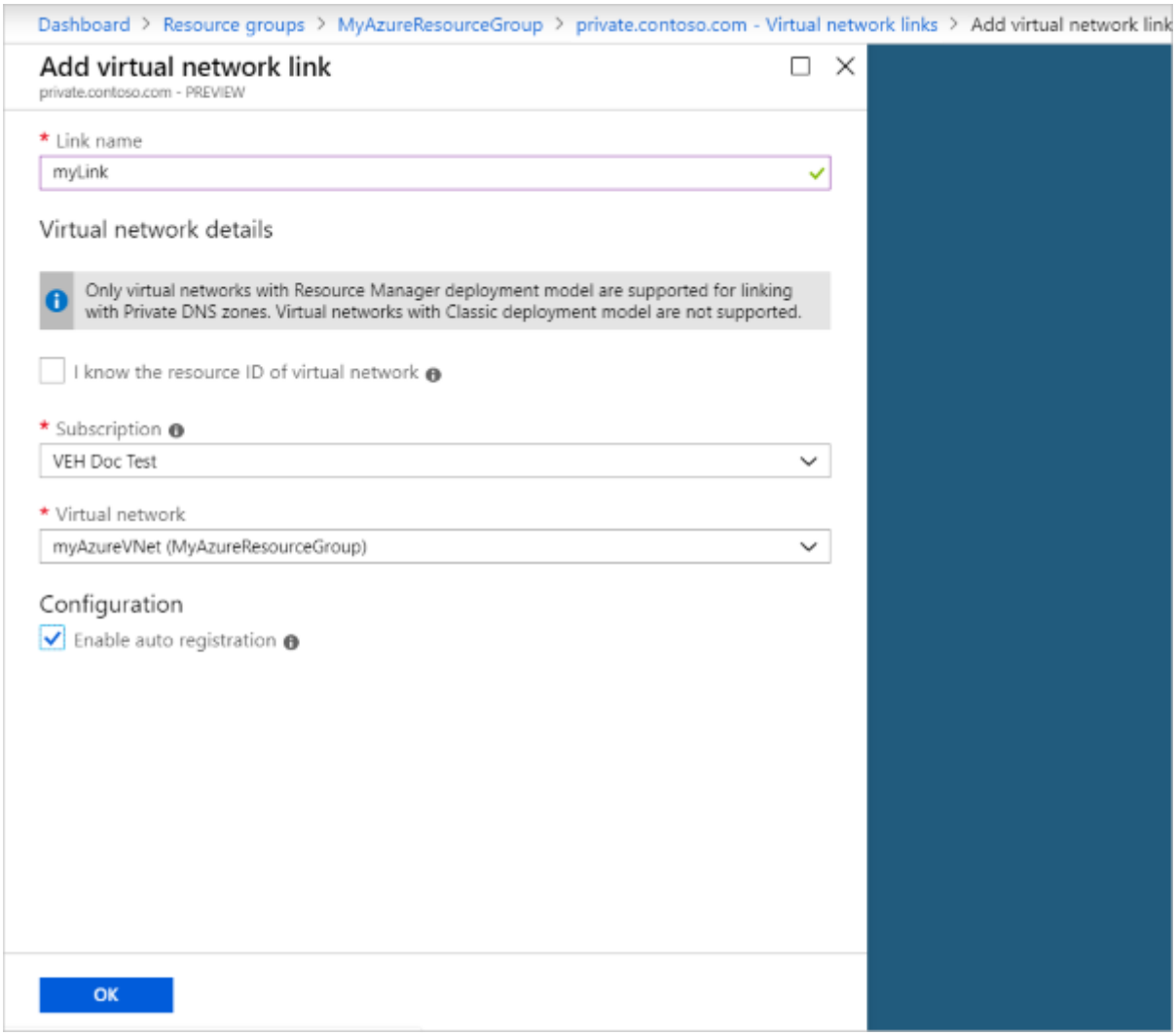
1. On the portal page upper left, select **Create a resource**, then **Networking**, then select **Virtual network**.
2. For **Name**, type **myAzureVNet**.
3. For **Resource group**, select **MyAzureResourceGroup**.
4. For **Location**, select **East US**.
5. Accept the other default values and select **Review + create**, then select **Create**.

39.1.3 Task 3: Link the virtual network

To link the private DNS zone to a virtual network, you create a virtual network link.

1. Open the **MyAzureResourceGroup** resource group and select the **private.contoso.com** private zone.
2. On the left pane, select **Virtual network links**.
3. Select **Add**.

4. Type **myLink** for the **Link name**.
5. For **Virtual network**, select **myAzureVNet**.
6. Select the **Enable auto registration** check box.
7. Select **OK**.



39.1.4 Task 4: Create the test virtual machines

Now, create two virtual machines so you can test your private DNS zone:

1. On the portal page upper left, select **Create a resource**, and then select **Windows Server 2016 Datacenter**.
2. Select **MyAzureResourceGroup** for the resource group.
3. Type **myVM01** - for the name of the virtual machine.
4. Select **East US** for the **Region**.
5. Type **LocalAdmin** for the administrator user name.
6. Type **Pa55w.rd1234** for the password and confirm the password.
7. For **Public inbound ports**, select **Allow selected ports**, and then select **RDP (3389)** for **Select inbound ports**.
8. Accept the other defaults for the page and then click **Next: Disks >**.
9. Accept the defaults on the **Disks** page, then click **Next: Networking >**.
10. Make sure that **myAzureVNet** is selected for the virtual network.

11. Accept the other defaults for the page, and then click **Next: Management >**.
12. For **Boot diagnostics**, select **Off**, accept the other defaults, and then select **Review + create**.
13. Review the settings and then click **Create**.
14. Repeat these steps and create another virtual machine named **myVM02**.

It will take a few minutes for both virtual machines to complete.

39.1.5 Task 5: Create an additional DNS record

The following example creates a record with the relative name **db** in the DNS Zone **private.contoso.com**, in resource group **MyAzureResourceGroup**. The fully qualified name of the record set is **db.private.contoso.com**. The record type is "A", with the IP address of **myVM01**.

1. Open the **MyAzureResourceGroup** resource group and select the **private.contoso.com** private zone.
2. Select **+ Record set**.
3. For **Name**, type **db**.
4. For **IP Address**, type the IP address you see for **myVM01**. This should be auto registered when the virtual machine started and should be **10.0.0.4**.
5. Select **OK**.

39.1.6 Task 6: Test the private zone

Now you can test the name resolution for your **private.contoso.com** private zone.

You can use the ping command to test name resolution. So, configure the firewall on both virtual machines to allow inbound ICMP packets.

1. Connect to myVM01, and open a Windows PowerShell window with administrator privileges.
2. Run the following command:

   ```
   New-NetFirewallRule -DisplayName "Allow ICMPv4-In" -Protocol ICMPv4
   ```

3. Repeat for myVM02.

39.1.7 Task 7: Ping the VMs by name

1. From the myVM02 Windows PowerShell command prompt, ping myVM01 using the automatically registered host name:

   ```
   ping myVM01.private.contoso.com
   ```

   You should see output that looks similar to this:

   ```
   Pinging myvm01.private.contoso.com [10.2.0.4] with 32 bytes of data:
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time=1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   ```

2. Now ping the **db** name you created previously:

   ```
   ping db.private.contoso.com
   ```

   You should see output that looks similar to this:

   ```
   PS C:\> ping db.private.contoso.com

   Pinging db.private.contoso.com [10.2.0.4] with 32 bytes of data:
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128
   Reply from 10.2.0.4: bytes=32 time<1ms TTL=128

   Ping statistics for 10.2.0.4:
   ```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
PS C:\>
```

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this lab.

40 Module 2: Lab 20 - Azure Blueprints

Scenario

When you learn how to create and assign blueprints, you can define common patterns to develop reusable and rapidly deployable configurations based on Azure Resource Manager templates, policy, security, and more. In this tutorial, you learn to use Azure Blueprints to do some of the common tasks related to creating, publishing, and assigning a blueprint within your organization. These tasks include:

- Create a new blueprint and add various supported artifacts
- Make changes to an existing blueprint still in Draft
- Mark a blueprint as ready to assign with Published
- Assign a blueprint to an existing subscription
- Check the status and progress of an assigned blueprint
- Remove a blueprint that has been assigned to a subscription

40.1 Exercise 1: Create a blueprint in the portal

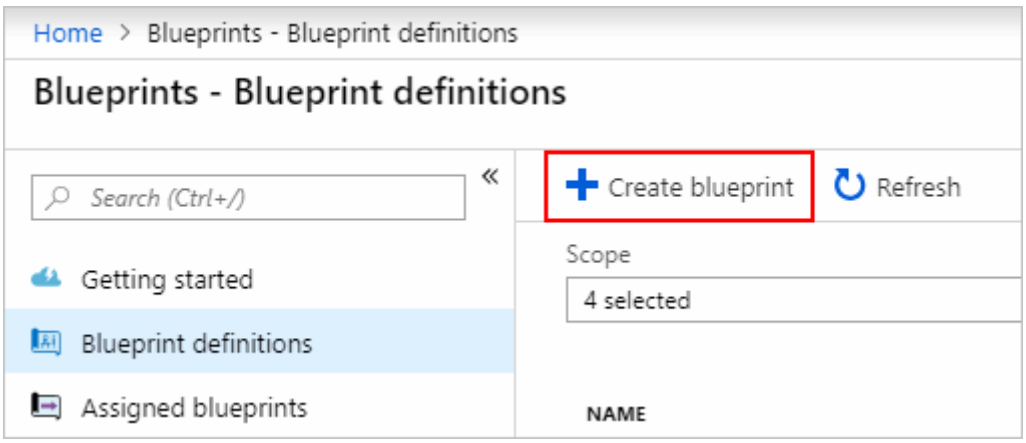
40.1.1 Task 1: Create a blueprint

The first step in defining a standard pattern for compliance is to compose a blueprint from the available resources. In this example, create a new blueprint named **MyBlueprint** to configure role and policy assignments for the subscription. Then add a new resource group, and create a Resource Manager template and role assignment on the new resource group.

1. Select **All services** in the left pane. Search for and select **Blueprints**.
2. Select **Blueprint definitions** from the page on the left and select the **+ Create blueprint** button at the top of the page.

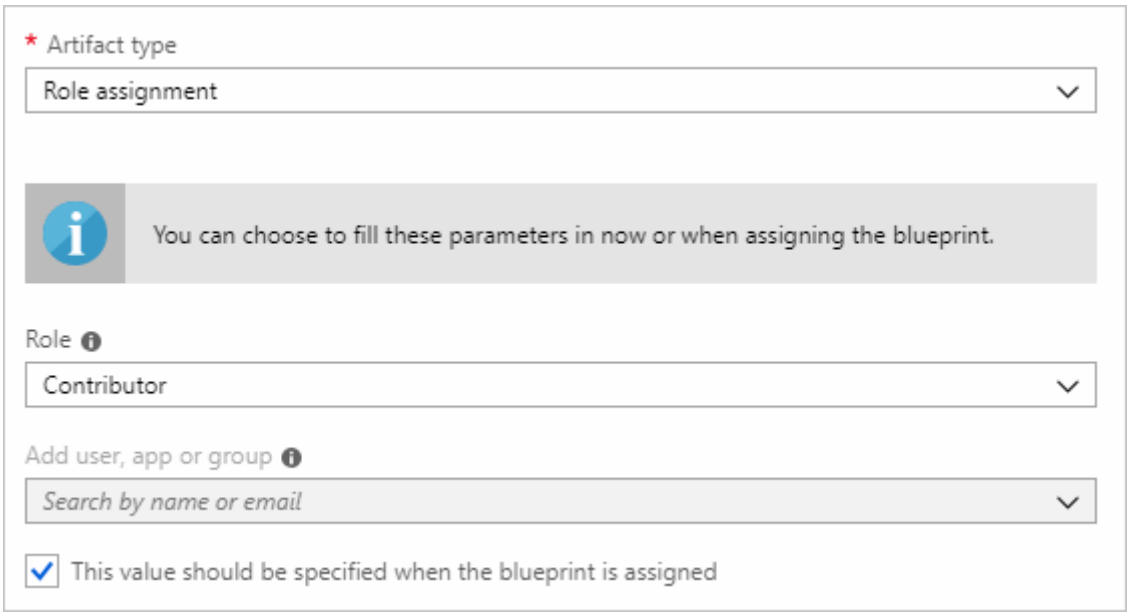
   Or, select **Create** from the **Getting started** page to go straight to creating a blueprint.

   If prompted, select Start with a blank blueprint.



3. Provide a **Blueprint name** such as **MyBlueprint**. (Use up to 48 letters and numbers, but no spaces or special characters). Leave **Blueprint description** blank for now.
4. In the **Definition location** box, select the ellipsis on the right, select your subscription where you want to save the blueprint, and choose **Select**.

5. Verify that the information is correct. The **Blueprint name** and **Definition location** fields can't be changed later. Then select **Next : Artifacts** at the bottom of the page or the **Artifacts** tab at the top of the page.
6. Add a role assignment at the subscription level:
7. Select the **+ Add artifact** row under **Subscription**. The **Add artifact** window opens on the right side of the browser.
8. Select **Role assignment** for **Artifact type**.
9. Under **Role**, select **Contributor**. Leave the **Add user, app or group** box with the check box that indicates a dynamic parameter.
10. Select **Add** to add this artifact to the blueprint.



Note: Most artifacts support parameters. A parameter that's assigned a value during blueprint creation is a *static parameter*. If the parameter is assigned during blueprint assignment, it's a *dynamic parameter*.

11. Add a policy assignment at the subscription level:
12. Select the **+ Add artifact** row under the role assignment artifact.
13. Select **Policy assignment** for **Artifact type**.
14. Change **Type** to **Built-in**. In **Search**, enter **tag**.
15. Click out of **Search** for the filtering to occur. Select **Append tag and its value to resource groups**.
16. Select **Add** to add this artifact to the blueprint.
17. Select the row of the policy assignment **Append tag and its value to resource groups**.
18. The window to provide parameters to the artifact as part of the blueprint definition opens and allows setting the parameters for all assignments (static parameters) based on this blueprint instead of during assignment (dynamic parameters). This example uses dynamic parameters during blueprint assignment, so leave the defaults and select **Cancel**.
19. Add a resource group at the subscription level:
20. Select the **+ Add artifact** row under **Subscription**.
21. Select **Resource group** for **Artifact type**.
22. Leave the **Artifact display name**, **Resource Group Name**, and **Location** boxes blank, but make sure that the check box is checked for each parameter property to make them dynamic parameters.
23. Select **Add** to add this artifact to the blueprint.
24. Add a template under the resource group:

25. Select the + **Add artifact** row under the **ResourceGroup** entry.

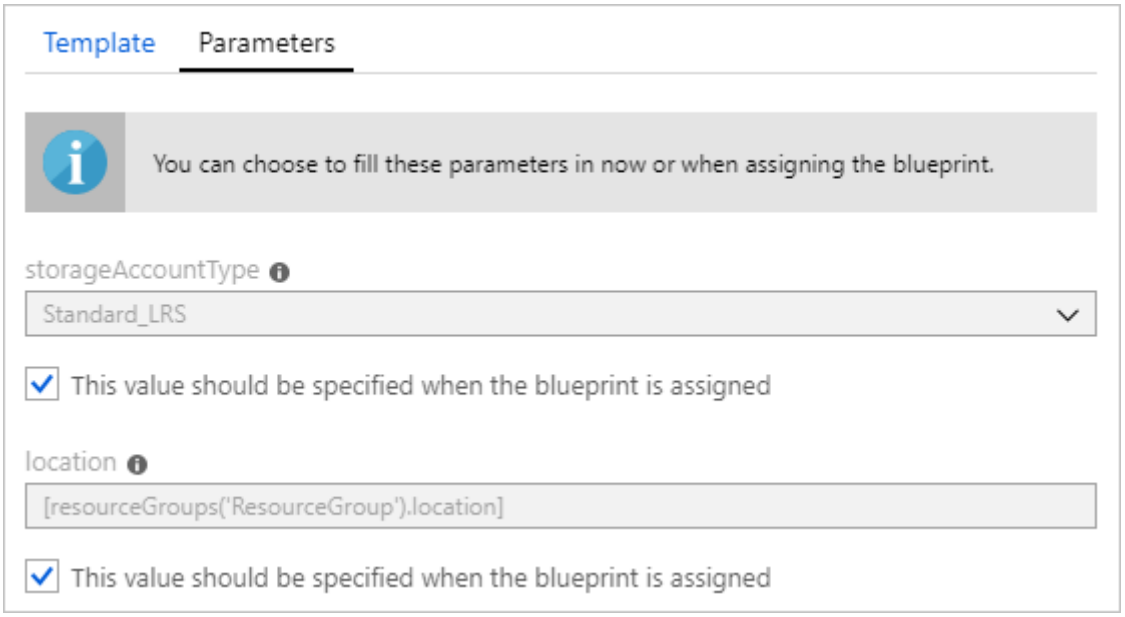
26. Select **Azure Resource Manager template** for **Artifact type**, set **Artifact display name** to **StorageAccount**, and leave **Description** blank.

27. On the **Template** tab in the editor box, paste the following Resource Manager template. After you paste the template, select the **Parameters** tab and note that the template parameters **storageAccountType** and **location** were detected. Each parameter was automatically detected and populated, but configured as a dynamic parameter.

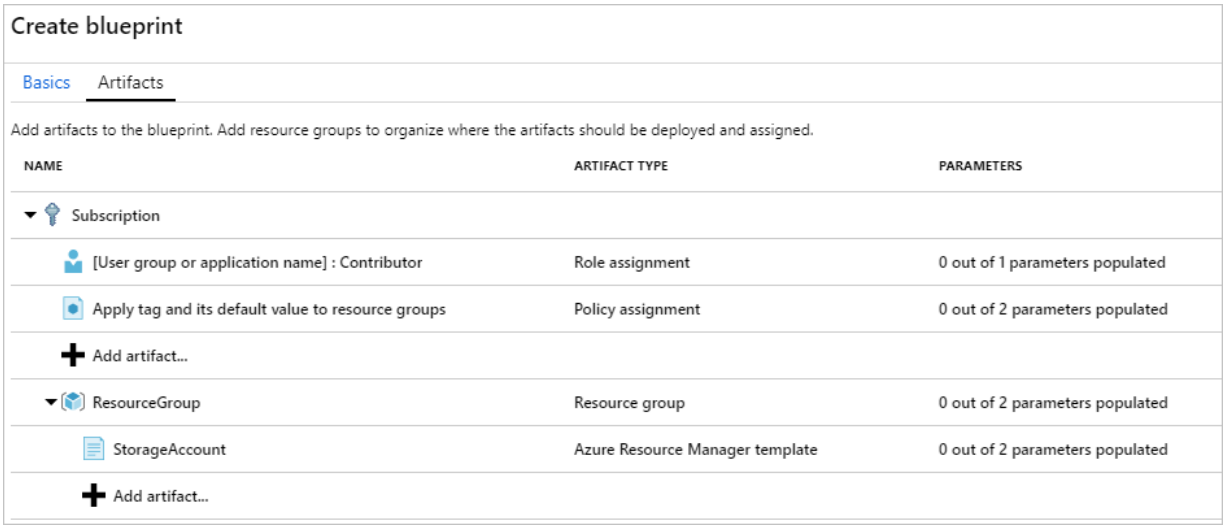
```json
        {
            "$schema": "https://schema.management.azure.com/schemas/      2015-01-01/deploymentTempla
            "contentVersion": "1.0.0.0",
            "parameters": {
                "storageAccountType": {
                    "type": "string",
                    "defaultValue": "Standard_LRS",
                    "allowedValues": [
                        "Standard_LRS",
                        "Standard_GRS",
                        "Standard_ZRS",
                        "Premium_LRS"
                    ],
                    "metadata": {
                        "description": "Storage Account type"
                    }
                },
                "location": {
                    "type": "string",
                    "defaultValue": "[resourceGroup().location]",
                    "metadata": {
                        "description": "Location for all resources."
                    }
                }
            },
            "variables": {
                "storageAccountName": "[concat('store', uniquestring      (resourceGroup().id))]"
            },
            "resources": [{
                "type": "Microsoft.Storage/storageAccounts",
                "name": "[variables('storageAccountName')]",
                "location": "[parameters('location')]",
                "apiVersion": "2018-07-01",
                "sku": {
                    "name": "[parameters('storageAccountType')]"
                },
                "kind": "StorageV2",
                "properties": {}
            }],
            "outputs": {
                "storageAccountName": {
                    "type": "string",
                    "value": "[variables('storageAccountName')]"
                }
            }
        }
```

28. Under the Parameters tab, clear the **storageAccountType** check box and note that the drop-down list contains only values included in the Resource Manager template under **allowedValues**. Select the box to set it back to a dynamic parameter.

29. Select **Add** to add this artifact to the blueprint.



30. Your completed blueprint should look similar to the following. Notice that each artifact has ***x* out of *y* parameters populated** in the **Parameters** column. The dynamic parameters are set during each assignment of the blueprint.



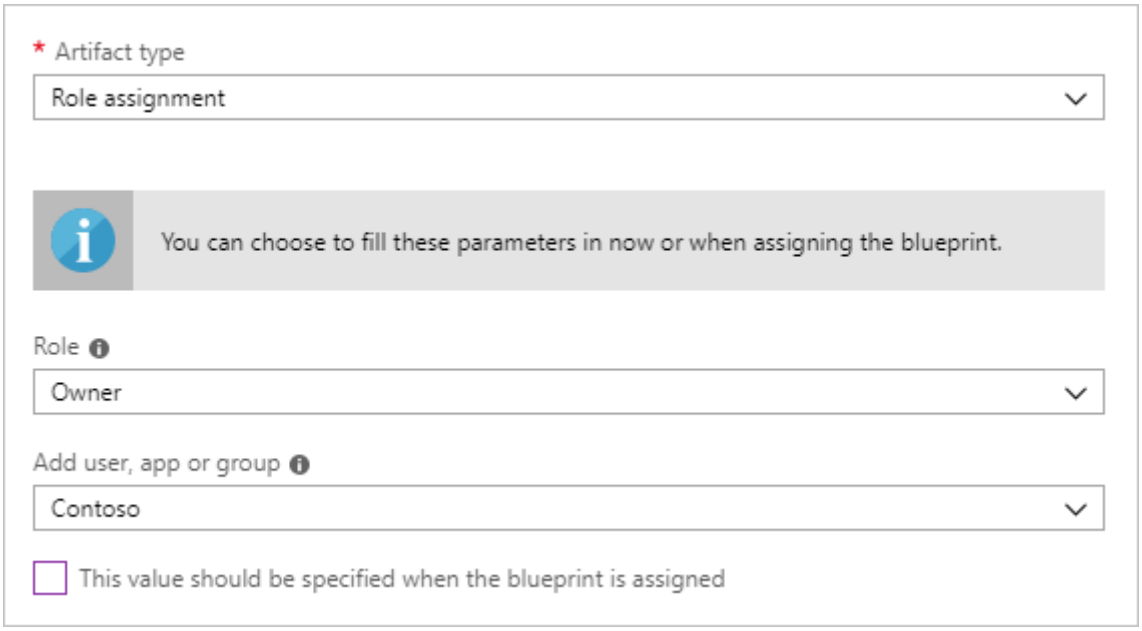
31. Now that all planned artifacts have been added, select **Save Draft** at the bottom of the page.

40.1.2 Task 2: Edit a blueprint

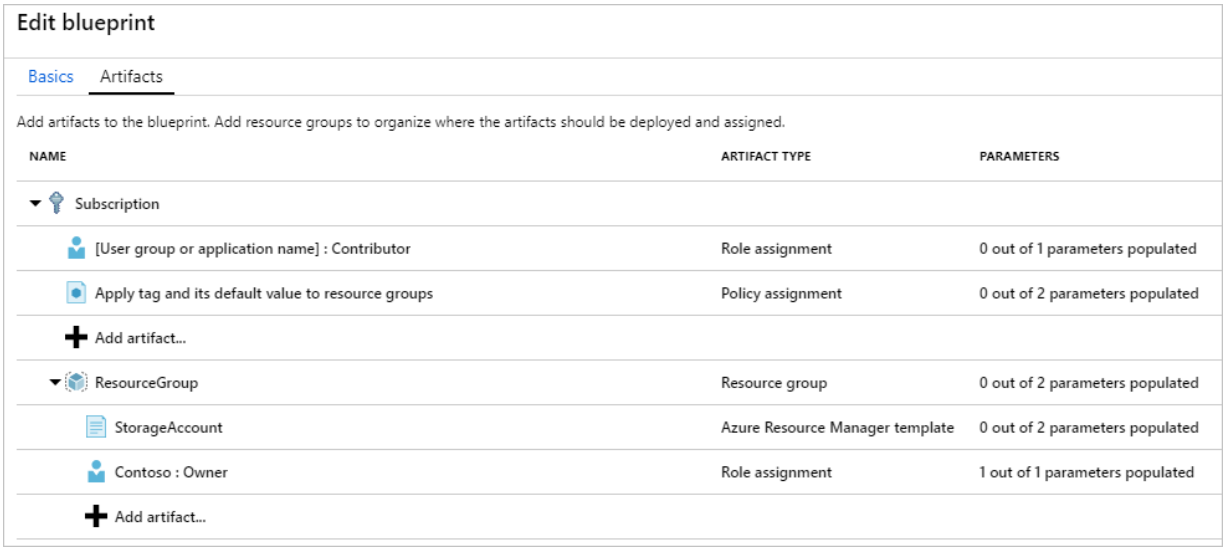
In Create a blueprint, you didn't provide a description or add the role assignment to the new resource group. You can fix both by following these steps:

1. Select **Blueprint definitions** from the page on the left.
2. In the list of blueprints, right-click the one that you previously created and select **Edit blueprint**.
3. In **Blueprint description**, provide some information about the blueprint and the artifacts that compose it. In this case, enter something like: **This blueprint sets tag policy and role assignment on the subscription, creates a ResourceGroup, and deploys a resource template and role assignment to that ResourceGroup.**
4. Select **Next : Artifacts** at the bottom of the page or the **Artifacts** tab at the top of the page.
5. Add a role assignment under the resource group:
6. Select the **+ Add artifact** row directly under the **ResourceGroup** entry.
7. Select **Role assignment** for **Artifact type**.

8. Under **Role**, select **Owner**, and clear the check box under the **Add user, app or group** box.
9. Search for and select a user, app, or group to add. This artifact uses a static parameter set the same in every assignment of this blueprint.
10. Select **Add** to add this artifact to the blueprint.



11. Your completed blueprint should look similar to the following. Notice that the newly added role assignment shows **1 out of 1 parameters populated**. That means it's a static parameter.



12. Select **Save Draft** now that it has been updated.

40.1.3 Task 3: Publish a blueprint

Now that all the planned artifacts have been added to the blueprint, it's time to publish it. Publishing makes the blueprint available to be assigned to a subscription.

1. Select **Blueprint definitions** from the page on the left.
2. In the list of blueprints, right-click the one you previously created and select **Publish blueprint**.
3. In the pane that opens, provide a **Version** (letters, numbers, and hyphens with a maximum length of 20 characters), such as **v1**. Optionally, enter text in **Change notes**, such as **First publish**.
4. Select **Publish** at the bottom of the page.

40.1.4 Task 4: Assign a blueprint

After a blueprint has been published, it can be assigned to a subscription. Assign the blueprint that you created to one of the subscriptions under your management group hierarchy. If the blueprint is saved to a subscription, it can only be assigned to that subscription.

1. Select **Blueprint definitions** from the page on the left.
2. In the list of blueprints, right-click the one that you previously created (or select the ellipsis) and select **Assign blueprint**.
3. On the **Assign blueprint** page, in the **Subscription** drop-down list, select the subscriptions that you want to deploy this blueprint to. *Skip this step if you are using an Azure Pass or lab Hoster solution*

   If there are supported Enterprise offerings available from Azure Billing, a Create new link is activated under the Subscription box. Follow these steps:

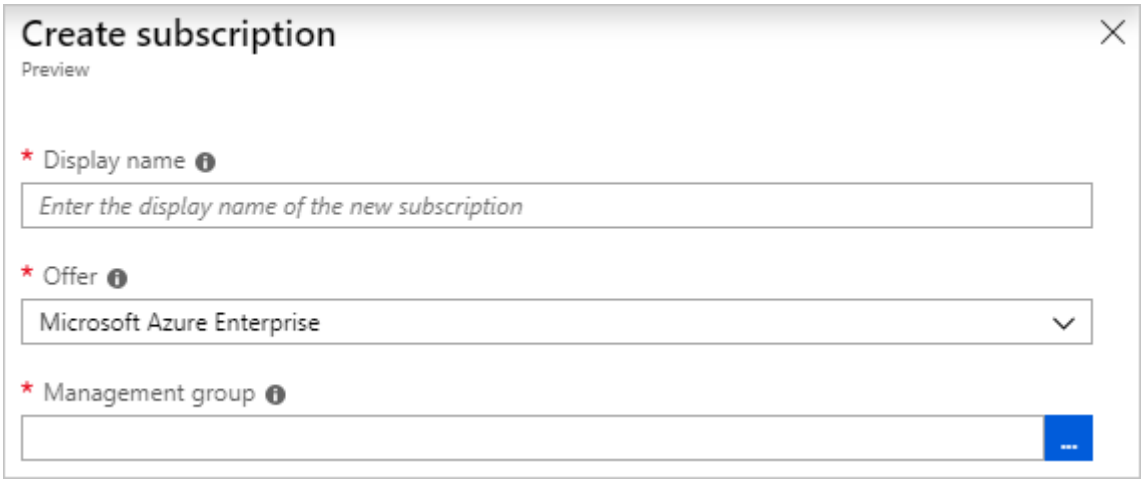
   a. Select the **Create new** link to create a new subscription instead of selecting existing ones.

   b. Provide a **Display name** for the new subscription.

   c. Select the available **Offer** from the drop-down list.

   d. Use the ellipsis to select the management group that the subscription will be a child of.

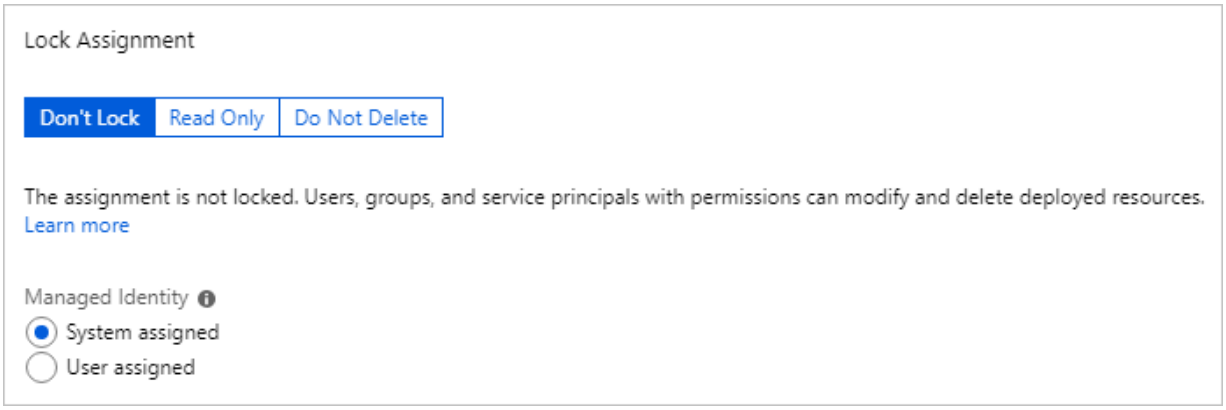
   e. Select **Create** at the bottom of the page.



   Important: The new subscription is created immediately after you select **Create**.

   Note: An assignment is created for each subscription that you select. You can make changes to a single subscription assignment at a later time without forcing changes on the remainder of the selected subscriptions.

4. For **Assignment name**, provide a unique name for this assignment.
5. In **Location**, select a region for the managed identity and subscription deployment object to be created in. Azure Blueprint uses this managed identity to deploy all artifacts in the assigned blueprint.
6. Leave the **Blueprint definition version** drop-down selection of **Published** versions on the **v1** entry. (The default is the most recently published version.)
7. For **Lock Assignment**, leave the default of **Don't Lock**.



8. Under **Managed Identity**, leave the default of **System assigned**.
9. For the subscription level role assignment [**User group or application name**] **: Contributor**, search for and select a user, app, or group. *You may skip this step if you have no users or groups*
10. For the subscription level policy assignment, set **Tag Name** to **CostCenter** and the **Tag Value** to **ContosoIT**.
11. For **ResourceGroup**, provide a **Name** of **StorageAccount** and a **Location** of **East US 2** from the drop-down list.

    Note: For each artifact that you added under the resource group during blueprint definition, that artifact is indented to align with the resource group or object that you'll deploy it with. Artifacts that either don't take parameters or have no parameters to be defined at assignment are listed only for contextual information.
12. On the Azure Resource Manager template **StorageAccount**, select **Standard_GRS** for the **storageAccountType** parameter.
13. Read the information box at the bottom of the page, and then select **Assign**. *Due to the limitations of lab environments you may receive an error. This can be ignored and continue to the next task*

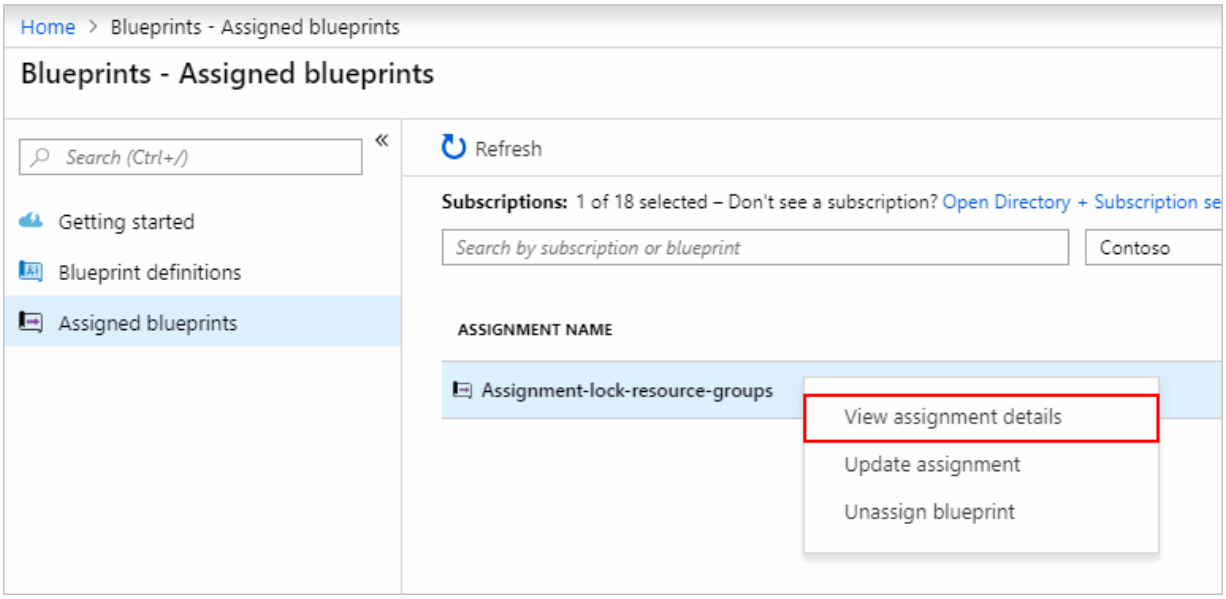
40.1.5 Task 5: Track deployment of a blueprint

When a blueprint has been assigned to one or more subscriptions, two things happen:

- The blueprint is added to the **Assigned blueprints** page for each subscription.
- The process of deploying all the artifacts defined by the blueprint begins.

Now that the blueprint has been assigned to a subscription, verify the progress of the deployment:

1. Select **Assigned blueprints** from the page on the left.
2. In the list of blueprints, right-click the one that you previously assigned and select **View assignment details**.



3. On the **Blueprint assignment** page, validate that all artifacts were successfully deployed and that there were no errors during the deployment. If errors occurred, see Troubleshooting blueprints for steps to determine what went wrong.

40.1.6 Task 6: Unassign a blueprint

If you no longer need a blueprint assignment, remove it from a subscription. The blueprint might have been replaced by a newer blueprint with updated patterns, policies, and designs. When a blueprint is removed, the artifacts assigned as part of that blueprint are left behind. To remove a blueprint assignment, follow these steps:

1. Select **Assigned blueprints** from the page on the left.
2. In the list of blueprints, select the blueprint that you want to unassign. Then select the **Unassign blueprint** button at the top of the page.
3. Read the confirmation message and then select **OK**.

40.1.7 Task 6: Delete a blueprint

1. Select **Blueprint definitions** from the page on the left.
2. Right-click the blueprint that you want to delete, and select **Delete blueprint**. Then select **Yes** in the confirmation dialog box.

Note: Deleting a blueprint in this method also deletes all published versions of the selected blueprint. To delete a single version, open the blueprint, select the **Published versions** tab, select the version that you want to delete, and then select **Delete This Version**. Also, you can't delete a blueprint until you've deleted all blueprint assignment of that blueprint definition.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this lab.

41 Module 2: Lab 1 - Monitor & Autoscale

Autoscale is a built-in feature of Cloud Services, Mobile Services, Virtual Machines, and Websites that helps applications perform their best when demand changes. Of course, performance means different things for different applications. Some apps are CPU-bound, others memory-bound. For example, you could have a web app that handles millions of requests during the day and none at night. Autoscale can scale your service by any of these-or by a custom metric you define.

41.1 Exercise 1: Lab setup

1. Go to the following URL in the browser:

   https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2FMicrosof
   500-Azure-Security%2Fmaster%2FAllfiles%2FLabs%2FMod2_Lab01%2Ftemplate.json

   This will deploy a new app and app service plan from a template that can then be used to demonstrate the scale up options in AZ500 Mod2 Lab 1.

2. Select **Create a new Resource Group**
3. Type a **unique** name for the **Site Name** and **Service Plan**.
4. Agree to the terms and click **Purchase**

41.2 Exercise 2: Create your first Autoscale setting

Let's now go through a simple step-by-step walkthrough to create your first Autoscale setting.

1. Open the **Autoscale** blade in Azure Monitor and select a resource that you want to scale. You can select the app service plan that you created during the setup
2. Note that the current instance count is 1. Click **Custom autoscale**.
3. Provide a name for the scale setting, and then click **Add a rule**.
4. Notice the scale rule options that open as a context pane on the right side. By default, this sets the option to scale your instance count by 1 if the CPU percentage of the resource exceeds 70 percent. Leave it at its default values and click **Add**.
5. You've now created your first scale rule. Note that the UI recommends best practices and states that "It is recommended to have at least one scale in rule." To do so:
   - Click **Add a rule**.
   - Set **Operator** to **Less than**.
   - Set **Threshold** to **20**.
   - Set **Operation** to **Decrease count by**.
6. Click **Add**
7. Click **Save**.

   Note: If you receive an error "Microsoft.insights not registered" (Add button grayed out) go to your Subscription blade and under Resource Providers register "Microsoft.insights" and wait a few minutes for registration then retry. If it does not register, continue to to the next exercise.

Congratulations! You've now successfully created your first scale setting to autoscale your web app based on CPU usage.

