Microsoft Cloud Workshop

Securing PaaS

Hands-on lab step-by-step

April 2018

Contents

Securing PaaS hands-on lab step-by-step

Abstract and learning objectives

This workshop is designed to provide exposure to many of Microsoft Azure's Platform-as-a-Service (PaaS) security features. The goal is to show a secure end-to-end solution that addresses concerns around sensitive data, controlling access to sensitive stores of information, controlling access to production systems and enabling secure processes for developers. The architecture includes:

- App Service Environments
- Application Gateway
- Web Application Firewall
- Azure Web Apps
- Azure Functions
- Azure API Apps
- Azure SQL DB and corresponding security features
- Azure Storage
- Cosmos DB
- Azure Search
- Azure Monitor
- Log Analytics
- App Insights
- Azure Security Center
- Azure Key Vault Integrations
- Azure Web Application Gateway
- Azure Active Directory

Attendees will learn how to:

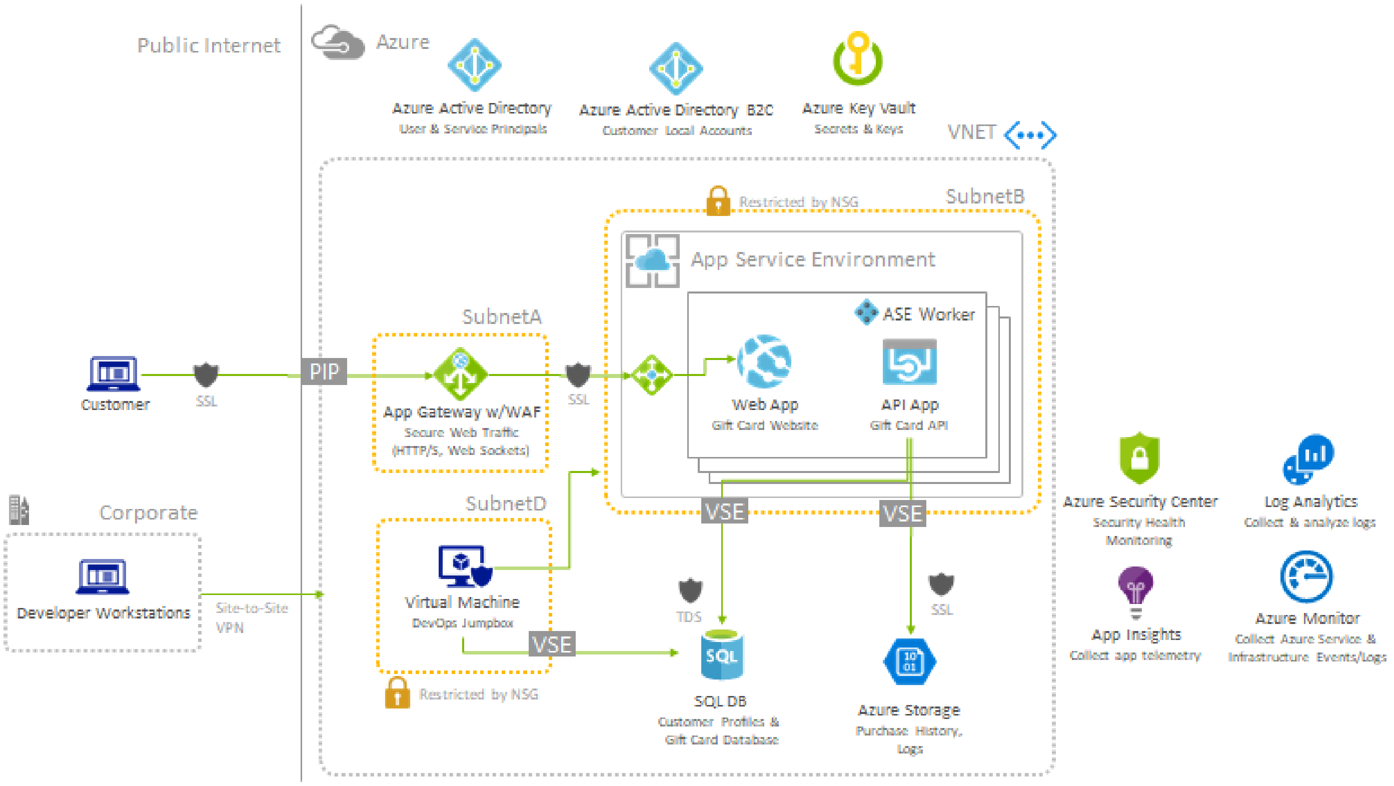
- Build secure solutions end to end with Azure PaaS services
- Control access to PaaS service
- Manage secrets and keys used by PaaS services

Overview

In this hands-on lab, attendees will implement several of the PaaS security features of Azure to help ensure a secure application environment.

Solution architecture

Below is a diagram of the solution architecture you will build in this lab. Please study this carefully, so you understand the whole of the solution as you are working on the various components.



The solution begins with a deployed template of typical and not so typical resources. Due to time restraints during deployment you will have an internal (versus an external facing) **App Service Environment** (ASE). The ASE will have no app service plans or apps deployed to it. It is also not accessible from the outside world. You will configure an application to be deployed using the **Azure DevOps** machine and Visual Studio to deploy to the ASE after creating an app service plan. Once deployed, you will then configure the **Application Gateway** to point to the new ASE hosted App. Once configured, you will perform a typical web-based attack on the environment in a detection-only mode to see the requests pass to your web application. Once you understand how this process works, you will then enable the **Web Application Firewall** to filter requests based on the **OWASP 3.0** standard and see that those requests are in fact blocked.

Separately, you will explore how Azure Identity Access and Management (Azure IAM) works and how those access permissions are separate from policies that may live within the actual Azure resource (such as with **Azure Key Vault**). You will learn how to remove sensitive information from your various resources such as **Azure Functions** and **Web Applications** and place them in the **Azure Key Vault** for both deployment and runtime use.

As a final step, you will learn how to perform queries against **Log Analytics** to populate a **Power BI** report based on your **Web Application Firewall** events.

Requirements

1. Microsoft Azure subscription must be pay-as-you-go or MSDN
    - Trial subscriptions will not work
2. A machine with the following software:
    - Visual Studio 2017 Community edition or greater
    - SQL Server Management Studio 2017
    - Power BI Desktop
    - Fiddler
3. **To ensure you can begin the course delivery on-time, you must take the following step at least 5-hours prior to the course start time:**
    - **Run the Azure resource template – The Application Service Environment can take more than 90-minutes to create.**

Before the hands-on lab

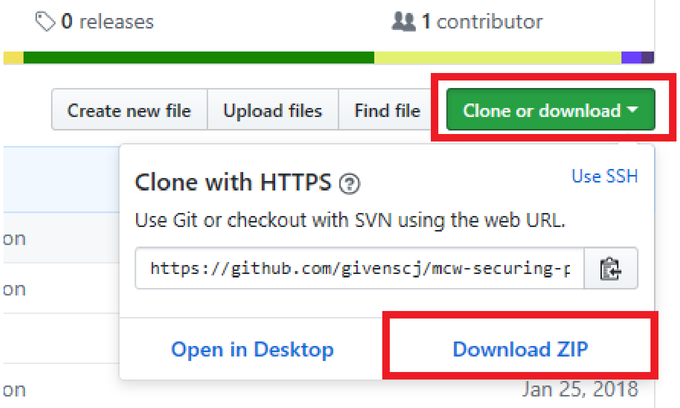
Duration: 30 minutes

Synopsis: In this exercise, you will set up your environment for use in the rest of the hands-on lab. You should follow all the steps provided in the Before the Hands-on Lab section to prepare your environment *before* attending the workshop.

Task 1: Download GitHub resources (Jump machine)

In this task, you will download the Azure Resource Manager (ARM) template required to setup this lab from a GitHub repository.

1. Open a browser window to the cloud workshop GitHub repository (https://github.com/givenscj/mcw-securing-paas).
2. Select **Clone or download**, then select **Download Zip**.

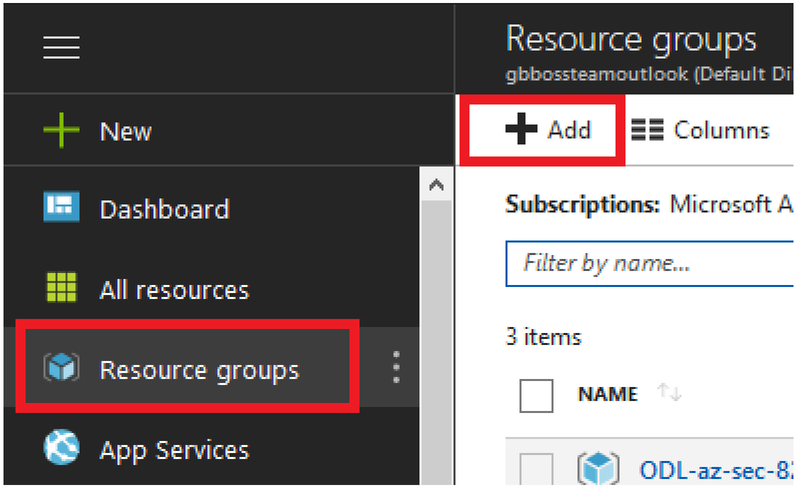


3. Extract the zip file to your local machine, be sure to keep note of where you have extracted the files.

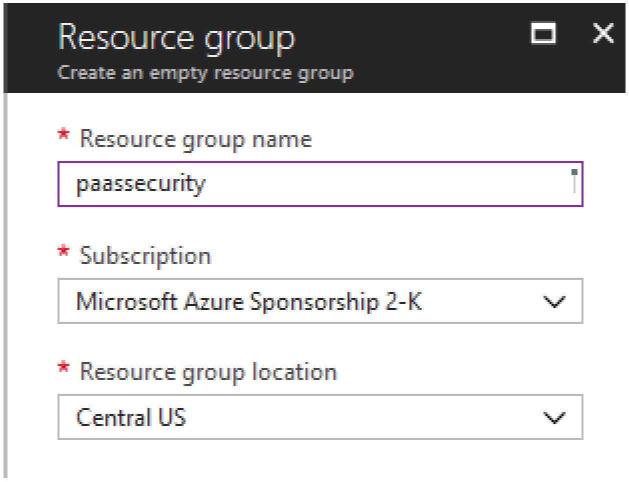
Task 2: Deploy resources (virtual machine, etc.) to Azure

In this task, you will run the ARM template downloaded in the previous task in the Azure portal to provision the resources you will be using throughout this hands-on lab.

1. In a browser, open the Azure Portal.
   - NOTE: If prompted, select **Maybe Later**.
2. Select **Resource groups** from the left-hand navigation menu, then select **+Add**.



3. Enter a **resource group** name, such as **paassecurity-[your initials or first name].**



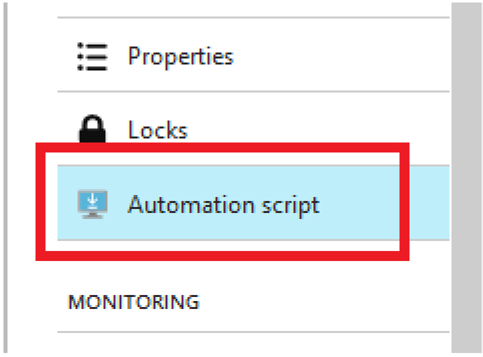
Resource group
Create an empty resource group

* Resource group name: paassecurity

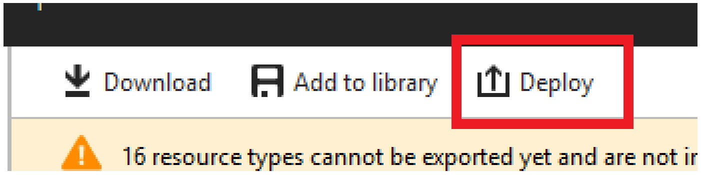
* Subscription: Microsoft Azure Sponsorship 2-K

* Resource group location: Central US

4. Select **Create**.
5. Select **Refresh** to see your new resource group displayed and select it.
6. Select **Automation Script**.