41.3 Exercise 3: Scale based on a schedule

In addition to scale based on CPU, you can set your scale differently for specific days of the week.

1. Click **Add a scale condition**.
2. Setting the scale mode and the rules is the same as the default condition.
3. Select **Repeat specific days** for the schedule.
4. Select the days and the start/end time for when the scale condition should be applied.

41.4 Exercise 4: Scale differently on specific dates

In addition to scale based on CPU, you can set your scale differently for specific dates.

1. Click **Add a scale condition**.
2. Setting the scale mode and the rules is the same as the default condition.
3. Select **Specify start/end dates** for the schedule.

4. Select the start/end dates and the start/end time for when the scale condition should be applied. Click **Save**.

41.5 Exercise 5: View the scale history of your resource

1. Whenever your resource is scaled up or down, an event is logged in the activity log. You can view the scale history of your resource for the past 24 hours by switching to the **Run history** tab.
2. If you want to view the complete scale history (for up to 90 days), select **View more details in the Activity Log**. The activity log opens, with Autoscale pre-selected for your resource and category.

41.6 Exercise 6: View the scale definition of your resource

1. Autoscale is an Azure Resource Manager resource. You can view the scale definition in JSON by switching to the **JSON** tab.
2. You can make changes in **JSON** directly, if required. These changes will be reflected after you save them.

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this lab.

42 Module 3: Classify a SQL Database

42.1 Exercise 1: Classify your SQL Database

42.1.1 Task 1: Lab Setup

1. In your browser, navigate to the following URL to open the ARM template:

   `https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2F`

2. Under **Resource group** click create new and use the default name "**Mod3Lab1**"
3. You can use the default populated **SQL server name** with a **unique** string added to make a **globaly unique** name
4. Click **Purchase**. warning **Note**: You must wait for the SQL database with the test data to deploy
5. Sign-in to the Azure portal.
6. **Select** the resource group of **Mod3Lab1**
7. Click **your unique SQL server name**
8. Under the **Security heading** in the **Azure SQL Database** pane, navigate to **Advanced Data Security**.
9. Select **ON** under **Advanced Data Security**
10. **Click Save**
11. Return to the **Mod3Lab1** resource group
12. **Select** the **SQL databse AZ500LabDb (your unique SQL server name/AZ500LabDb)**
13. **Click** Advanced Data Security again
14. **Click** the bar at the top **Complete Advanced Data Security setup by selecting a storage account for Vulnerability Assesment**
15. Make sure the option for **Advanced Data Security** is set to **On**
16. **Click** storage account
17. **Click** create new
18. For the name use **mod3lab1yourname** replacing **yourname** with your name to make it unique but memorable

19. Click **OK**
20. **Select** the **save** option
21. **Return** to the previous **Advanced Data Security** pane
22. **Select** the **Data discovery** and **classification** card.

42.2 Exercise 2: Begin Classification

1. To begin classifying your data, select the **Classification tab** at the top of the window.
2. The classification engine scans your database for columns containing potentially sensitive data and provides a list of recommended column classifications.
3. To view and apply classification recommendations:
   - View the list. To view the list of recommended column classifications, select the **recommendations** panel at the top of the window. (**Blue Bar**)
4. Click **Select all** in the top left to select all reccomendations
   - To manually classify columns as an alternative to or in addition to the recommendation-based classification, in the top menu of the window, select Add classification.
   - In the Add classification blade, configure the five fields that display, and then select Add classification:
     - **Schema name**
     - **Table name**
     - **Column name**
     - **Information type**
     - **Sensitivity label.**
5. Click **Accept Selected Reccomendations**
6. To complete your classification and persistently label (**tag**) the database columns with the new classification metadata, select **Save**.
7. Upon returning to the **Advanced Data Security** pane you will be able to see in the overview pane for Data classification an overview of the data that was classified.

Results: You have now classified information in a SQL database on Azure for GPDR and data protection compliance

43 Module 3: Auditing a Database

43.1 Exercise 1: Enable auditing on your database

43.1.1 Task 1: Lab Setup

1. In your browser, navigate to the following URL to open the ARM template:

   `https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2F`

2. **Under Resource** group click create new and use the default name "**Mod3Lab2**"
3. You can use the default **populated SQL server** name with a **unique** string added to make a **globaly unique** name
4. Click **Purchase** warning **Note**: You must wait for the SQL database with the test data to deploy

43.1.2 Task 2: Enable auditing on your database

1. Select your resource group created in the lab setup
2. **Select** the SQL server **your unique SQL Server name**
3. **Under Security**, select **Auditing**
4. **Switch Auditing** to **ON**.
5. Select **storage** as the location to send the audit logs to

6. Click **Configure**
7. Select **Your Subscription**
8. Click **Storage account** then if necessary click **Create New**. The Create storage account blade should open..
9. Name the storage account **mod3lab2yourname** ensuring you replace **yourname** with a unique name using lowercase letters
10. **Click OK**.
11. Change the retention days to **5** and click **OK**
12. Click **Save** to save the **auditing configuration**

43.2 Exercise 2: Review audit logs

43.2.1 Task 1: Review audit logs on the SQL DB.

1. To review audit logs for a database return to the resource group created in the lab setup
2. Click **AZ500LabDb (your unique SQL Server name/AZ500LabDb)** to select your test database
3. **Under Security**, select **Auditing**

   Note: The Auditing looks off here but it is set on the underlying server level so it is turned on for this database
4. Click **View Audit Logs**.

Note: Here you will review the output of the audit logs of the database including any attempted SQL injections. Since this is a test database created recently, there will be minimal audits if any in the log at the current time.

If server auditing is enabled, the database-configured audit will exist side-by-side with the server audit. Notice that you can select for audit logs to be written to an Azure storage account, to a Log Analytics workspace for consumption by Azure Monitor logs, or to Event Hub for consumption using an event hub. You can configure any combination of these options, and audit logs will be written to each.

Results: You have now setup up auditing on a SQL database and reviewed where to view the auditing output

44 Module 3: Analyze audit logs and reports

44.1 Exercise 1: Get started with SQL database auditing

44.1.1 Task 0: Lab Setup

1. In your browser, navigate to the following URL to open the ARM template:

   `https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2F`
2. Under **Resource group** click create new and use the default name "**Mod3Lab3**"
3. You can use the default populated SQL server name with a unique string added to make a globaly unique name.
4. Click **Purchase**

44.1.2 Task 1 - Set up auditing for your database

1. **Navigate** to **Resource Groups**
2. **Select** your resource group created above ("**Mod3Lab3**" if you chose the same name as the instructions)
3. **Click** your **SQL Server** name
4. On the left pane click auditing
5. **Change** auditing to **ON** from **OFF**. warning **Note**: You now have the option to select where you wish audit logs to be written to.
6. **Select** all **3 options**, **Storage**, **Log analytics**, **Event hub**.

7. **Under Storage**, **Click** configure.
8. **Click subscription** and select your subscription
9. **Click** storage account
10. Under create storage account input a unique name (**e.g. mod3lab3yourname**)
11. **Click OK**.
12. When validated click **OK** again
13. Under Log analytics click configure
14. **Click** create new workspace
15. Enter the following settings

Log Analytics Workspace	Subscription	Resource Group	Location	Pricing Tier
Mod3Lab3YOURNAME	Your Subscription	The RG you created in the lab setup	East US	Per GB (2018)

16. **Click OK**. warning **Note**: Setting up Event Hub requires extra steps as the Azure portal does not allow you to create an event hub from this location
17. To set up an **Event Hub** for configuration, click **Azure Cloud Shell** at the top of the **Portal**.
18. **Enter** the following **Powershell Commands**.

    Note Replace the section **{GlobalUniqueName}** with a globally unique name

    ```
    New-AzEventHubNamespace -ResourceGroupName Mod3Lab3  -NamespaceName {GlobalUniqueName} -Location e
    ```

    ```
    New-AzEventHub -ResourceGroupName Mod3Lab3 -NamespaceName {GlobalUniqueName}  -EventHubName Mod3La
    ```

    Return to the Advanced Data Security blade in portal. (**Note** The Event Hub Namespace will be the unique name you specified).
19. When these commands have completed click **configure under event hubs**
20. **Select** the following information

Subscription	Hub Namespace	Hub Name	Hub Policy Name
Your Subscription	Mod3Lab3	mod3lab3	RootManageSharedAccessKey

21. **Click OK**
22. You can now click **Save** on the **Auditing Settings** page

    Result: You have now turned on auditing for your SQL database
23. To access the logs return to the **Resource group** where the **SQL Database** and **Server reside**
24. **Select** the **Mod3Lab3** Log analytics workspace you created earlier
25. **Click** logs
26. **Click Get Started**
27. In the query space enter the following code and **click Run**.

    ```
    Event | where Source  == "MSSQLSERVER"
    ```

28. You will not see any results, please read the below warning warning **Note**: Because we have set up logs on a new database with test data, there are minimal log entries available to see. To show how logs are displayed, we can use the example log analytics website that is populated with example data.

44.1.3 Task 2 - Analyze audit logs and reports

1. Visit `https://portal.loganalytics.io/demo` in a new web browser tab, this will direct you to a demo log analytics workspace with demo data populated

2. In the query space enter the following code and **click Run**.

   ```
   Event | where Source  == "MSSQLSERVER"
   ```

3. From here you can expand some of the example audit logs to view what they would look like in a live system

4. In the query space enter the following code and **Click Run**.

   ```
   Event
   | where EventLevelName == "Error"
   | where TimeGenerated > ago(1d)
   | where Source != "HealthService"
   | where Source != "Microsoft-Windows-DistributedCOM"
   | summarize count() by Source
   ```

5. **Click chart**

6. From here you can review how log data can be displayed as chart data

7. **Click** the **Stacked Column** drop down and select **Pie**

8. Here you can see the same data as a different chart

9. From the top right of the window you can **Click Export** to export the data as **CSV**.

The query language used by log analytics is called the Kusto query language. The full documentation for this language can be found here **`https://docs.microsoft.com/en-us/azure/kusto/query/`**

Results: You have now completed the setup of logs on a SQL database and run queries against the log analytics workspace to view the audit logs that will be generated

45 Module 4: Lab 1 - Azure Monitor

Azure Monitor maximizes the availability and performance of your applications and services by delivering a comprehensive solution for collecting, analyzing, and acting on telemetry from your cloud and on-premises environments. It helps you understand how your applications are performing and proactively identifies issues affecting them and the resources they depend on.

In this lab you will configure Azure Monitor to:

- Collect data from an Azure virtual machine.
- Use Application Insights to monitor your website.

45.1 Exercise 1: Collect data from an Azure virtual machine with Azure Monitor

Azure Monitor can collect data directly from your Azure virtual machines into a Log Analytics workspace for detailed analysis and correlation. Installing the Log Analytics VM extension for Windows and Linux allows Azure Monitor to collect data from your Azure VMs. This exercise shows you how to configure and collect data from your Azure Linux or Windows VMs using the VM extension with a few easy steps.

45.1.1 Task 1: Deploy an Azure VM to monitor.

1. Open the Azure Cloud Shell and run the following two commands to create a Resource Group and Azure VM that you will use to monitor:

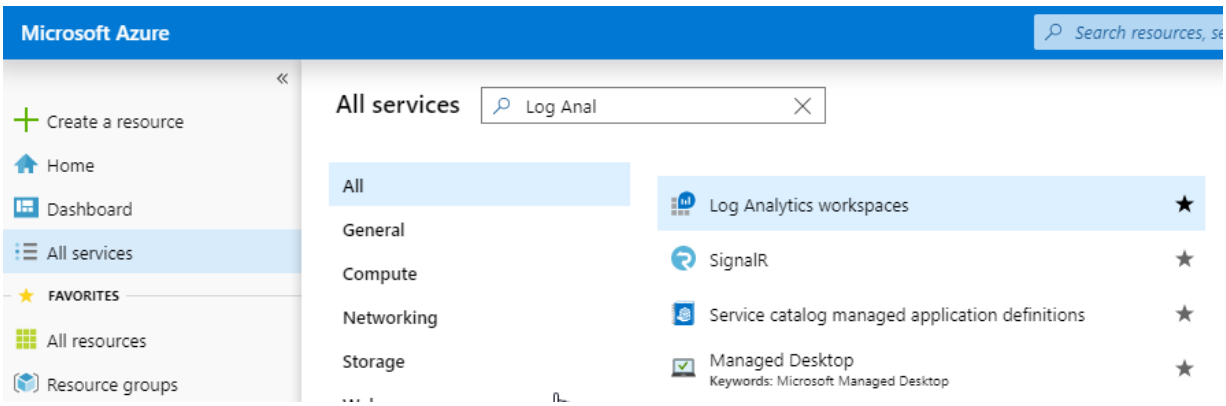
   ```
   New-AzResourceGroup -Name myResourceGroup -Location EastUS
   ```

   ```
   New-AzVm -ResourceGroupName "myResourceGroup" -Name "myVM" -Location "East  US" -VirtualNetworkNam
   ```

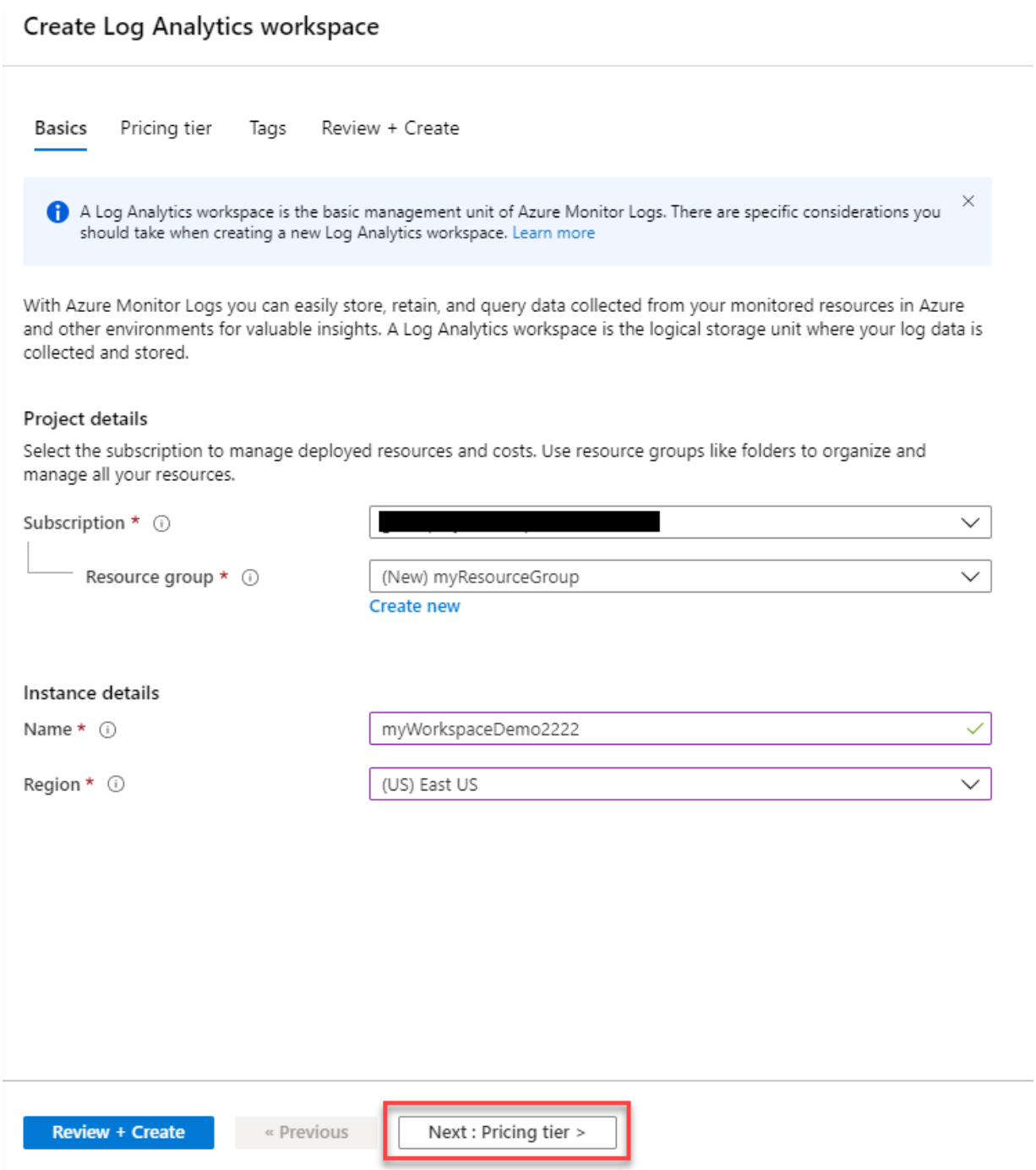
2. When prompted for credentials enter **LocalAdmin** as the User and use the password **Pa55w.rd1234**

45.1.2 Task 2: Create a workspace

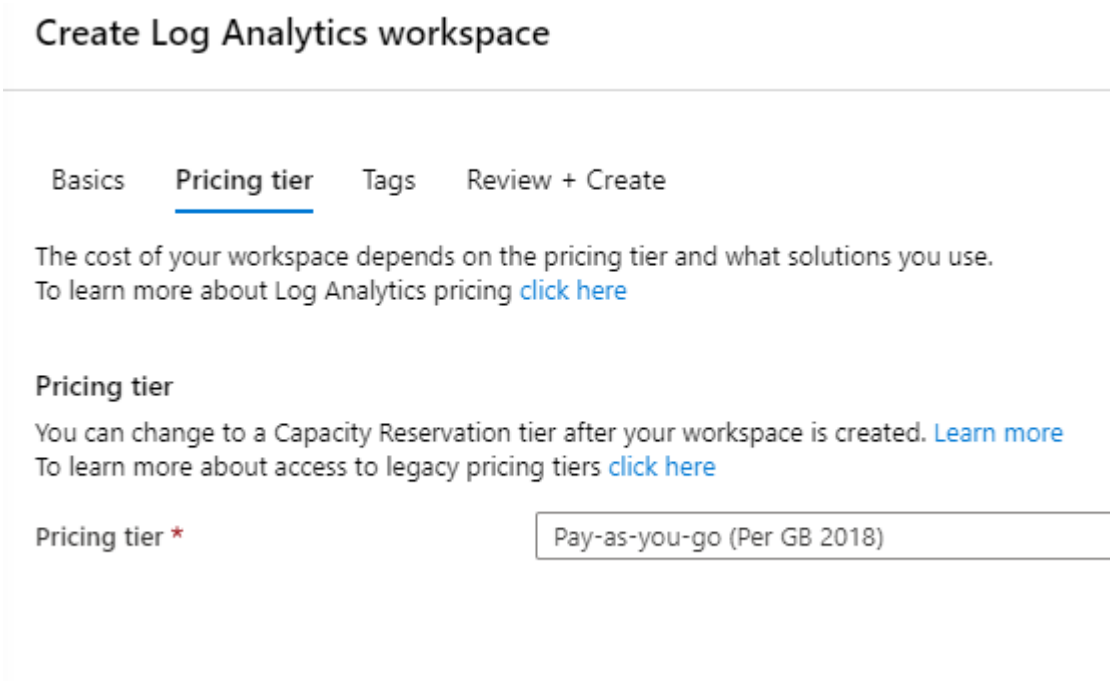
1. In the Azure portal, select **All services**. In the list of resources, type **Log Analytics**. As you begin typing, the list filters based on your input. Select **Log Analytics workspaces**.



2. Select + **Add**, and then select choices for the following items:
   - Select a **Subscription** to link to by selecting from the drop-down list if the default selected is not appropriate.
   - For **Resource Group**, select **myResourceGroup** which is the Resource Group that contains the VM you created in Task 1.
   - Provide a name for the new **Log Analytics workspace**, such as *myWorkspaceDemo*.
   - Select the **EastUS** as the location.
   - Click **Next: Pricing tier**



- Leave the pricing Tier as **Per Gb (2018)**



3. After providing the required information on the **Log Analytics workspace** pane, select **Review + Create** then click **Create**.
4. While the information is verified and the workspace is created, you can track its progress under **Notifications** from the menu.

45.1.3 Task 2: Enable the Log Analytics VM Extension

For Windows and Linux virtual machines already deployed in Azure, you install the Log Analytics agent with the Log Analytics VM Extension. Using the extension simplifies the installation process and automatically configures the agent to send data to the Log Analytics workspace that you specify. The agent is also upgraded automatically when a newer version is released, ensuring that you have the latest features and fixes. Before proceeding, verify the VM is running otherwise the process will fail to complete successfully.

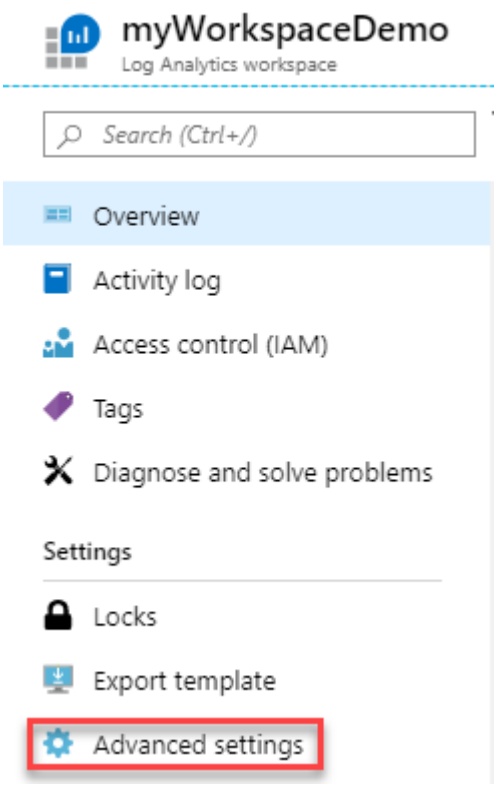
1. In the Azure portal, select **All services** found in the upper left-hand corner. In the list of resources, type **Log Analytics**. As you begin typing, the list filters based on your input. Select **Log Analytics workspaces**.
2. In your list of Log Analytics workspaces, select **myWorkspaceDemo** created earlier.

   Note: The name of your workspace may be different to **myWorkspaceDemo**.
3. On the left-hand menu, under Workspace Data Sources, select **Virtual machines**.
4. In the list of **Virtual machines**, select a virtual machine you want to install the agent on. Notice that the **Log Analytics connection status** for the VM indicates that it is **Not connected**.
5. In the details for your virtual machine, select **Connect**. The agent is automatically installed and configured for your Log Analytics workspace. This process takes a few minutes, during which time the **Status** shows **Connecting**.
6. After you install and connect the agent, the **Log Analytics connection status** will be updated with **This workspace**.

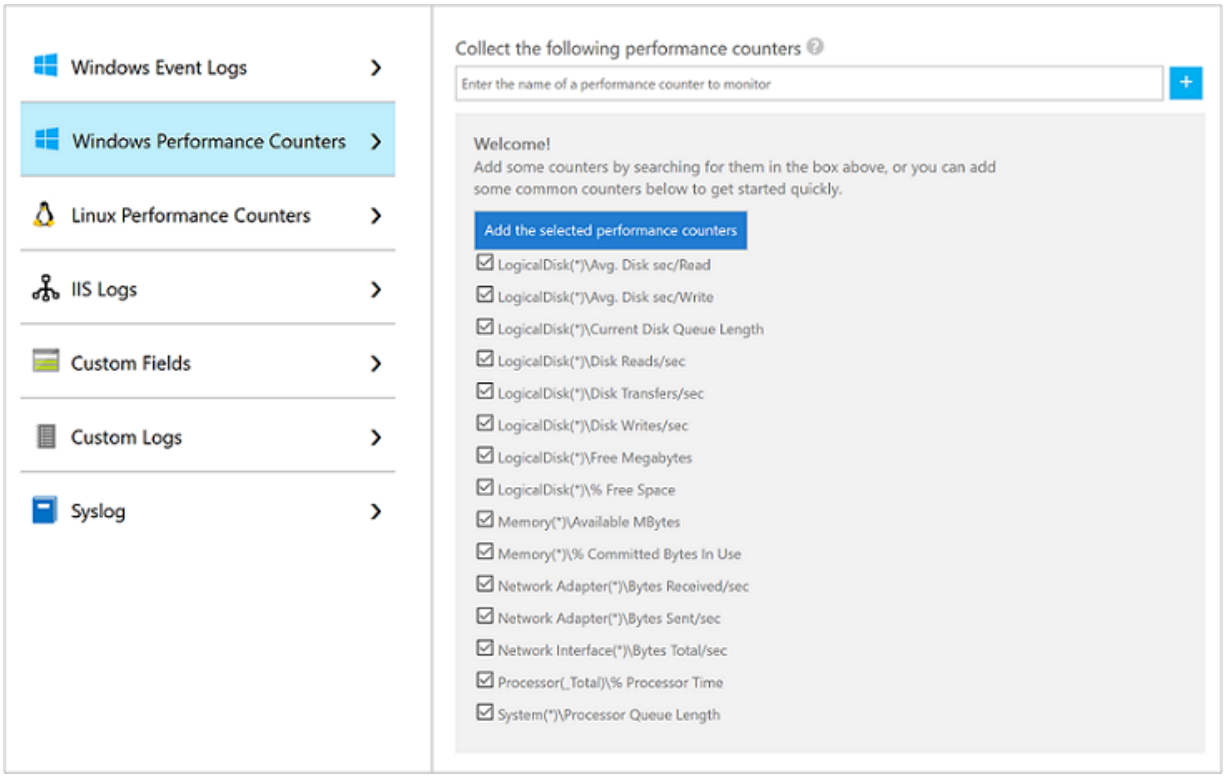
45.1.4 Task 3: Collect event and performance of a Windows VM.

Azure Monitor can collect events from the Windows event logs or Linux Syslog and performance counters that you specify for longer term analysis and reporting, and take action when a particular condition is detected. Follow these steps to configure collection of events from the Windows system log and Linux Syslog, and several common performance counters to start with.

1. On the Log Analytics workspaces blade, select **Advanced settings**.



2. Select **Data**, and then select **Windows Event Logs**.
3. You add an event log by typing in the name of the log. Type **System** and then select the plus sign +.
4. In the table, check the severities **Error** and **Warning**.
5. Select **Save** at the top of the page to save the configuration.
6. Select **Windows Performance Counters** to enable collection of performance counters on a Windows computer.
7. When you first configure Windows Performance counters for a new Log Analytics workspace, you are given the option to quickly create several common counters. They are listed with a checkbox next to each.



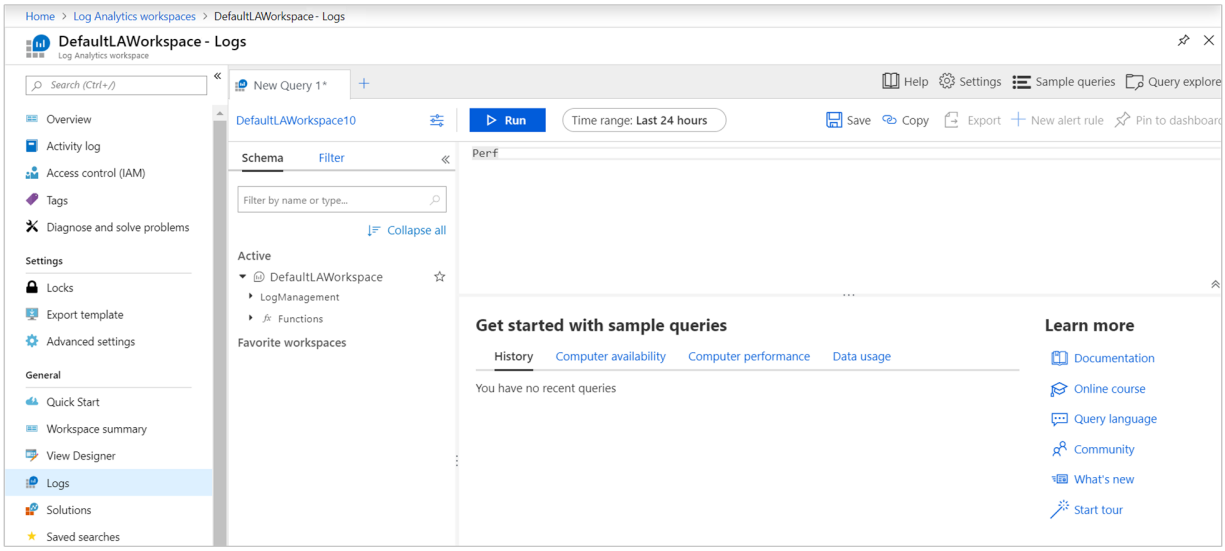
Select **Add the selected performance counters**. They are added and preset with a ten second collection sample interval.

8. Select **Save** at the top of the page to save the configuration.

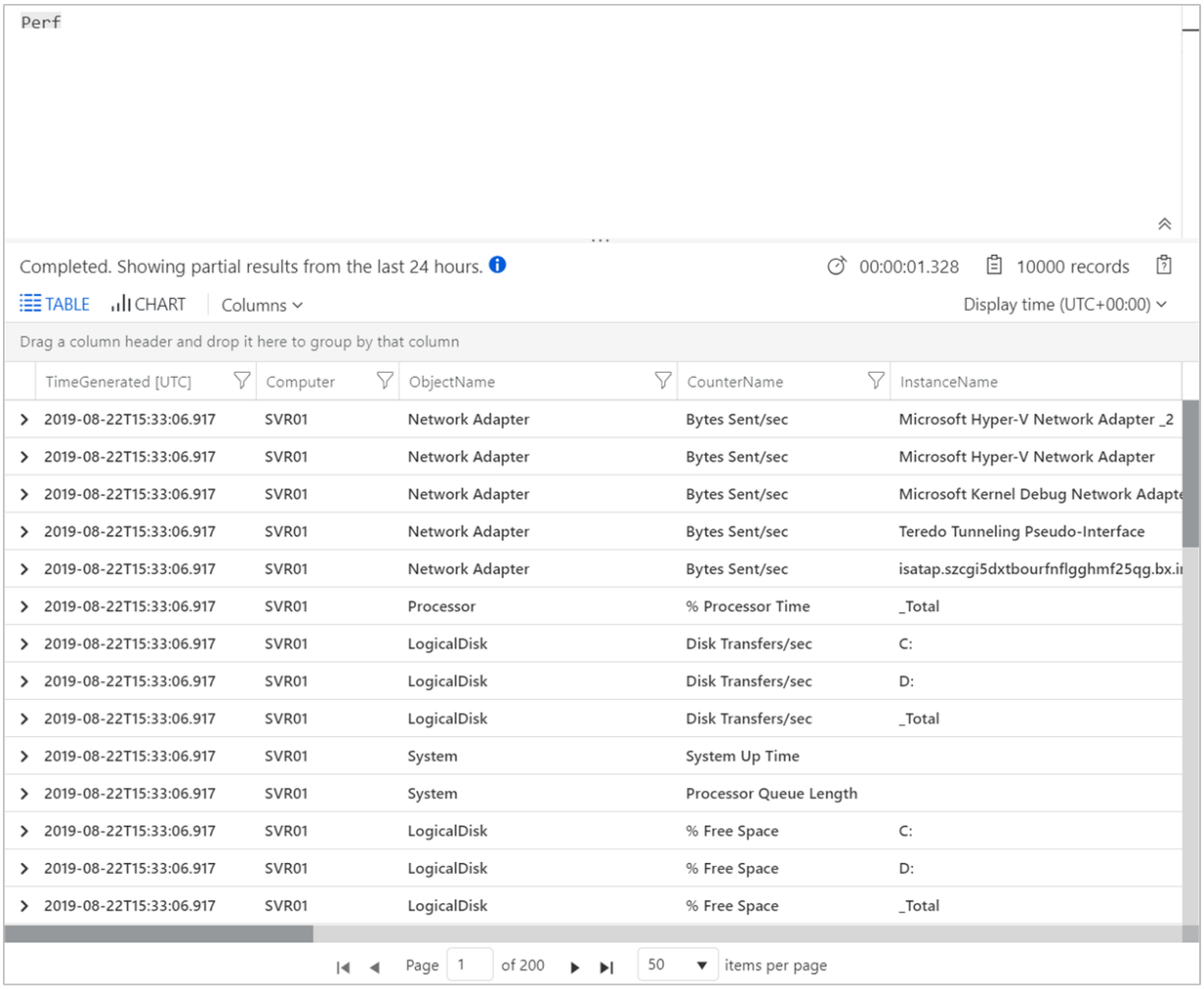
45.1.5 Task 4: View data collected

Now that you have enabled data collection, lets run a simple log search example to see some data from the target VMs.

1. In the selected workspace, from the left-hand pane, select **Logs**.
2. Click Get started. On the Logs query page, type `Perf` in the query editor and select **Run**.



For example, the query in the following image returned 10,000 performance records. Your results will be significantly less due to the VM having only being ran for a few minutes.



45.2 Exercise 2: Monitor Websites with Azure Monitor Application Insights

With Azure Monitor Application Insights, you can easily monitor your website for availability, performance, and usage. You can also quickly identify and diagnose errors in your application without waiting for a user to report them. Application Insights provides both server-side monitoring as well as client/browser-side monitoring capabilities.

This exercise guides you through adding the open source Application Insights JavaScript SDK which allows you to understand the client/browser-side experience for visitors to your website.

45.2.1 Task 1: Enable Application Insights

Application Insights can gather telemetry data from any internet-connected application, running on-premises or in the cloud. Use the following steps to start viewing this data.

1. Select **Create a resource** > **IT & Management tools** > **Application Insights**.

   A configuration box appears; use the following table to fill out the input fields.

Settings	Value
Name	Enter a Globally Unique Value
Resource Group	myResourceGroup
Location	East US

2. Click **Review + Create** then click **Create**.

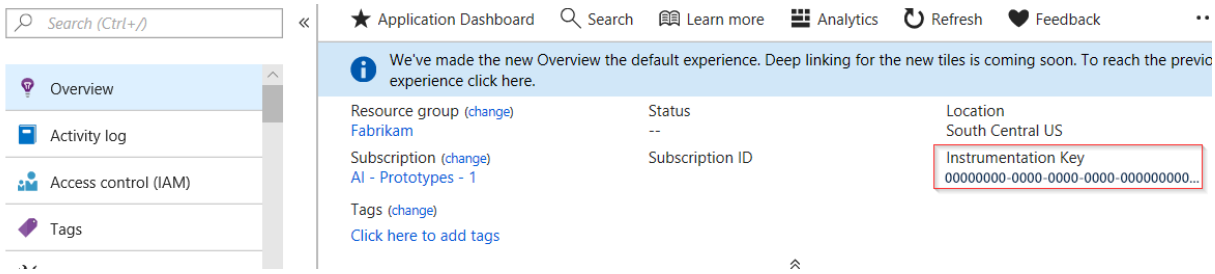
45.2.2 Task 2: Create an HTML file

1. On your local computer, create a file called `hello_world.html`. For this example the file will be placed on the root of the C: drive at `C:\hello_world.html`.
2. Copy the script below into `hello_world.html`:

```
<!DOCTYPE html>
<html>
<head>
<title>
Azure Monitor Application Insights
</title>
<script>
    var appInsights=window.appInsights||function(config)
{
function r(config){ t[config] = function(){ var i = arguments; t.queue.push(function(){ t[config].
var t = { config:config},u=document,e=window,o='script',s=u.createElement(o),i,f;for(s.src=config.
}({
instrumentationKey:'xxxxxx-xxxxxxxx-xxxxxxxx-xxxxxxxxx' // REMOVE xxxx-xxx... REPLACE WITH INSTRUM
});
window.appInsights=appInsights;
appInsights.trackPageView();
</script>
</head>
<body>
<h1>Azure Monitor Application Insights Hello World!</h1>
<p>You can use the Application Insights JavaScript SDK to perform client/browser-side monitoring o
</body>
</html>
```

45.2.3 Task 3: Configure App Insights SDK

1. Navigate to the Applications Insights blade. Select **Overview** > **Essentials** > Copy your application's **Instrumentation Key**.



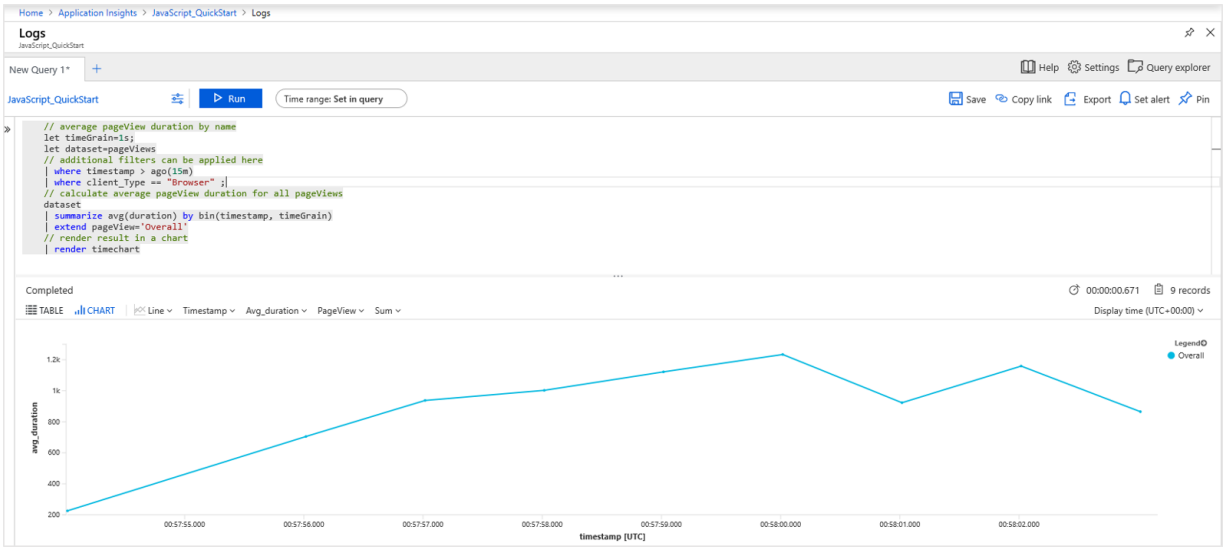
2. Edit `hello_world.html` and add your instrumentation key and save the file.
3. Open `hello_world.html` in a local browser session. This will create a single pageview. You can refresh your browser to generate multiple test page views.

45.2.4 Task 4: Start monitoring in the Azure portal

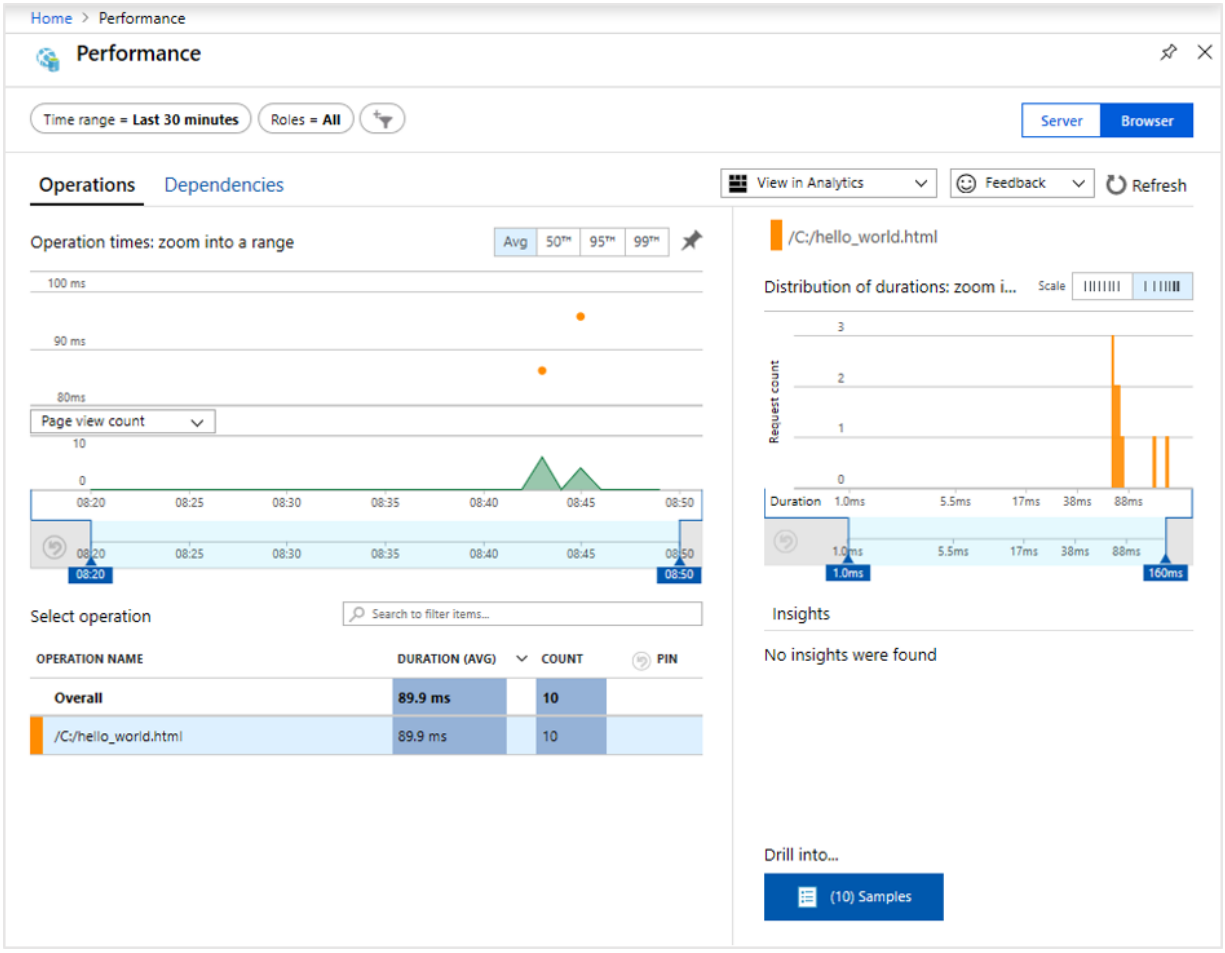
1. You can now reopen the Application Insights **Overview** page in the Azure portal, where you retrieved your instrumentation key, to view details about your currently running application. The four default charts on the overview page are scoped to server-side application data. Since we are instrumenting the client/browser-side interactions with the JavaScript SDK this particular view doesn't apply unless we also have a server-side SDK installed.
2. Click on **Logs (Analytics)**. This opens **Logs**, which provides a rich query language for analyzing all data collected by Application Insights. To view data related to the client-side browser requests run the following query then click **Run**:

```
// average pageView duration by name
let timeGrain=1s;
let dataset=pageViews
```

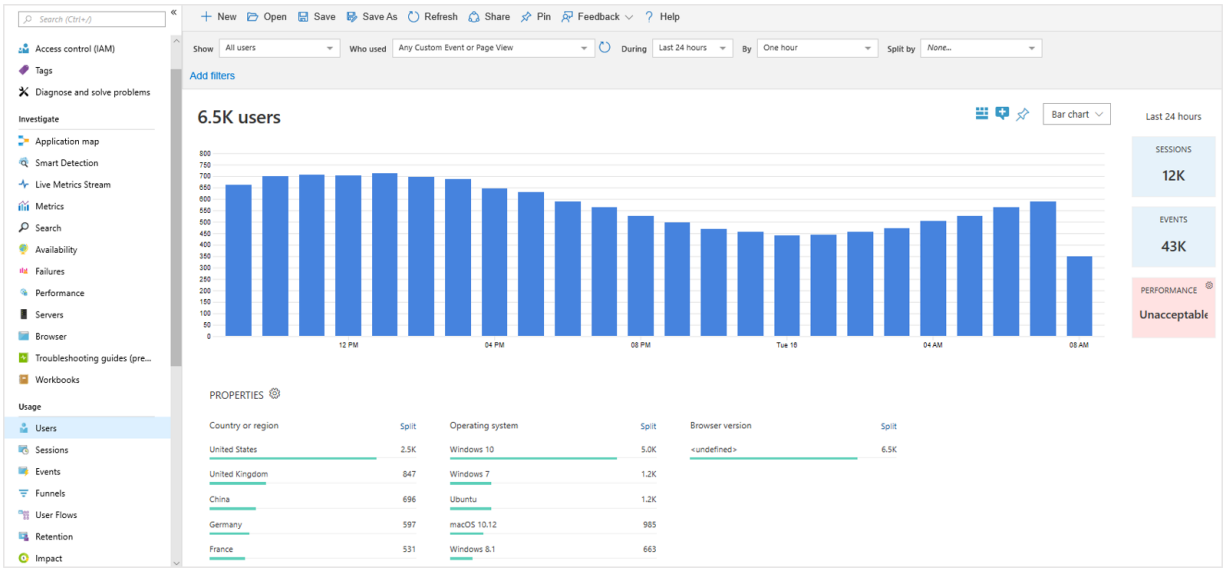
```
// additional filters can be applied here
| where timestamp > ago(15m)
| where client_Type == "Browser" ;
// calculate average pageView duration for all pageViews
dataset
| summarize avg(duration) by bin(timestamp, timeGrain)
| extend pageView='Overall'
// render result in a chart
| render timechart
```



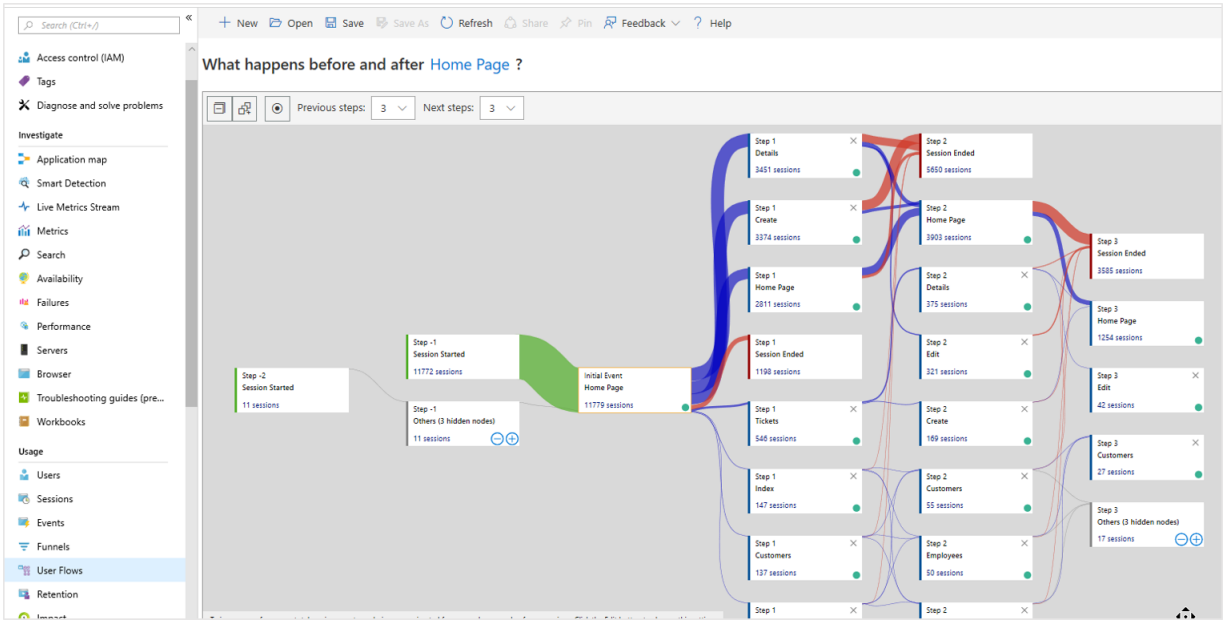
3. Go back to the **Overview** page. Click on **Performance** from under the **Investigate** header. Here you find metrics related to the performance of your website. There is also a corresponding view for analyzing failures and exceptions in your website. You can click **Samples** to drill into individual transaction details. From here, you can access the end-to-end transaction details. Change Local Time button from Last 24 hours to Last 30 Minutes. Change Server/Browser button to Browser.



4. To begin exploring the user behavior analytics tools, from the main Application Insights menu select **Users** under the **Usage** header. Since we are testing from a single machine, we will only see data for one user. For a live website, the distribution of users might look as follows:



5. If we had instrumented a more complex website with multiple pages, another useful tool is **User Flows**. With **User Flows** you can track the pathway visitors takes through the various parts of your website.



6. Leave all resources. You will use these in a future lab.

Results: In this lab, you learned how to monitor resources with Azure Monitor.

46 Module 4: Lab 2 -Security Center

Azure Security Center is a unified infrastructure security management system that strengthens the security posture of your data centers, and provides advanced threat protection across your hybrid workloads in the cloud - whether they're in Azure or not - as well as on premises.

Keeping your resources safe is a joint effort between your cloud provider, Azure, and you, the customer. You have to make sure your workloads are secure as you move to the cloud, and at the same time, when you move to IaaS (infrastructure as a service) there is more customer responsibility than there was in PaaS (platform as a service), and SaaS (software as a service). Azure Security Center provides you the tools needed to harden your network, secure your services and make sure you're on top of your security posture.

46.1 Exercise 1: Onboard your Azure subscription to Security Center Standard

Azure Security Center provides unified security management and threat protection across your hybrid cloud workloads. While the Free tier offers limited security for your Azure resources only, the Standard tier extends these capabilities to on-premises and other clouds. Security Center Standard helps you find and fix security vulnerabilities, apply access and application controls to block malicious activity, detect threats using analytics and intelligence, and respond quickly when under attack. You can try Security Center Standard at no cost. To learn more, see the pricing page.

In this Exercise, you upgrade to the Standard tier for added security and install the Microsoft Monitoring Agent on your virtual machines to monitor for security vulnerabilities and threats.

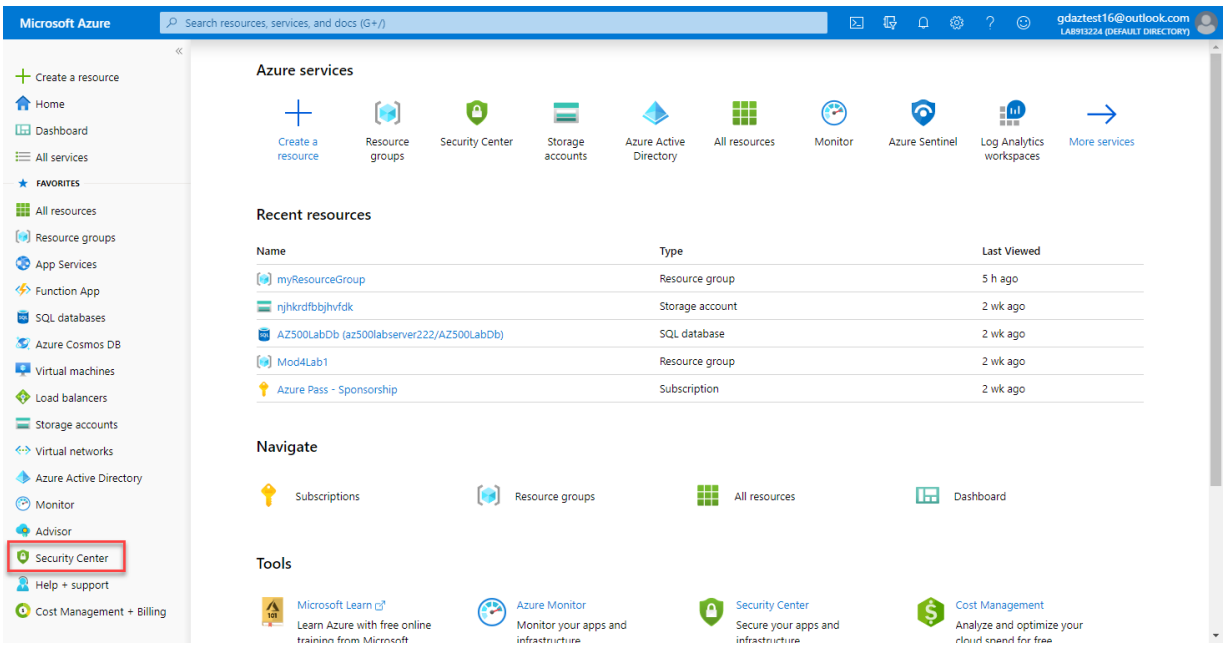
46.1.1 Task 1: Automate data collection

Security Center collects data from your Azure VMs and non-Azure computers to monitor for security vulnerabilities and threats. Data is collected using the Microsoft Monitoring Agent, which reads various security-related configurations and event logs from the machine and copies the data to your workspace for analysis. By default, Security Center will create a new workspace for you.

When automatic provisioning is enabled, Security Center installs the Microsoft Monitoring Agent on all supported Azure VMs and any new ones that are created. Automatic provisioning is strongly recommended.

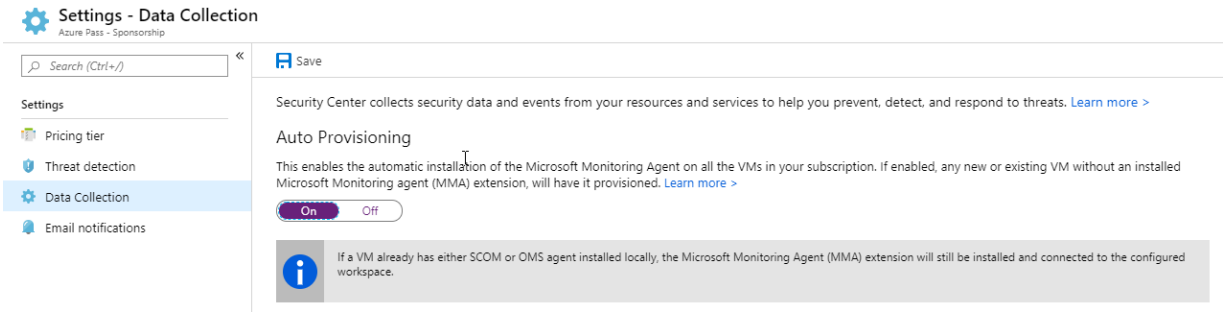
To enable automatic provisioning of the Microsoft Monitoring Agent:

1. In the Azure Portal, select the **Security Center** from the Hub menu.



2. On the **Getting started** blade click **Upgrade**.
3. Under the Security Center main menu, select **Pricing & settings**.
4. On the row of the subscription, click on the subscription on which you'd like to change the settings.
5. In the **Data Collection** tab, set **Auto provisioning** to **On**.
6. Exit the blade **without** saving.

   Note: Ensure you do not click save otherwise the following exercises will not function as expected.



With this new insight into your Azure VMs, Security Center can provide additional Recommendations related to system update status, OS security configurations, endpoint protection, as well as generate additional Security alerts.

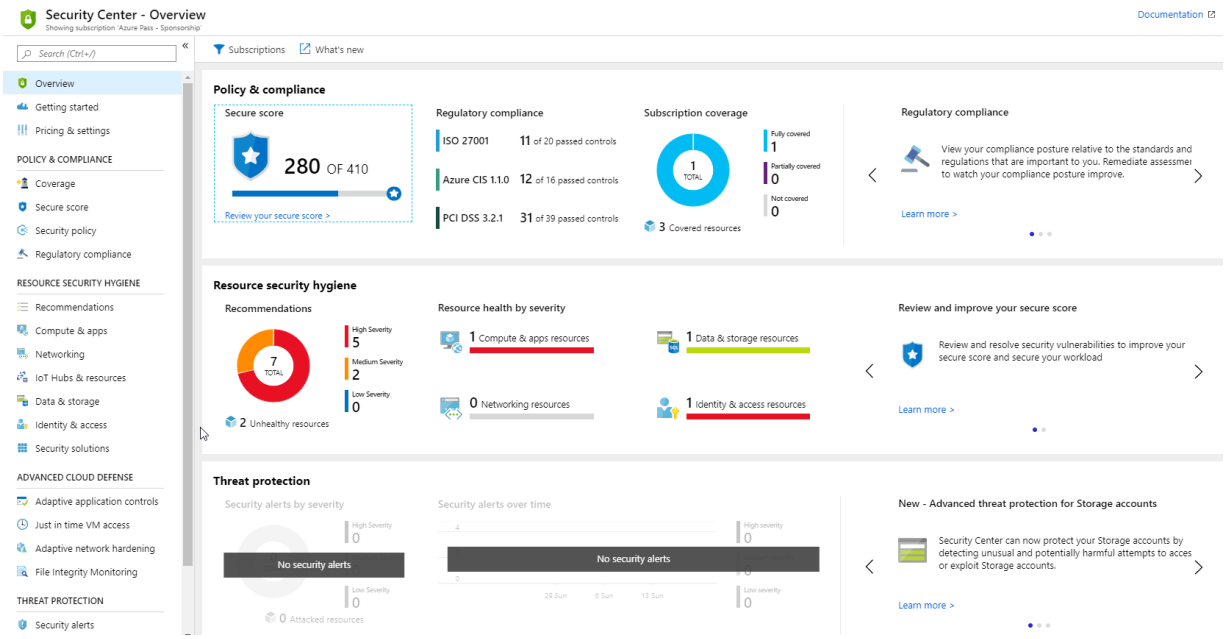
46.2 Exercise 2: Onboard Windows computers to Azure Security Center

After you onboard your Azure subscriptions, you can enable Security Center for resources running outside of Azure, for example on-premises or in other clouds, by provisioning the Microsoft Monitoring Agent.

This exercise shows you how to install the Microsoft Monitoring Agent on a Windows computer.

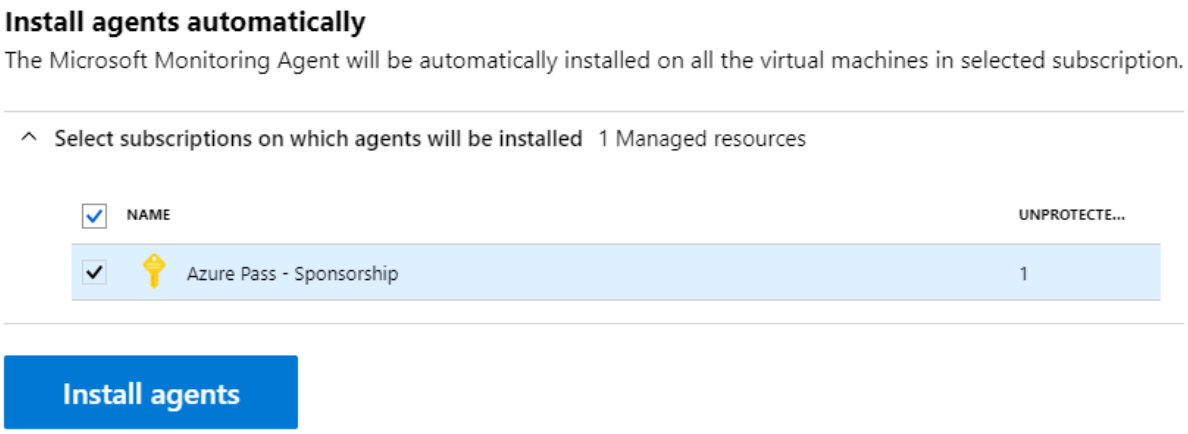
46.2.1 Task 1: Add new Windows computer

1. In the Azure Portal, select **Security Center**. **Security Center - Overview** opens.

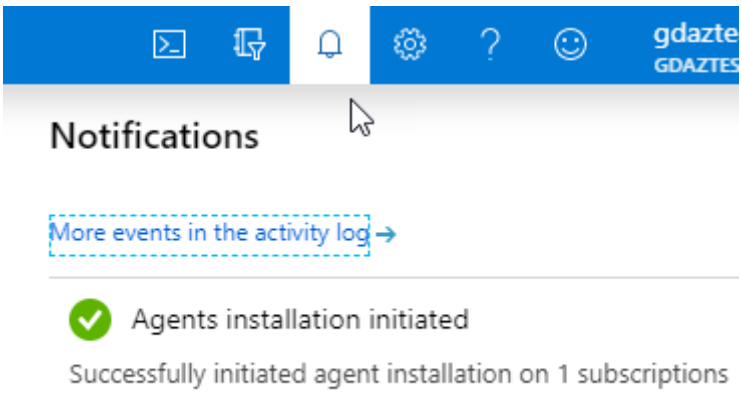


2. Under the Security Center main menu, select **Getting started**.
3. Select the **Install Agents** tab.

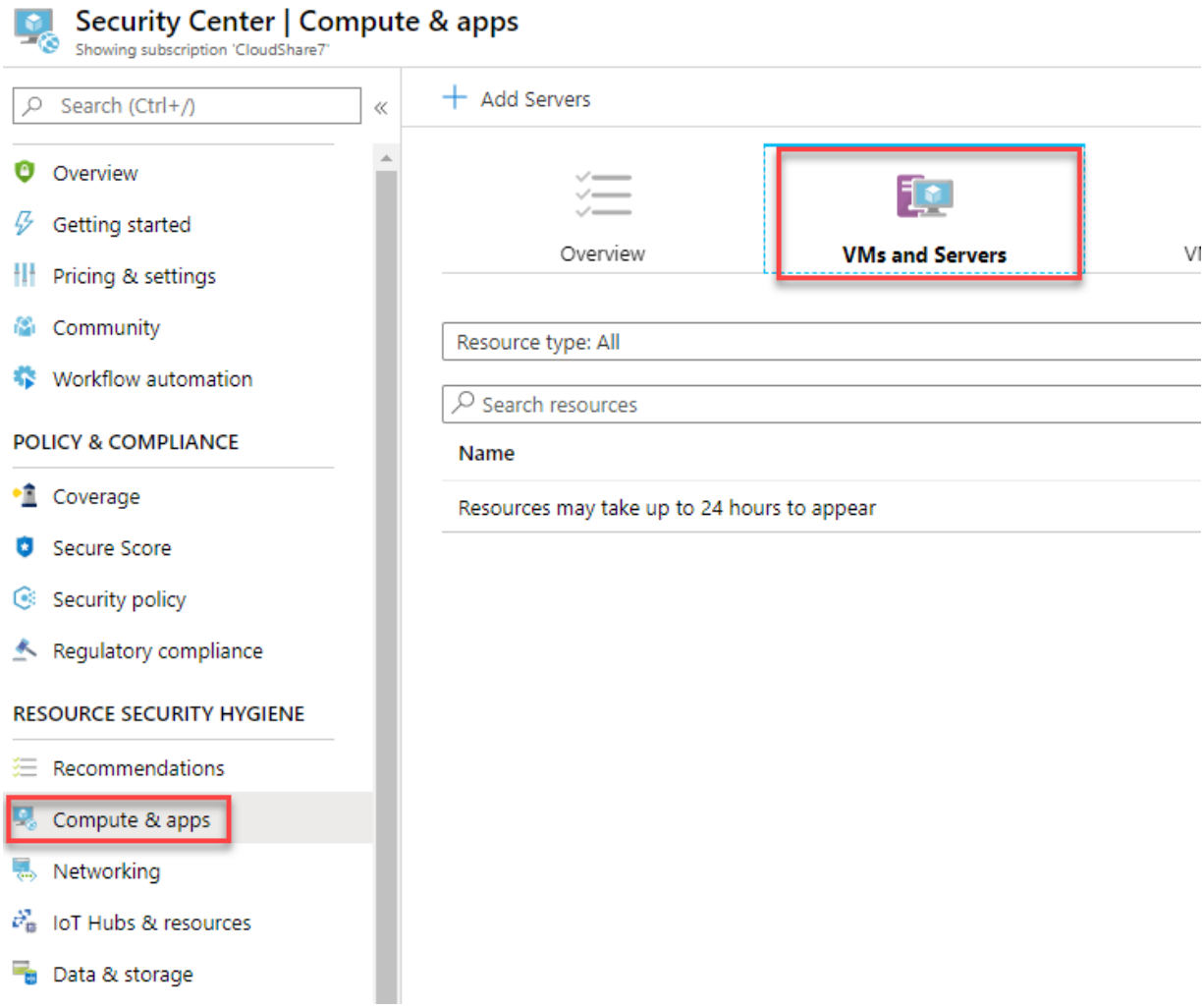
4. Scroll down to the Install agents automatically section and click **Install agents**.



5. Wait until the agent is install by monitoring the deployment.



6. Open the **Security Center** and click on **Compute & apps** then click on **VMs and Servers**.



7. Notice your Virtual Machine is now monitored.

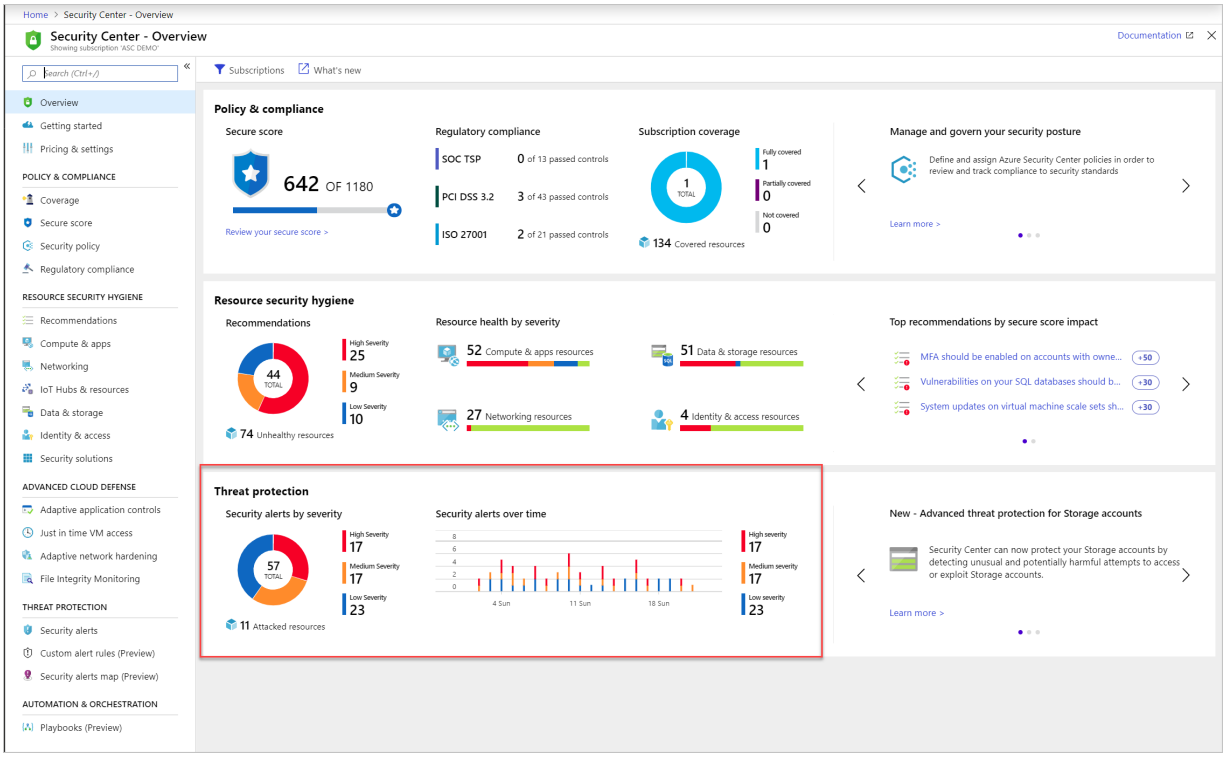
46.3 Exercise 3: Manage and respond to alerts in Azure Security Center

Security Center automatically collects, analyzes, and integrates log data from your Azure resources, the network, and connected partner solutions, like firewall and endpoint protection solutions, to detect real threats and reduce false positives. A list of prioritized security alerts is shown in Security Center along with the information you need to quickly investigate the problem and recommendations for how to remediate an attack.

46.3.1 Task 1: Manage your alerts

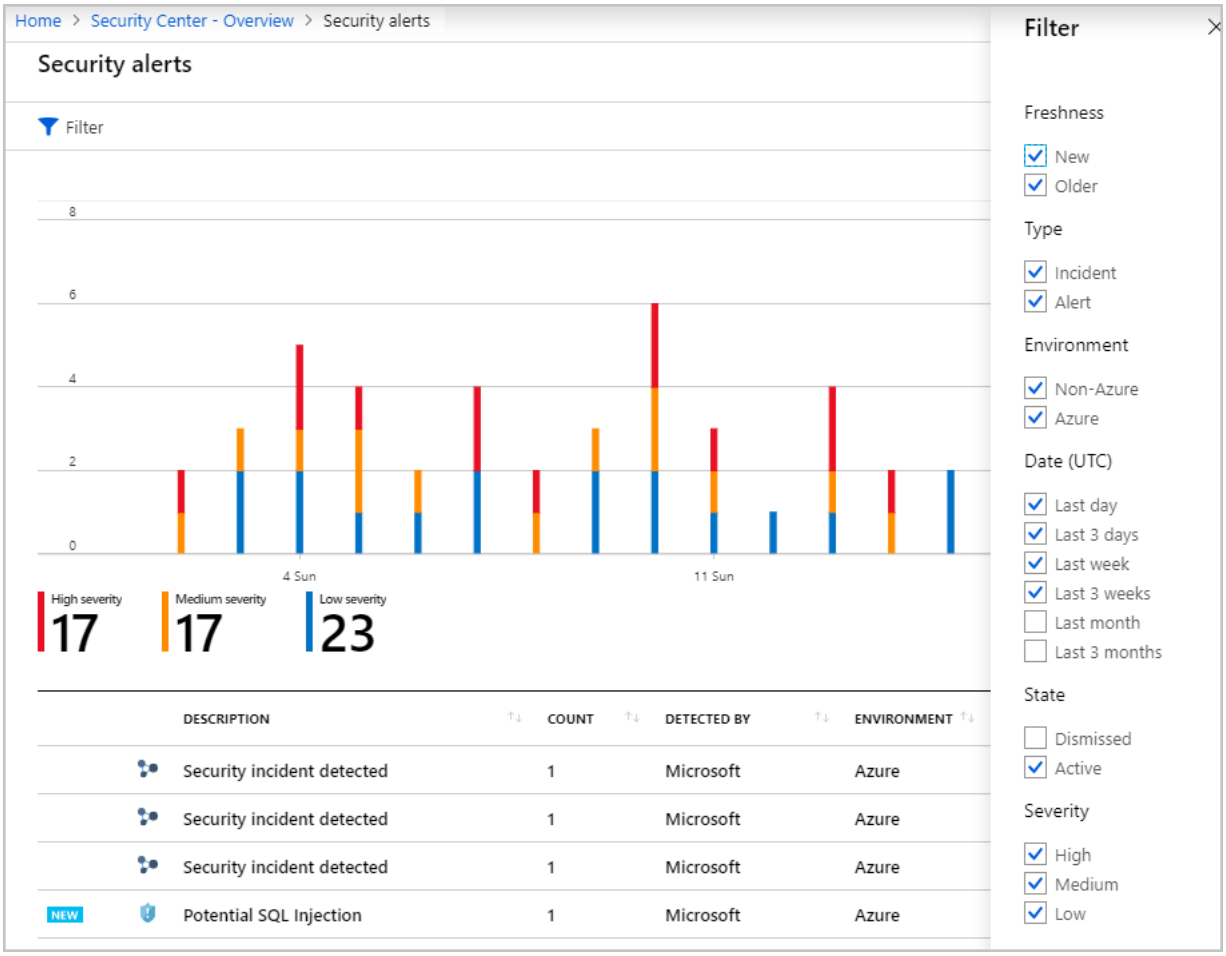
1. From the Security Center dashboard, see the **Threat protection** tile to view and overview of the alerts.

Note: If the tile displays **No security alerts**, you may have to wait some time for the evaluation to run.



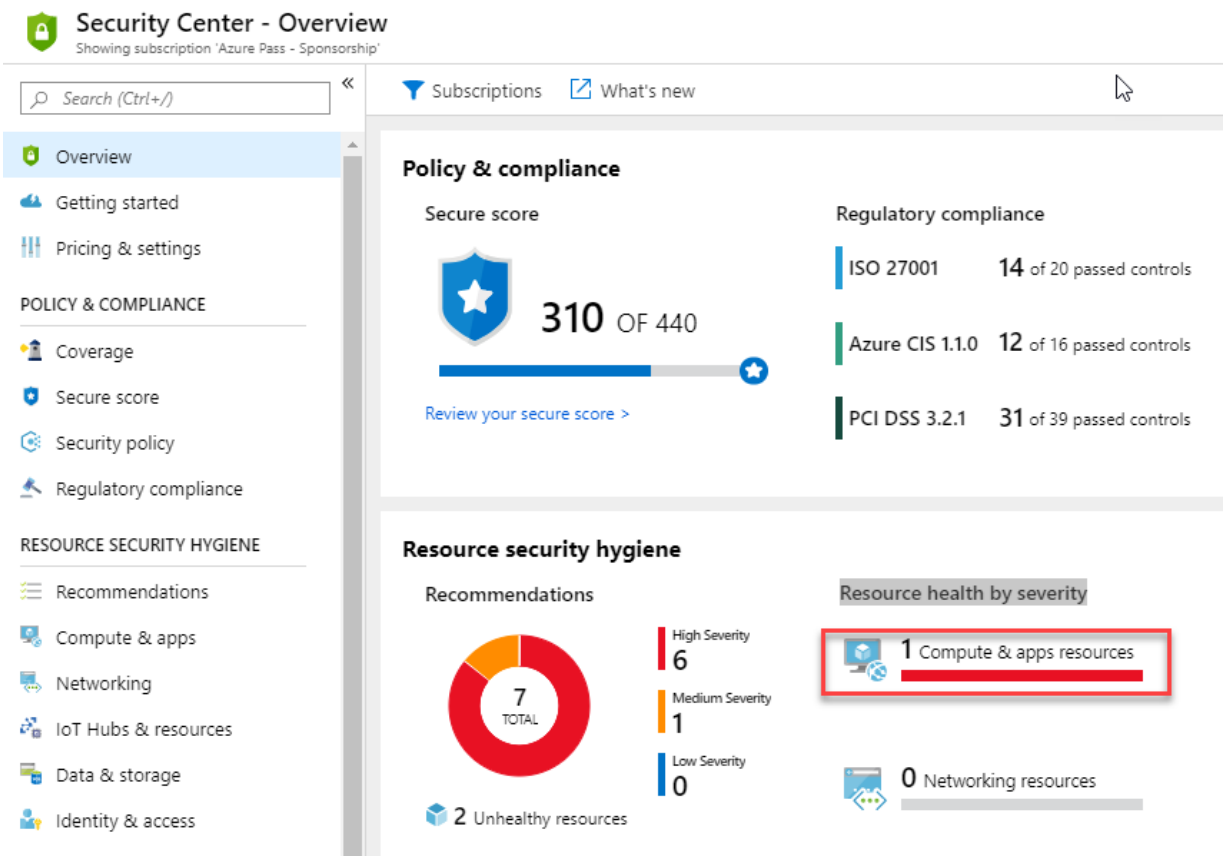
2. To see more details about the alerts, click the tile. The screenshot belown shows potential alerts you would see in the real world:

3. To filter the alerts shown, click **Filter**, and from the **Filter** blade that opens, select the filter options that you want to apply. The list updates according to the selected filter. Filtering can be very helpful. For example, you might you want to address security alerts that occurred in the last 24 hours because you are investigating a potential breach in the system.

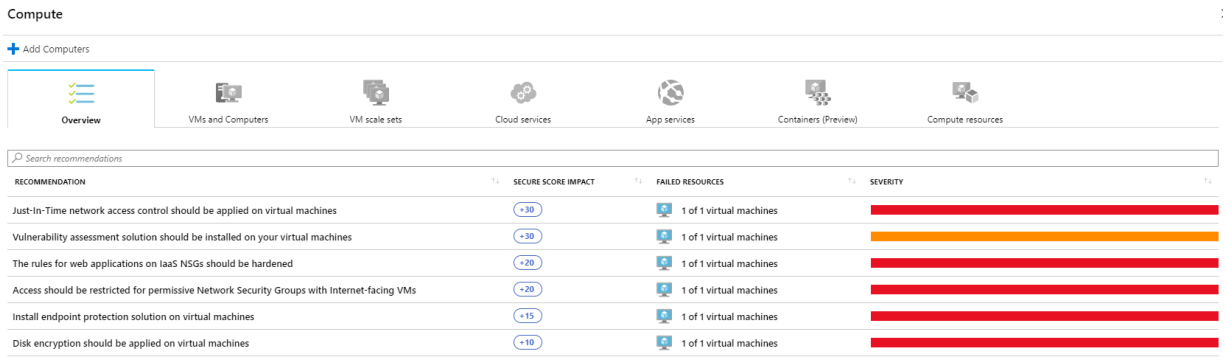


46.3.2 Task 2: Respond to recommendations

1. In the Azure Security Center click **Overview**.
2. From the **Resource secuity hygiene** list, in the **Resource health by severity** section select **Compute & apps resources**



3. Review the recommendations.



WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this lab and can move onto the next lab in the series

47 Module 4: Lab 3 - Event hub

Connect sending and receiving applications with Event Hubs so you can handle extremely high loads without losing data.

In this lab, you will:

- Create an Event Hub using the Azure portal
- Configure applications to send or receive messages through an Event Hub
- Evaluate Event Hub performance using the Azure portal

47.1 Exercise 1: Implementing Event Hub

47.1.1 Task 1: Enabling Event Hubs Namespace

1. Log into the Azure portal
2. In the search bar type **Event Hubs** and select **Event Hubs**
3. Click **+ Add** to add a new Event Hubs Namespace
    - Populate the fields with the following details:
        - **Name** : yourUniqueName
        - **Pricing Tier** : Standard
        - **Subscription**: yourSubscription
        - **Resource group**: Create New with name "EventHubRG"
        - **Location**: East US
        - **Throughput Units**: 2
4. Click Select **Review + Create** then select **Create**.