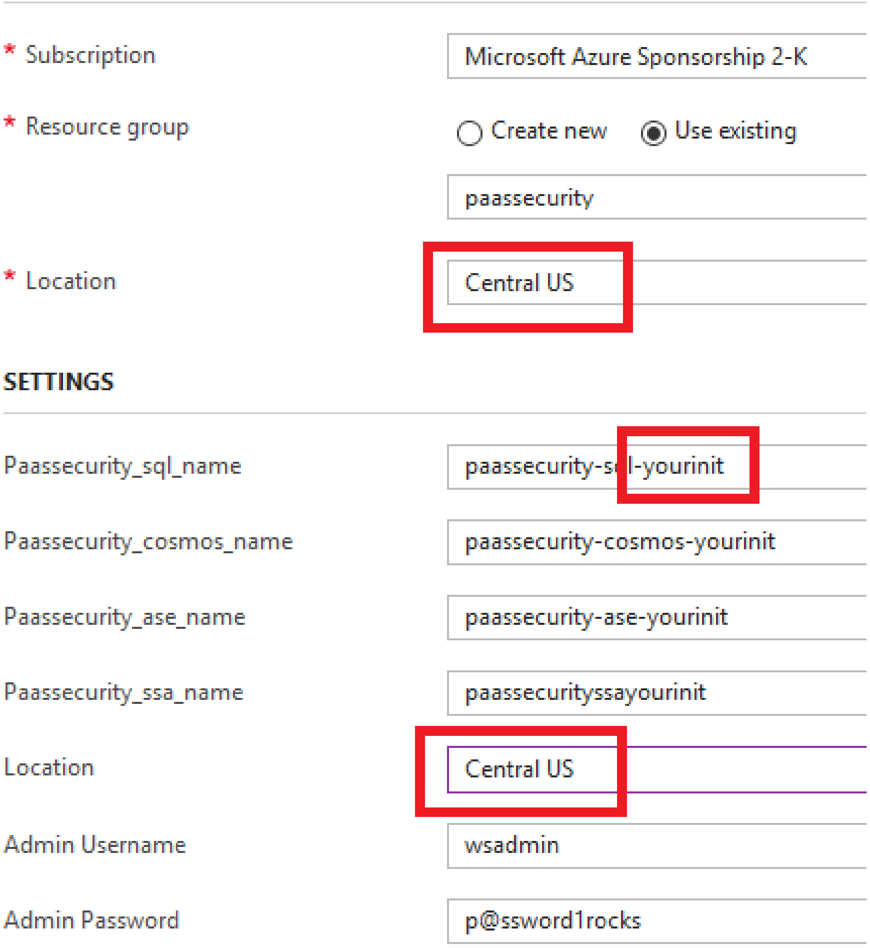


7. Select **Deploy**.



16 resource types cannot be exported yet and are not ir

8. Select **Build your own template in the editor**.
9. In the extracted folder, open the **\AzureTemplate\azure-deploy.json**.
10. Copy and paste it into the window.
11. Select **Save**, you will see the dialog with the input parameters. Fill out the form:
    a. **Subscription**: Select your subscription.
    b. **Resource group**: Use an existing Resource group or create a new one by entering a unique name, such as **paassecurity-[your initials or first name]**.
    c. **Location**: Select a location for the Resource group. Recommend using East US, East US 2, West Central US, or West US 2.
    d. Modify the **parameters** to be something unique by replacing with your initials or something similar.
    e. Fill in the remaining parameters, but if you change anything be sure to note it for future reference throughout the lab.
    f. **Be sure your resource group location matches the location you select in the settings window**
        i. NOTE: This field and matching is due to a limitation of the resource templates not resolving the resource group location for some template types.

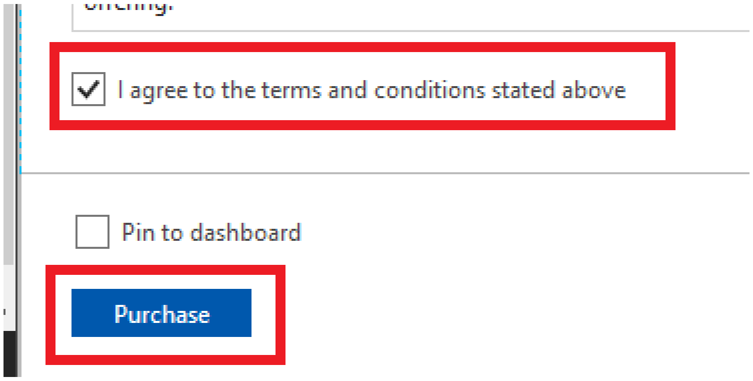


* Subscription	Microsoft Azure Sponsorship 2-K
* Resource group	○ Create new ● Use existing
	paassecurity
* Location	Central US

SETTINGS

Paassecurity_sql_name	paassecurity-sql-yourinit
Paassecurity_cosmos_name	paassecurity-cosmos-yourinit
Paassecurity_ase_name	paassecurity-ase-yourinit
Paassecurity_ssa_name	paassecurityssayourinit
Location	Central US
Admin Username	wsadmin
Admin Password	p@ssword1rocks

12. Check the **I agree to the terms and conditions stated above** checkbox.
13. Select **Purchase**.



offering.

☑ I agree to the terms and conditions stated above

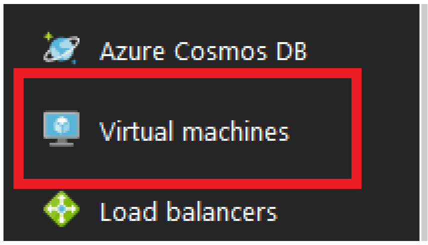
□ Pin to dashboard

14. The deployment will take about 90 minutes to complete. To view the progress, select the **Deployments** link.
    a. As part of the deployment, you will see the following items created:
        i. App Service Environment v2
        ii. Virtual Networks and Machines
        iii. Cosmos DB
        iv. Azure SQL Server and Databases
        v. Application Gateway with Firewall
15. See Appendix A for detailed steps on creating these components without using an ARM template.

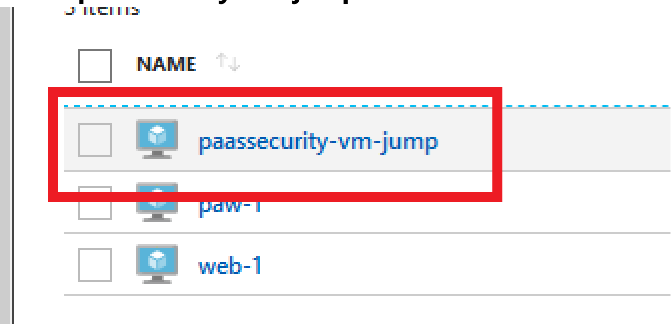
Task 3: Download GitHub resources (Jump machine)

In this task, you will log into the lab VM that was created by the ARM template you executed in the previous task and download the GitHub resources needed to complete this hands-on lab.

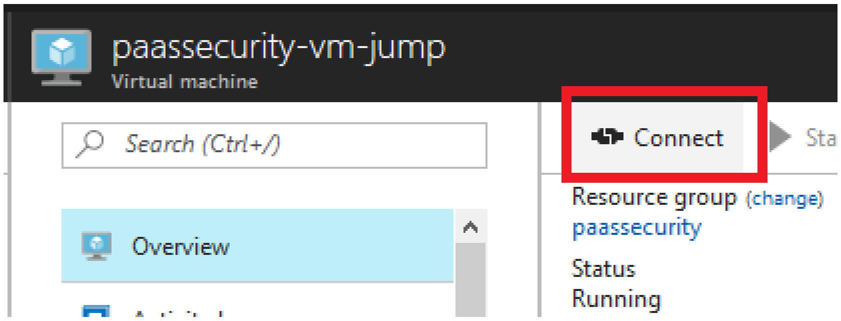
1. Login to the paassecurity-vm-jump virtual machine.
   a. Select **Virtual machines**.



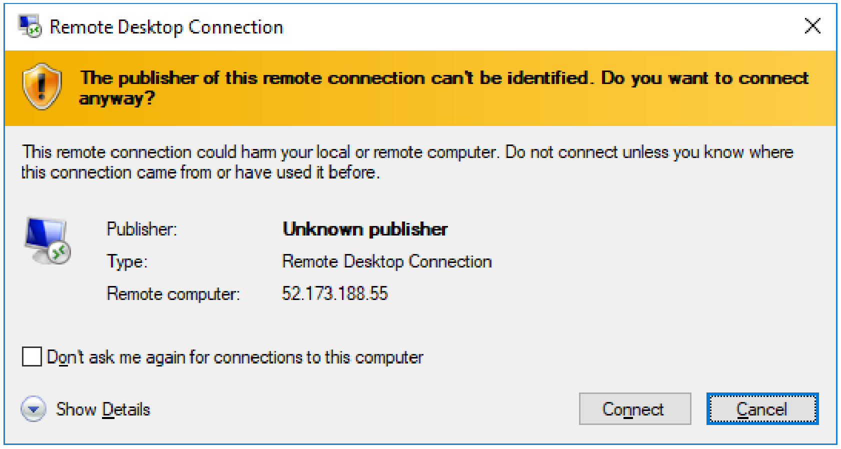
   b. Select **paassecurity-vm-jump**.



   c. Select **Connect**.



   d. Select to open the RDP connection.
   e. Enter the VM credentials (**wsadmin – p@ssword1rocks**).



Remote Desktop Connection

The publisher of this remote connection can't be identified. Do you want to connect anyway?

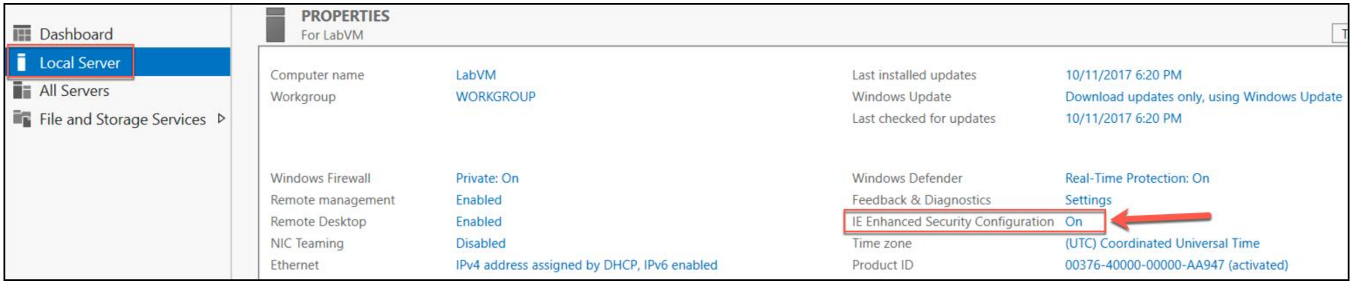
This remote connection could harm your local or remote computer. Do not connect unless you know where this connection came from or have used it before.

Publisher: **Unknown publisher**
Type: Remote Desktop Connection
Remote computer: 52.173.188.55

□ Don't ask me again for connections to this computer

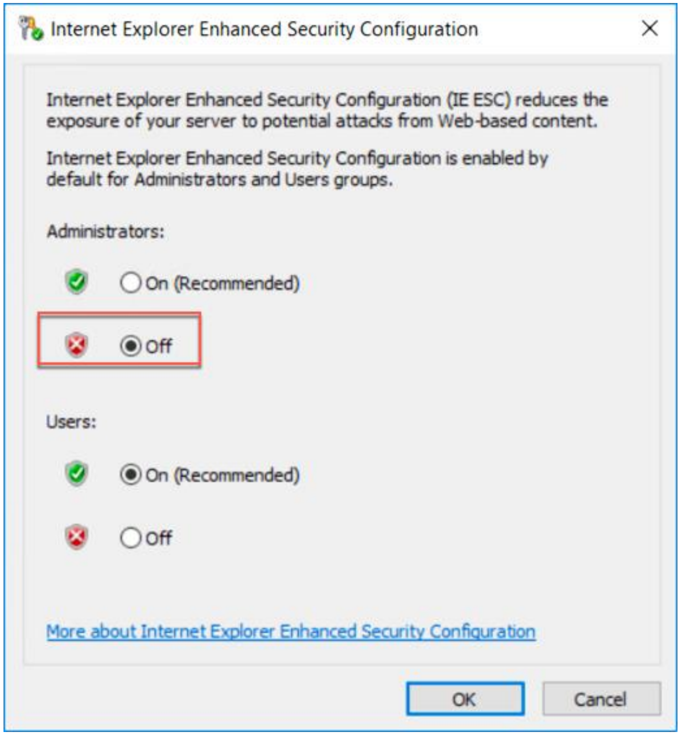
   f. Select **Connect**.

2. Once logged in, launch the Server Manager. This should start automatically, but you can access it via the Start menu if it does not start.
3. Select Local Server, the select On next to IE Enhanced Security Configuration.



Computer name	LabVM	Last installed updates	10/11/2017 6:20 PM
Workgroup	WORKGROUP	Windows Update	Download updates only, using Windows Update
		Last checked for updates	10/11/2017 6:20 PM
Windows Firewall	Private: On	Windows Defender	Real-Time Protection: On
Remote management	Enabled	Feedback & Diagnostics	Settings
Remote Desktop	Enabled	IE Enhanced Security Configuration	On
NIC Teaming	Disabled	Time zone	(UTC) Coordinated Universal Time
Ethernet	IPv4 address assigned by DHCP, IPv6 enabled	Product ID	00376-40000-00000-AA947 (activated)

4. In the Internet Explorer Enhanced Security Configuration dialog, select Off under Administrators, then select OK.


Internet Explorer Enhanced Security Configuration
More about Internet Explorer Enhanced Security Configuration
OK
Cancel

Internet Explorer Enhanced Security Configuration (IE ESC) reduces the exposure of your server to potential attacks from Web-based content.

Internet Explorer Enhanced Security Configuration is enabled by default for Administrators and Users groups.

Administrators:

○ On (Recommended)

● Off

Users:

● On (Recommended)

○ Off

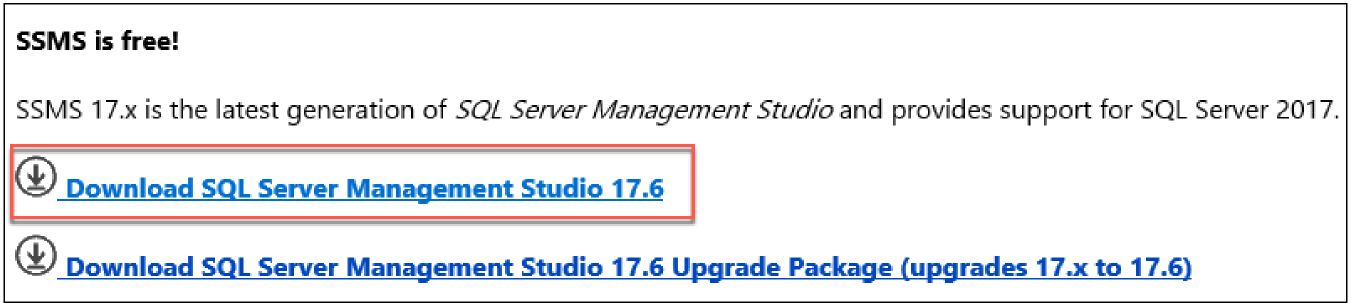
5. Close the Server Manager.
6. Repeat the steps you completed in Task 1 to download or copy the GitHub folders to the virtual machine.