47.1.2 Task 2: Create a storage account for later user

Note: We also need to create a storage account and a blob store container to store events that will be sent to the Event Hubs later on

1. Search for **Storage Accounts** in the search bar and click on **Storage Accounts**
2. Click **Add**
3. Choose the Resource Group of **EventHubsRG** (Or the name of your resource group if you chose to use another)
4. Set the following options:
    - **Storage Account Name**: uniqueName (unique across all of azure)
    - **location**: east US
    - **Performance**: Standard
    - **Account Kind**: General purpose v2
    - **Replication**: Locally-redundent storage (LRS)
    - **Access tier**: Hot
5. Click **Review + Create** then click **Create**
6. Wait for the storage account to create.
7. Return to storage accounts
8. Select the storage account you created
9. In the overview pane click **Containers**
10. Click **+ Container**
11. For the name type **events**
12. Set the **Public Access Level** to **Container**
13. Click **Create**

47.1.3 Task 3: Create new event hub

1. Return to event Hubs click the name of your newly created Event Hub namespace
2. In the event hub namespace click **Event Hubs** underneath Entities in the selection pane
3. Click **+ Event Hub**
4. Enter the name of "events"
5. Click **On** underneath **Capture**

    Note: This will turn on the dumping of events to the Blob store we created erlier
6. Click **Select Container**

7. Select the **Storage account** name you created earlier
8. Select the **Blob storage (Container)** name you created earlier
9. Click **Select**
10. Click **Create**

47.1.4 Task 4: Collect data to be able to send events into event hubs

1. Under event hubs namespace click **Shared access policies**
2. Click **RootManageSharedAccessKey**
3. Click the copy to clipboard icon next to Primary Key
4. Open notepad and paste it in there for later use
5. Copy the name of the Event Hubs namespace and the name of the Event Hub you created to the same notepad document

Note: You will need this primary key and other information for the scripts that will be run later to enter some data into the event hubs system

47.1.5 Task 5: Download the script files

We will now download the scripts that will be used to create some events to be sent into the Event Hub, this will simulate the Event Hub receiving data from an application in the environment that has been written to communicate with Event Hubs, or from other systems that communicate with Azure Event Hubs. The script files have been developed and published on the PowerShell gallery

1. Open **PowerShell as administrator** (right click on the PowerShell icon) and run the following command

   ```
   install-script get-blobevents, send-blobevents
   ```

Note: If prompted confirm the installation. The script files have now been downloaded and are available for use in PowerShell

47.1.6 Task 6: Send some events to Event Hub

For this section you will need the primary key copied from the portal erlier

1. Open PowerShell
2. Run the following command

   ```
   send-blobevents
   ```

3. This command will prompt you to enter the following data:
   - **primaryKey**: your primary key copied earlier in the lab
   - **eventhubnamespace**: the name of your namespace
   - **eventhub**: the name of your event hub
   - **numberOfEvents**: 10 (you can choose more, but they will take longer to send to the Event Hub)

If you recieve a series if **401 Unauthorised errors** from the script check the clock on the machine you have executed the code, if using a virtual machine it should be set to UTC. If you have corrected the time you will need to restart the PowerShell shell and re-run the code.

Note: This will send a series of events to the Event Hub with randomised data

47.1.7 Task 7: Review the Events in EventHub and Blob Storage

1. Return to Event Hubs and your **Event hubs namespace**
2. Click **Event Hubs** and select your **Event Hub**
3. In the overview pane review the Event Hub graphs to see the data spikes on incoming messages
4. Search for and open **Storage Accounts**
5. Select your **event hubs Storage Account**

6. Click **Containers**
7. Open your **Blob containter** for events storage

Information: Here you can review the raw .avro events that have been created by the Event Hubs system, for more information about the .avro format used by Event Hubs see the following wiki article `https://en.wikipedia.org/wiki/Apache_Avro`

47.1.8 Task 8: Review the Events from a REST query to the Blob storage

1. Launch **PowerShell**
2. Run `get-blobevents` and enter the following when prompted:
   - **blobName**: Name of the storage account
   - **containterName**: Name of the Blob container where the events are stored

Note: You can see this information in your storage account. The script will make a REST query to the Blob storage to list out all the .avro event that are in the storage Blob

---

WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this lab and can move onto the next lab in the series

48 Module 4: Lab 4 - Azure Sentinel

Azure Sentinel is your bird's-eye view across the enterprise. Put the cloud and large-scale intelligence from decades of Microsoft security experience to work. Make your threat detection and response smarter and faster with artificial intelligence (AI)

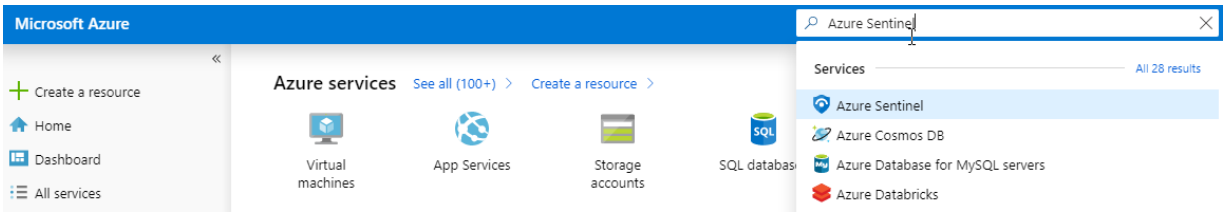
48.1 Exercise 1: On-board Azure Sentinel

To on-board Azure Sentinel, you first need to enable Azure Sentinel, and then connect your data sources. Azure Sentinel comes with a number of connectors for Microsoft solutions, available out of the box and providing real-time integration, including Microsoft Threat Protection solutions, Microsoft 365 sources, including Office 365, Azure AD, Azure ATP, and Microsoft Cloud App Security, and more. In addition, there are built-in connectors to the broader security ecosystem for non-Microsoft solutions. You can also use common event format, Syslog or REST-API to connect your data sources with Azure Sentinel.

After you connect your data sources, choose from a gallery of expertly created workbooks that surface insights based on your data. These workbooks can be easily customized to your needs.

48.1.1 Task 1: Enable Azure Sentinel

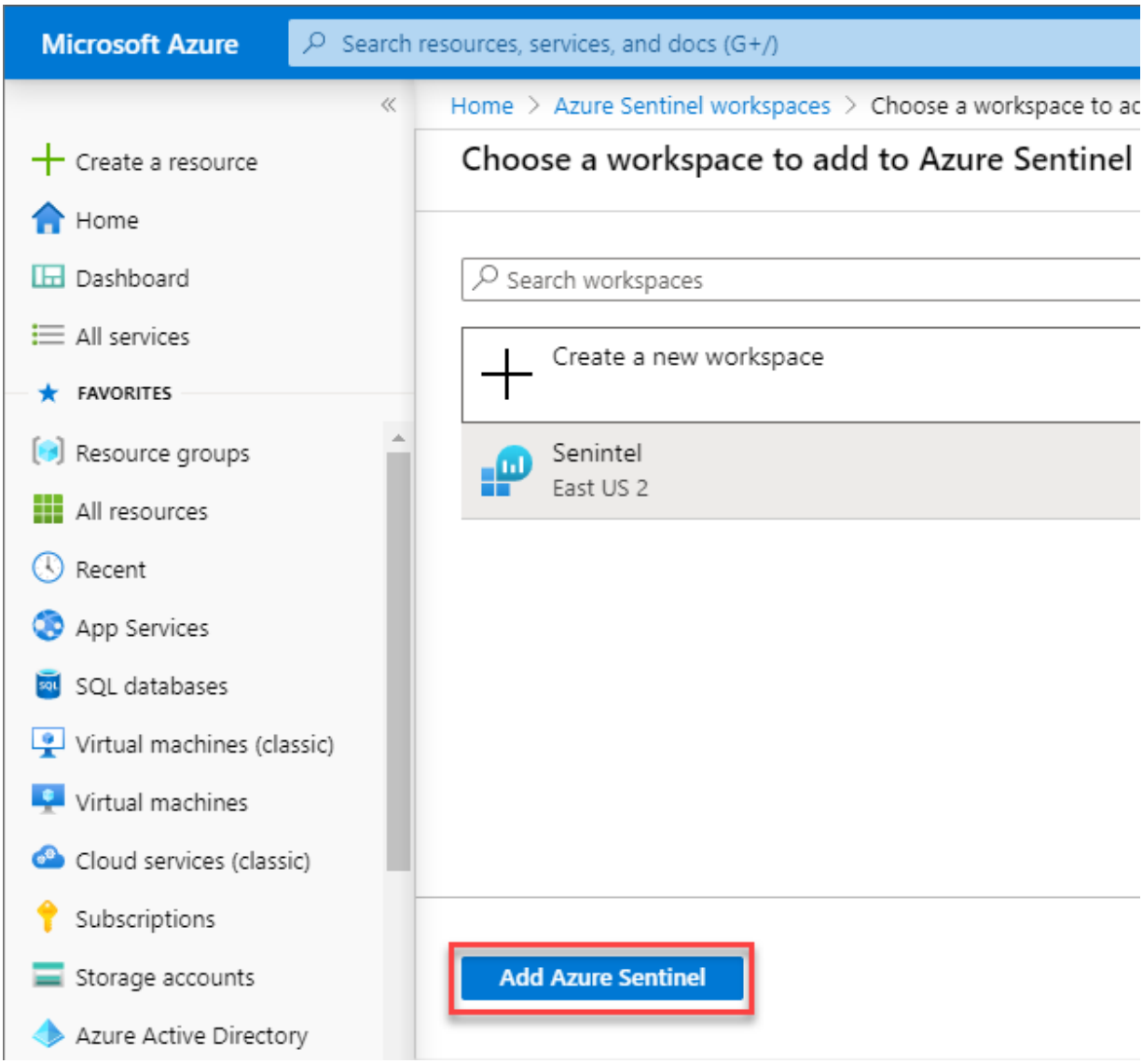
1. In the Azure portal, search for Azure Sentinel.



2. Click **+Add**.
3. Create a new workspace in a new resource group using the East US region if necessary.

   Note: - Default workspaces created by Azure Security Center will not appear in the list; you can't install Azure Sentinel on them. - Azure Sentinel can run on workspaces in any GA region of Log Analytics except the China, Germany and Azure Government regions. Data generated by Azure Sentinel (such as incidents, bookmarks, and alert rules, which may contain some customer data sourced from these workspaces) is saved either in West Europe (for workspaces located in Europe) or East US (for all US-based workspaces, as well as any other region except Europe).

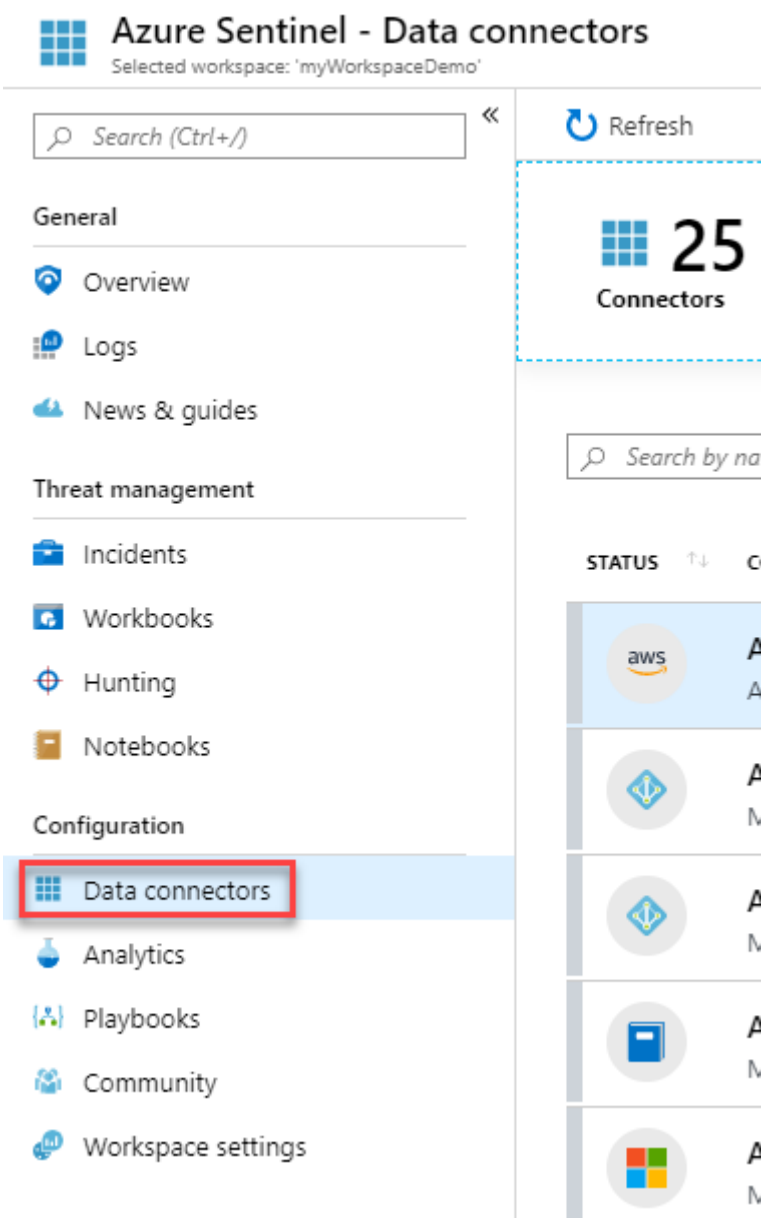
4. To finish creating the workspace, click **Review and Create**, then click **Create**.
5. Once the workspace has been created, click **Add Azure Sentinel**.



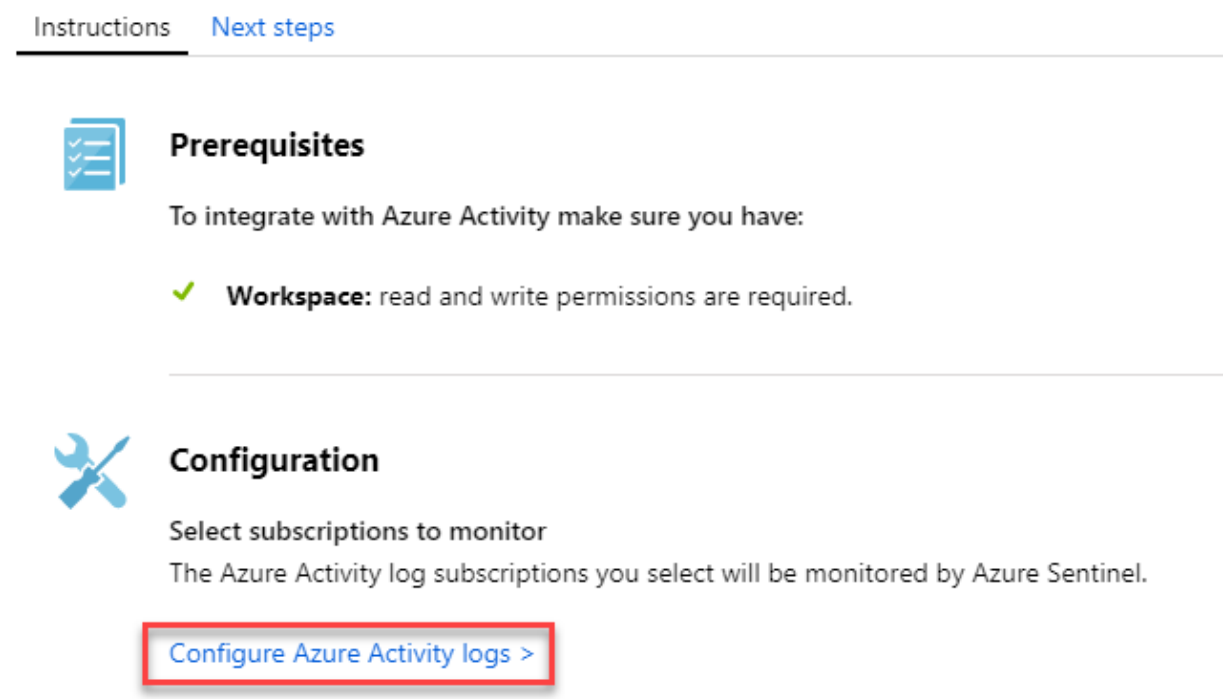
48.1.2 Task 3: Connect data sources

Azure Sentinel creates connections to services and apps by connecting to the service and forwarding the events and logs to Azure Sentinel. For machines and virtual machines, you can install the Azure Sentinel agent that collects the logs and forwards them to Azure Sentinel. For Firewalls and proxies, Azure Sentinel utilizes a Linux Syslog server. The agent is installed collects the log files and forwards them to Azure Sentinel.

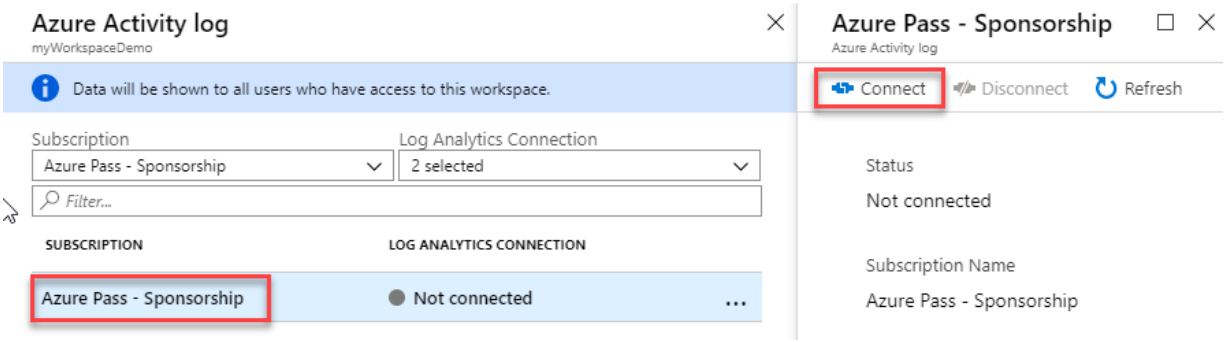
1. In the Azure Portal select All resources and select the Log Analytics workspace you created in the previous task.
2. On the menu, select **Data connectors**. This page lets you see the full list of connectors that Azure Sentinel provides and their status.



3. Select **Azure Activity** and click **Open connector page**.
4. Select **Configure Azure Activity logs**.



5. On the specific connector page, make sure you have fulfilled all the prerequisites and follow the instructions to connect the data to Azure Sentinel. It may take some time for the logs to start syncing with Azure Sentinel. After you connect, you see a summary of the data in the **Data received** graph, and connectivity status of the data types.
6. Select your Azure subscription then click **Connect**.



Results: You have now completed this lab.

49 Module 4: Lab 5 - Manage endpoint protection issues with Azure Security Center

Azure Security Center monitors the status of antimalware protection and reports this under the Endpoint protection issues blade. Security Center highlights issues, such as detected threats and insufficient protection, which can make your virtual machines (VMs) and computers vulnerable to antimalware threats. By using the information under **Endpoint protection issues**, you can identify a plan to address any issues identified.

Security Center reports the following endpoint protection issues:

- Endpoint protection not installed on Azure VMs - A supported antimalware solution is not installed on these Azure VMs.
- Endpoint protection not installed on non-Azure computers - A supported antimalware is not installed on these non-Azure computers.
- Endpoint protection health:

- Signature out of date - An antimalware solution is installed on these VMs and computers, but the solution does not have the latest antimalware signatures.
- No real time protection - An antimalware solution is installed on these VMs and computers, but it is not configured for real-time protection.
- Not reporting - An antimalware solution is installed but not reporting data.
- Unknown - An antimalware solution is installed but its status is unknown or reporting an unknown error.

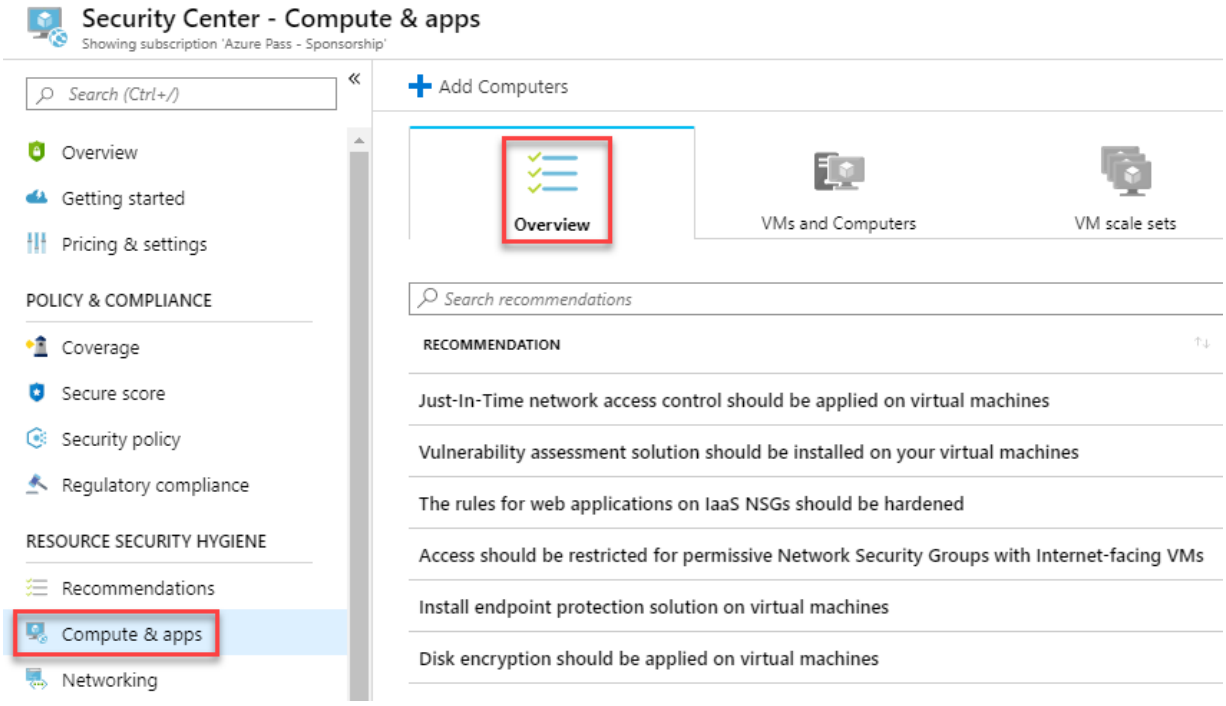
49.1 Exercise 1: Implement the recommendation

Endpoint protection issues is presented as a recommendation in Security Center. If your environment is vulnerable to antimalware threats, this recommendation will be displayed under **Recommendations** and under **Compute**. To see the **Endpoint protection issues dashboard**, you need to follow the Compute workflow.

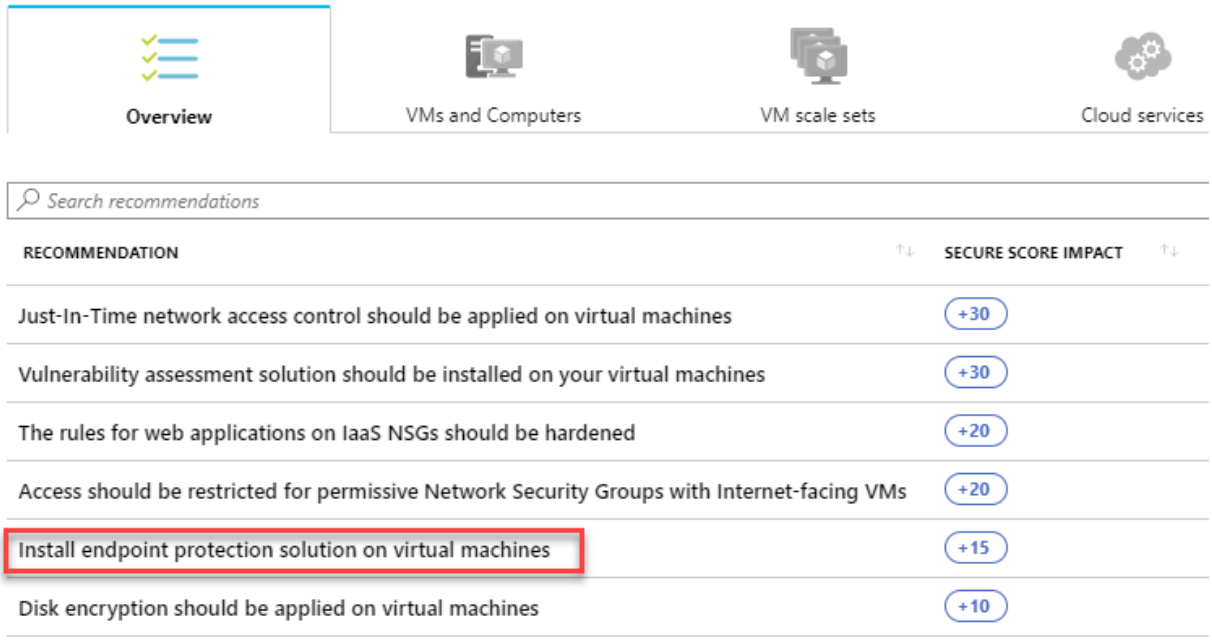
In this exercise, we will use **Compute**. We will look at how to install antimalware on Azure VMs and on non-Azure computers.

49.1.1 Task 1: Install antimalware on Azure VMs

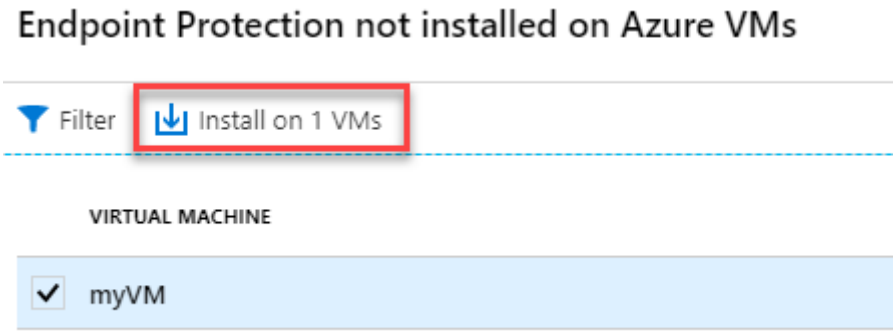
1. Select **Compute & apps** under the Security Center main menu or **Overview**.



2. Under **Compute**, select **Install endpoint protection solution on virtual machines**. The **Endpoint protection issues** dashboard opens.



3. On the **Endpoint Protection not installed on Azure VMs** blade click **Install on 1 VMs**.



4. Under **Select Endpoint protection**, select the endpoint protection solution you want to use. In this example, select **Microsoft Antimalware**.
5. Additional information about the endpoint protection solution is displayed. Select **OK**.

Results: You have now completed this lab.

50 Module 4: Lab 6 - Security Playbook in Azure Sentinel

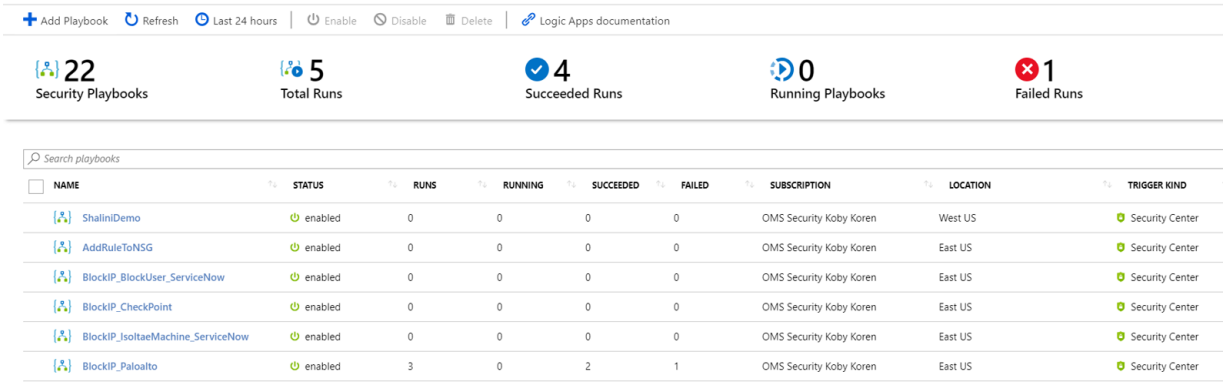
A security playbook is a collection of procedures that can be run from Azure Sentinel in response to an alert. A security playbook can help automate and orchestrate your response, and can be run manually or set to run automatically when specific alerts are triggered. Security playbooks in Azure Sentinel are based on Azure Logic Apps, which means that you get all the power, customizability, and built-in templates of Logic Apps. Each playbook is created for the specific subscription you choose, but when you look at the Playbooks page, you will see all the playbooks across any selected subscriptions.

50.1 Exercise 1: Create and manage a Security Playbook in Azure.

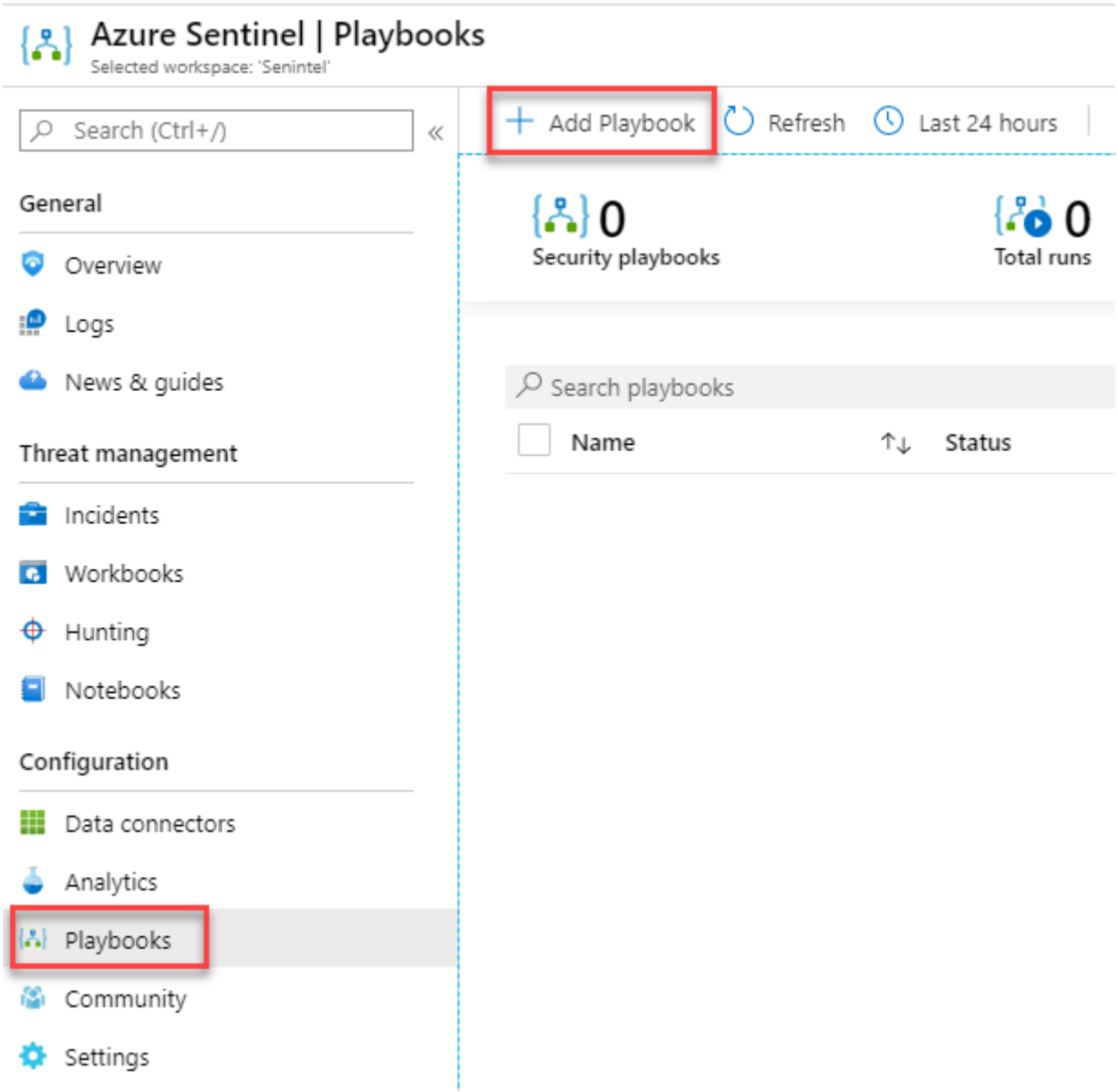
50.1.1 Task 1: How to create a security playbook.

Follow these steps to create a new security playbook in Azure Sentinel:

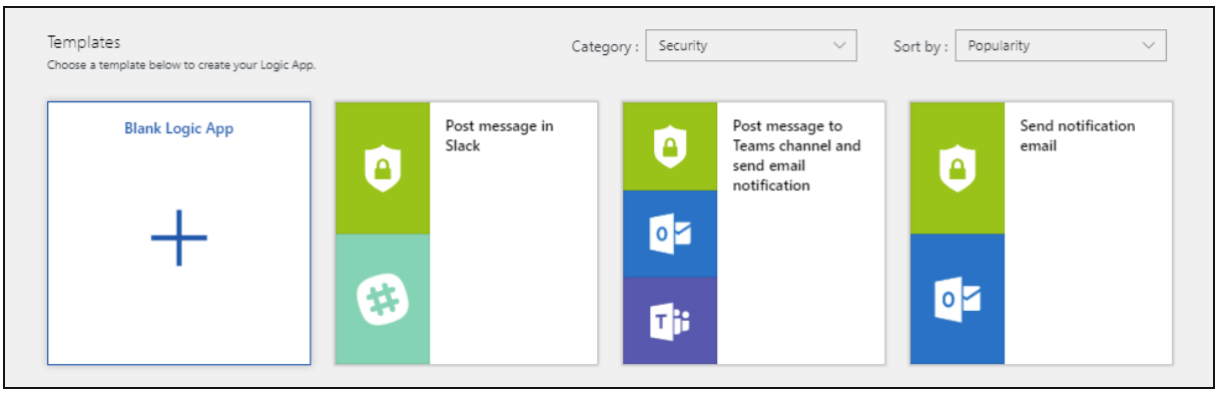
1. Open **Azure Sentinel** dashboard.
2. Under **Configuration**, select **Playbooks**.



3. In the **Azure Sentinel - Playbooks** page, click + **Add Playbook** button.

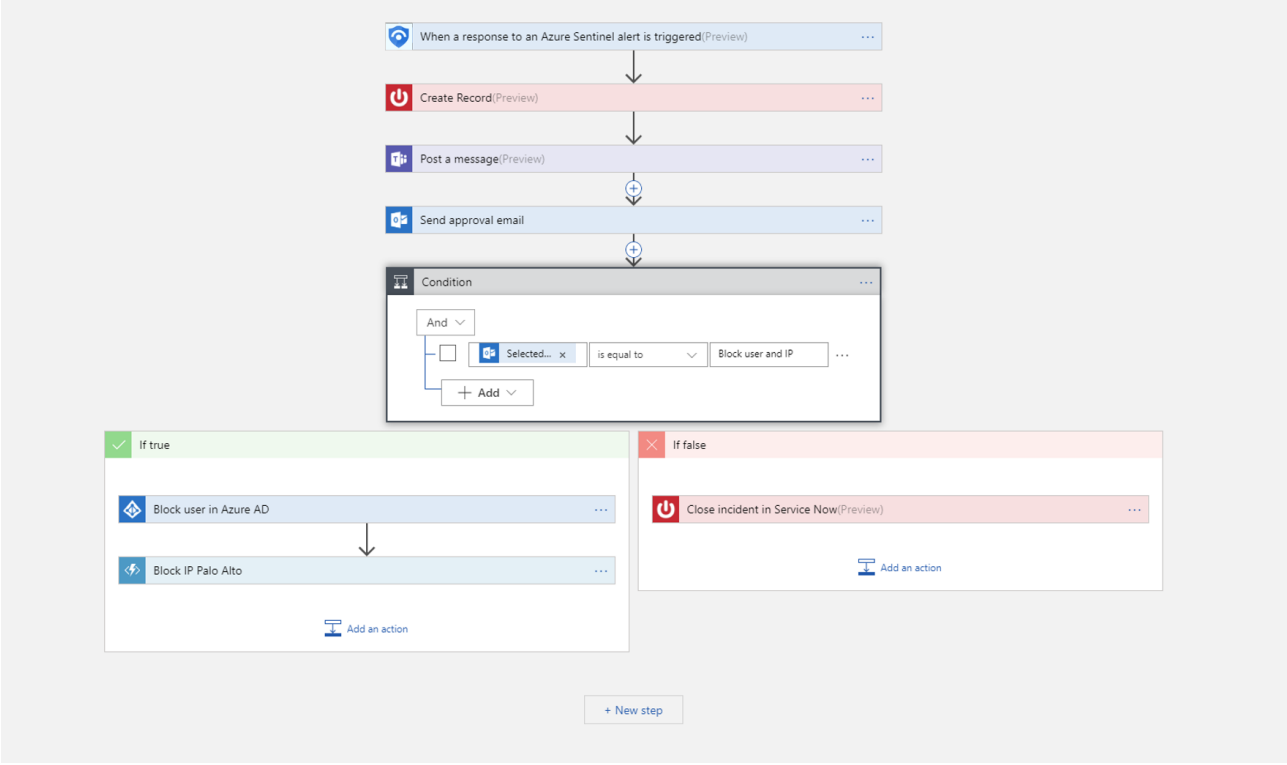


4. In the **Create Logic app** page, type the requested information to create your new logic app, and click **Review + create** then click **Create**.
5. Once created click **Go to resource**.
6. In the **Logic App Designer** select the template you want to use. If you select a template that necessitates credentials, you will have to provide them. Alternatively, you can create a new blank playbook from scratch. Select **Blank Logic App**.



7. You are taken to the Logic App Designer where you can either build new or edit the template.
8. If you are creating a blank playbook, in the **Search all connectors and triggers** field, type *Azure Sentinel*, and select **When a response to an Azure Sentinel alert is triggered**.

   Note: You may be required to re-authenticate. If so, click **Sign in** and authenticate with your Azure credentials.
9. Click **Save** and return to the Azure Sentinel blade. After it is created, the new playbook appears in the **Playbooks** list. If it doesn't appear, click **Refresh**.
10. Use the **Get entities** functions, which enable you to get the relevant entities from inside the **Entities** list, such as accounts, IP addresses and hosts. This will enable you to run actions on specific entities.
11. Now you can define what happens when you trigger the playbook. You can add an action, logical condition, switch case conditions, or loops.



50.1.2 Task 2: How to run a security playbook

You can run a playbook on demand.

To run a playbook on-demand:

1. In the **Incidents** page, select an incident and click on **View full details**.
2. In the **Alerts** tab, click on the alert you want to run the playbook on, and scroll all the way to the right and click **View playbooks** and select a playbook to **run** from the list of available playbooks on the subscription.

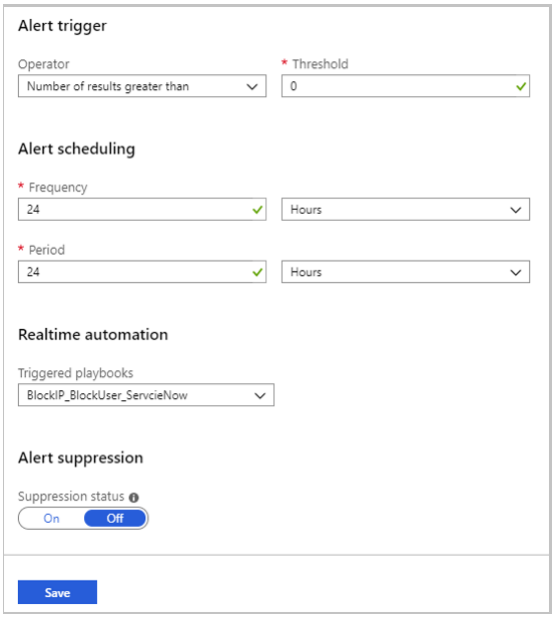
50.2 Exercise 2: Automate threat responses

SIEM/SOC teams can be inundated with security alerts on a regular basis. The volume of alerts generated is so huge, that available security admins are overwhelmed. This results all too often in situations where many alerts can't be investigated, leaving the organization vulnerable to attacks that go unnoticed.

Many, if not most, of these alerts conform to recurring patterns that can be addressed by specific and defined remediation actions. Azure Sentinel already enables you to define your remediation in playbooks. It is also possible to set real-time automation as part of your playbook definition to enable you to fully automate a defined response to particular security alerts. Using real-time automation, response teams can significantly reduce their workload by fully automating the routine responses to recurring types of alerts, allowing you to concentrate more on unique alerts, analyzing patterns, threat hunting, and more.

50.2.1 Task 1: Automate Responses

1. Select the alert for which you want to automate the response.
2. In the **Edit alert rule** page, under **Real-time automation**, choose the **Triggered playbook** you want to run when this alert rule is matched.
3. Select **Save**.



Results: In this lab, you learned how to use playbooks in Azure Sentinel.

51 Module 4: Lab 7 - Secure score in Azure Security Center

With so many services offering security benefits, it's often hard to know what steps to take first to secure and harden your workload. The Azure secure score reviews your security recommendations and prioritizes them for you, so you know which recommendations to perform first. This helps you find the most serious security vulnerabilities so you can prioritize investigation. Secure score is a tool that helps you assess your workload security posture.

Secure score calculation

Security Center mimics the work of a security analyst, reviewing your security recommendations, and applying advanced algorithms to determine how crucial each recommendation is. Azure Security center constantly reviews you active recommendations and calculates your secure score based on them, the score of a recommendation is derived from its severity and security best practices that will affect your workload security the most.

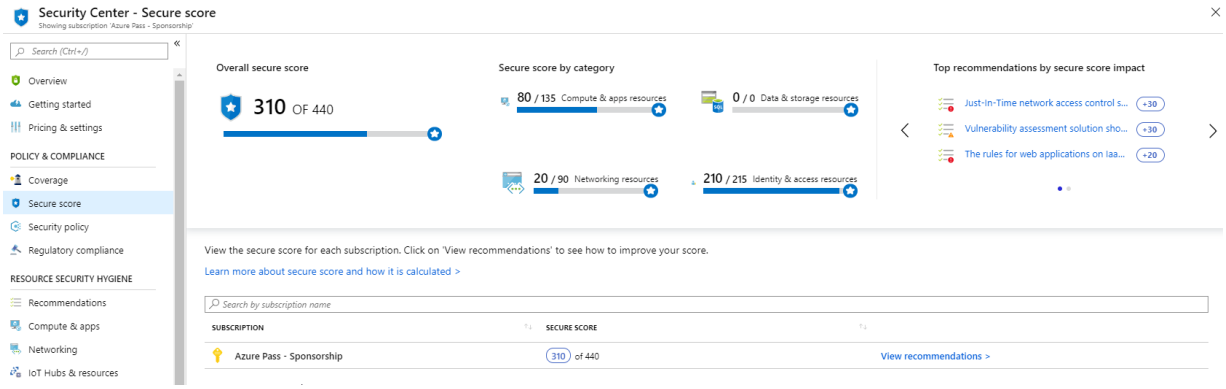
Security Center also provides you with an **Overall secure score**.

Overall secure score is an accumulation of all your recommendation scores. You can view your overall secure score across your subscriptions or management groups, depending on what you select. The score will vary based on subscription selected and the active recommendations on these subscriptions.

51.1 Exercise 1: Improve your secure score in Azure Security Center.

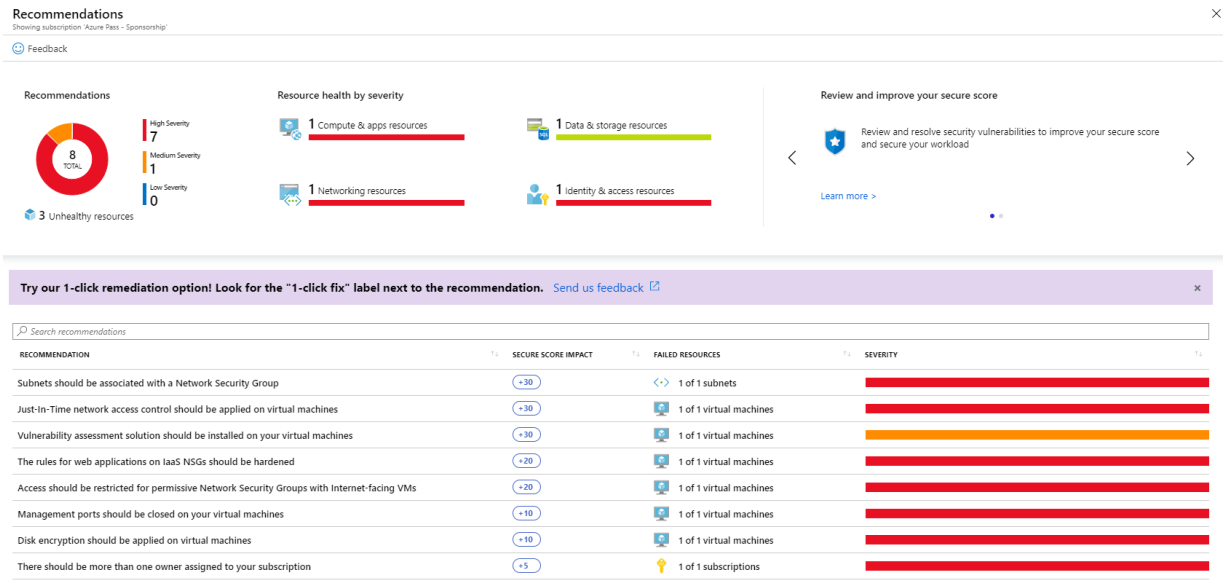
51.1.1 Task 1: View the secure score in the Azure Portal.

1. In the Azure dashboard, click **Security Center** and then click **Secure score**.
2. At the top you can see Secure score highlights:
   - The **Overall secure score** represents the score per policies, per selected subscription
   - **Secure score by category** shows you which resources need the most attention
   - **Top recommendations by secure score impact** provides you with a list of the recommendations that will improve your secure score the most if you implement them.



Note: The sum of the secure score of each subscription does not equal the overall secure score. The secure score is a calculation based on the ratio between your healthy resources and your total resources per recommendation, not a sum of secure scores across your subscriptions.

3. Click **View recommendations** to see the recommendations for that subscription that you can remediate to improve your secure score.
4. In the list of recommendations, you can see that for each recommendation there is a column that represents the **Secure score impact**. This number represents how much your overall secure score will improve if you follow the recommendations. For example, in the screen below, if you **Remediate vulnerabilities in container security configurations**, your secure score will increase by 35 points.



51.1.2 Task 2: View the individual secure scores.

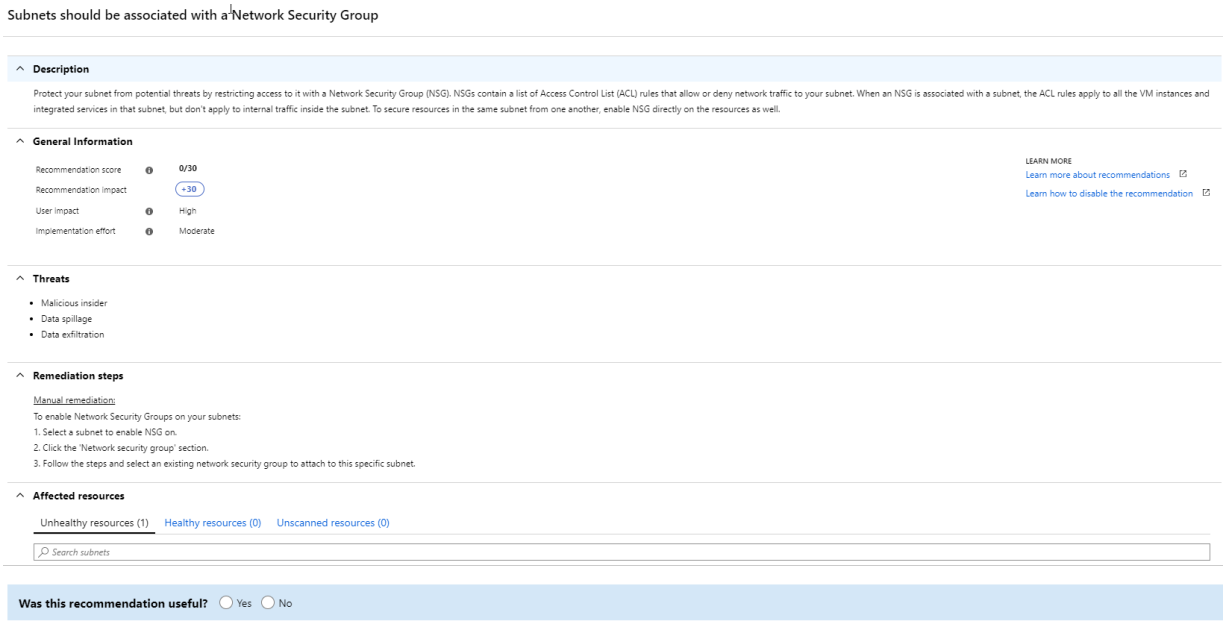
In addition, to view individual secure scores, you can find these within the individual recommendation blade.

The **Recommendation secure score** is a calculation based on the ratio between your healthy resources and your total resources. If the number of healthy resources is equal to the total number of resources, you get the maximum secure score of the recommendation of 50. To try to get your secure score closer to the max score, fix the unhealthy resources by following the recommendations.

The **Recommendation impact** lets you know how much your secure score improves if you apply the recommendation steps. For example, if your secure score is 42 and the **Recommendation impact** is +3, performing the steps outlined in the recommendation improve your score to become 45.

1. Click any of the recommendations in the Secure Score blade.

   The recommendation shows which threats your workload is exposed to if the remediation steps are not taken.



Results: In this lab, you learned how to improve your secure score in Azure Security Center

52 Module 4: Lab 8 - Create security baselines

Azure doesn't monitor security or respond to security incidents within the customer's area of responsibility. Azure does provide many tools (such as Azure Security Center, Azure Sentinel) that are used for this purpose. There is also an effort to help make every service as secure as possible by default. That is, every service comes with a baseline that is already designed to help provide security for most common-use cases. However, because Azure cannot predict how a service will be used, you should always review these security controls to evaluate whether they adequately mitigate risks.

This lab will guide you through a security baseline for Azure services. Each unit will provide a checklist of things to verify about the services you are using in your architecture.

In this lab, you will:

- Learn Azure platform security baselines and how they were created
- Create and validate a security baseline for the most commonly used Azure services

52.1 Exercise 1: Create an Identity & Access Management (IAM) baseline

Identity management is key to granting access and to the security enhancement of corporate assets. To secure and control your cloud-based assets you must manage identity and access for your Azure administrators, application developers, and application users.

IAM recommendations

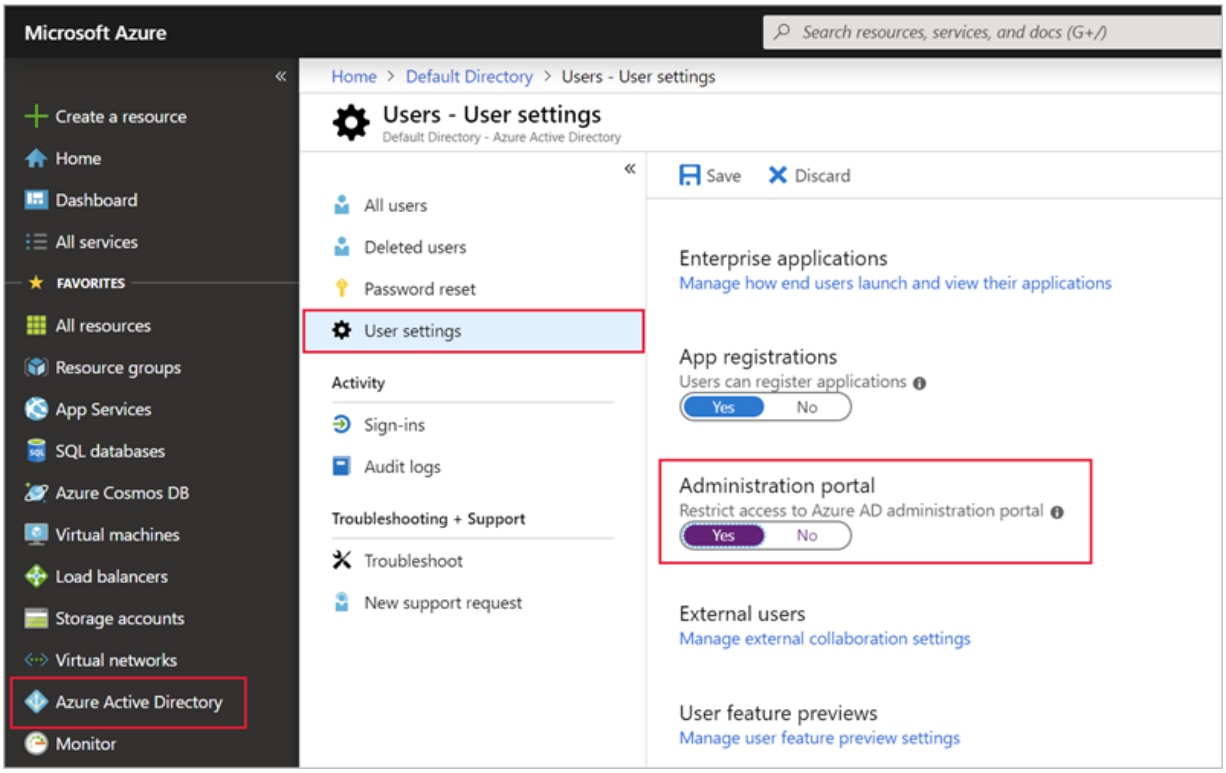
Here are the recommendations for identity and access management. Included with each recommendation are the basic steps to follow in the Azure portal.

52.1.1 Task 1: Restrict access to the Azure AD administration portal

All non-Administrators should not have access due to the sensitive data and the rules of least privilege.

1. Sign in to the Azure portal.

2. On the left, select **Azure Active Directory** > **Users**.
3. Go to **User settings**.
4. Ensure that Restrict access to Azure AD administration portal is set to Yes. Setting this value to Yes restricts all non-administrators from accessing any Azure AD data in the administration portal, but does not restrict such access using PowerShell or another client such as Visual Studio.

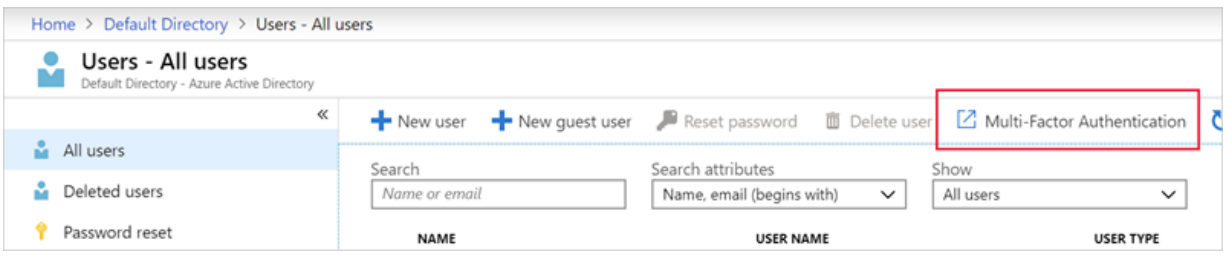


5. Click **Save**.

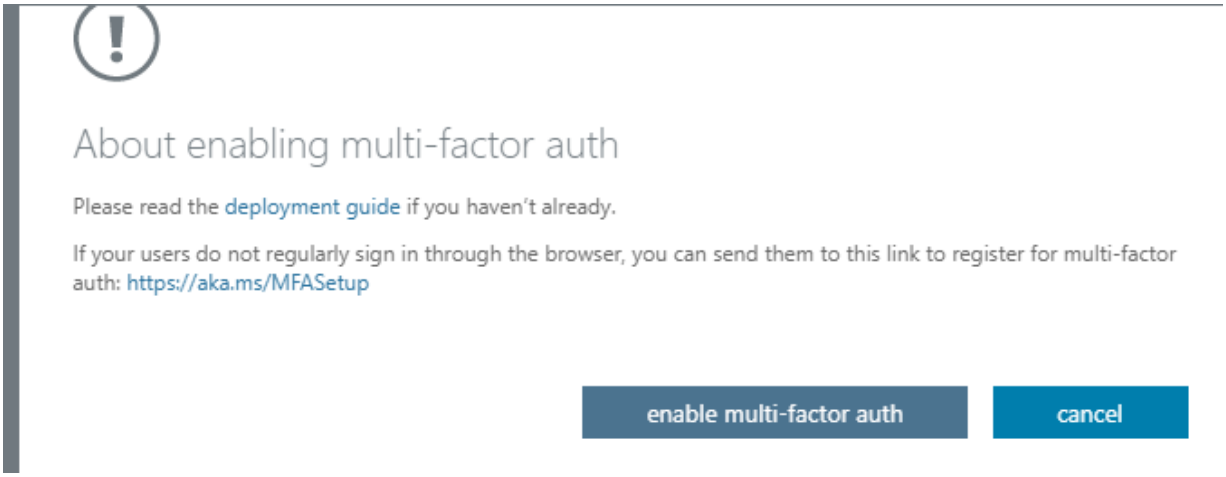
52.1.2 Task 2: Enable Azure Multi-Factor Authentication (MFA)

Enable it for privileged and non-privileged users.

1. Sign in to the Azure portal.
2. On the left, select **Azure Active Directory** > **Users**.
3. Click + **New User**.
4. On the New user blade, enter the following details and click Create:
   - **Username**: Abbi
   - **Name**: Abbi Skinner
   - **First name**: Abbi
   - **Last Name**: Skinner
   - **Roles**: Select Global Admin
5. On the left, select **Azure Active Directory** > **Users** > **All users**.
6. Select Multi-Factor Authentication. This will open a new window.
7. Select Abbi Skinner and click **Enable**



8. Select **enable multi-factor auth** then click **Close**.

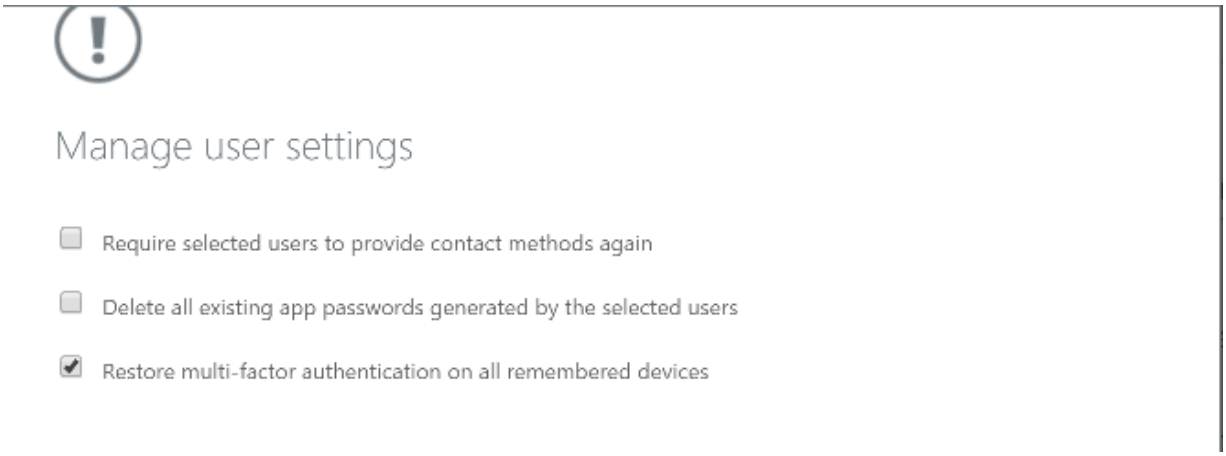


9. Abbi is now enabled for MFA.

52.1.3 Task 3: Block remembering MFA on trusted devices

Remember Multi-Factor Authentication feature for devices and browsers that are trusted by the user is a free feature for all Multi-Factor Authentication users. Users can bypass subsequent verifications for a specified number of days, after they've successfully signed-in to a device by using Multi-Factor Authentication. If an account or device is compromised, remembering Multi-Factor Authentication for trusted devices can negatively affect security.

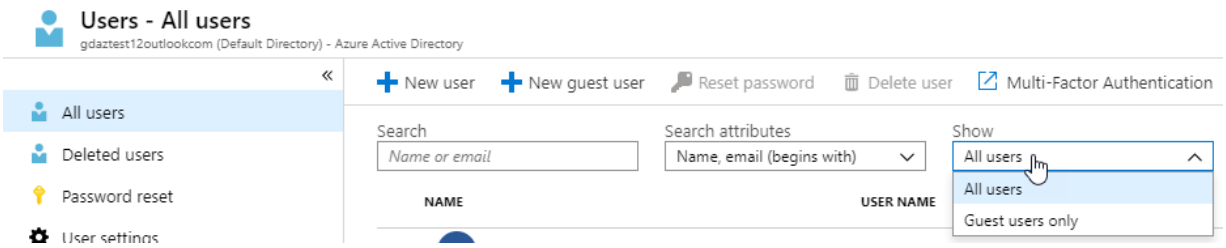
1. Sign in to the Azure portal.
2. On the left, select **Azure Active Directory** > **Users** > **All users**.
3. Select Multi-Factor Authentication.
4. Select **Abi Skinner**, then click **Manage users settings**.
5. Ensure that **Restore multi-factor authentication on all remembered devices** is Selected then click **Save**.



52.1.4 Task 4: About guests

In this task you will ensure that no guest users exist, or alternatively if the business requires guest users, ensure to limit their permissions.

1. Sign in to the Azure portal.
2. On the left, select **Azure Active Directory** > **Users** > **All users**.
3. Select the Show drop down and select **Guest users only**.
4. Verify that there are no guest users listed (`USER TYPE=Guest`).



52.1.5 Task 5: Password options

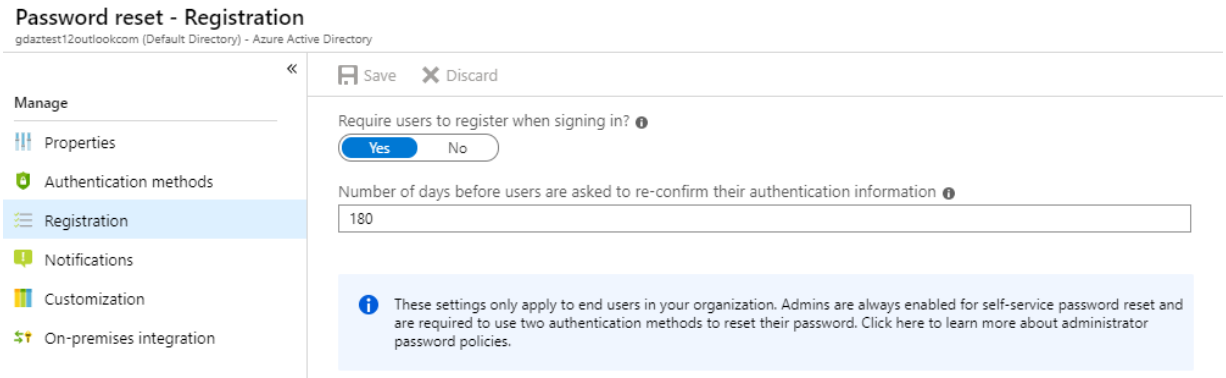
With dual identification set, an attacker would require compromising both the identity forms before they could maliciously reset a user's password.

1. Sign in to the Azure portal.
2. On the left, select **Azure Active Directory** > **Users**.
3. Select **Password reset**.
4. Go to Authentication methods.
5. Set the Number of methods required to reset to 2.
6. Select two methods and click **Save**.

52.1.6 Task 6: Establish an interval for reconfirming user authentication methods

If authentication reconfirmation is set to disabled, register users will never be prompted to re-confirm their authentication information.

1. Sign in to the Azure portal.
2. On the left, select **Azure Active Directory** > **Users**.
3. Go to **Password reset**.
4. Go to **Registration**.
5. Ensure that Number of days before users are asked to re-confirm their authentication information is not set to 0. The default is 180 days.

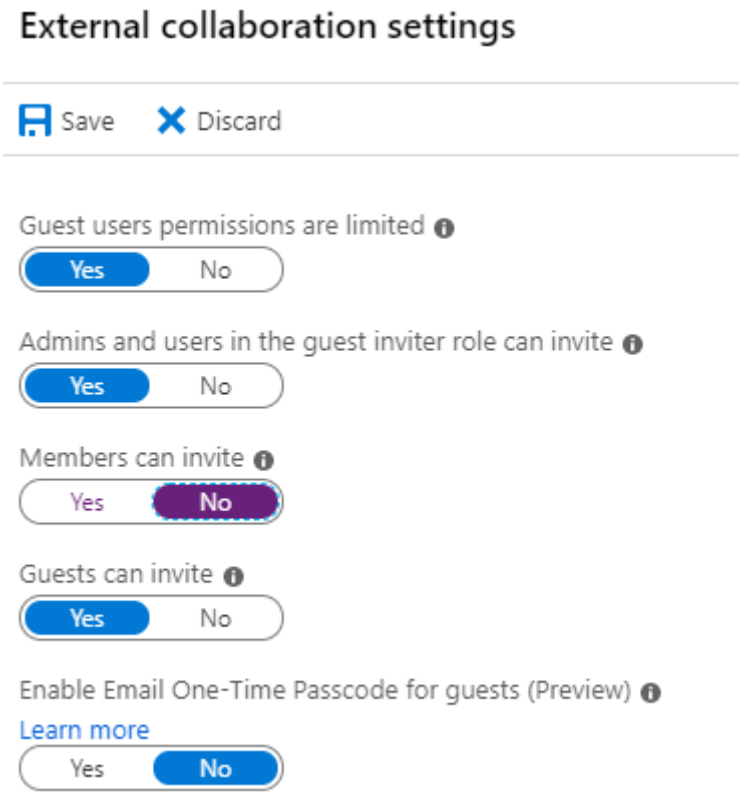


52.1.7 Task 7: Disable Members invitations

Restricting invitations through administrators only ensures that only authorized accounts have access Azure resources.

1. Sign in to the Azure portal.
2. On the left, select **Azure Active Directory** > **Users**.
3. Go to **User settings**.
4. Go to External users, click **Manage external collaboration settings**.

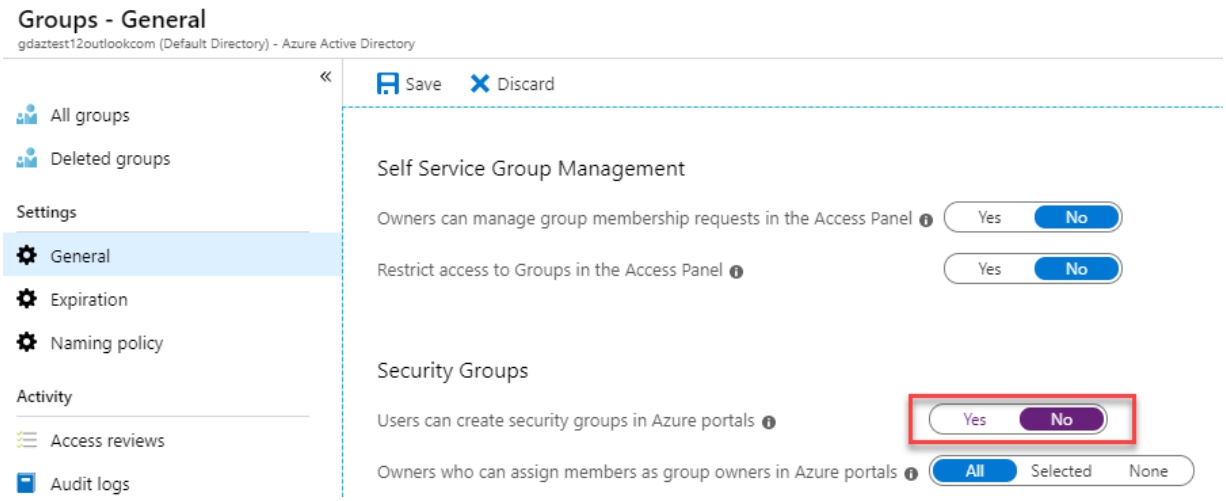
5. Ensure that Members can invite is set to **No**.



52.1.8 Task 8: Users to create and manage security groups

When this feature is enabled, all users in AAD are allowed to create new security groups. Security Group creation should be restricted to administrators.

1. Sign in to the Azure portal.
2. On the left, select **Azure Active Directory** > **Groups**.
3. Go to **General** under the settings section.
4. Ensure that Users can create security groups is set to **No**.

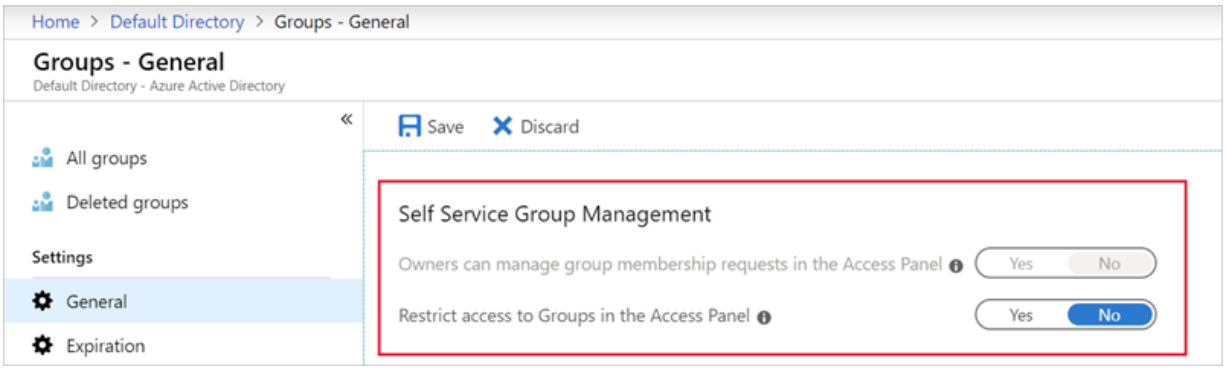


52.1.9 Task 9: Self-service group management enabled

Until your business requires this delegation to various users, it is a best practice to disable this feature.

1. Sign in to the Azure portal.
2. On the left, select **Azure Active Directory** > **Groups**

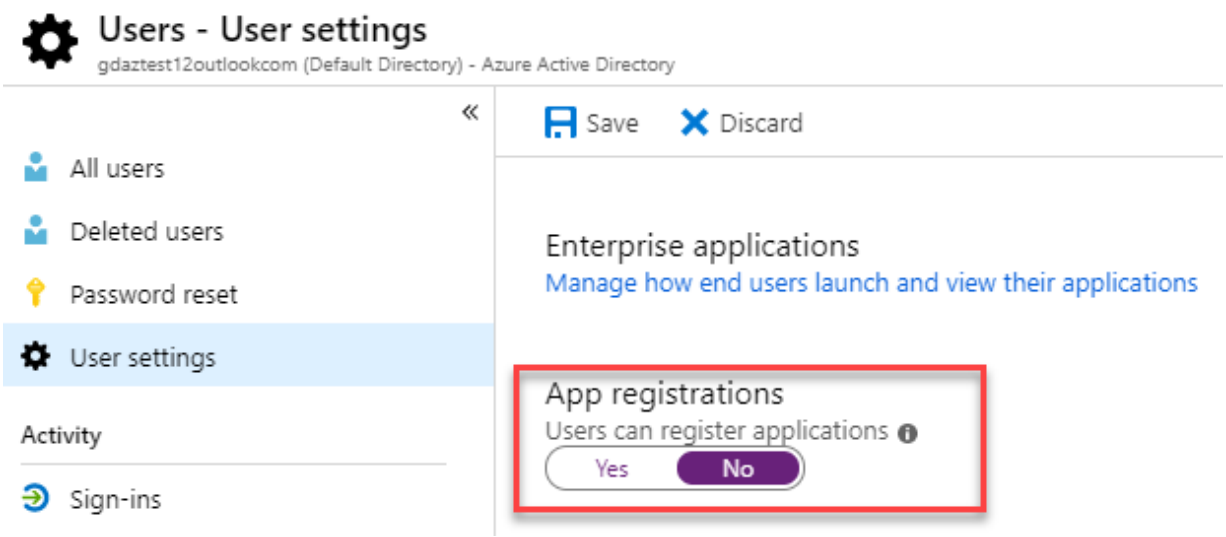
3. Go to **General** under the settings section.
4. Ensure that Self-service group management enabled is set to **No**.



52.1.10 Task 10: Application options - Allow users to register apps

Require administrators to register custom applications.

1. Sign in to the Azure portal.
2. On the left, select **Azure Active Directory** > **Users**
3. Go to User settings.
4. Ensure that User can register applications is set to **No** then click **Save**.



52.2 Exercise 2: Create an Azure Security Center baseline

Azure Security Center (ASC) provides unified security management and advanced threat protection for workloads running in Azure, on-premises, and in other clouds. The following are Security Center recommendations that, if followed, will set various security policies on an Azure subscription.

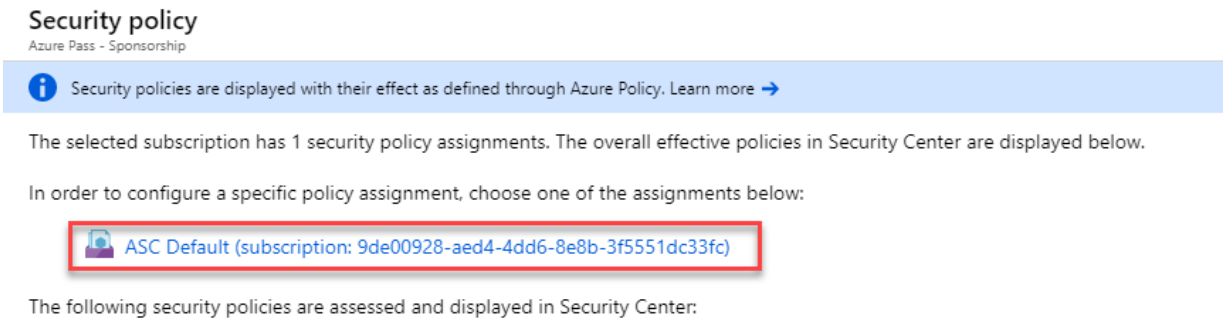
These policies define the set of controls that are recommended for your resources with an Azure subscription.

52.2.1 Task 1: Enable System Updates

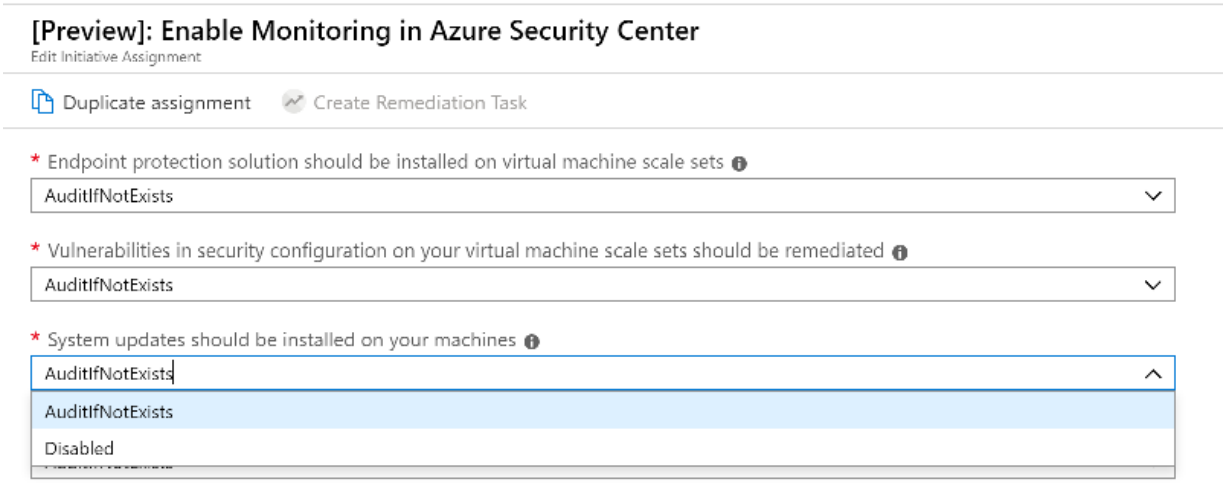
Azure Security Center monitors daily Windows and Linux virtual machines (VMs) and computers for missing operating system updates. Security Center retrieves a list of available security and critical updates from Windows Update or Windows Server Update Services (WSUS), depending on which service is configured on a Windows computer. Security Center also checks for the latest updates in Linux systems. If your VM or computer is missing a system update, Security Center will recommend that you apply system updates.

1. Sign in to the Azure portal.
2. Select **Security Policy** on the **Security Center** main menu.

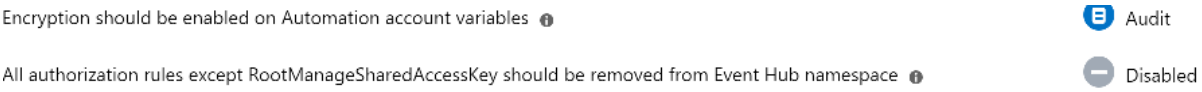
3. The **Policy Management** screen is displayed.
4. Choose your subscription from the displayed list.
5. Select **View effective policy**.
6. Check that **System updates should be installed on your machines** is one of the policies.
7. Click the Enable Monitoring in Azure Security Center link (This may also be displayed as ASC Default with a GUID).



8. In this example, the ASC agent has not been deployed to a VM or physical machine so the message AuditIfNotExists is displayed. AuditIfNotExists enables auditing on resources that match the if condition. If the resource is not deployed, NotExists is displayed.