Task 4: Install SQL Server Management Studio

In this task, you will install SQL Server Management Studio (SSMS) on your Jump machine VM.

1. On your jump machine VM, open a web browser and navigate to https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms.

2. Select Download SQL Server Management Studio 17.x.

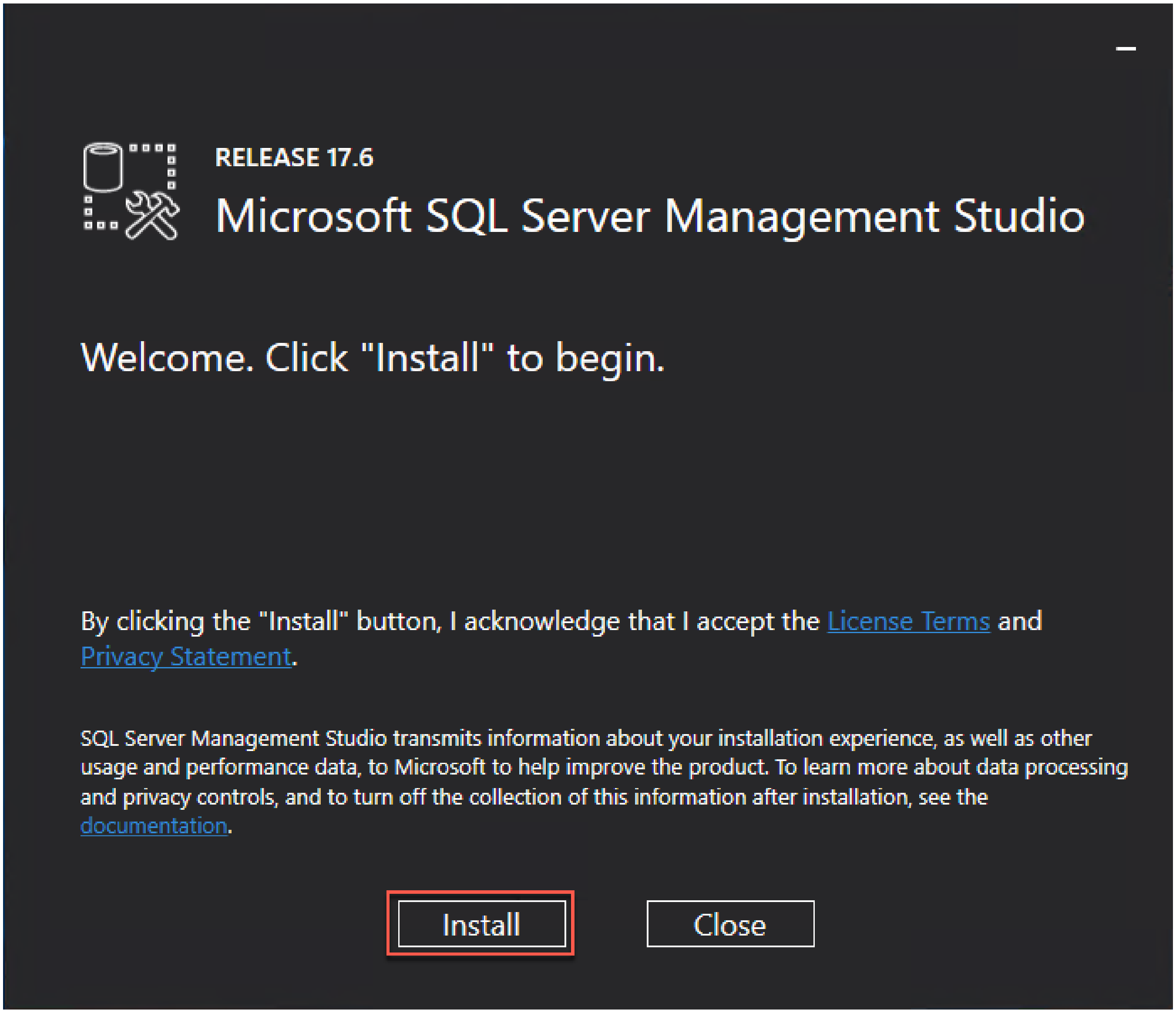


SSMS is free!

SSMS 17.x is the latest generation of *SQL Server Management Studio* and provides support for SQL Server 2017.

Download SQL Server Management Studio 17.6
Download SQL Server Management Studio 17.6 Upgrade Package (upgrades 17.x to 17.6)

3. Run the downloaded installer.
4. On the Welcome screen, select Install.



RELEASE 17.6

Microsoft SQL Server Management Studio

Welcome. Click "Install" to begin.

By clicking the "Install" button, I acknowledge that I accept the License Terms and Privacy Statement.

SQL Server Management Studio transmits information about your installation experience, as well as other usage and performance data, to Microsoft to help improve the product. To learn more about data processing and privacy controls, and to turn off the collection of this information after installation, see the documentation.

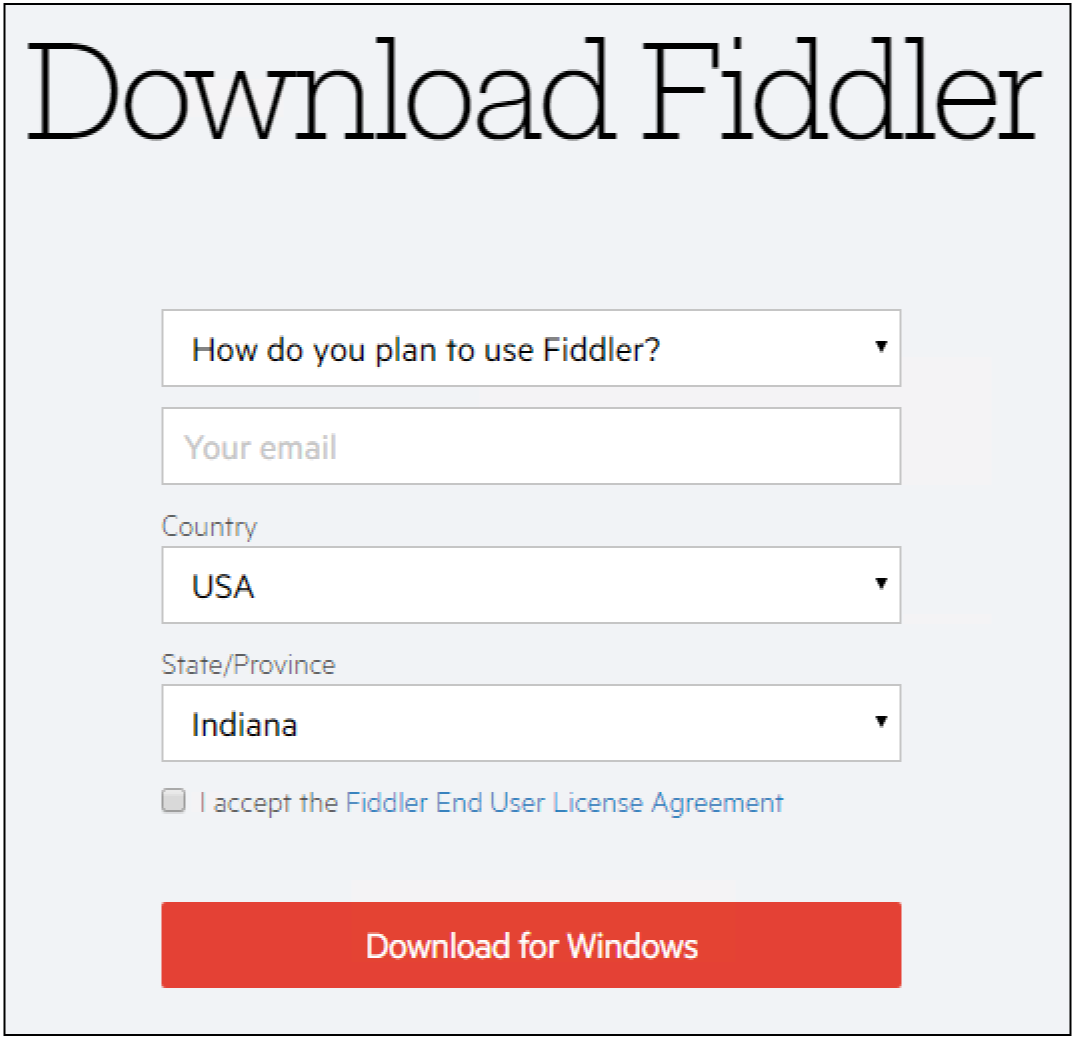
5. **Close** the SSMS installer once setup is completed and **restart the VM** to complete the installation of SSMS.

Task 5: Install Fiddler

In this task, you will download and install Fiddler, which will enable you to watch network traffic from your lab VM.

1. In a web browser, navigate to https://www.telerik.com/download/fiddler.

2. Complete the form, accepting the license agreement, and select Download for Windows.



Download Fiddler

How do you plan to use Fiddler? ▾

Your email

Country

USA ▾

State/Province

Indiana ▾

□ I accept the Fiddler End User License Agreement

Download for Windows

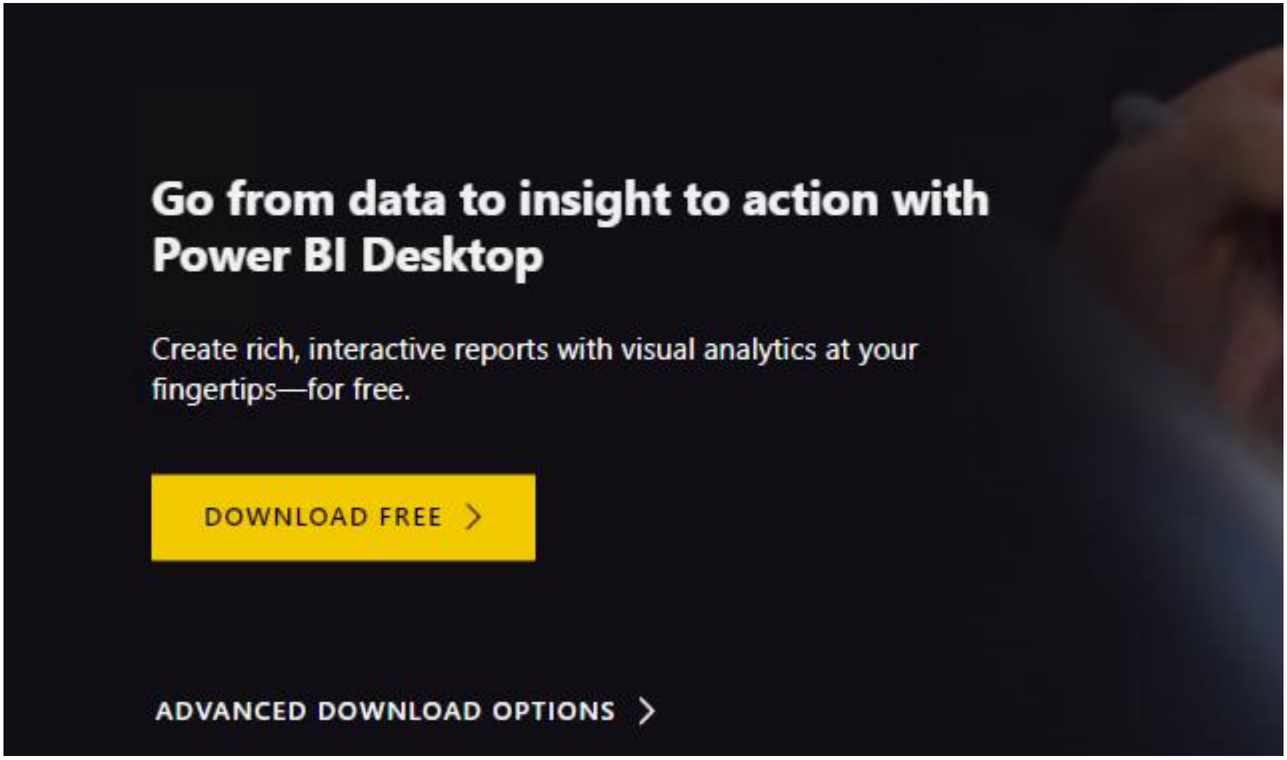
3. Run the download installer, accepting all the default values.
4. Close the installer when completed.

Task 6: Install Power BI Desktop

Below, you will install Power BI on the jump VM, which will be used in Exercise 8.

1. In a web browser on you jump VM navigate to the Power BI Desktop download page (https://powerbi.microsoft.com/en-us/desktop/).

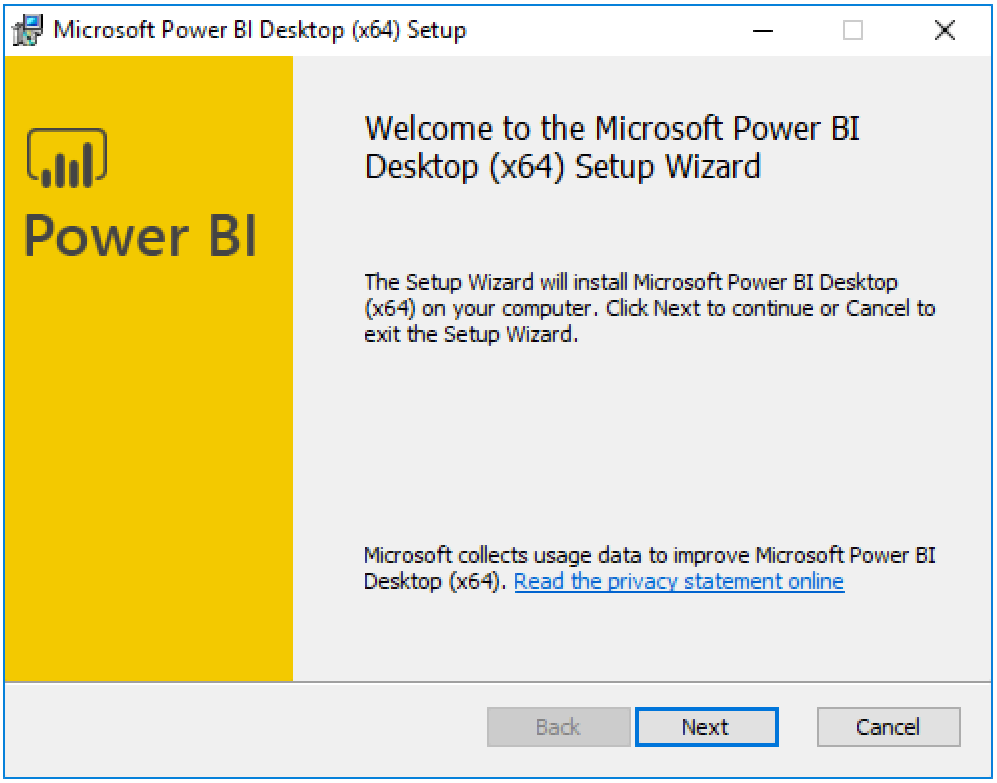
2. Select the Download Free link in the middle of the page.



Go from data to insight to action with Power BI Desktop

Create rich, interactive reports with visual analytics at your fingertips—for free.

3. Run the installer.
4. Select Next on the welcome screen.



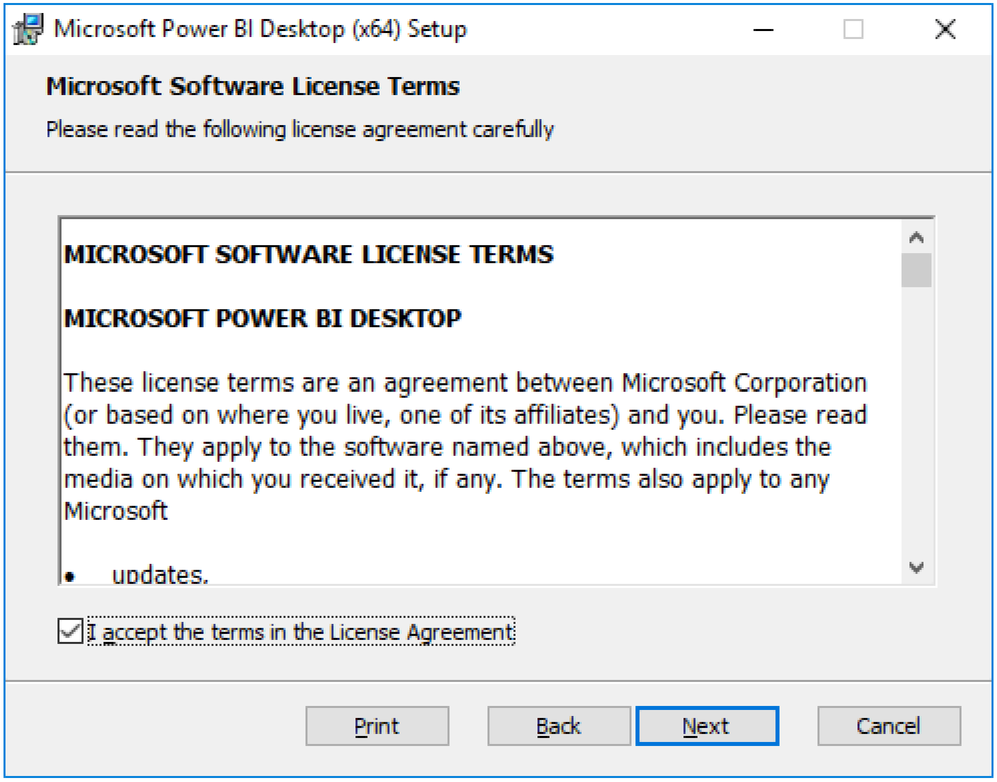
Microsoft Power BI Desktop (x64) Setup

Welcome to the Microsoft Power BI Desktop (x64) Setup Wizard

The Setup Wizard will install Microsoft Power BI Desktop (x64) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

Microsoft collects usage data to improve Microsoft Power BI Desktop (x64). Read the privacy statement online

5. Accept the license agreement, and select Next.



Microsoft Power BI Desktop (x64) Setup

Microsoft Software License Terms

Please read the following license agreement carefully

MICROSOFT SOFTWARE LICENSE TERMS

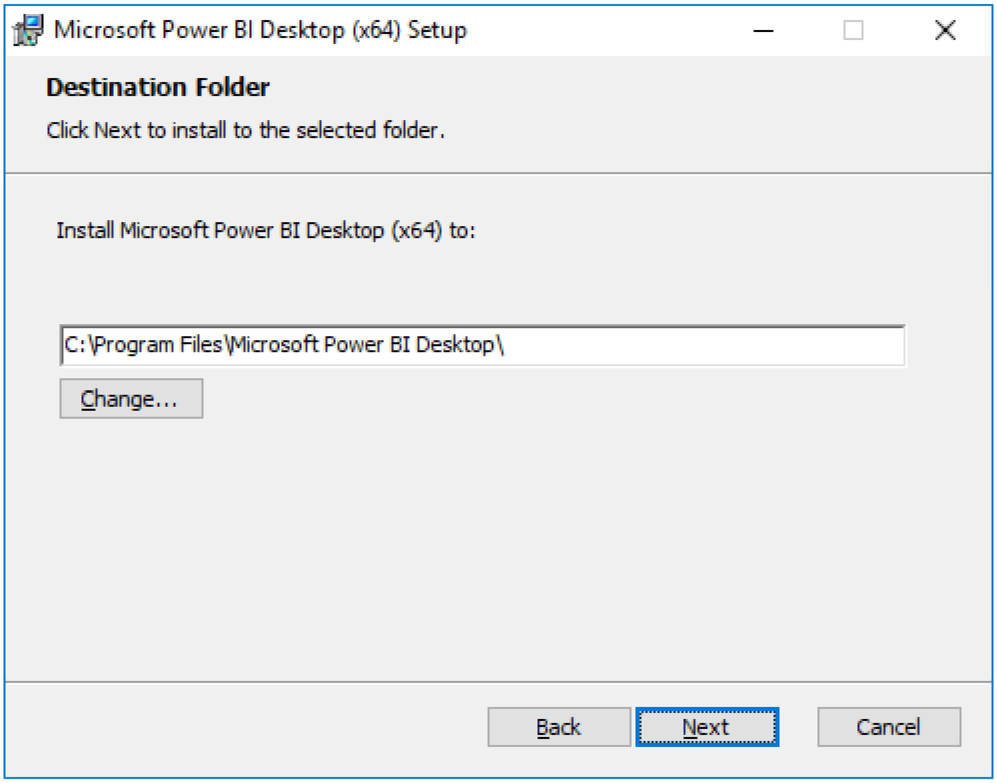
MICROSOFT POWER BI DESKTOP

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to the software named above, which includes the media on which you received it, if any. The terms also apply to any Microsoft

- updates,

☑ I accept the terms in the License Agreement

6. Leave the default destination folder, and select Next.



Microsoft Power BI Desktop (x64) Setup

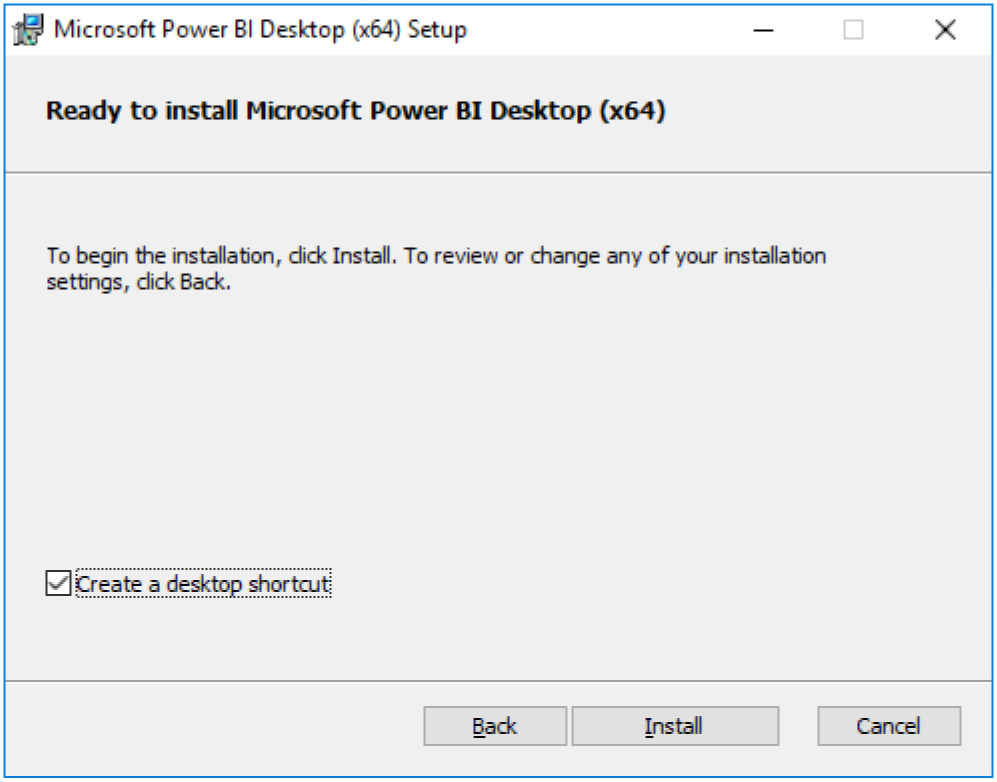
Destination Folder

Click Next to install to the selected folder.

Install Microsoft Power BI Desktop (x64) to:

C:\Program Files\Microsoft Power BI Desktop\

7. Make sure the Create a desktop shortcut box is checked, and select Install.



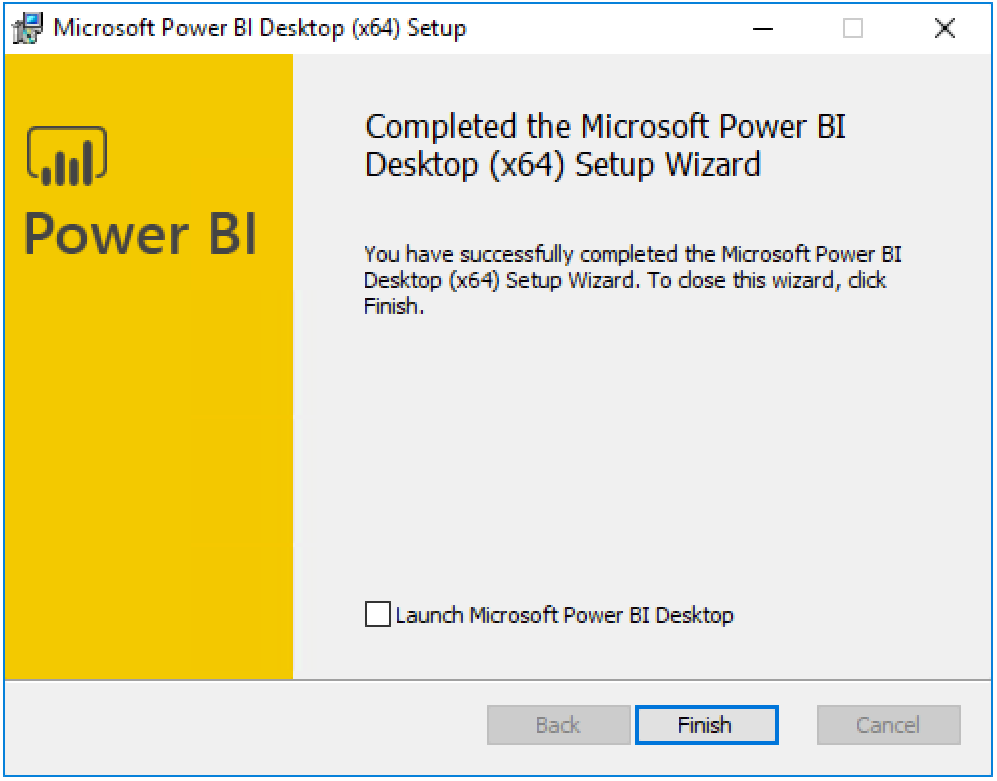
Microsoft Power BI Desktop (x64) Setup

Ready to install Microsoft Power BI Desktop (x64)

To begin the installation, click Install. To review or change any of your installation settings, click Back.

☑ Create a desktop shortcut

8. Uncheck Launch Microsoft Power BI Desktop, and select Finish.



Microsoft Power BI Desktop (x64) Setup

Completed the Microsoft Power BI Desktop (x64) Setup Wizard

You have successfully completed the Microsoft Power BI Desktop (x64) Setup Wizard. To close this wizard, click Finish.

□ Launch Microsoft Power BI Desktop

You should follow all steps provided *before* attending the Hands-on lab.

Exercise 1: Creating and securing Azure Active Directory accounts

Duration: 45 minutes

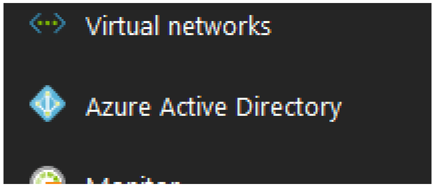
Synopsis: In this exercise, attendees will learn how to create Azure Active Directory (Azure AD) groups and users and then securing them using multi-factor authentication.