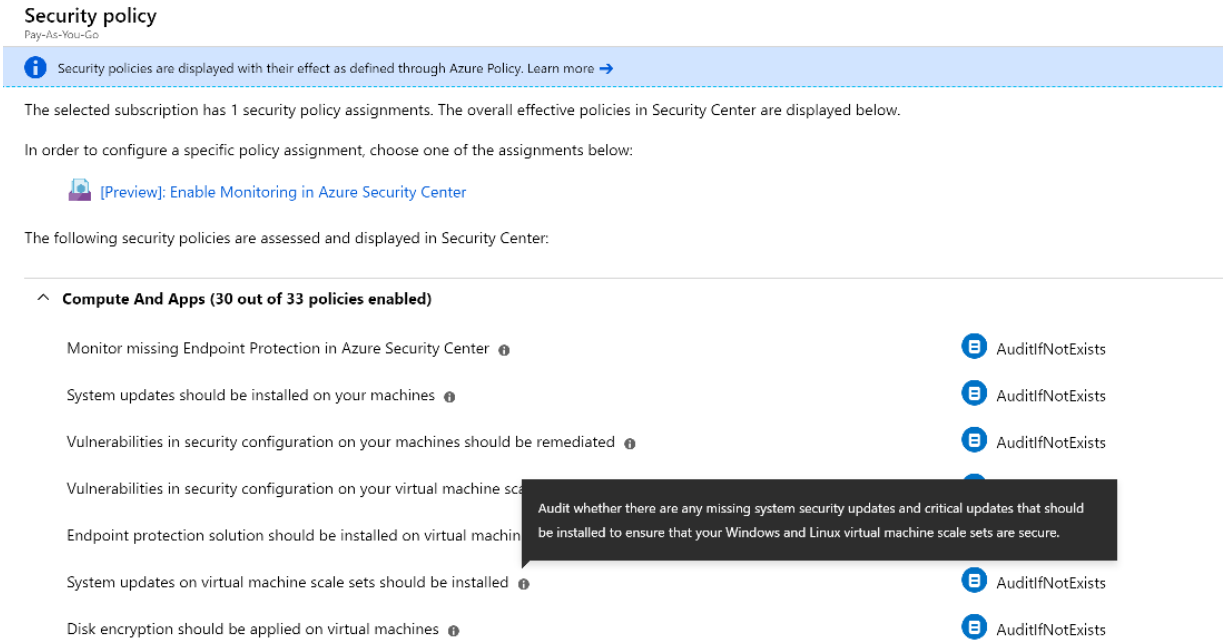
If enabled, Audit is displayed. If deployed but disabled, Disabled is displayed.



52.2.2 Task 2: Enable Security Configurations

Azure Security Center monitors security configurations by applying a set of over 150 recommended rules for hardening the OS, including rules related to firewalls, auditing, password policies, and more. If a machine is found to have a vulnerable configuration, Security Center generates a security recommendation.

1. Sign in to the Azure portal.
2. On the Hub menu select **Security Center**.
3. Select **Security Policy** then select your **Subscription**.
4. Click **View effective policy**.
5. The Policy Management screen is displayed.
6. Check that **Vulnerabilities in security configuration on your virtual machine scale sets should be remediated** is one of the policies.



Note: All of the following policies that have a (*) in their title are listed in the Security policies blade as described above

- **Enable Endpoint Protection** - *Endpoint protection is recommended for all virtual machines.*
- **Enable Disk Encryption** - *Azure Security Center recommends that you apply disk encryption if you have Windows or Linux VM disks that are not encrypted using Azure Disk Encryption. Disk Encryption lets you encrypt your Windows and Linux IaaS VM disks. Encryption is recommended for both the OS and data volumes on your VM.*
- **Enable Network Security Groups** *Azure Security Center recommends that you enable a network security group (NSG) if one is not already enabled. NSGs contain a list of Access Control List (ACL) rules that allow or deny network traffic to your VM instances in a Virtual Network. NSGs can be associated with either subnets or individual VM instances within that subnet. When an NSG is associated with a subnet, the ACL rules apply to all the VM instances in that subnet. In addition, traffic to an individual VM can be restricted further by associating an NSG directly to that VM.*
- **Enable Web Application Firewall** - *Azure Security Center may recommend that you add a web application firewall (WAF) from a Microsoft partner to secure your web applications.*
- **Enable Vulnerability Assessment** - *The vulnerability assessment in Azure Security Center is part of the Security Center virtual machine (VM) recommendations. If Security Center doesn't find a vulnerability assessment solution installed on your VM, it recommends that you install one. A partner agent, after being deployed, starts reporting vulnerability data to the partner's management platform. In turn, the partner's management platform provides vulnerability and health monitoring data back to Security Center.*
- **Enable Storage Encryption** - *When this setting is enabled, any new data in Azure Blobs and Files will be encrypted.*
- **Enable JIT Network Access** - *Just-in-time (JIT) virtual machine (VM) access can be used to lock down inbound traffic to your Azure VMs, reducing exposure to attacks while providing easy access to connect to VMs when needed.*
- **Enable Adaptive Application Controls** - *Adaptive application control is an intelligent, automated end-to-end application whitelisting solution from Azure Security Center. It helps you control which applications can run on your Azure and non-Azure VMs (Windows and Linux), which, among other benefits, helps harden your VMs against malware. Security Center uses machine learning to analyze the applications running on your VMs and helps you apply the specific whitelisting rules using this intelligence. This capability greatly simplifies the process of configuring and maintaining application whitelisting policies.*
- **Enable SQL Auditing & Threat Detection** - Azure Security Center will recommend that you turn on auditing and threat detection for all databases on your Azure SQL servers if auditing is not already enabled. Auditing and threat detection can help you maintain regulatory compliance, understand database activity, and gain insight into discrepancies and anomalies that could indicate business concerns or suspected security violations.
- **Enable SQL Encryption** - *Azure Security Center will recommend that you enable Transparent*

Data Encryption (TDE) on SQL databases if TDE is not already enabled. TDE protects your data and helps you meet compliance requirements by encrypting your database, associated backups, and transaction log files at rest, without requiring changes to your application.

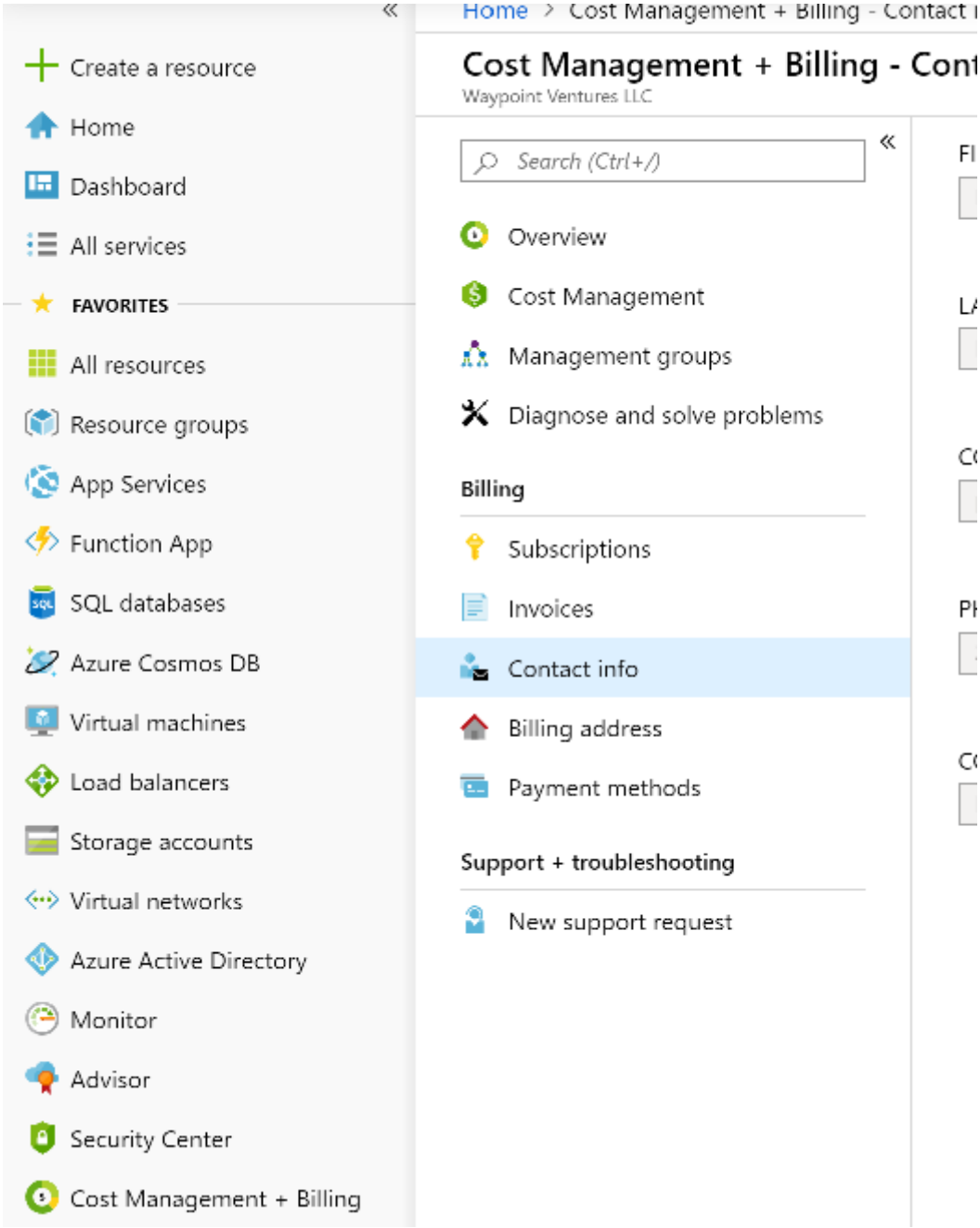
- **Set Security Contact Email and Phone Number** - *Azure Security Center will recommend that you provide security contact details for your Azure subscription if you haven't already. This information will be used by Microsoft to contact you if the Microsoft Security Response Center (MSRC) discovers that your customer data has been accessed by an unlawful or unauthorized party. MSRC performs select security monitoring of the Azure network and infrastructure and receives threat intelligence and abuse complaints from third parties.*

7. Select **Cost Management + Billing**.

   Note: The following steps will not work with an Azure Pass subscription but reamins in this lab in order to identify the steps requried in a Real World scenrio.

8. The Contact info screen is displayed.

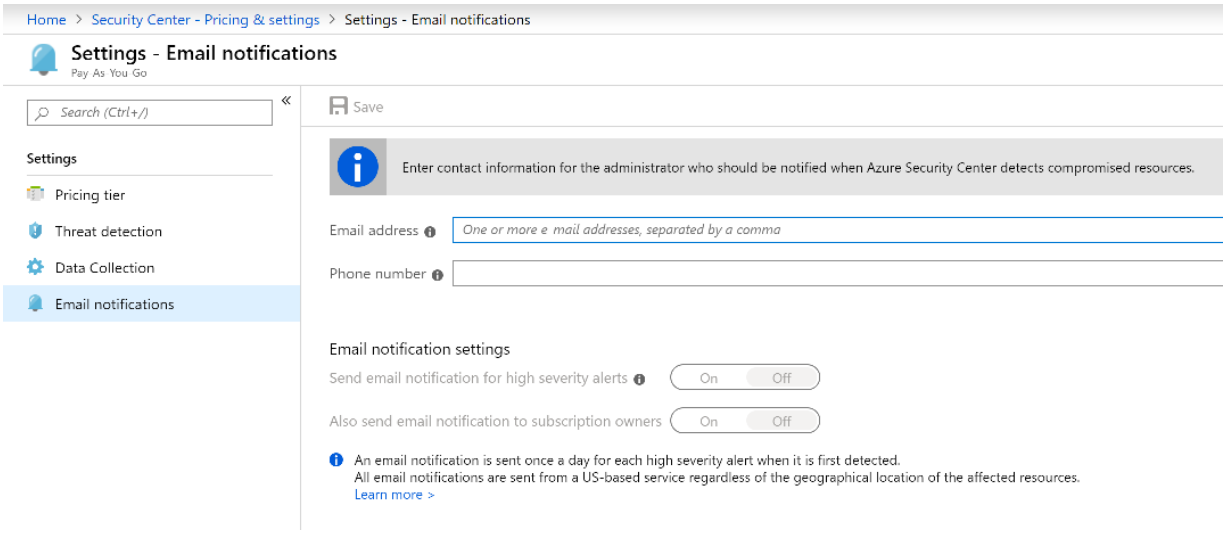
9. Enter or validate the contact information displayed.



52.2.3 Task 3: Enable Send me emails about alerts

Azure Security Center will recommend that you provide security contact details for your Azure subscription if you haven't already.

1. On the Hub menu select **Security Center**.
2. Select **Pricing & settings**.
3. The Pricing & settings screen is displayed.
4. Click on your Subscription.
5. Click **Email notifications**.
6. Select **Save**.



52.2.4 Task 4: Enable Send email also to subscription owners

Azure Security Center will recommend that you provide security contact details for your Azure subscription if you haven't already.

1. Using the above Email notifications form, additional emails can be added separated by commas.
2. Click **Save**.

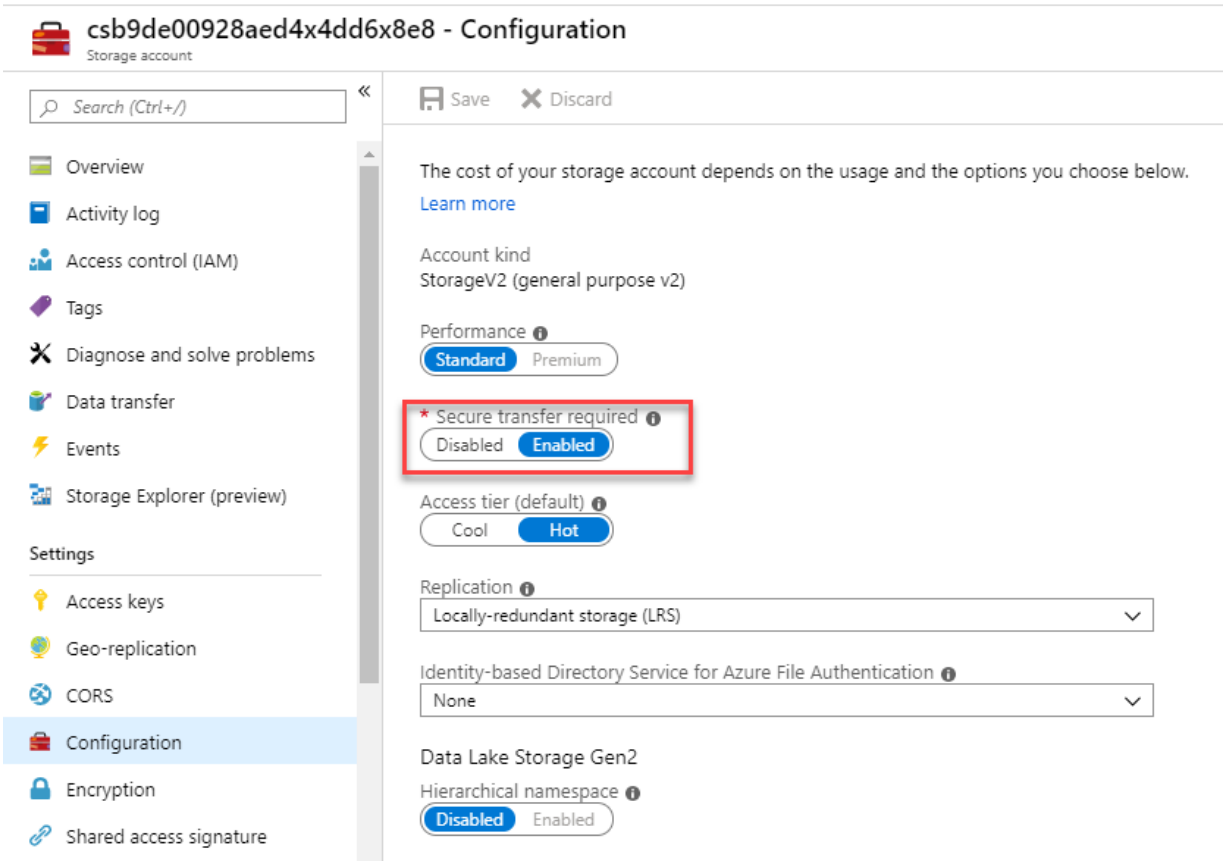
52.3 Exercise 3: Create an Azure storage accounts baseline

An Azure storage account provides a unique namespace to store and access your Azure Storage data objects. Storage Accounts also need to be secured.

52.3.1 Task 1: Require security-enhanced transfers

Another step you should take to ensure the security of your Azure Storage data is to encrypt the data between the client and Azure Storage. The first recommendation is to always use the HTTPS protocol, which ensures secure communication over the public Internet. You can enforce the use of HTTPS when calling the REST APIs to access objects in storage accounts by enabling Secure transfer required for the storage account. Connections using HTTP will be refused once this is enabled.

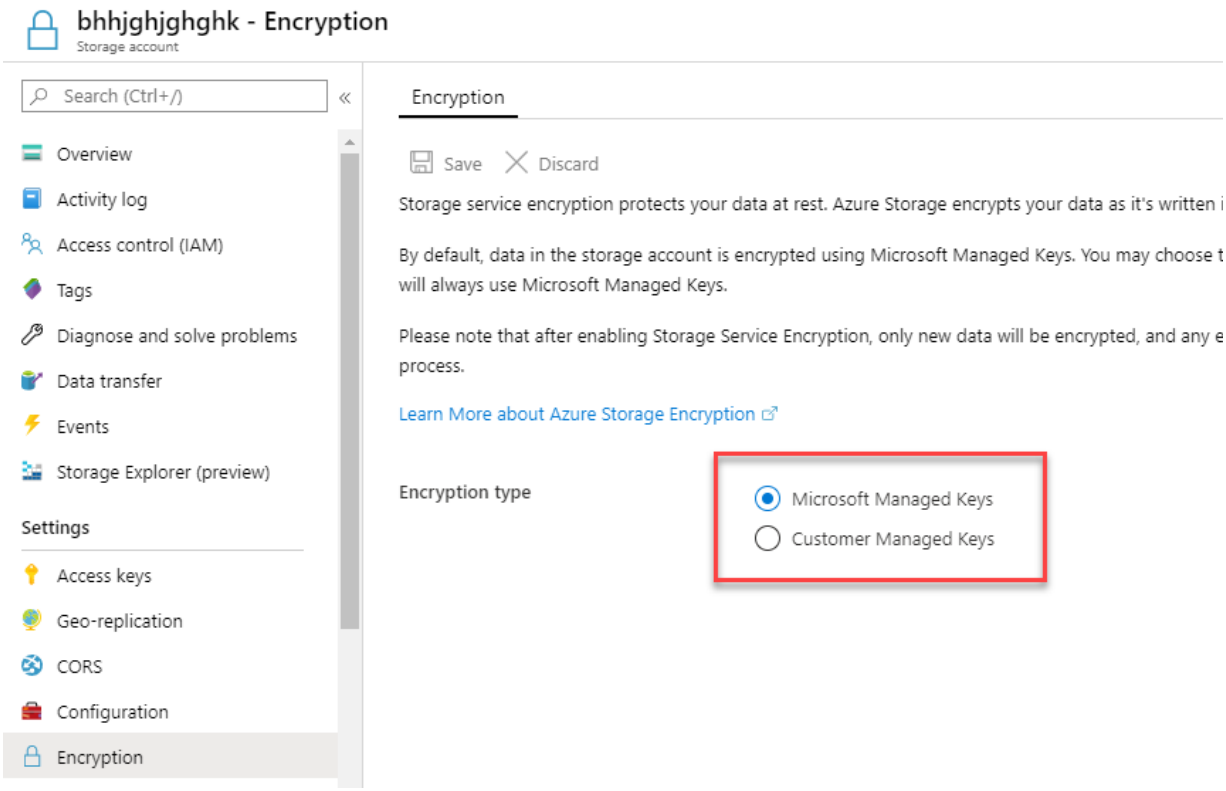
1. Go to **Storage Accounts** under **All services**.
2. Select the storage account.
3. Under **Settings**, select **Configuration**.
4. Ensure **Secure Transfer required** is set to **Enabled**.



52.3.2 Task 2: Enable binary large object (blob) encryption

Azure Blob storage is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing massive amounts of unstructured data. Unstructured data is data that does not adhere to a particular data model or definition, such as text or binary data. Storage service encryption protects your data at rest. Azure Storage encrypts your data as it's written in its datacenters, and automatically decrypts it for you as you access it.

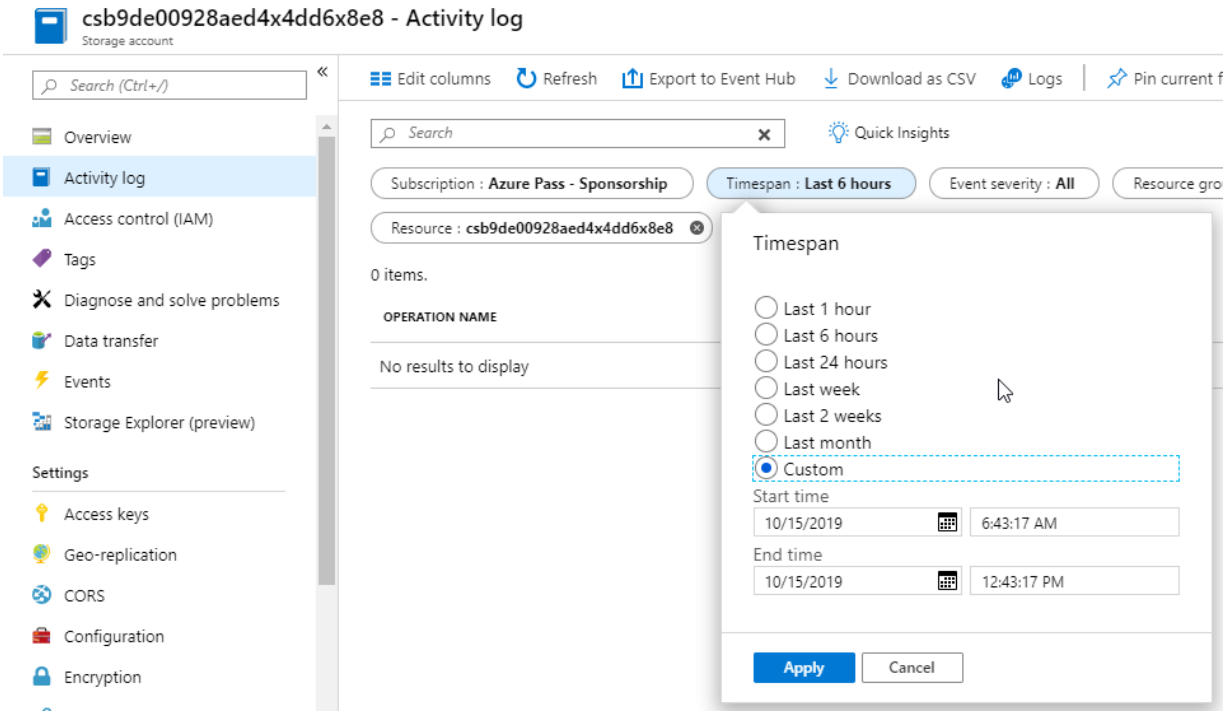
1. Go to Storage Accounts under Azure services.
2. Select the storage account.
3. Under **Settings**, select **Encryption**.
4. Azure Storage encryption is enabled by default and cannot be disabled.



52.3.3 Task 3: Periodically regenerate access keys

When you create a storage account, Azure generates two 512-bit storage access keys, which are used for authentication when the storage account is accessed. Rotating these keys periodically ensures that any inadvertent access or exposure to these keys could be undermined.

1. Go to **Storage Accounts** under Azure services.
2. Select the storage account.
3. For the storage account, go to **Activity log**.
4. Under Timespan drop-down, select **Custom** and choose Start Time and End Time so it creates a 90 day range.
5. Click **Apply**.



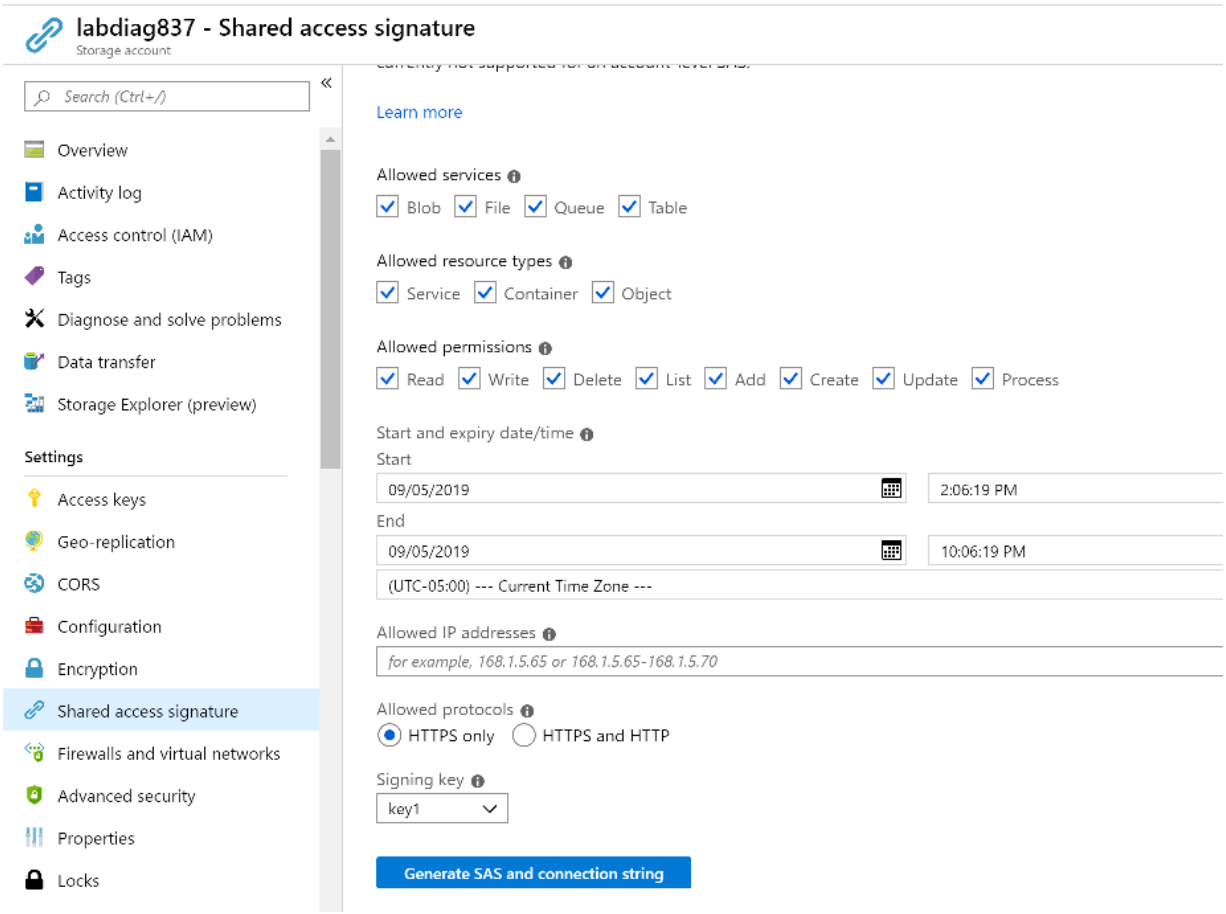
52.3.4 Task 4: Require Shared Access Signature (SAS) tokens to expire within an hour

A shared access signature (SAS) is a URI that grants restricted access rights to Azure Storage resources. You can provide a shared access signature to clients who should not be trusted with your storage account key but to whom you wish to delegate access to certain storage account resources. By distributing a shared access signature URI to these clients, you can grant them access to a resource for a specified period of time, with a specified set of permissions.

Currently verification of a SAS token expiry times cannot be accomplished. Until Microsoft makes token expiry time as a setting rather than a token creation parameter, this recommendation would require a manual verification.

1. Go to **Storage Accounts**.
2. Select the existing account.
3. For the storage account, go to **Shared Access signature**.
4. Set the Start and expiry date/time.
5. Set Allowed protocols to HTTPS only.

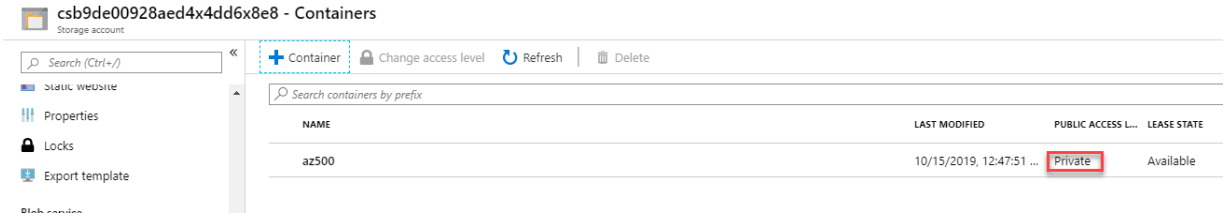
   Both SAS features are shown below.



52.3.5 Task 5: Require only private access to blob containers

You can enable anonymous, public read access to a container and its blobs in Azure Blob storage. By doing so, you can grant read-only access to these resources without sharing your account key, and without requiring a shared access signature (SAS). By default, a container and any blobs within it may be accessed only by a user that has been given appropriate permissions. To grant anonymous users read access to a container and its blobs, you can set the container public access level. When you grant public access to a container, then anonymous users can read blobs within a publicly accessible container without authorizing the request.

1. Go to **Storage Accounts**.
2. For the storage account, select **Containers** under **Blob Service**.
3. Click + **Container**.
4. Give the container the name **az500** and click **OK**.
5. Ensure that Public access level to Private.



52.4 Exercise 4: Create an Azure SQL Database baseline

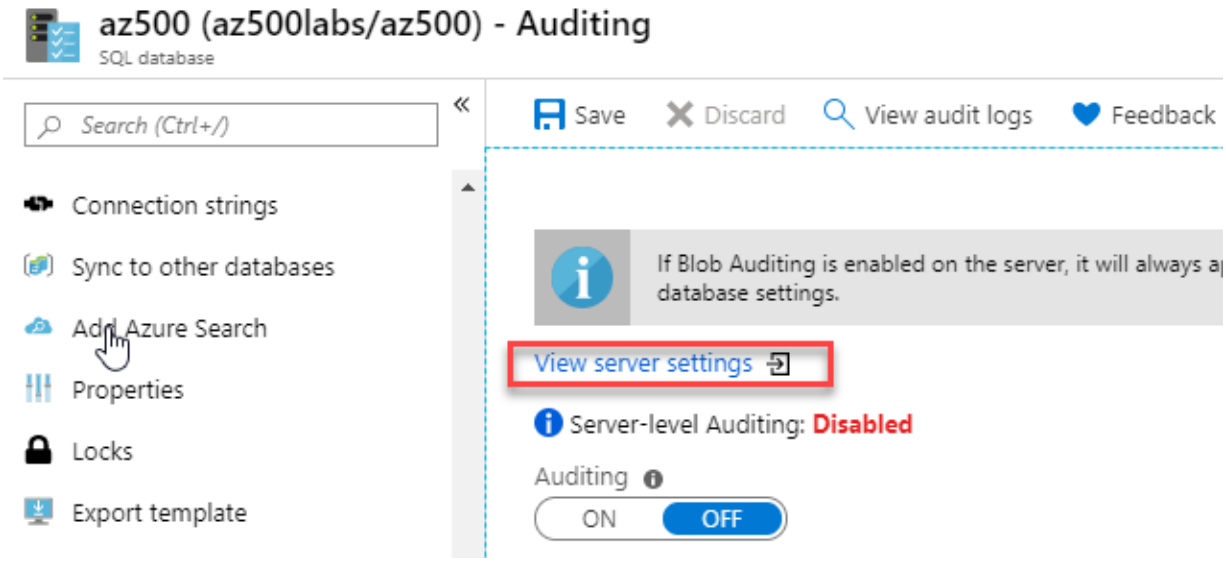
Azure SQL Server is a cloud-based relational database server that supports many of the same features as Microsoft SQL Server. It provides an easy transition from an on-premises database into a cloud-based one with built-in diagnostics, redundancy, security and scalability. This exercise looks at the security recommendations to set Azure SQL Server policies.

52.4.1 Task 1: Enable auditing

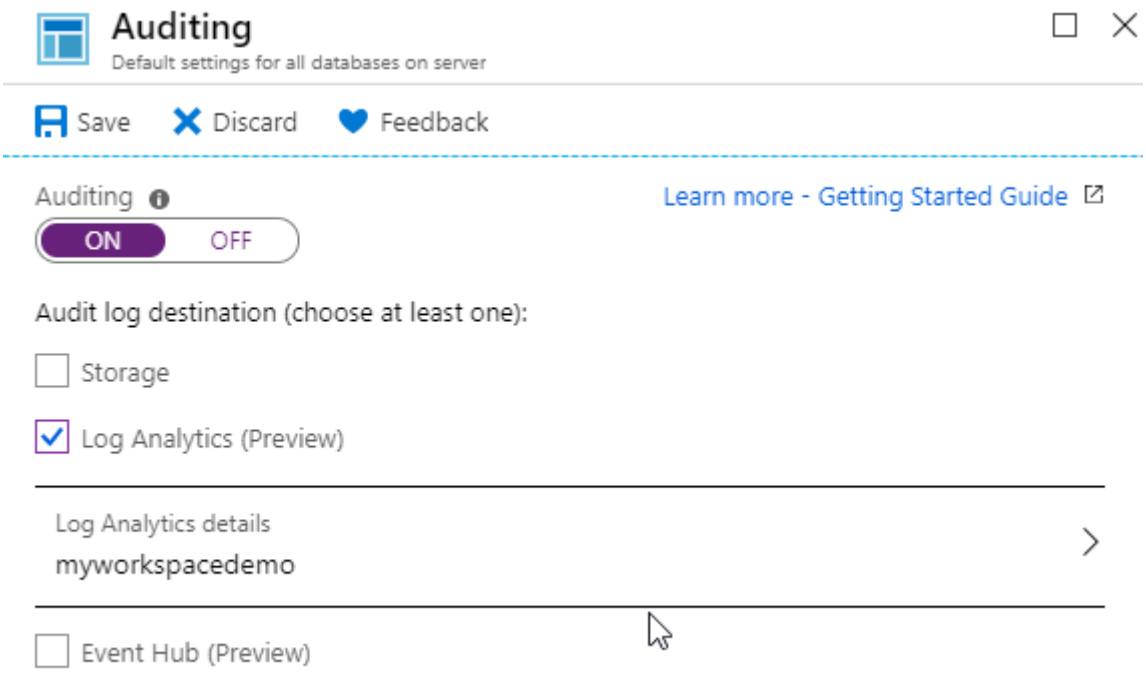
Auditing for Azure SQL Database and SQL Data Warehouse tracks database events and writes them to an audit log in your Azure storage account, OMS workspace or Event Hubs. Auditing also:

- Helps you maintain regulatory compliance, understand database activity, and gain insight into discrepancies and anomalies that could indicate business concerns or suspected security violations.
- Enables and facilitates adherence to compliance standards, although it doesn't guarantee compliance.

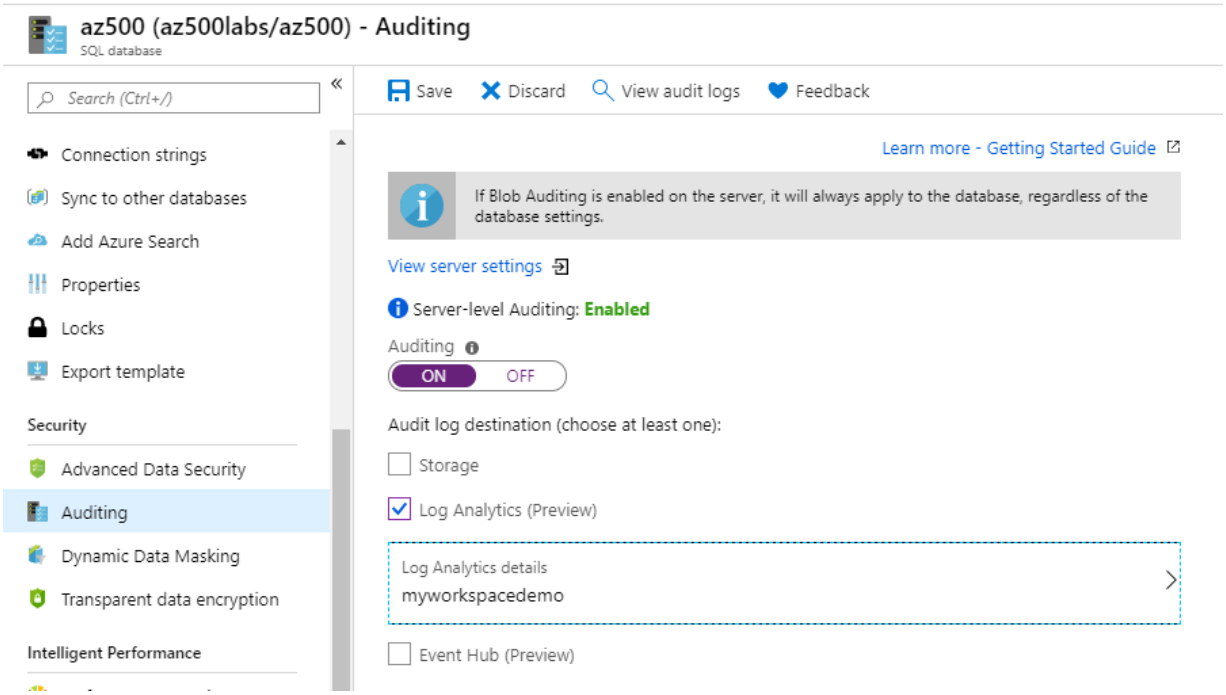
1. In the Azure Portal go to **SQL databases**.
2. Click + **Add**.
3. Create the database with the following settings then click **Review + create** and click **Create**:
   - **Resource Group**: Select myResourceGroup
   - **Database name**: az500
   - **Server**: **Create new**
     - **Server Name** Give the server a unique name
     - **Server admin login**: localadmin
     - **Password**: Pa55w.rd1234
     - **Location**: EastUS
4. Once the deployment is complete, click **Go to resource**.
5. Select **Auditing**, under the **Security** section.
6. Click **View server settings**.



7. Select **On** and check the box next to **Log Analytics**.
8. Select your Log Analyics workspace created in earlier labs and then click **Save**.



9. Exit the Auditing blade.
10. Ensure that Auditing is set to **On** and check the box next to **Log Analytics**.
11. Select your Log Analyics workspace created in earlier labs and then click **Save**.



52.4.2 Task 2: Enable a threat detection service

Threat detection for single and pooled databases detects anomalous activities indicating unusual and potentially harmful attempts to access or exploit databases. Threat detection can identify Potential SQL injection, Access from unusual location or data center, Access from unfamiliar principal or potentially harmful application, and Brute force SQL credentials. Threat detection is part of the advanced data security (ADS) offering, which is a unified package for advanced SQL security capabilities. Threat detection can be accessed and managed via the central SQL ADS portal.

1. In the Azure portal go to **SQL databases**.

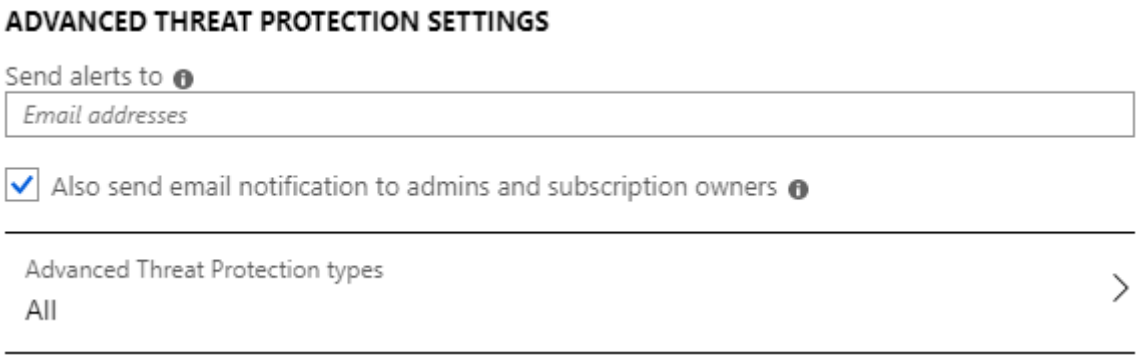
2. Under **Security**, then navigate to **Advanced Data Security**.
3. Click **Settings**.
4. Select **Enable Advanced Data Security on the server** and click **Yes** then click **Save**.