NOTE: If you are using a corporate Azure instance and do not have access to Active Directory, you will not be able to complete this exercise, and should skip to [Exercise 3](#).

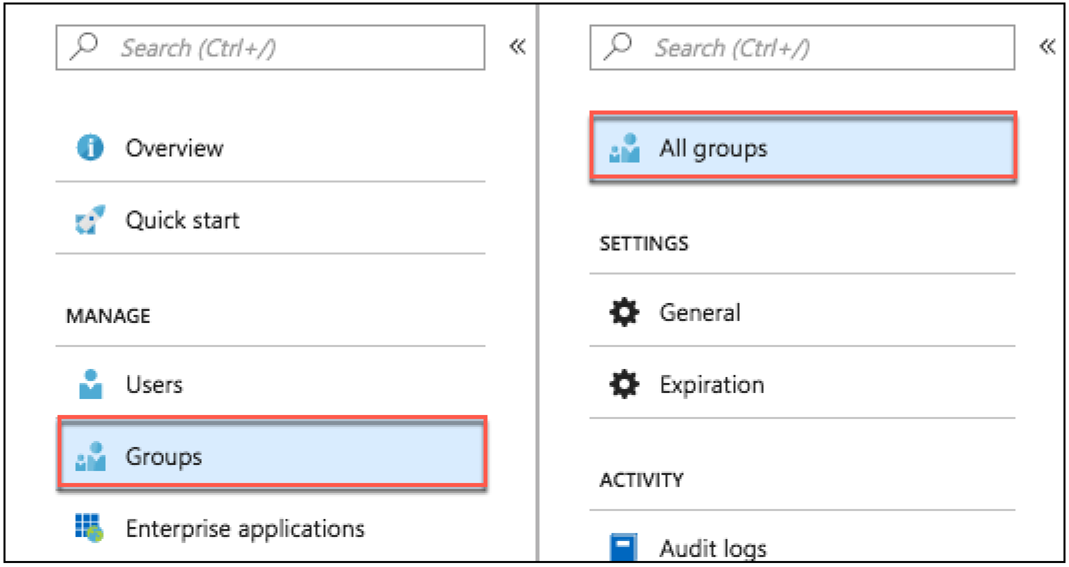
Task 1: Create Azure Active Directory groups

In this task, you will create security groups in Azure AD to be used in exercises later in this hands-on lab.

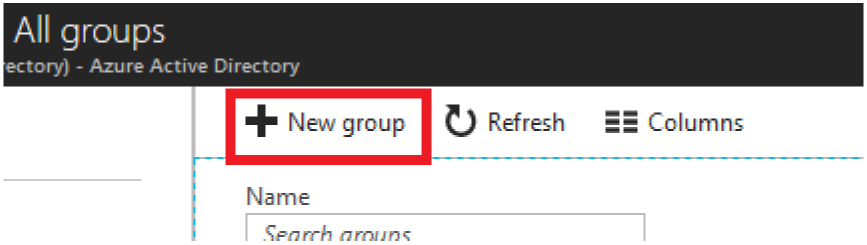
1. Open your Azure Portal (https://portal.azure.com).
2. Select **Azure Active Directory**.



3. Select **Groups,** then select **All groups**.

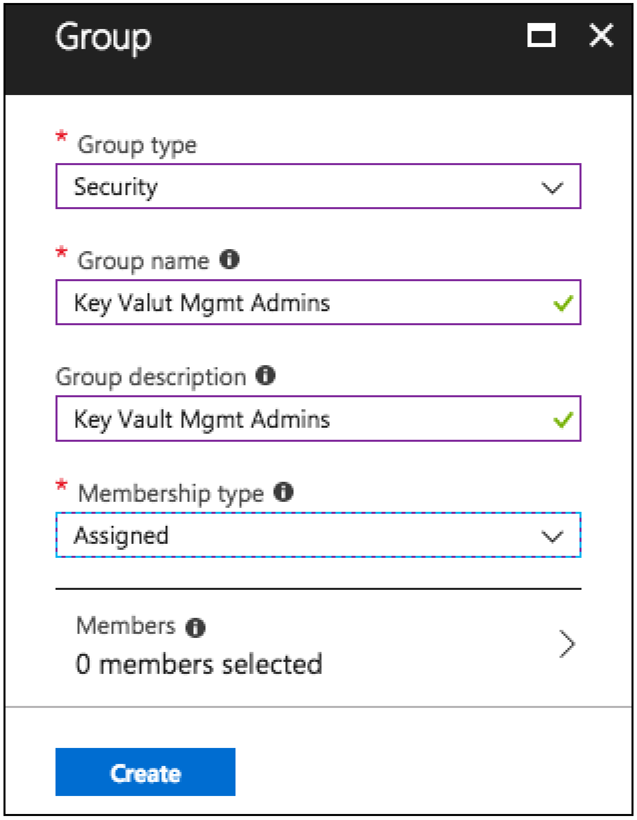


4. Select **+New group**.



5. On the Group blade, enter the following:
   a. **Group type**: Select **Security**
   b. **Group name**: Enter **Key Vault Mgmt Admins**
   c. **Group description**: Enter **Key Vault Mgmt Admins**

d. **Membership type**: select **Assigned**



Group

* Group type

Security

* Group name

Key Valut Mgmt Admins

Group description

Key Vault Mgmt Admins

* Membership type

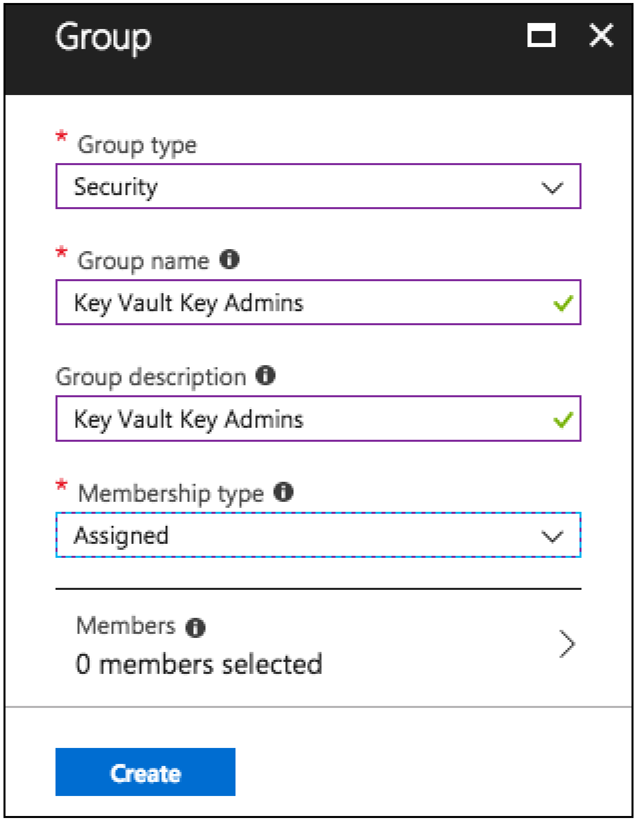
Assigned

Members

0 members selected

6. Select **Create** and close the dialog window if it does not close.
7. Select **+New group** again.
8. On the Group blade, enter the following:
    a. **Group type**: Select **Security**
    b. **Group name**: Enter **Key Vault Key Admins**
    c. **Group description**: Enter **Key Vault Key Admins**

d. **Membership type**: select **Assigned**



Group

* Group type
Security

* Group name
Key Vault Key Admins

Group description
Key Vault Key Admins

* Membership type
Assigned

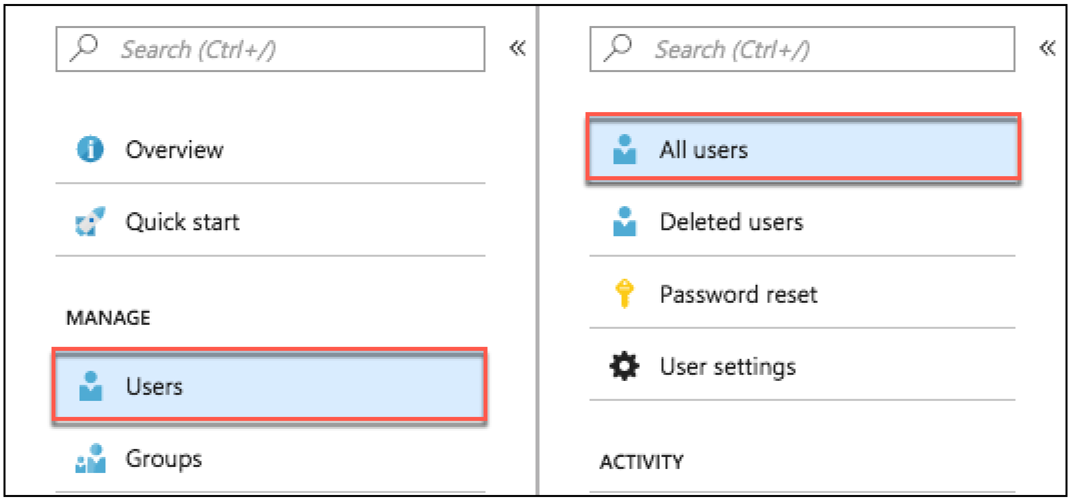
Members
0 members selected

9. Select **Create** and close the dialog window if it does not close.

Task 2: Create Azure Active Directory accounts

In this task, you will create multiple Azure AD user accounts that will be used within the exercises in this hands-on lab to demonstrate the various levels of permissions and access control with Azure resources.

1. Determine your Active Directory domain name.
   a. Select **Azure Active Directory**.
   b. Select **Custom domain names**.
   c. Record the ***.microsoftonline.com** domain name, you will use this later.
2. Select **Users**, then select **All users**.

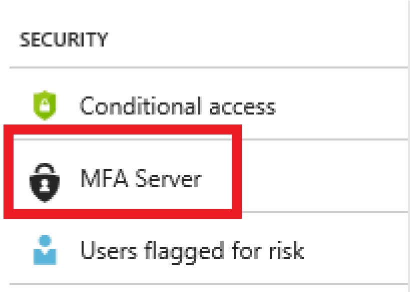


3. Select +**New user**.
4. On the User blade, enter the following:
    a. **Name**: enter **Key Vault Admin**
    b. **User name**, enter KeyVaultAdmin@<yourdomain>.microsoftonline.com
        - NOTE: Use the domain you recorded earlier.
    c. Select **Groups**.
        i. Select **Key Vault Mgmt Admins**, select.
    d. Select **Create**.
5. Select +**New user** again.
6. On the User blade, enter the following:
    a. **Name**: Enter **Key Vault Auditor**.
    b. **User name**, enter KeyVaultAuditor@<yourdomain>.microsoftonline.com
        - NOTE: Use the domain you recorded earlier.
    c. Select **Groups**.
        i. Select **Key Vault Mgmt Admins**, select.
    d. Select **Create**.
7. Select +**New user** again.
    a. **Name**, enter **Key Vault Developer**.
    b. **User name**, enter KeyVaultDeveloper@<yourdomain>.microsoftonline.com
        i. NOTE: Use the domain you recorded earlier.
    c. NOTE: No groups will be assigned to this user.
    d. Select **Create**.

Task 3: Enable Azure Identity Protection features

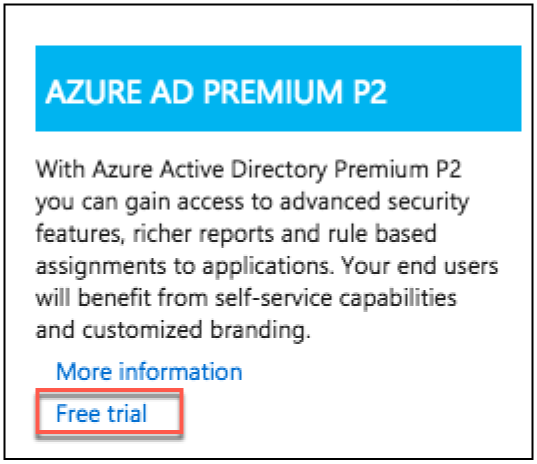
In this task, you will enable multi-factor authentication on the Key Vault Admin account you created in the previous task to demonstrate the Identity Protection features of Azure.

1. Select your Active Directory.
2. Select **MFA Server**.



3. Select **Get Free Premium Trial**.

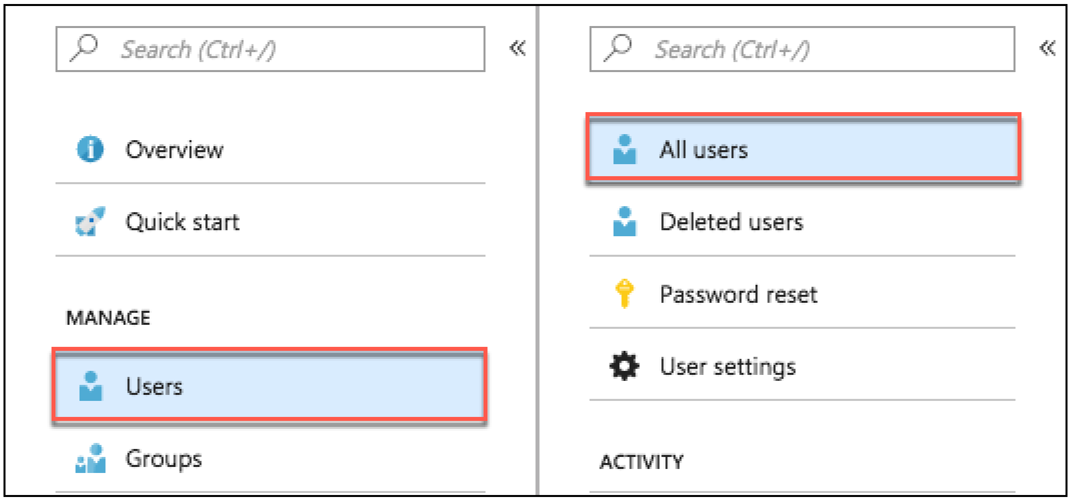
4. Select the **AZURE AD PREMIUM P2** option, select **Free trial**.



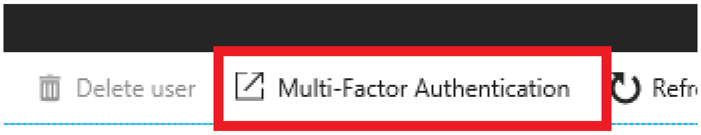
AZURE AD PREMIUM P2

With Azure Active Directory Premium P2 you can gain access to advanced security features, richer reports and rule based assignments to applications. Your end users will benefit from self-service capabilities and customized branding.

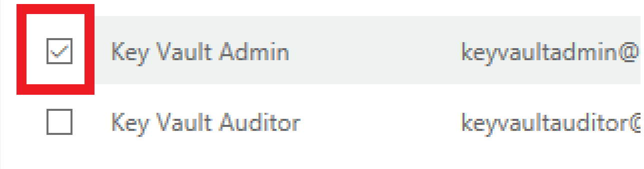
5. Select **Activate**.
6. Select **Users**, the select **All users**.



7. Select **Multi-Factor Authentication**.



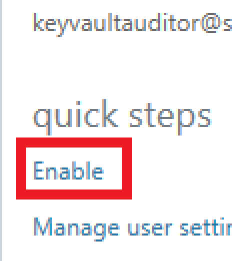
8. Check the check box for the **Key Vault Admin** user



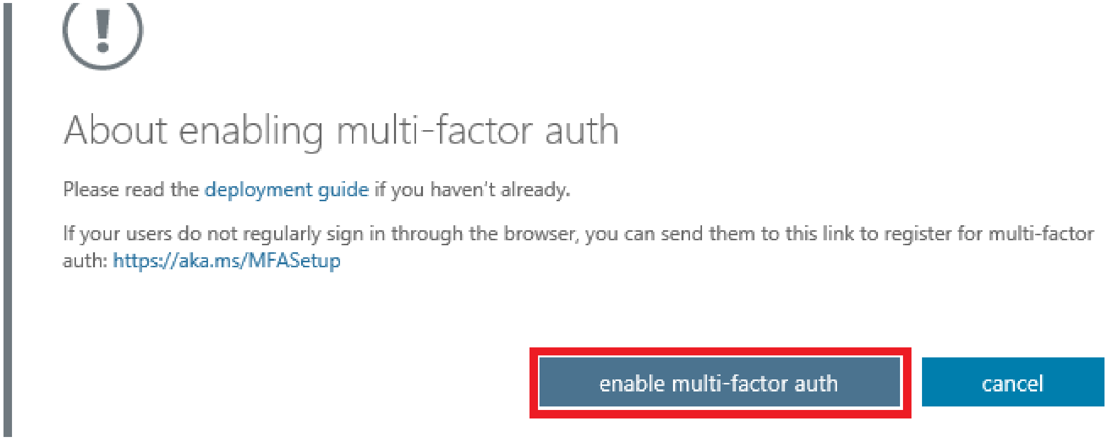
☑ Key Vault Admin keyvaultadmin@

☐ Key Vault Auditor keyvaultauditor@

9. Select **Enable**.



10. In the dialog, select **enable multi-factor auth**.

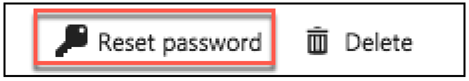


About enabling multi-factor auth

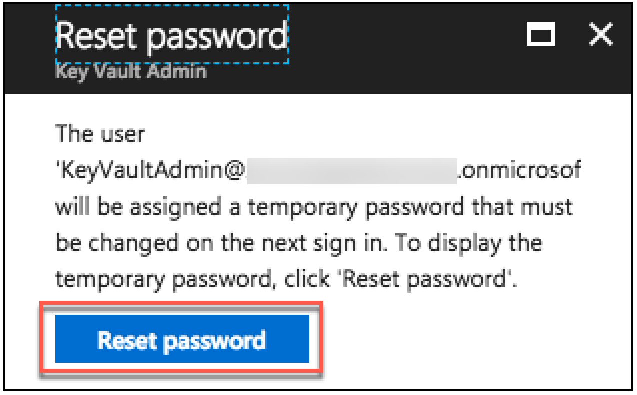
Please read the deployment guide if you haven't already.

If your users do not regularly sign in through the browser, you can send them to this link to register for multi-factor auth: https://aka.ms/MFASetup

11. In the dialog, select **close**.
12. Attempt to sign-in as the **KeyVaultAdmin user**.
13. In the Azure portal, select Azure Active Directory.
    a. Select Users, All Users, and select the Key Vault Admin user from the list.
    b. On the Key Vault Admin user blade, select **Reset Password**.



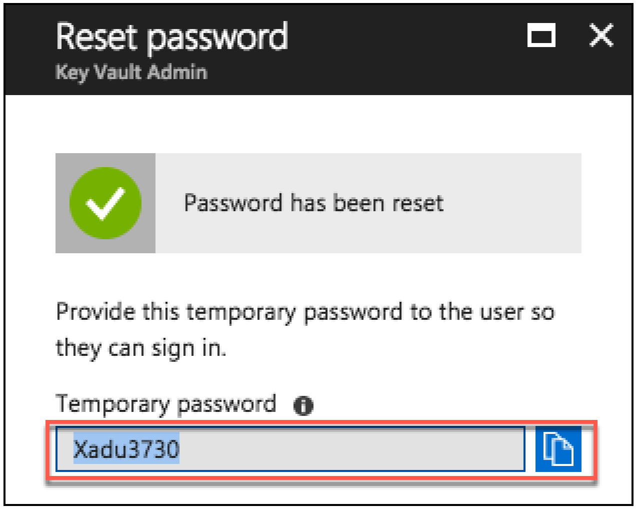
    c. On the Reset password blade, select **Reset password**.



Reset password
Key Vault Admin

The user 'KeyVaultAdmin@[redacted].onmicrosof will be assigned a temporary password that must be changed on the next sign in. To display the temporary password, click 'Reset password'.

    d. Copy the Temporary password for use in the next step.



Reset password
Key Vault Admin

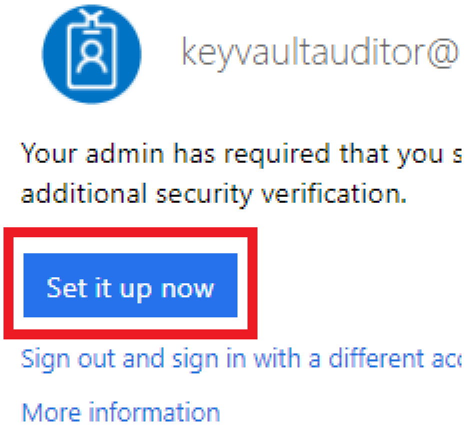
Password has been reset

Provide this temporary password to the user so they can sign in.

Temporary password: Xadu3730

14. Open an InPrivate or Incognito browser window, navigate to http://login.microsoftonline.com and enter the **username** and **password** for the KeyVaultAdmin account.

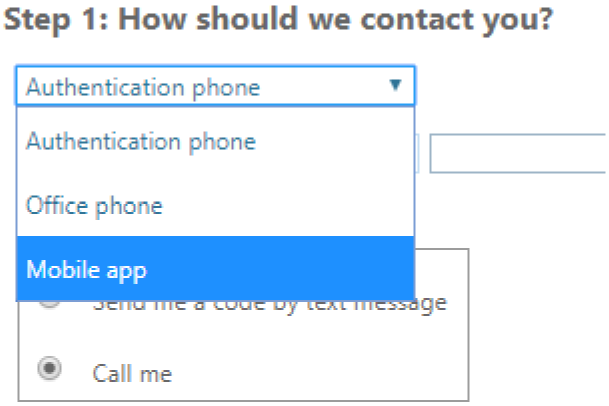
15. You will be prompted to setup additional security, select **Set it up now**.



keyvaultauditor@

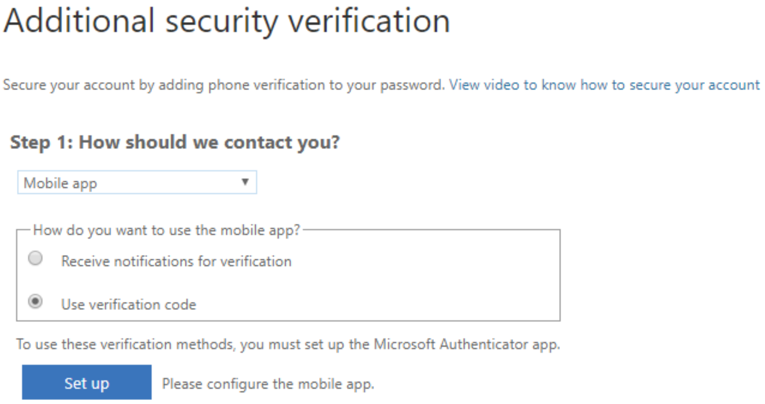
Your admin has required that you s
additional security verification.

16. Select **Mobile app in the dropdown**.



Step 1: How should we contact you?

17. Select **Use verification code**.



Additional security verification

Secure your account by adding phone verification to your password. View video to know how to secure your account

Step 1: How should we contact you?

Mobile app ▾

How do you want to use the mobile app?

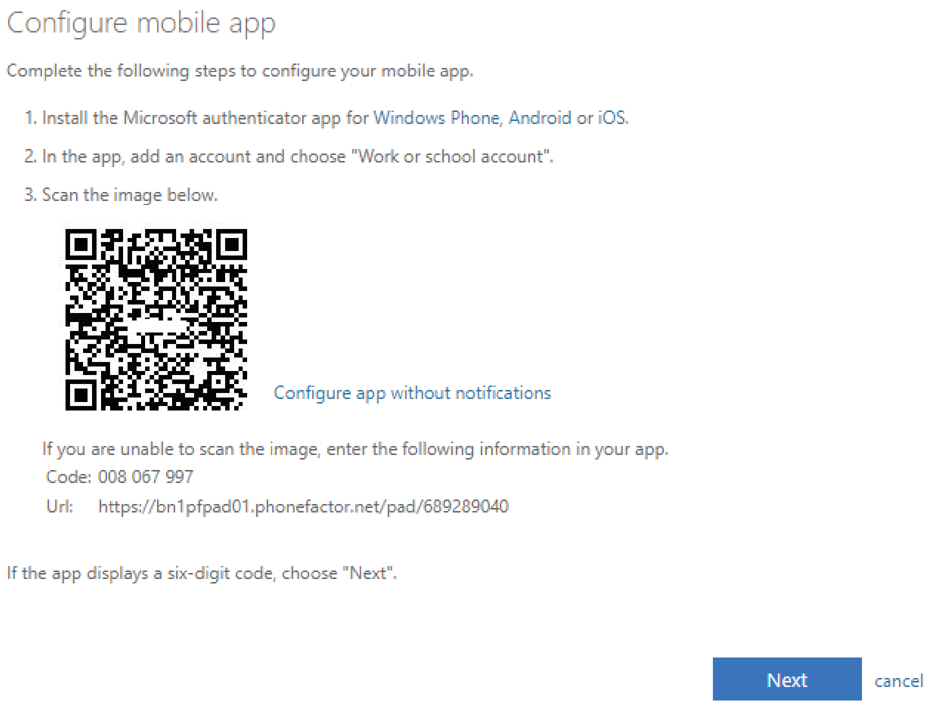
○ Receive notifications for verification

● Use verification code

To use these verification methods, you must set up the Microsoft Authenticator app.

Please configure the mobile app.

18. Select **Set up**.
19. Depending on your mobile device, download the Microsoft Authenticator application from the respective app store.
20. Scan the image on the page to add the credentials to your authenticator app.



Configure mobile app

Complete the following steps to configure your mobile app.

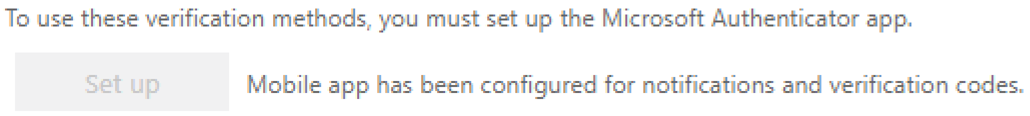
1. Install the Microsoft authenticator app for Windows Phone, Android or iOS.
2. In the app, add an account and choose "Work or school account".
3. Scan the image below.

Configure app without notifications

If you are unable to scan the image, enter the following information in your app.
Code: 008 067 997
Url: https://bn1pfpad01.phonefactor.net/pad/689289040

If the app displays a six-digit code, choose "Next".

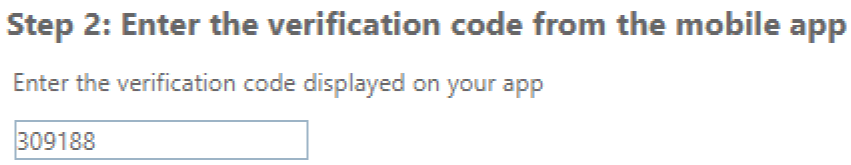
21. Select **Next**, the page will validate that you in fact added the account.