52.4.3 Task 3: Enable all threat detection types

Advanced data security (ADS) provides a set of advanced SQL security capabilities, including data discovery & classification, vulnerability assessment, and Advanced Threat Protection (ATP).

Advanced Threat Protection is part of the advanced data security (ADS) offering, which is part of the defense in depth SQL security strategy. Advanced Threat Protection can be accessed and managed via the central SQL ADS portal.

1. In the Azure portal go to **SQL databases**.
2. Under **Security**, then navigate to **Advanced Data Security**.
3. Click **Settings**.
4. Ensure that **Send alerts** is set as appropriate.



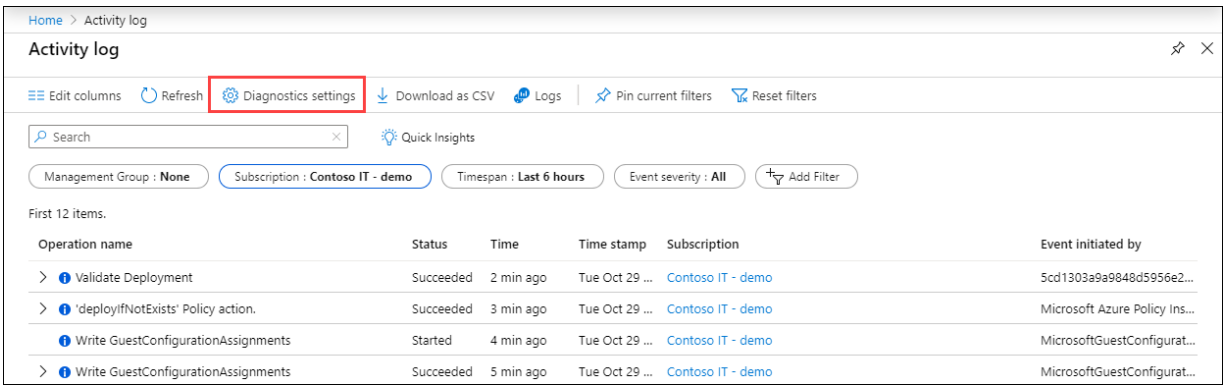
52.5 Exercise 5: Create a logging and monitoring baseline

Logging and monitoring are a critical requirement when trying to identify, detect, and mitigate security threats. Having a proper logging policy can ensure you can determine when a security violation has occurred, but also potentially identify the culprit responsible. Azure Activity logs provide data about both external access to a resources and diagnostic logs, which provide information about the operation of that specific resource.

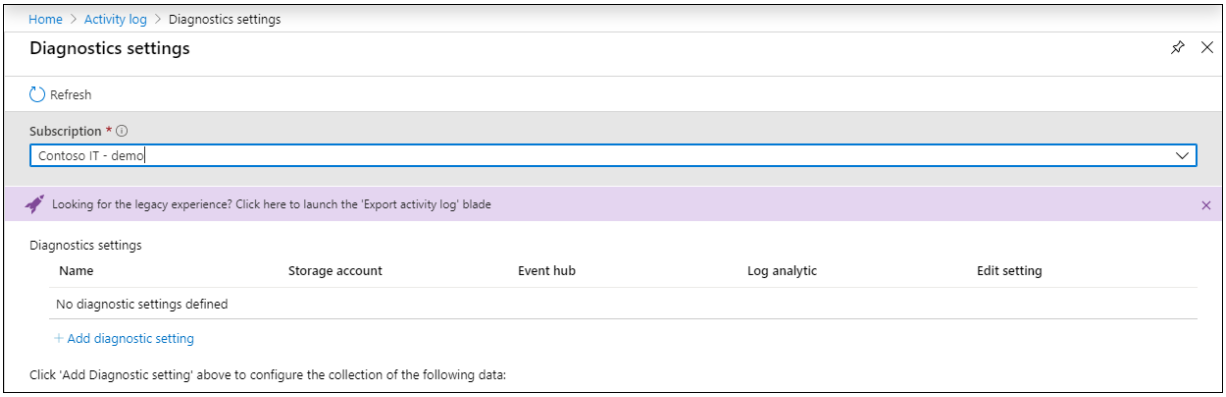
52.5.1 Task 1: Ensure that a log profile exists

The Azure Activity Log provides insight into subscription-level events that have occurred in Azure. This includes a range of data, from Azure Resource Manager operational data to updates on Service Health events. The Activity Log was previously known as Audit Logs or Operational Logs, since the Administrative category reports control-plane events for your subscriptions. There is a single Activity Log for each Azure subscription. It provides data about the operations on a resource from the outside. Diagnostic Logs are emitted by a resource and provide information about the operation of that resource. You must enable diagnostic settings for each resource.

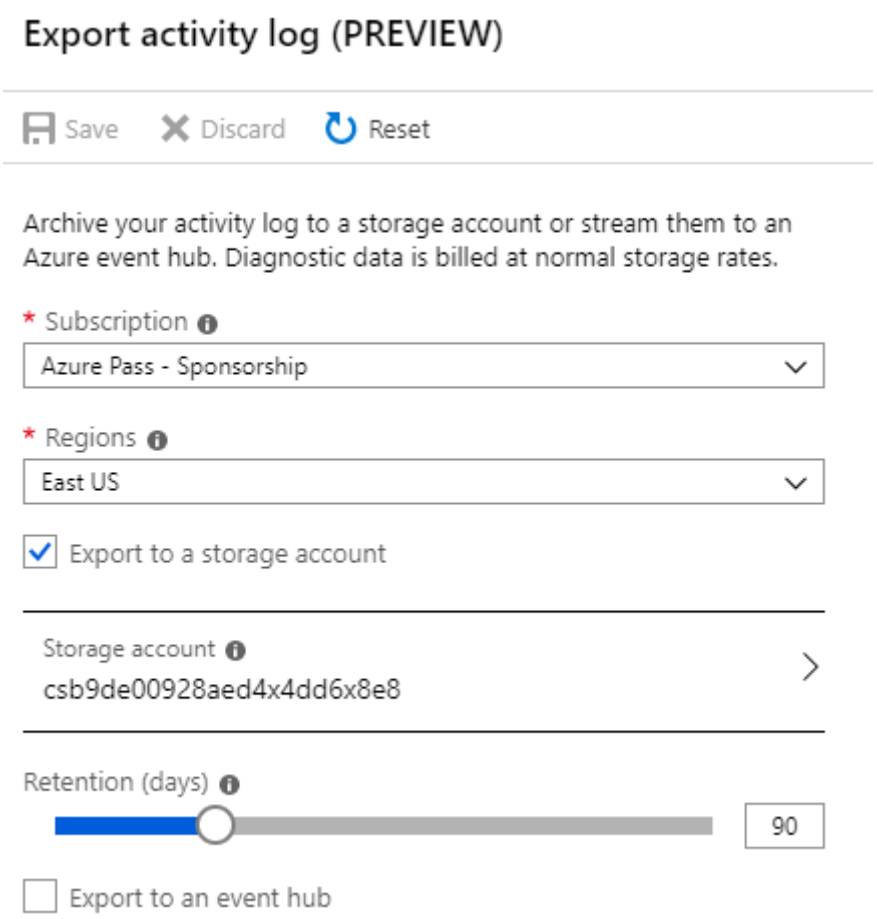
1. In the Azure Portal go to **Monitor**, then select **Activity log**.
2. Click **Diagnostic settings**.



3. Click the purple banner for the legacy experience.



4. Configure the following settings then click **Save**.
   - **Region**: EastUS
   - **Select**: Export to Storage Account
   - **Storage Account**: Select your storage account and click OK
   - **Retention**: 90 days
5. Select **Save**.



52.5.2 Task 2: Change activity log retention is set to 365 days or more

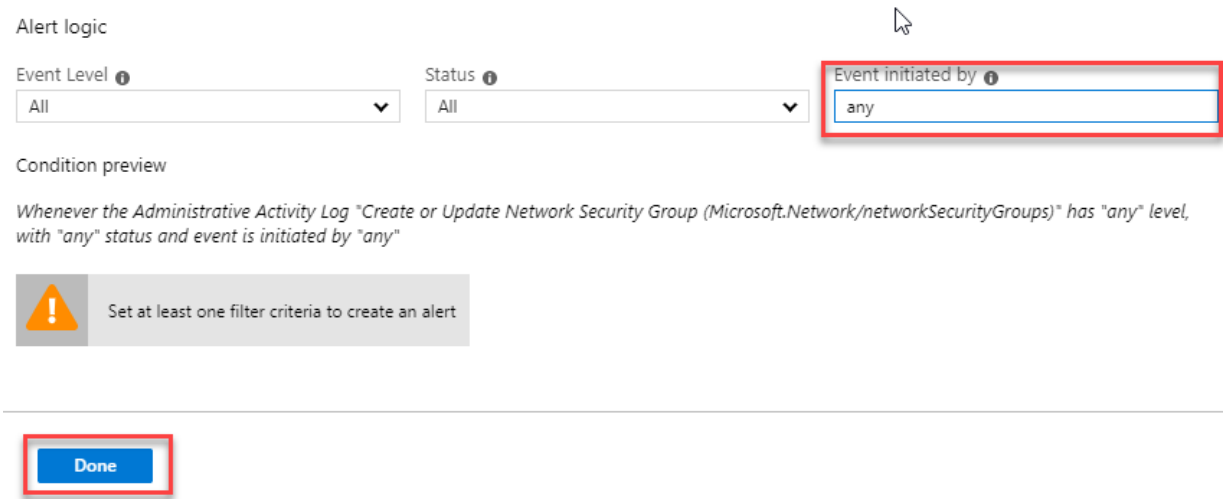
Setting the Retention (days) to 0 retains the data forever.

1. Follow the steps listed above. Adjust the Retention days slider bar.

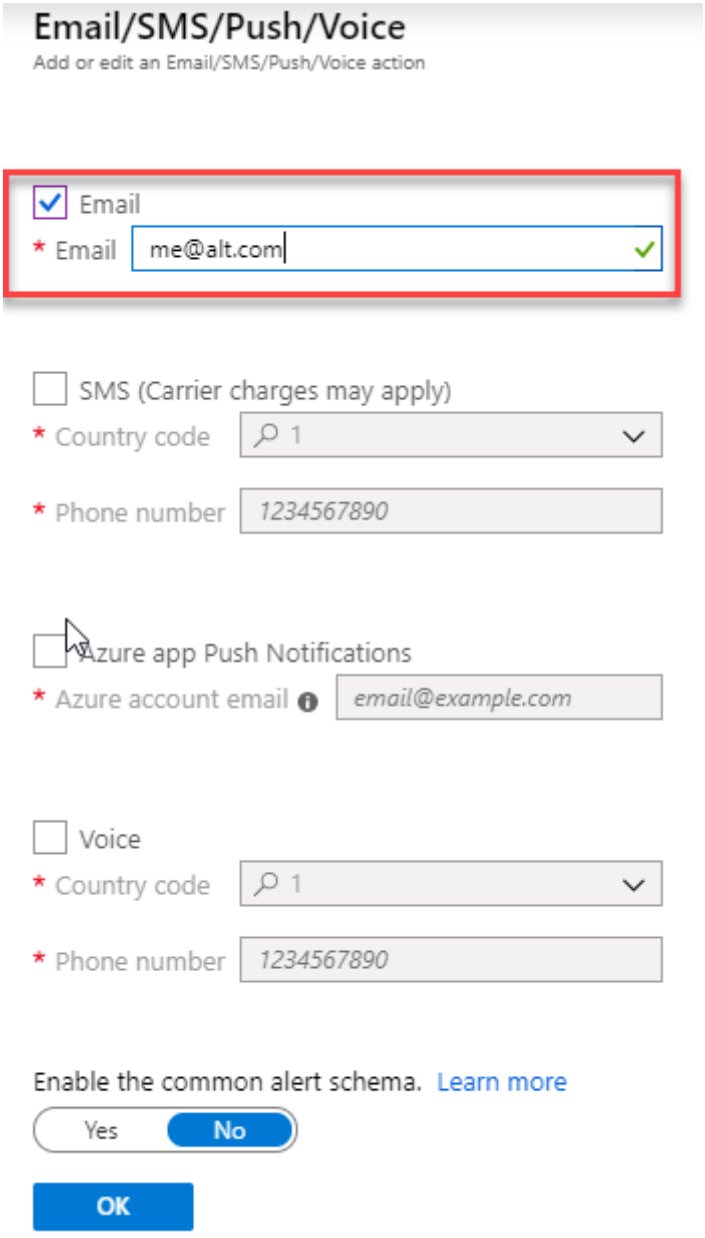
52.5.3 Task 3: Create an activity log alert for "Creating, updating, or deleting a Network Security Group"

By default, no monitoring alerts are created when NSGs are created/updated/deleted. Changing or deleting a security group can allow internal resources to be accessed from improper sources, or for unexpected outbound network traffic.

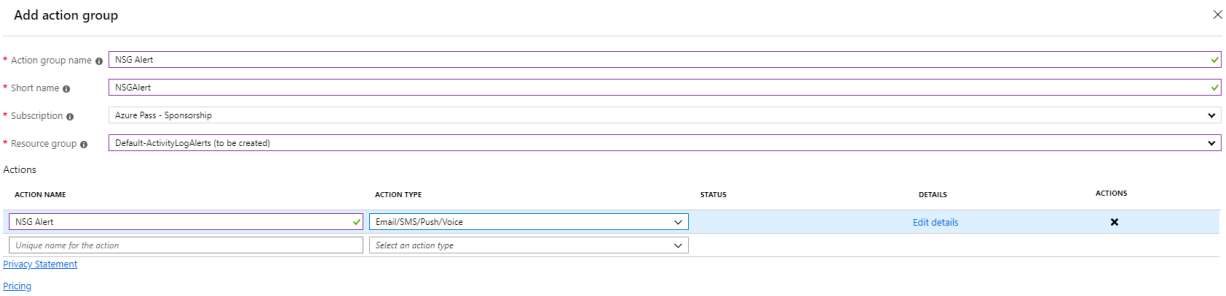
1. In to the Azure portal go to **Monitor**, then select **Alerts**.
2. Select **+ New alert rule**.
3. In the **Resource** section click **Select**.
4. Select your subscription and click **Done**.
5. In the **Condition** section click **Add**.
6. Search for **Create or Update Network Security Group** and select it.
7. On the Configure signal logic blade, in the Event initiated by enter **any** and click **Done**.



8. In the **Actions** section click **Create action group**.
9. On the Add action group blade enter the following details:
   - **Action group name**: NSG Alert
   - **Short name**: NSGAlert
   - **Action Name**: NSG Alert
   - **Action type**: Email/SMS/Push/Voice
10. On the **Email/SMS/Push/Voice** blade check the email box and enter your email address and click **OK**.

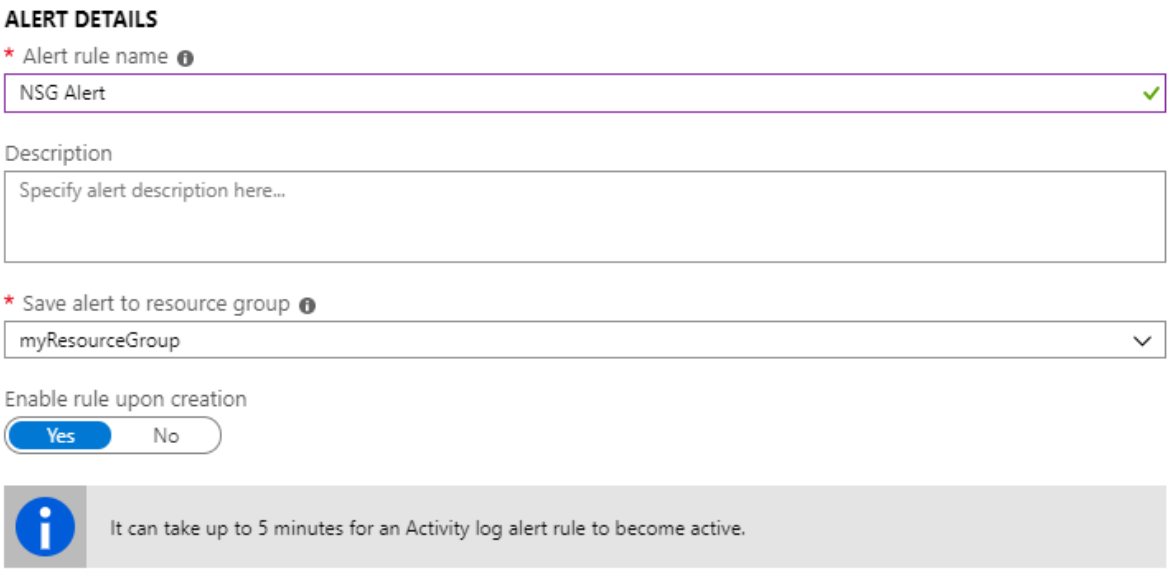


11. On the Add action group blade click **OK**.



12. On the Create rule blade, in the **Alert Details** section enter the following details:

    - **Alert rule name**: NSG Alert
    - **Save to resource group**: myResourceGroup



13. Click **Create alert rule**

52.6 Exercise 6: Create a Networking baseline

Azure networking services maximize flexibility, availability, resiliency, security, and integrity by design. Network connectivity is possible between resources located in Azure, between on-premises and Azure-hosted resources, and to and from the Internet and Azure.

52.6.1 Task 1: Restrict RDP and SSH access from the Internet

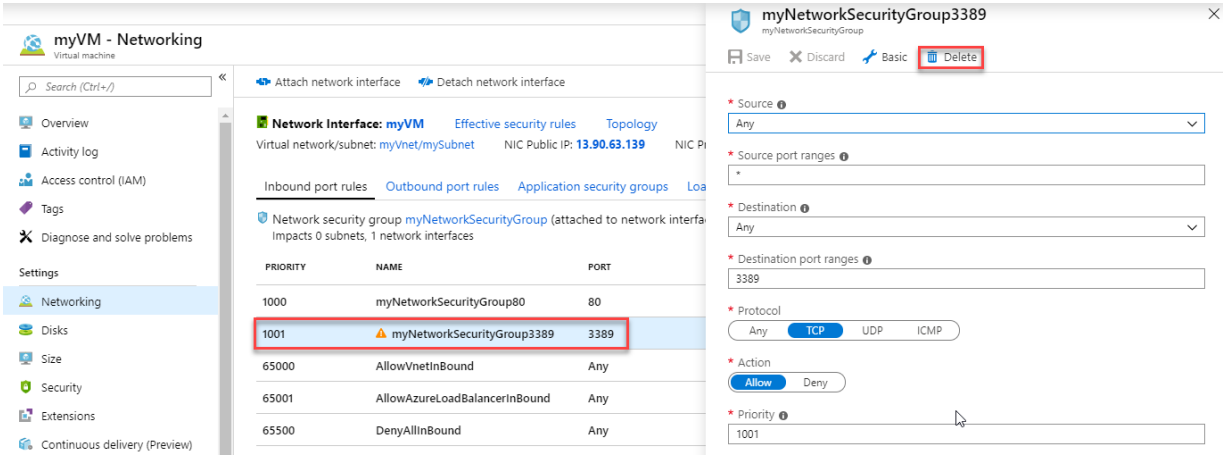
It's possible to reach Azure virtual machines by using Remote Desktop Protocol (RDP) and the Secure Shell (SSH) protocol. These protocols enable the management VMs from remote locations and are standard in datacenter computing.

The potential security problem with using these protocols over the Internet is that attackers can use brute force techniques to gain access to Azure virtual machines. After the attackers gain access, they can use your VM as a launch point for compromising other machines on your virtual network or even attack networked devices outside Azure.

It's recommended that you disable direct RDP and SSH access to your Azure VMs from the Internet. After direct RDP and SSH access from the Internet is disabled, you have other options that you can use to access these VMs for remote management:

- Point-to-site VPN
- Site-to-site VPN
- Azure ExpressRoute
- Azure Bastion Host

1. In the Azure portal click **Virtual machines**.
2. Select **myVM**.
3. open the **Networking** blade.
4. Select the rule which allows RDP (Port 3389) then click **Delete**.

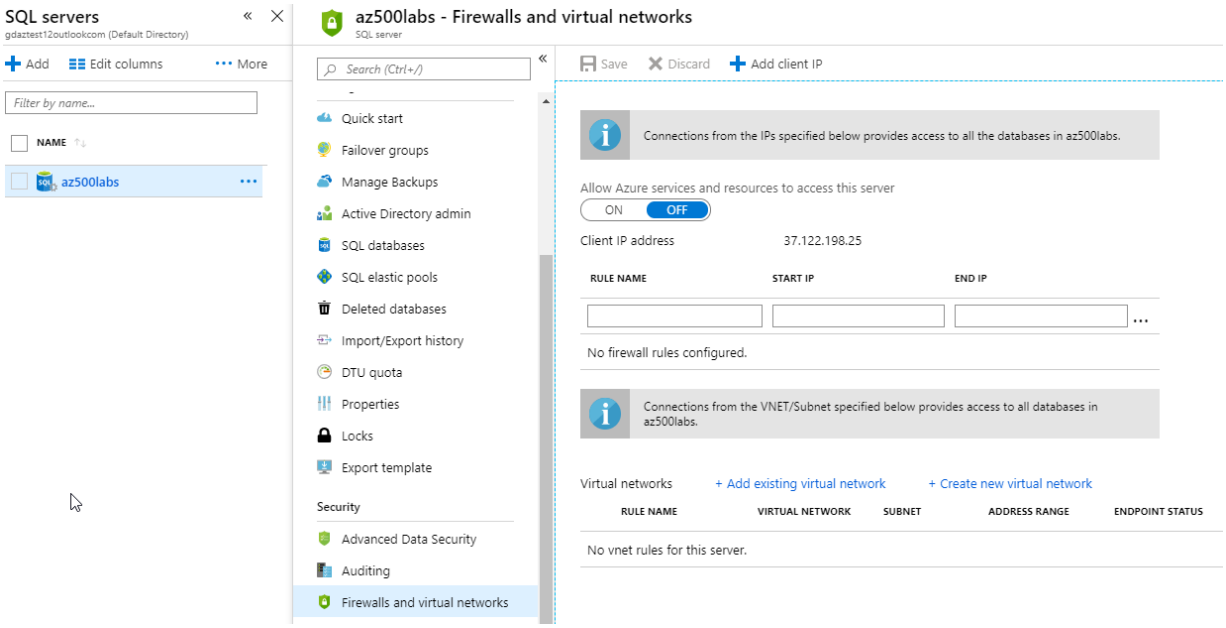


52.6.2 Task 2: Restrict SQL Server access from the Internet

Firewall systems help prevent unauthorized access to computer resources. If a firewall is turned on but not correctly configured, attempts to connect to SQL Server might be blocked.

To access an instance of the SQL Server through a firewall, you must configure the firewall on the computer that is running SQL Server. Allowing ingress for the IP range `0.0.0.0/0` (Start IP of `0.0.0.0` and End IP of `0.0.0.0`) allows open access to any/all traffic potentially making the SQL Database vulnerable to attacks. Ensure that no SQL Databases allow ingress from the Internet.

1. In the Azure portal go to **SQL servers** and select your SQL Server.
2. Click on **Firewalls and virtual networks**.



3. Ensure that the firewall rules exist, and no rule has a Start IP of `0.0.0.0` and End IP of `0.0.0.0` or other combinations which allows access to wider public IP ranges.
4. Close the blade.

52.6.3 Task 3: Configure the NSG flow logs

When you create or update a virtual network in your subscription, Network Watcher will be enabled automatically in your Virtual Network's region. There is no impact to your resources or associated charge for automatically enabling Network Watcher.

Network security group (NSG) flow logs are a feature of Network Watcher that allows you to view information about ingress and egress IP traffic through an NSG. Flow logs are written in JSON format, and show outbound and inbound flows on a per rule basis, the network interface (NIC) the flow applies to, 5-tuple information about the flow (Source/destination IP, source/destination port, and protocol), if the traffic was allowed or denied, and

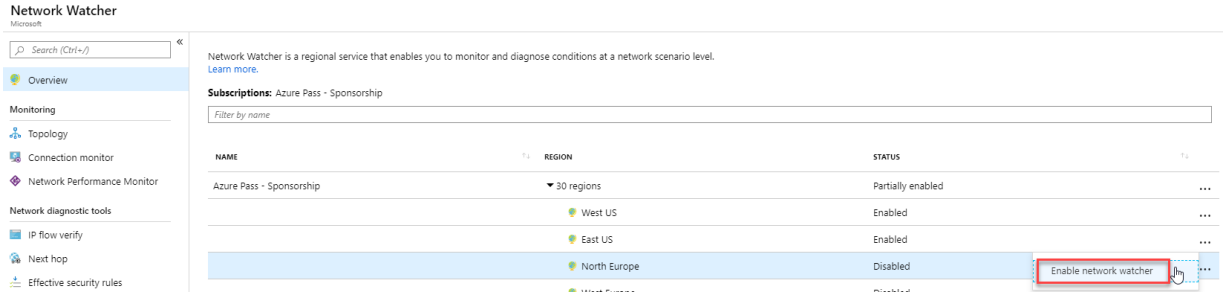
in Version 2, throughput information (Bytes and Packets). Logs can be used to check for anomalies and give insight into suspected breaches.

1. In the Azure portal select **All services**.
2. Select **Networking**.
3. Select **Network Watcher**.
4. Select **NSG flow logs** under Logs.
5. Select **On**.
6. Select a storage account and click **Save**.

52.6.4 Task 4: Enable Network Watcher

Network security group (NSG) flow logs are a feature of Network Watcher that allows you to view information about ingress and egress IP traffic through an NSG.

1. In the Azure portal select All services. In the Filter box, enter **Network Watcher**. When Network Watcher appears in the results, select it.
2. Select Regions, to expand it, and then select the elipsis (...) button on a region which is not enabled.
3. Select **Enable Network Watcher**.



52.7 Exercise 7: Create an Azure VM baseline

Azure Policy is a service in Azure that you use to create, assign, and manage policies. These policies enforce different rules and effects over your resources, so those resources stay compliant with your corporate standards and service level agreements. Azure Policy meets this need by evaluating your resources for non-compliance with assigned policies. For example, you can have a policy to allow only a certain SKU size of virtual machines in your environment. Once this policy is implemented, new and existing resources are evaluated for compliance. With the right type of policy, existing resources can be brought into compliance.

Azure networking security recommendations

Here are the security recommendations you should follow to set Virtual Machine (VM) policies on your Azure subscription. Included with each recommendation are the basic steps to follow in the Azure portal. You should perform these steps on your own subscription with your own resources to validate the security for each. Keep in mind that Level 2 options might restrict some features or activity, so carefully consider which security options you decide to enforce.

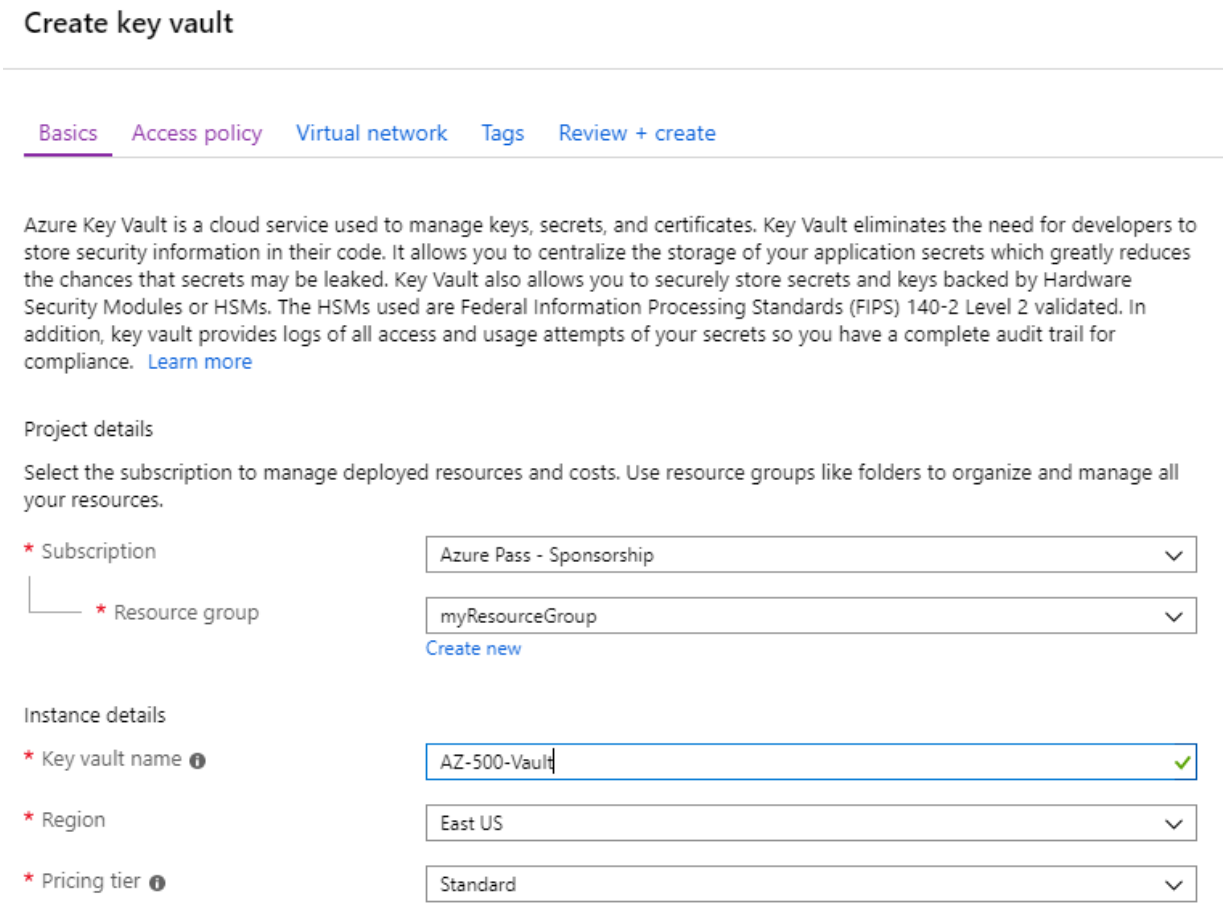
52.7.1 Task 1: Ensure that OS disk are encrypted

Azure Disk Encryption helps protect and safeguard your data to meet your organizational security and compliance commitments. It uses the BitLocker feature of Windows and the DM-Crypt feature of Linux to provide volume encryption for the OS and data disks of Azure virtual machines (VMs). It is also integrated with Azure Key Vault to help you control and manage the disk encryption keys and secrets, and ensures that all data on the VM disks are encrypted at rest while in Azure storage. Azure Disk Encryption for Windows and Linux VMs is in General Availability in all Azure public regions and Azure Government regions for Standard VMs and VMs with Azure Premium Storage.

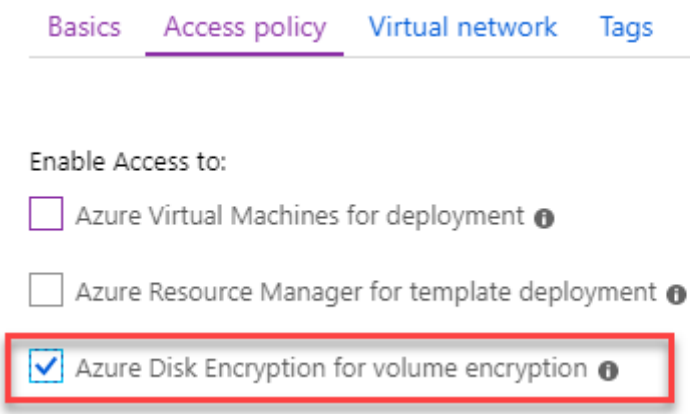
If you use Azure Security Center (recommended), you're alerted if you have VMs that aren't encrypted.

1. In the Azure portal search for **KeyVault**.
2. Select **Key Vault** and click **Create**.

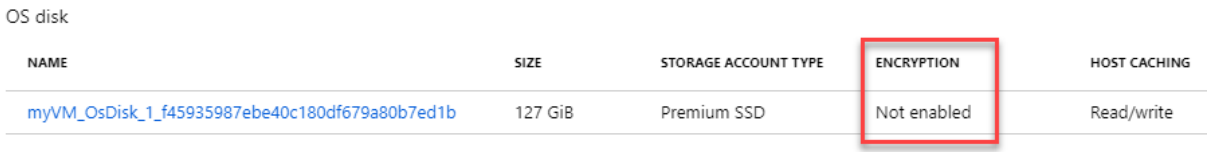
3. Enter the following details:
   - **Resource Group**: myResourceGroup
   - **Key vault name**: *Enter something unique*
   - **Region**: EastUS



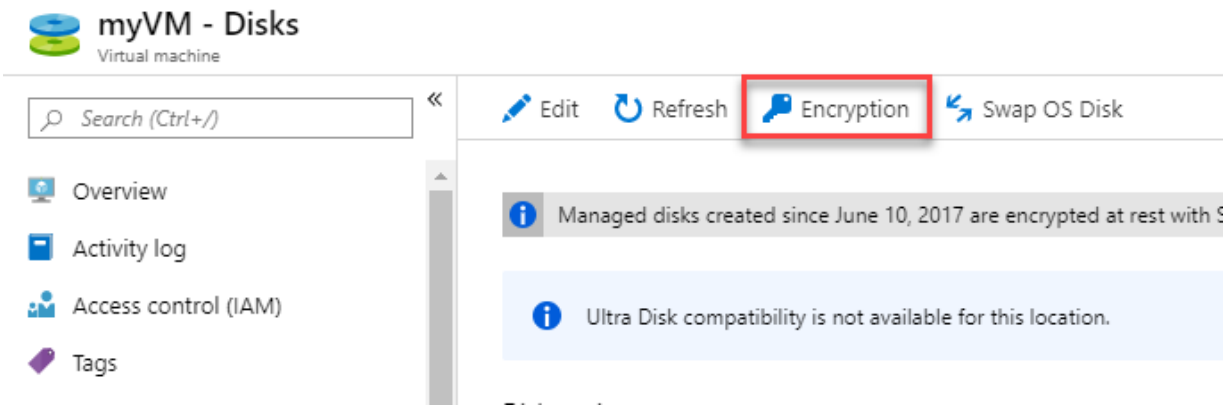
4. Select the **Access policy** tab and select **Azure Disk Encryption** for volume encryption.



5. Click **Review + create** then click **Create**.
6. Wait for the deployment to complete before continuing.
7. In the Azure portal select **Virtual machines**.
8. Select **myVM** virtual machine.
9. Under the **Settings** section select **Disks**.
10. Notice the disk is not encrypted.



11. Click **Encryption**.

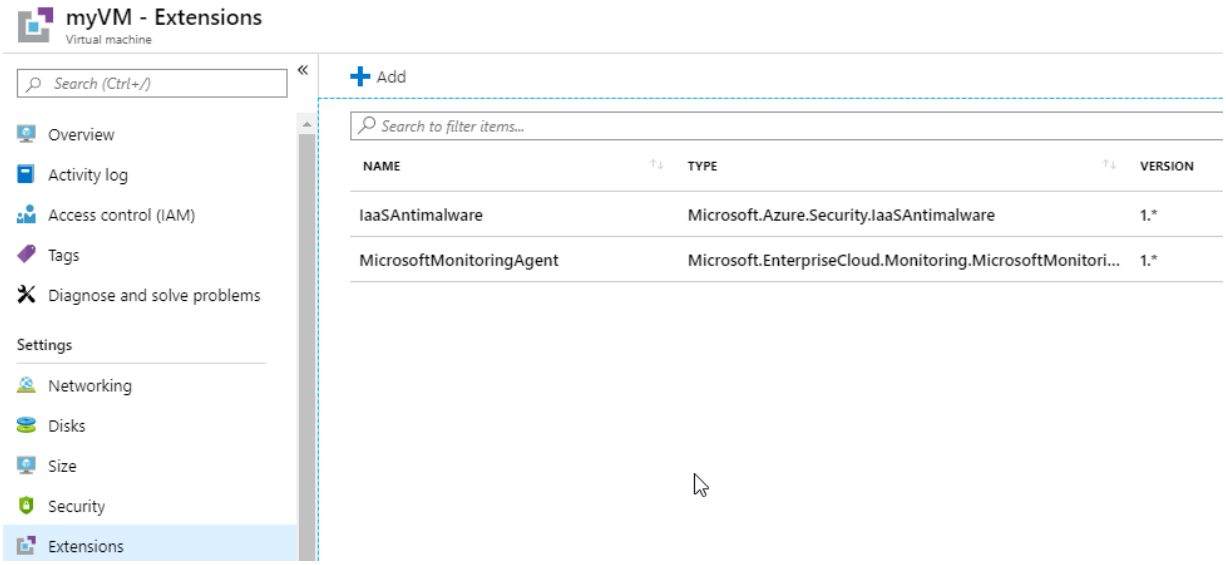


12. Select **OS & data disks** to be encrypted.
13. Click **Select a key vault and key for encryption** and select your vault and click **Select**.
14. Click **Save** and click **Yes** to confirm.

52.7.2 Task 2: Ensure only approved extensions are installed

Azure virtual machine (VM) extensions are small applications that provide post-deployment configuration and automation tasks on Azure VMs. For example, if a virtual machine requires software installation, anti-virus protection, or to run a script inside of it, a VM extension can be used. Azure VM extensions can be run with the Azure CLI, PowerShell, Azure Resource Manager templates, and the Azure portal. Extensions can be bundled with a new VM deployment, or run against any existing system.

1. In the Azure portal select **Virtual machines**.
2. Select **myVM** and then in the **Settings** section click **Extensions**.
3. Ensure that the listed extensions are approved for use.



Results: You have now completed this lab.

53 Module 4: Lab 9 - JIT

Scenario

Just-in-time (JIT) virtual machine (VM) access can be used to lock down inbound traffic to your Azure VMs, reducing exposure to attacks while providing easy access to connect to VMs when needed.

Brute force attacks commonly target management ports as a means to gain access to a VM. If successful, an attacker can take control over the VM and establish a foothold into your environment.

One way to reduce exposure to a brute force attack is to limit the amount of time that a port is open. Management ports don't need to be open at all times. They only need to be open while you're connected to the VM, for example to perform management or maintenance tasks. When just-in-time is enabled, Security Center uses network security group (NSG) and Azure Firewall rules, which restrict access to management ports so they cannot be targeted by attackers.

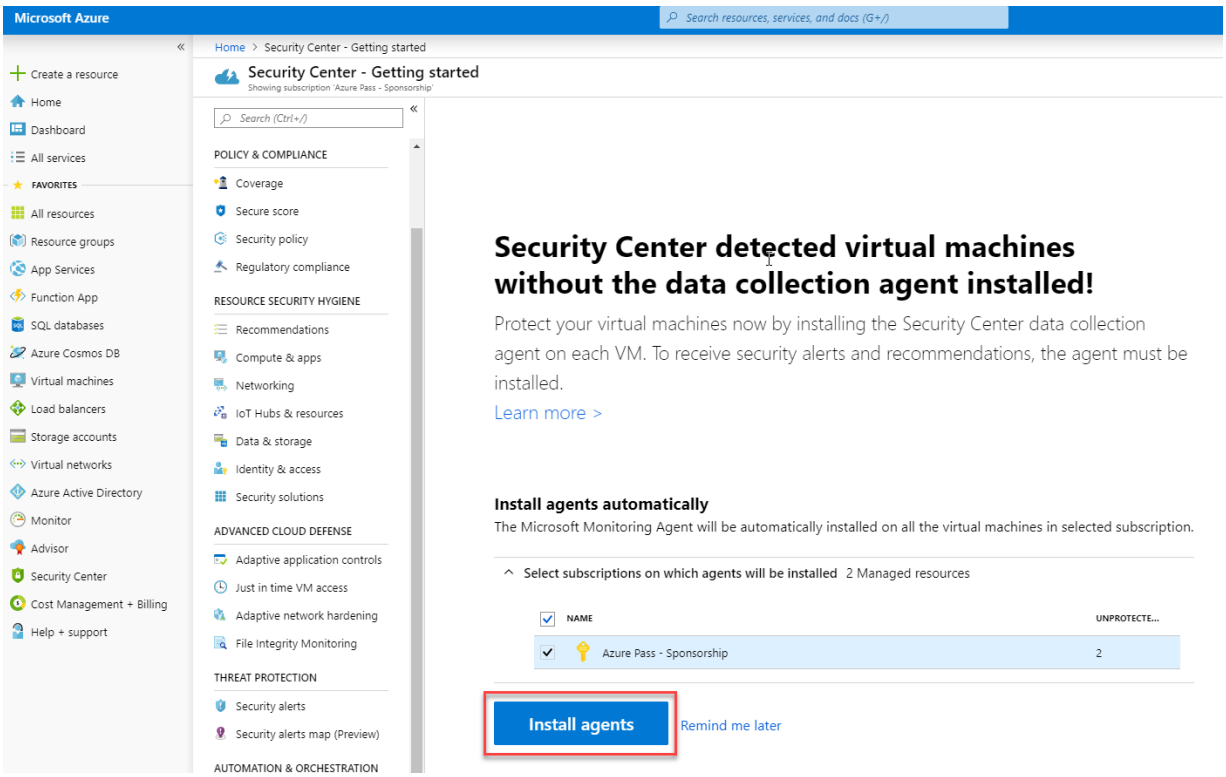
53.1 Exercise 1: Manage virtual machine access using just-in-time

There are three ways to configure a JIT policy on a VM:

- Configure JIT access in Azure Security Center
- Configure JIT access in an Azure VM blade
- Configure a JIT policy on a VM programmatically

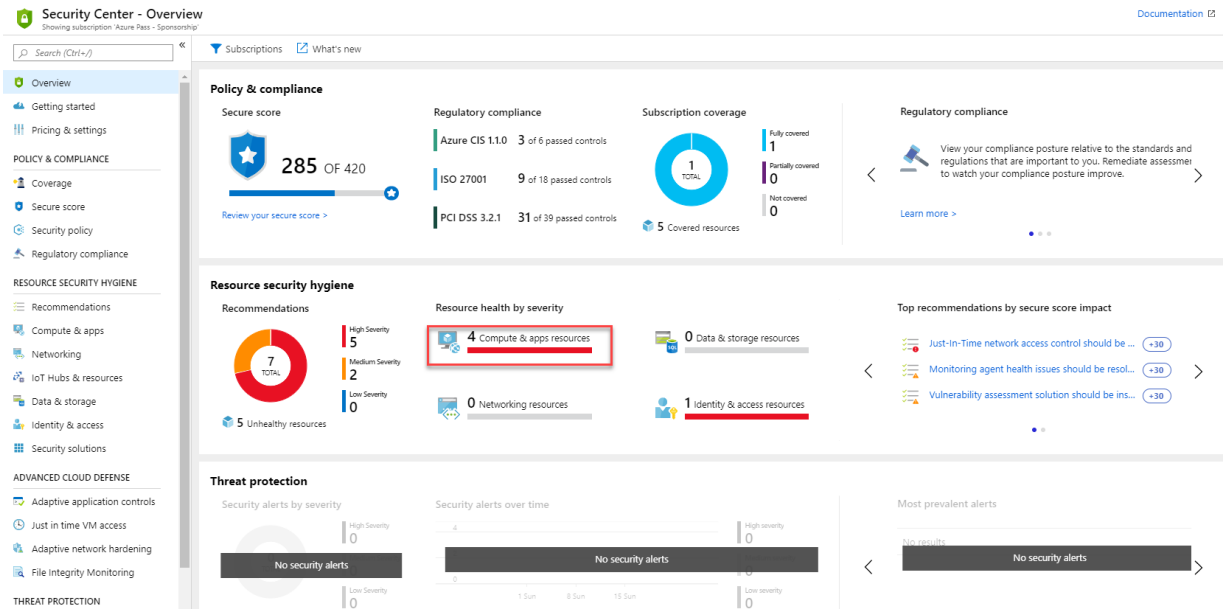
53.1.1 Task 1: Configure JIT access on a VM in Azure Security Center

1. In the Azure Portal open the **Security Center** and then click **Getting Started**.
2. Click **Install agents**.

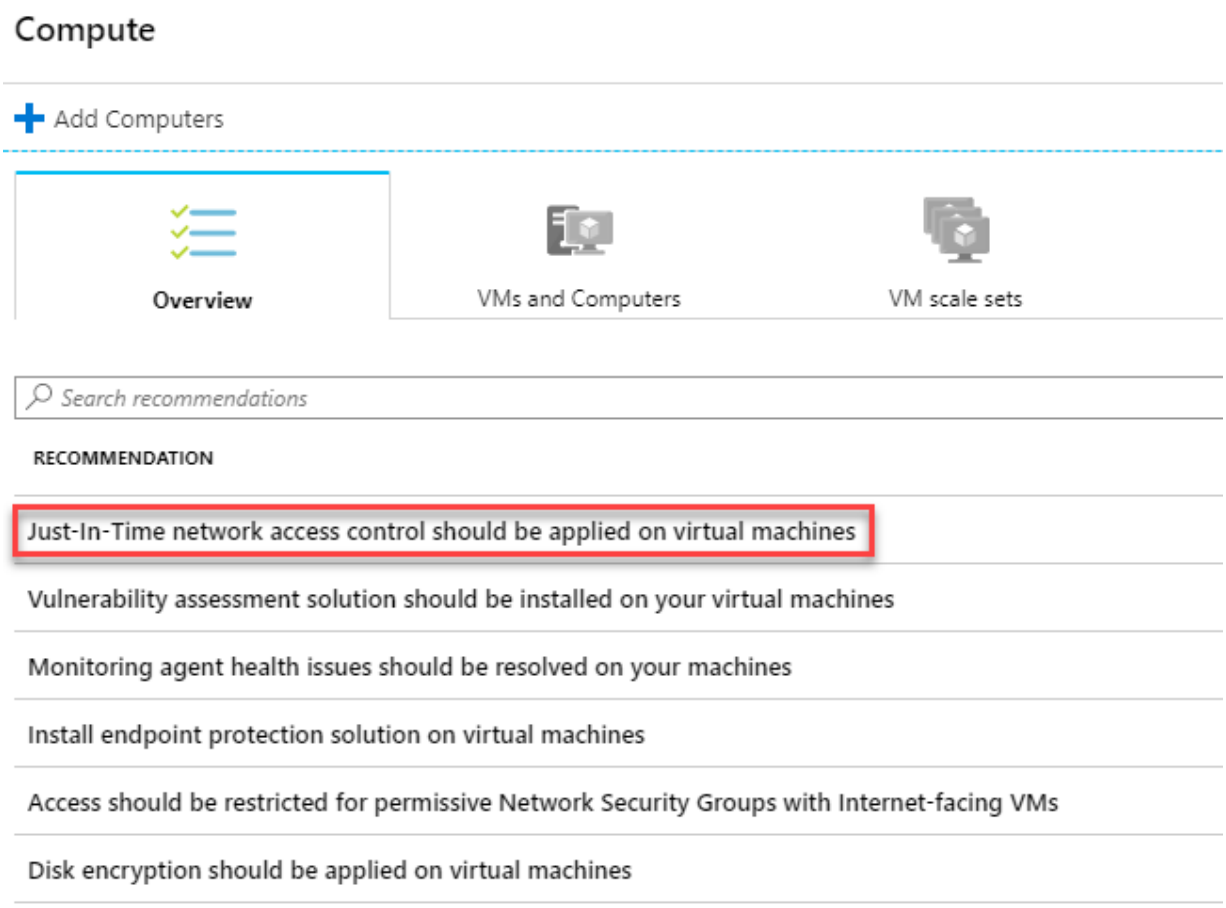


Note: You may have to wait upto 5 minutes for the agents to deploy.

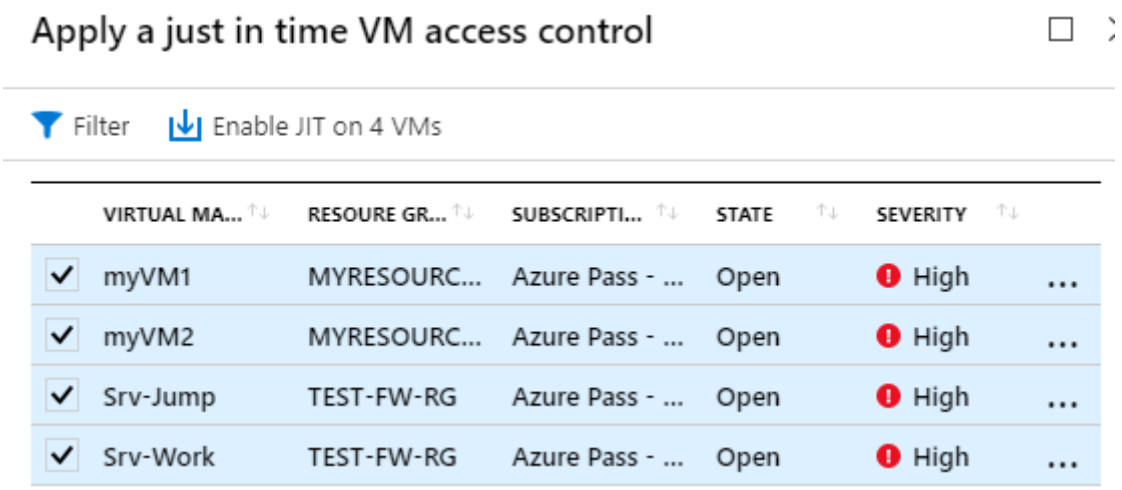
2. In the left pane, select **Overview**.
3. Select **Compute & app resources**.



4. On the Compute blade, note the recommendations.
5. Select the **Just-In-Time network access control should be applied on virtual machines**.

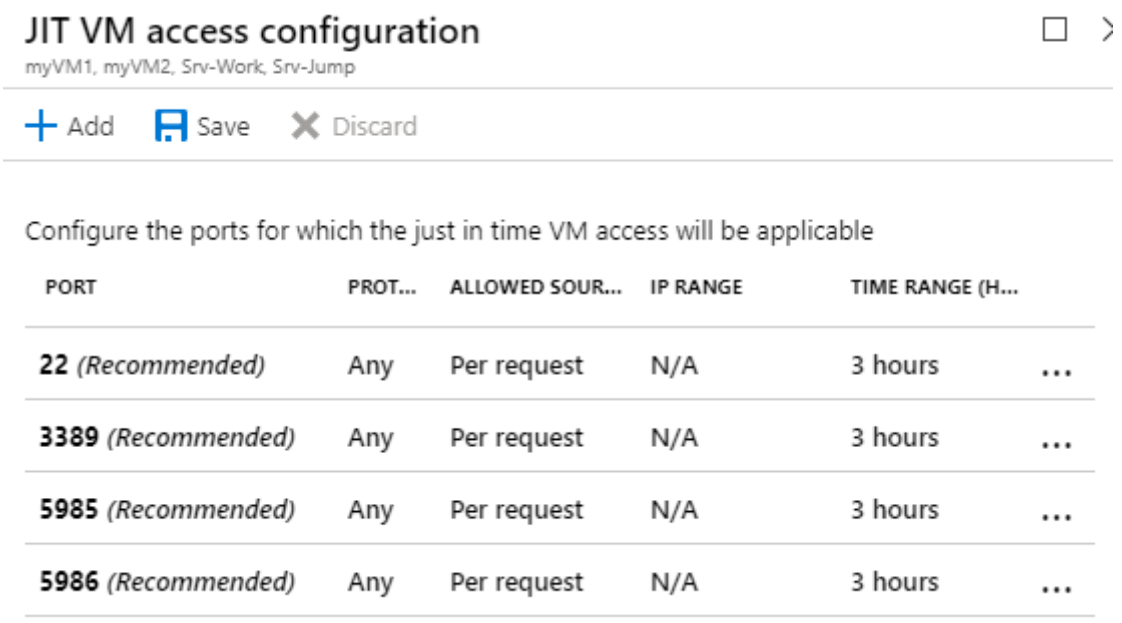


6. Select all 4 virtual machines and click **Enable JIT on 4 VMs**.

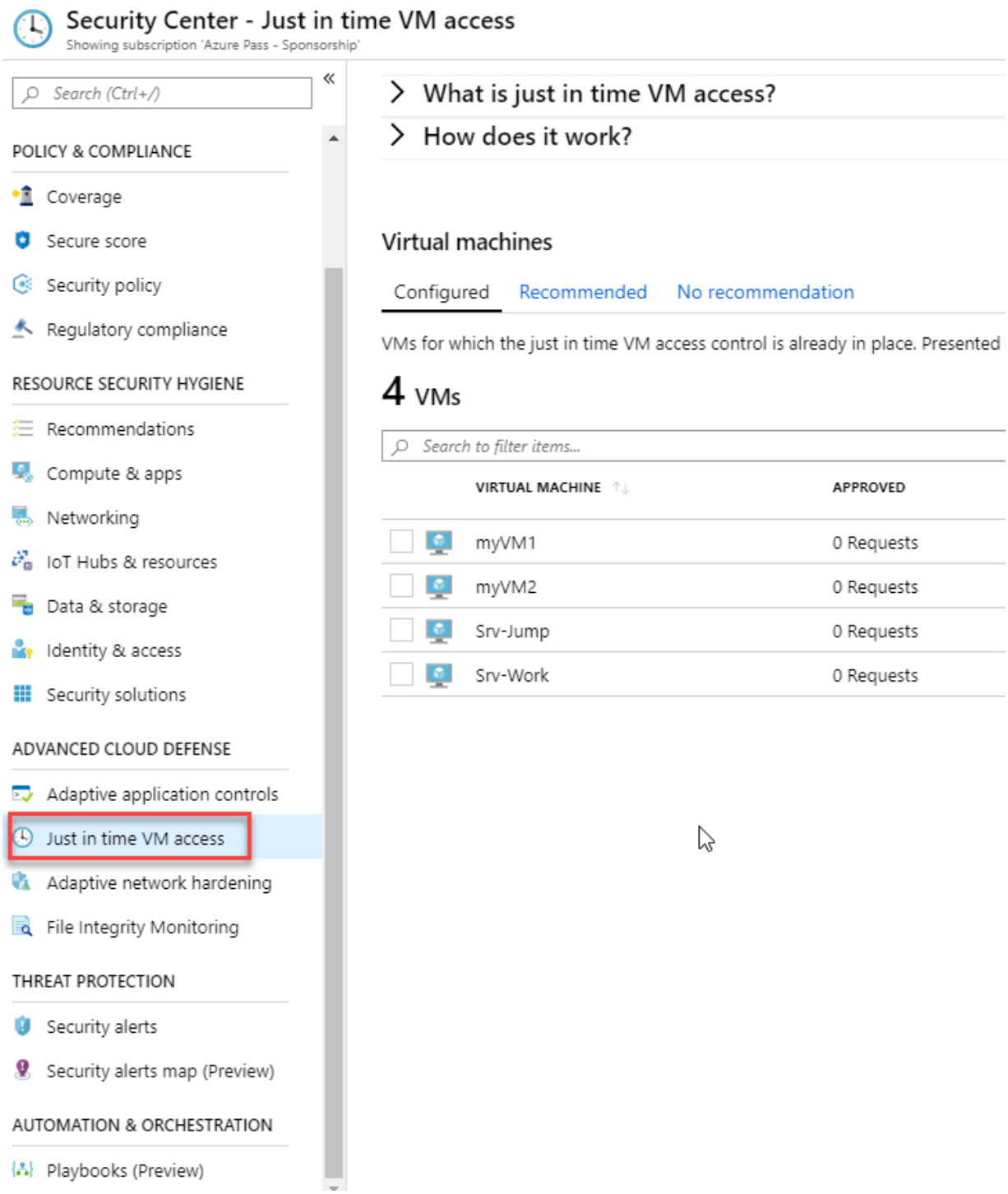


7. On the **JIT VM access configuration** blade click **Save**.

- This blade displays the default ports recommended by Azure Security Center:
  - 22 - SSH
  - 3389 - RDP
  - 5985 - WinRM
  - 5986 - WinRM



1. Close all the blade and on the Security Center blade click **Just in time VM access**.



The **Just-in-time VM access** window opens.

Just-in-time VM access provides information on the state of your VMs:

- **Configured** - VMs that have been configured to support just-in-time VM access. The data presented is for the last week and includes for each VM the number of approved requests, last access date and time, and last user.
- **Recommended** - VMs that can support just-in-time VM access but haven't been configured to. We recommend that you enable just-in-time VM access control for these VMs.
- **No recommendation** - Reasons that can cause a VM not to be recommended are:
  - Missing NSG - The just-in-time solution requires an NSG to be in place.
  - Classic VM - Security Center just-in-time VM access currently supports only VMs deployed through Azure Resource Manager. A classic deployment is not supported by the just-in-time solution.
  - Other - A VM is in this category if the just-in-time solution is turned off in the security policy of the subscription or the resource group, or if the VM is missing a public IP and doesn't have

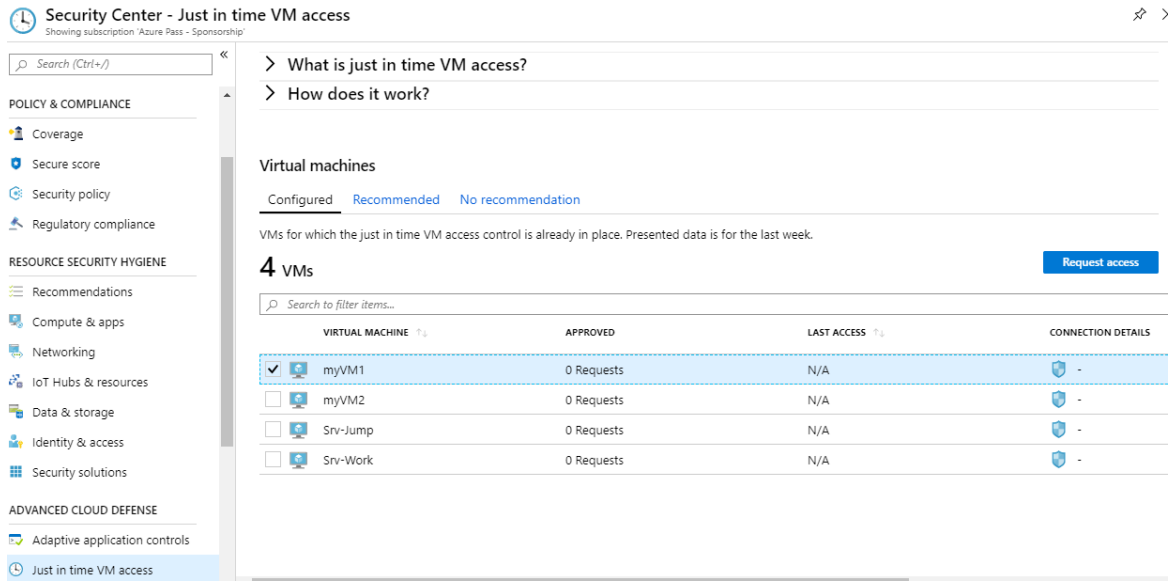
an NSG in place.

Note: When JIT VM Access is enabled for a VM, Azure Security Center creates ”deny all inbound traffic” rules for the selected ports in the network security groups associated and Azure Firewall with it. If other rules had been created for the selected ports, then the existing rules take priority over the new ”deny all inbound traffic” rules. If there are no existing rules on the selected ports, then the new ”deny all inbound traffic” rules take top priority in the Network Security Groups and Azure Firewall.

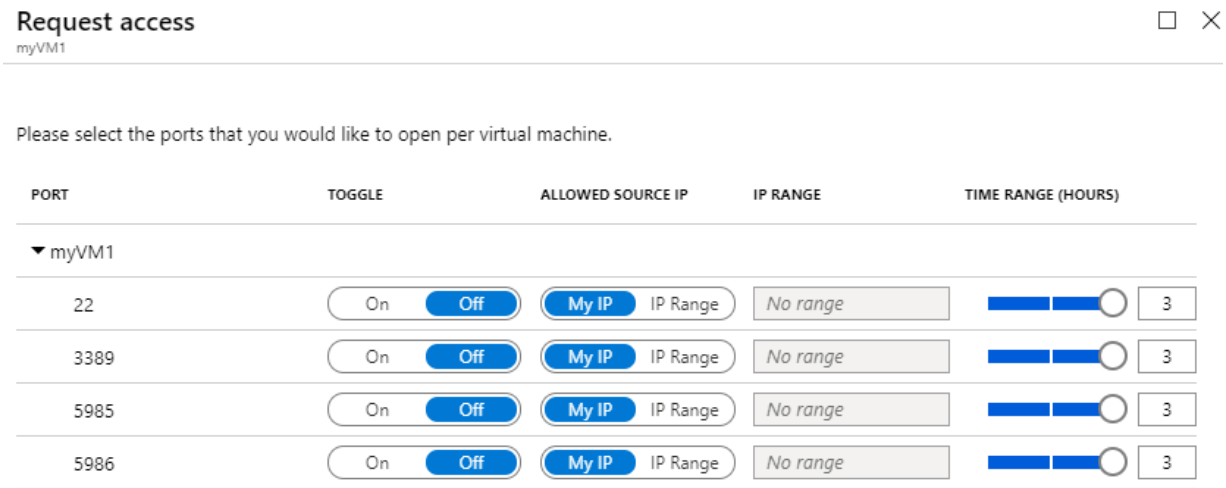
53.1.2 Task 2: Request JIT access via ASC

To request access to a VM via ASC:

1. Under **Just in time VM access**, select the **Configured** tab.
2. Under **Virtual Machine**, select one of the VMs that you want to request access for. This puts a checkmark next to the VM.
   - The icon in the **Connection Details** column indicates whether JIT is enabled on the NSG or FW. If it's enabled on both, only the Firewall icon appears.
   - The **Connection Details** column provides the information required to connect the VM, and its open ports.



3. Click **Request access**. The **Request access** window opens.



4. Under **Request access**, for each VM, configure the ports that you want to open and the source IP addresses that the port is opened on and the time window for which the port will be open. It will only be possible to request access to the ports that are configured in the just-in-time policy. Each port has a maximum allowed time derived from the just-in-time policy.

5. Click **Open ports**.

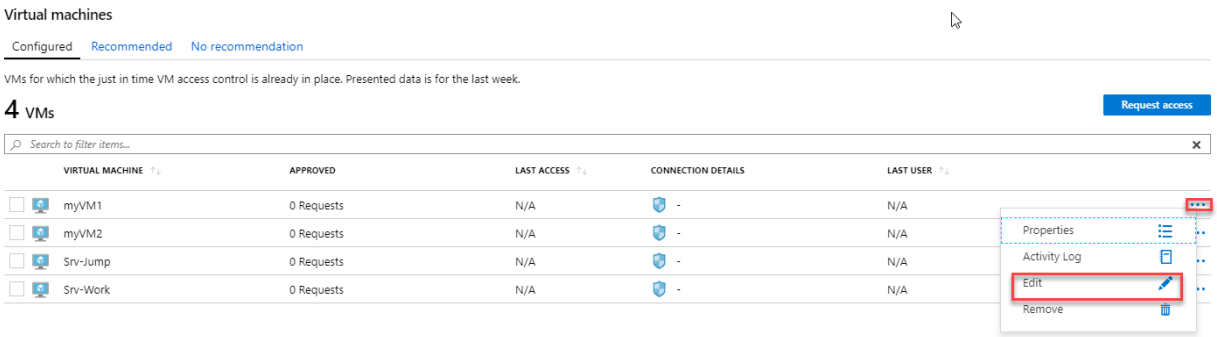
Note: If a user who is requesting access is behind a proxy, the option **My IP** may not work. You may need to define the full IP address range of the organization.

53.1.3 Task 3: Edit a JIT access policy via ASC

You can change a VM's existing just-in-time policy by adding and configuring a new port to protect for that VM, or by changing any other setting related to an already protected port.

To edit an existing just-in-time policy of a VM:

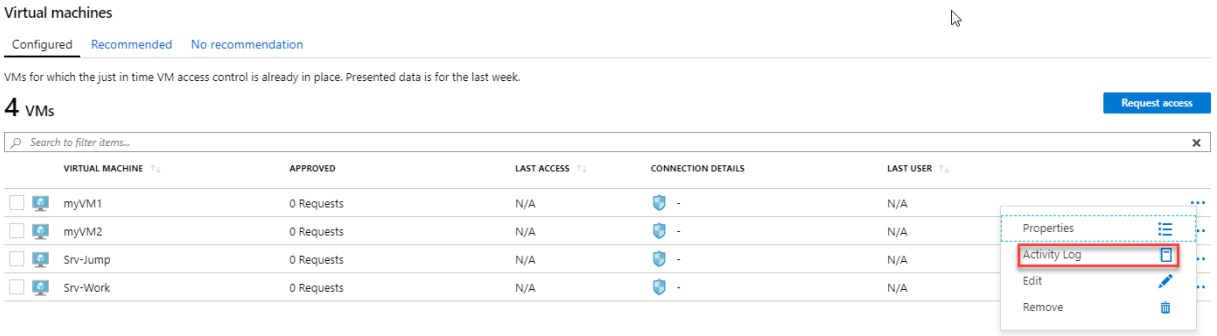
1. In the **Configured** tab, under **VMs**, select a VM to which to add a port by clicking on the three dots within the row for that VM.
2. Select **Edit**.
3. Under **JIT VM access configuration**, you can either edit the existing settings of an already protected port or add a new custom port.



53.1.4 Task 4: Audit JIT access activity in ASC

You can gain insights into VM activities using log search. To view logs:

1. Under **Just-in-time VM access**, select the **Configured** tab.
2. Under **VMs**, select a VM to view information about by clicking on the three dots within the row for that VM and select **Activity Log** in the menu. The **Activity log** opens.



Activity log provides a filtered view of previous operations for that VM along with time, date, and subscription.

You can download the log information by selecting **Click here to download all the items as CSV**.

Modify the filters and click **Apply** to create a search and log.

53.1.5 Task 5: Configure JIT access on a VM via the Azure VM blade

To make it easy to roll out just-in-time access across your VMs, you can set a VM to allow only just-in-time access directly from within the VM.

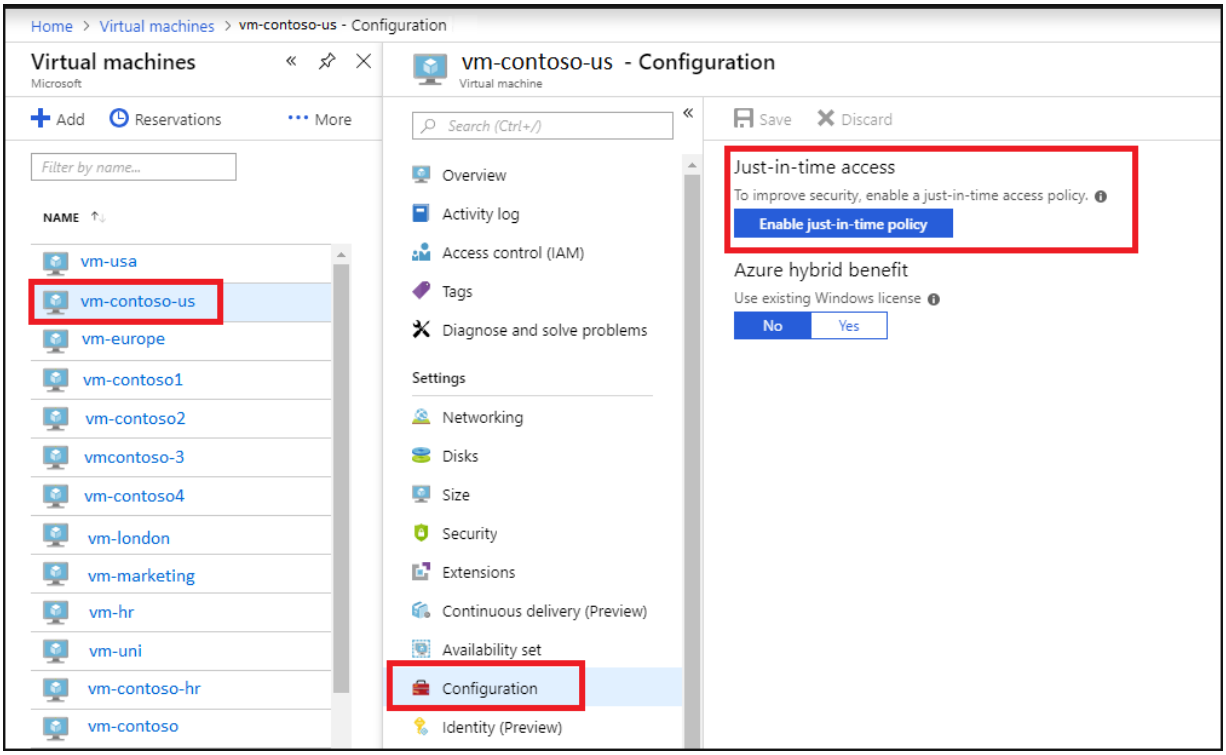
1. In the Azure portal, select **Virtual machines**.
2. Click on the virtual machine you want to limit to just-in-time access.
3. In the menu, click **Configuration**.

4. Under **Just-in-time-access** click **Enable just-in-time policy**.

   This enables just-in-time access for the VM using the following settings:

   - Windows servers:
     - RDP port 3389
     - Three hours of maximum allowed access
     - Allowed source IP addresses is set to Any
   - Linux servers:
     - SSH port 22
     - Three hours of maximum allowed access
     - Allowed source IP addresses is set to Any

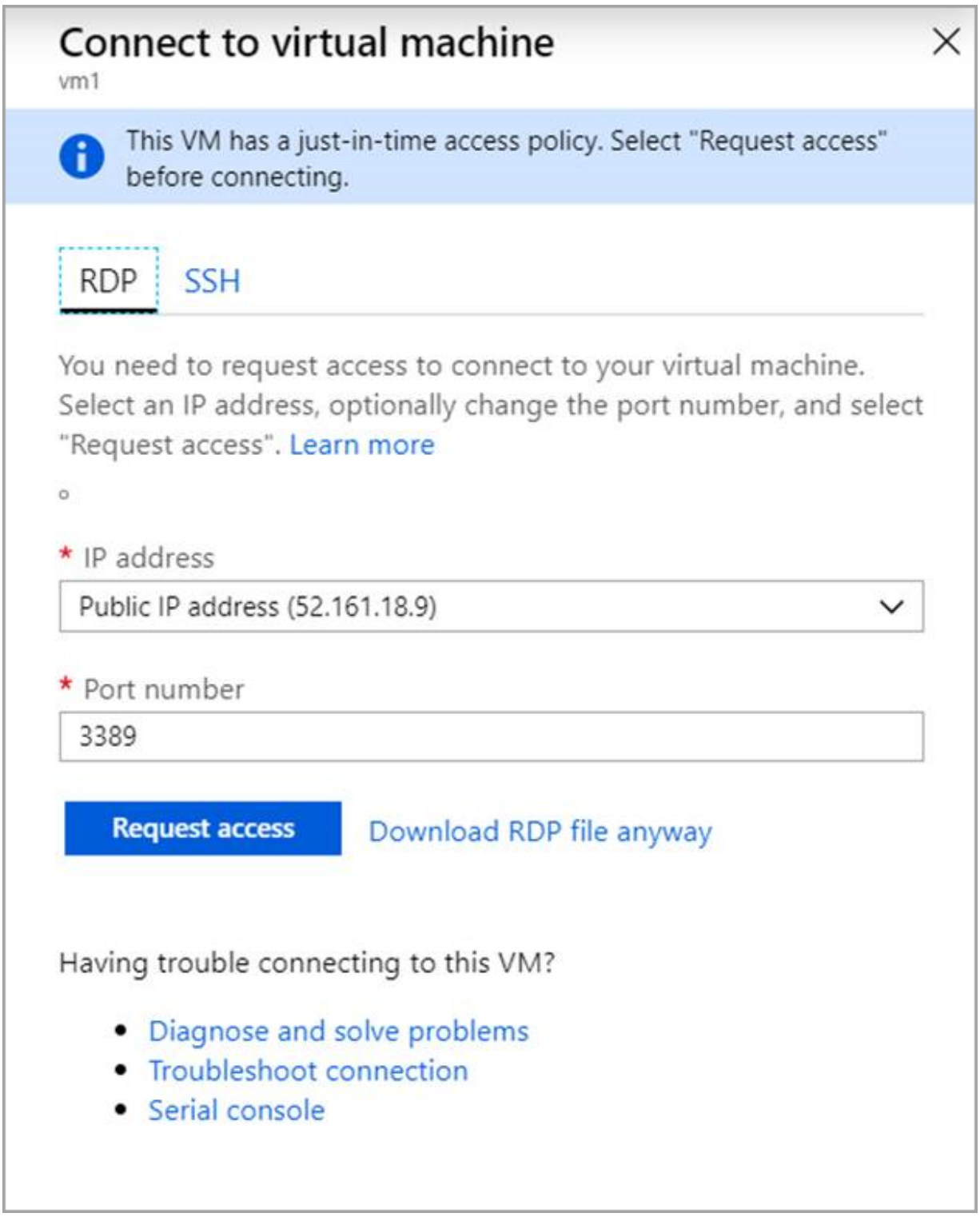
   If a VM already has just-in-time enabled, when you go to its configuration page you will be able to see that just-in-time is enabled and you can use the link to open the policy in Azure Security Center to view and change the settings.



53.1.6 Task 5: Request JIT access to a VM via the Azure VM blade

In the Azure portal, when you try to connect to a VM, Azure checks to see if you have a just-in-time access policy configured on that VM.

- If you do have a JIT policy configured on the VM, you can click **Request access** to enable you to have access in accordance with the JIT policy set for the VM.

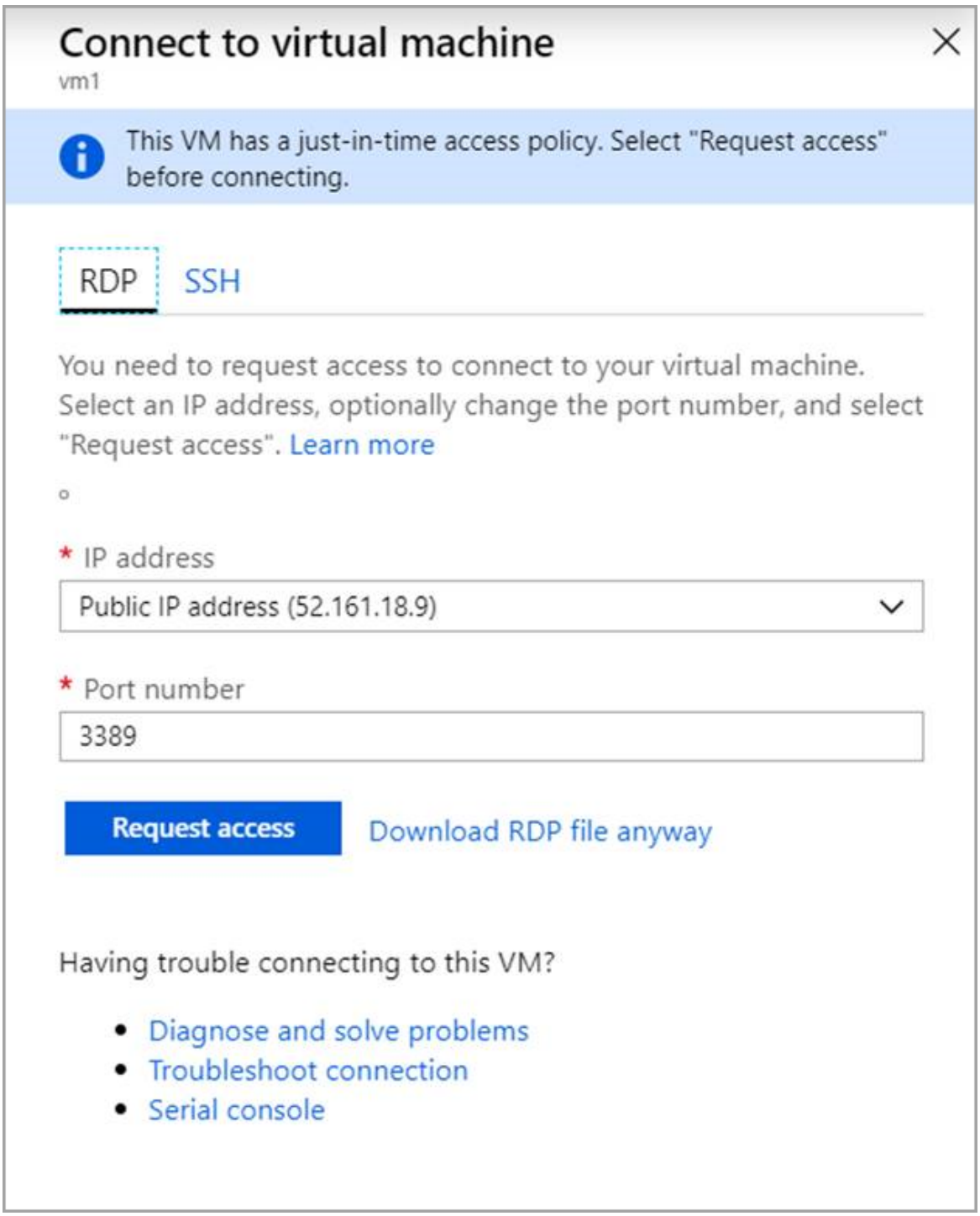


The access is requested with the following default parameters:

- **source IP**: 'Any' (*) (cannot be changed)
- **time range**: Three hours (cannot be changed)
- **port number** RDP port 3389 for Windows / port 22 for Linux (can be changed)

Note: After a request is approved for a VM protected by Azure Firewall, Security Center provides the user with the proper connection details (the port mapping from the DNAT table) to use to connect to the VM.

- If you do not have JIT configured on a VM, you will be prompted to configure a JIT policy it.



WARNING: Prior to continuing you should remove all resources used for this lab. To do this in the **Azure Portal** click **R**

Results: You have now completed this lab.