To use these verification methods, you must set up the Microsoft Authenticator app.

Mobile app has been configured for notifications and verification codes.

22. Select **Next**, enter the **validation code** from the mobile app.



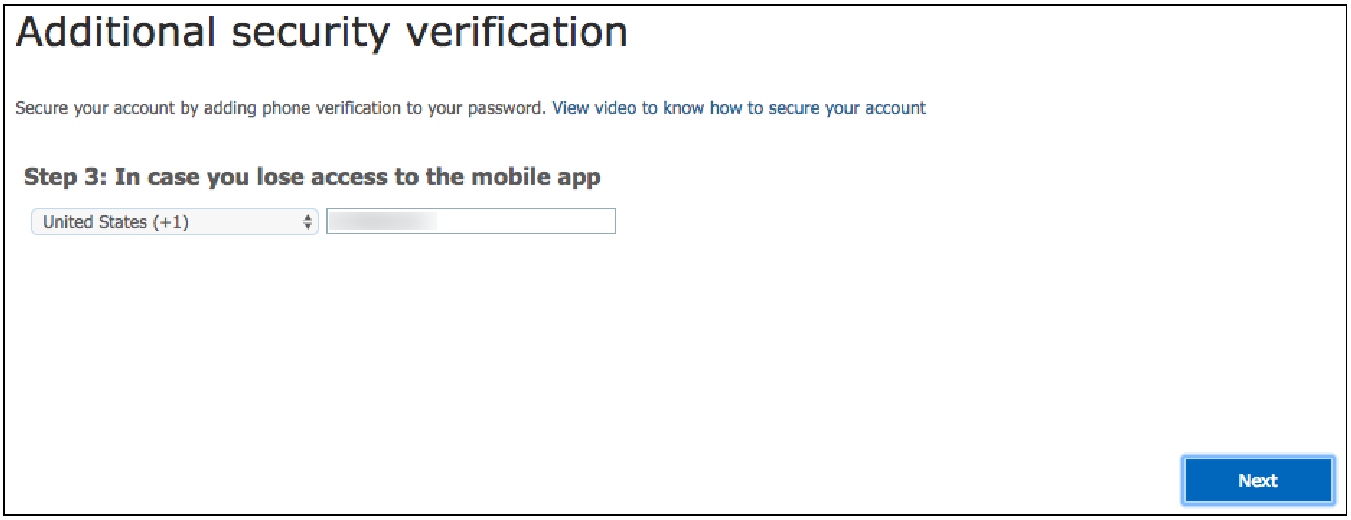
Step 2: Enter the verification code from the mobile app

Enter the verification code displayed on your app

309188

23. Select **Verify**.

24. On the Additional security verification, select your country, and enter your mobile phone number, then select **Next**.



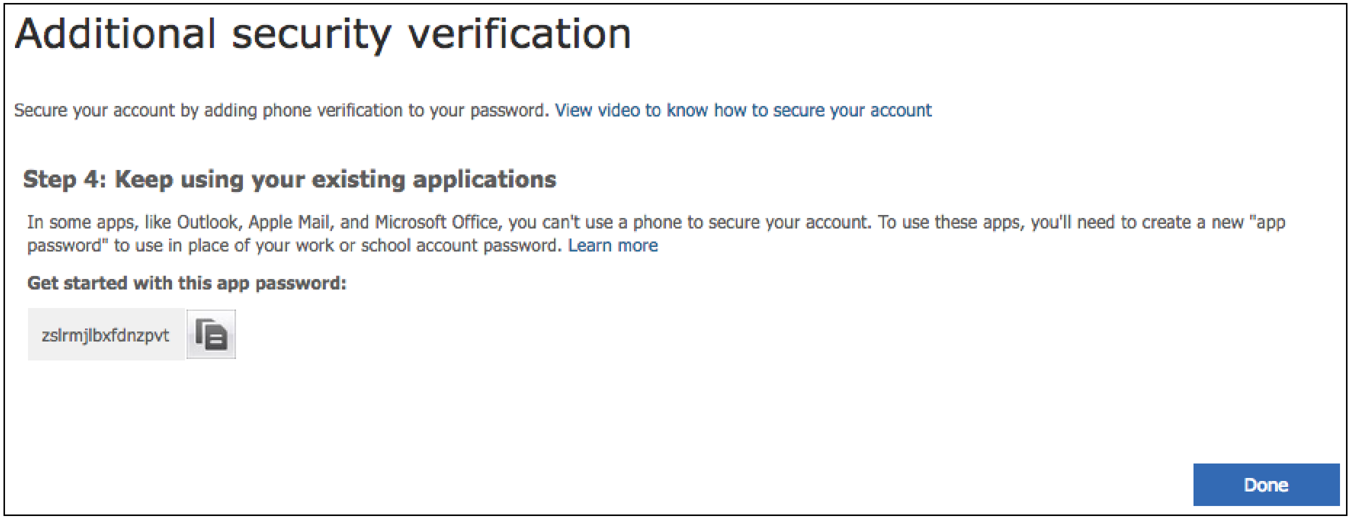
Additional security verification

Secure your account by adding phone verification to your password. View video to know how to secure your account

Step 3: In case you lose access to the mobile app

United States (+1) [redacted]

25. On the next screen, copy the password provided, and select **Done**.



Additional security verification

Secure your account by adding phone verification to your password. View video to know how to secure your account

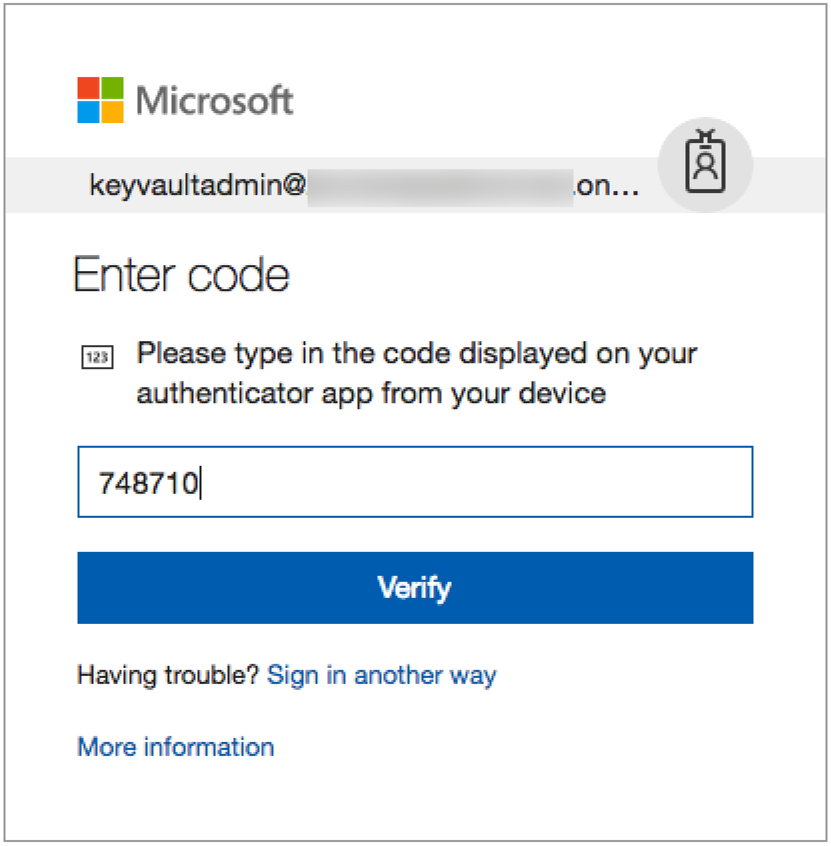
Step 4: Keep using your existing applications

In some apps, like Outlook, Apple Mail, and Microsoft Office, you can't use a phone to secure your account. To use these apps, you'll need to create a new "app password" to use in place of your work or school account password. Learn more

Get started with this app password:

zslrmjlbxfdnzpvt

26. Enter the Authenticator app code on the next screen and select **Verify**.



Microsoft

keyvaultadmin@[redacted]on...

Enter code

Please type in the code displayed on your authenticator app from your device

748710

Having trouble? Sign in another way

More information

27. If prompted, on the **Update your password page**, update your password.
    a. NOTE: The Current password will be the value you copied after resetting the password in Azure AD.
28. Select **Sign in**.
29. If prompted, close the **Welcome to Microsoft Azure** dialog.

Exercise 2: Securing Azure Key Vault with Azure IAM

Duration: 45 minutes

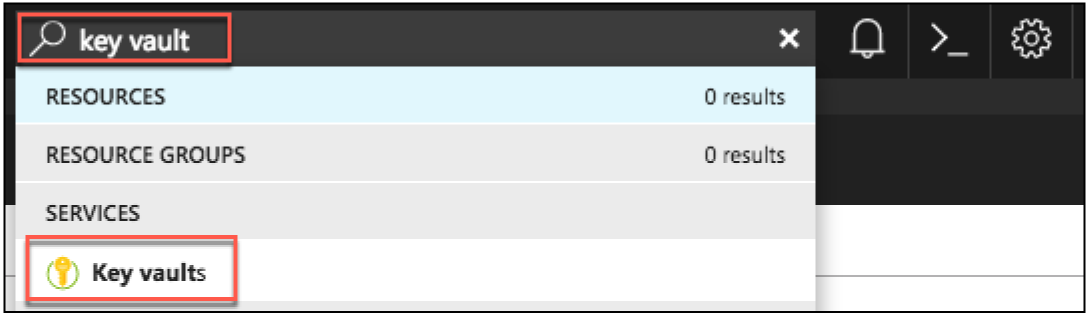
Synopsis: In this exercise, attendees will learn how to create various roles for managing the Azure Key Vault.

NOTE: If you are using a corporate Azure instance and do not have access to Active Directory, you must skip this Exercise and move to Exercise 3.

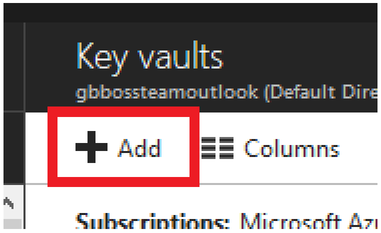
Task 1: Create a new Azure Key Vault

In this task, you will create a new Azure Key Vault.

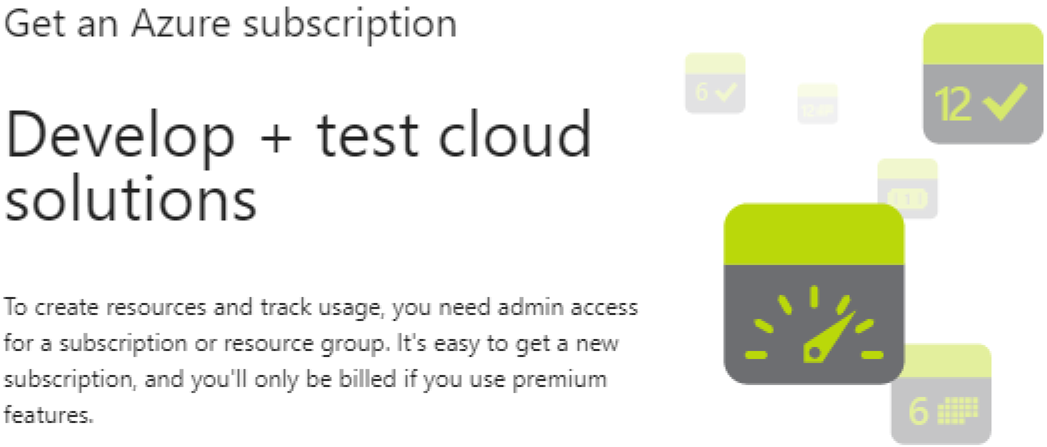
1. In your InPrivate or Incognito browser window, log into the Azure portal using the **KeyVaultAdmin** account.
2. In the Search box at the top of the Azure portal, search for “key vault” box, and select **Key vaults** from the results.



3. On the Key vaults blade, select +**Add**.



4. You should get a message that you must have admin access to create a key vault.



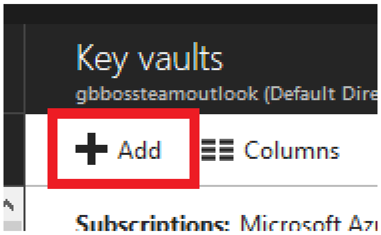
Get an Azure subscription

Develop + test cloud solutions

To create resources and track usage, you need admin access for a subscription or resource group. It's easy to get a new subscription, and you'll only be billed if you use premium features.

5. Return to the Azure portal browser window where you are logged in with your subscription admin account, not the Incognito window where the Key Vault Admin account is logged in.
6. As in step 2 above, search for **Key vaults** and navigate to the Key Vaults blade.

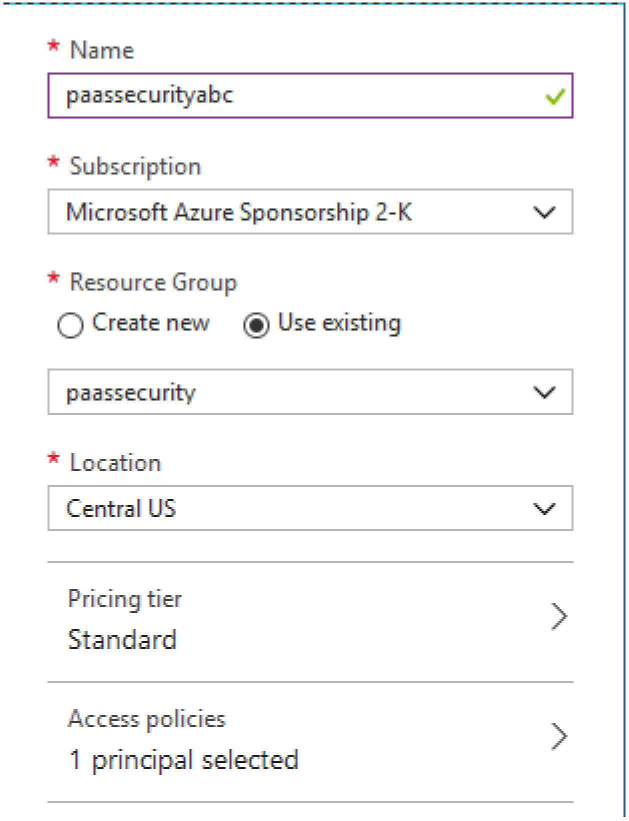
7. Select **+Add**.



Key vaults

gbbossteamoutlook (Default Dire

8. On the Create key vault blade, enter the following:
   a. **Name**: Enter something similar to **paassecuritykeyvault[Your Initials]**
   b. **Subscription**: Select the subscription you are using for this lab
   c. **Resource group**: Select your existing resource group
   d. Leave Pricing tier and Access policies set to their default values



* Name: paassecurityabc

* Subscription: Microsoft Azure Sponsorship 2-K

* Resource Group: ○ Create new ● Use existing

paassecurity

* Location: Central US

Pricing tier: Standard

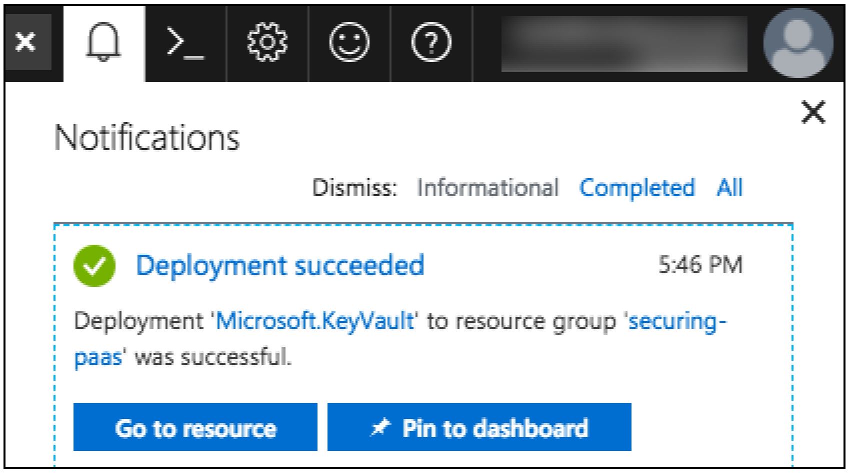
Access policies: 1 principal selected

9. Select **Create**.

Task 2: Assign IAM based Azure Key Vault permissions

In this task, you use Access control (IAM) to assign role-based access control (RBAC) permissions to the key vault you created in the previous task.

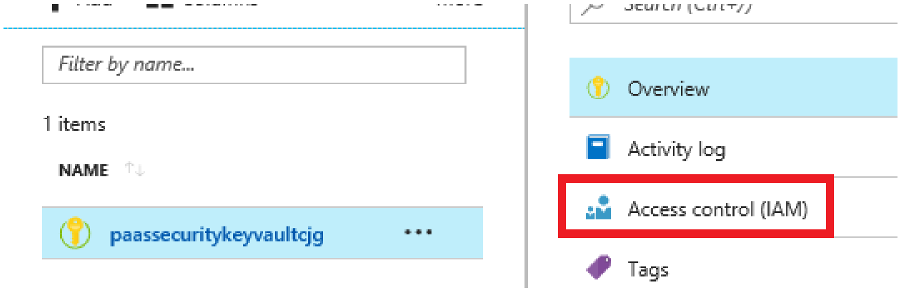
1. When the Key vault has finished provisioning, you will receive a notification in the Azure portal. In the notification, select **Go to resource**.



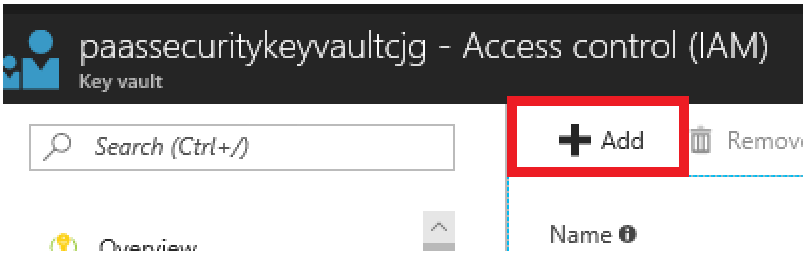
Deployment succeeded 5:46 PM

Deployment 'Microsoft.KeyVault' to resource group 'securing-paas' was successful.

2. Select **Access control (IAM)**.

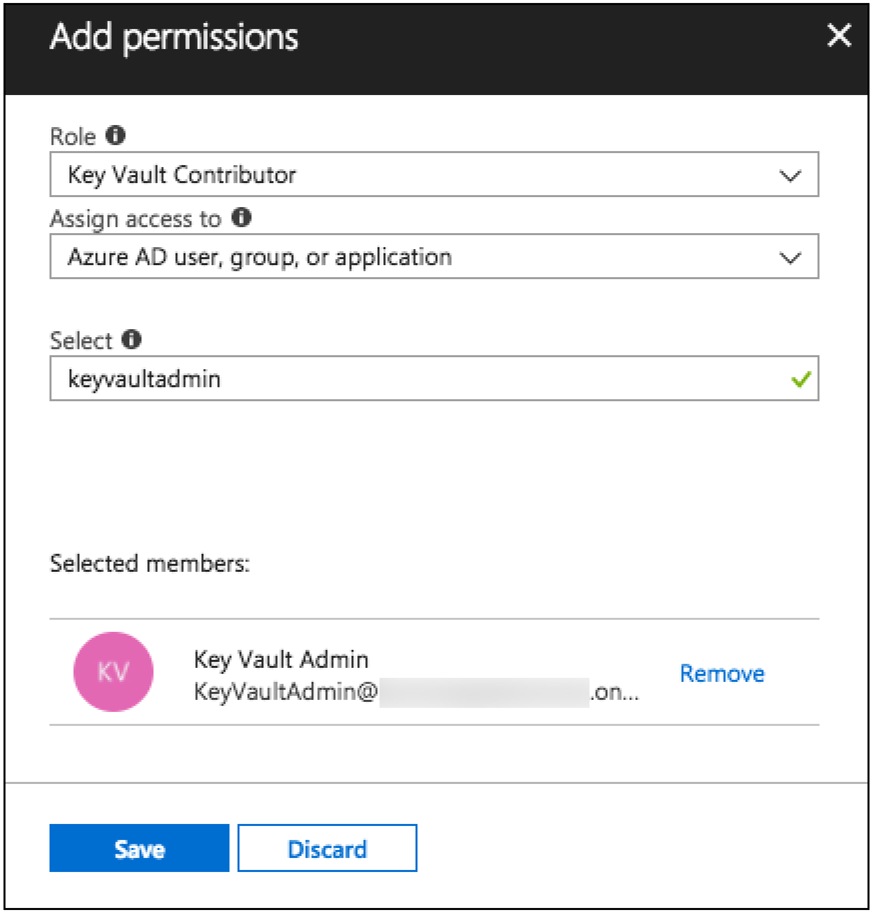


3. Select **+Add**.



4. In the Add permissions blade, enter:
   a. **Role:** Select **Key Vault Contributor**
   b. **Assign access to**: Leave set to Azure AD user, group, or application

Select: Search for and select the **KeyVaultAdmin** user.



Add permissions

Role: Key Vault Contributor

Assign access to: Azure AD user, group, or application

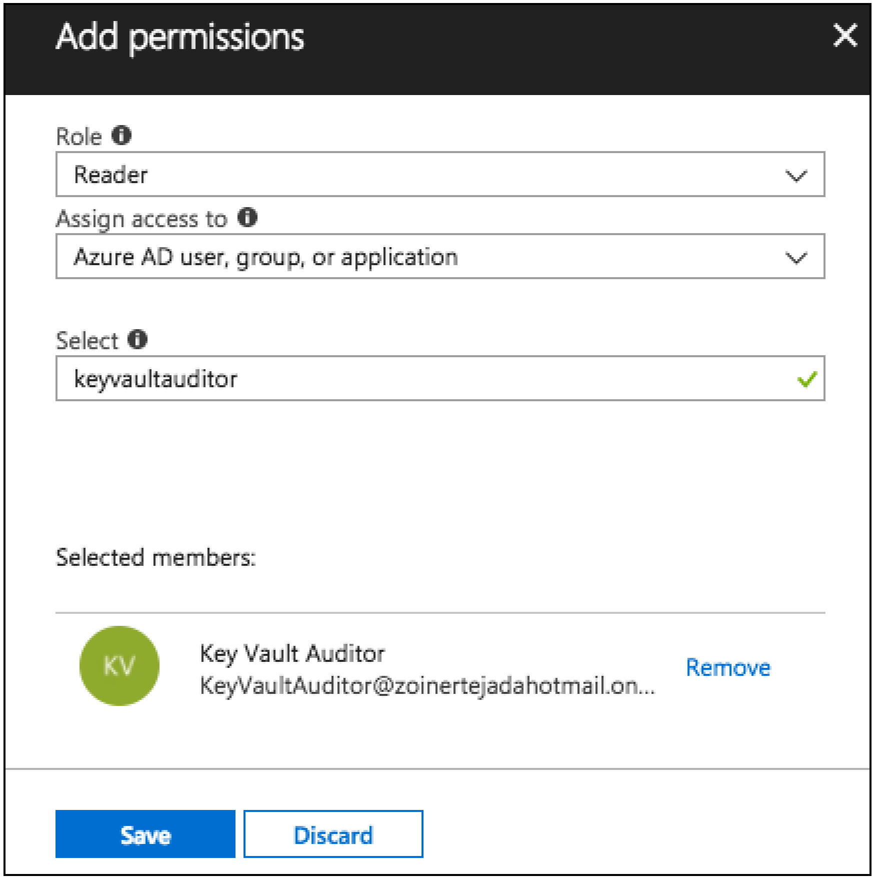
Select: keyvaultadmin

Selected members:

Key Vault Admin
KeyVaultAdmin@[redacted].on...

5. Select **Save**.
6. Select **+Add** again.
7. In the Add permissions blade, enter:
   a. **Role:** Select **Reader**
   b. **Assign access to**: Leave set to Azure AD user, group, or application

Select: Search for and select the **KeyVaultAuditor** user.



Add permissions

Role: Reader

Assign access to: Azure AD user, group, or application

Select: keyvaultauditor

Selected members:

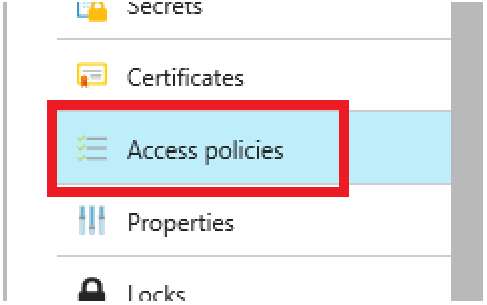
Key Vault Auditor
KeyVaultAuditor@zoinertejadahotmail.on...

8. Select **Save**.

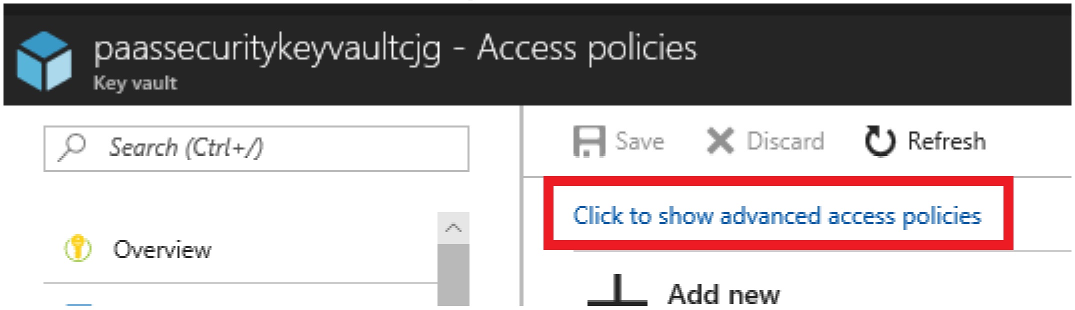
Task 3: Assign access policy based Azure Key Vault permissions

In this task, you will add Access policies to the Azure Key Vault, to set the permissions of individual users within the key vault.

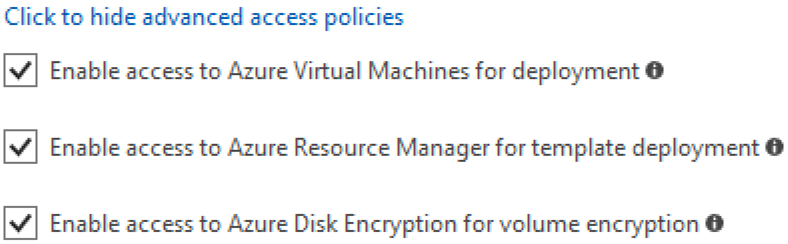
1. On the new key vault blade, select **Access Policies** from the left-hand menu under SETTINGS.



2. Select **Click to show advanced access policies**.


paassecuritykeyvaultcjg - Access policies
Key vault
Search (Ctrl+/)
Overview
Save
Discard
Refresh
Click to show advanced access policies
Add new

3. Check the boxes for all items.



Click to hide advanced access policies

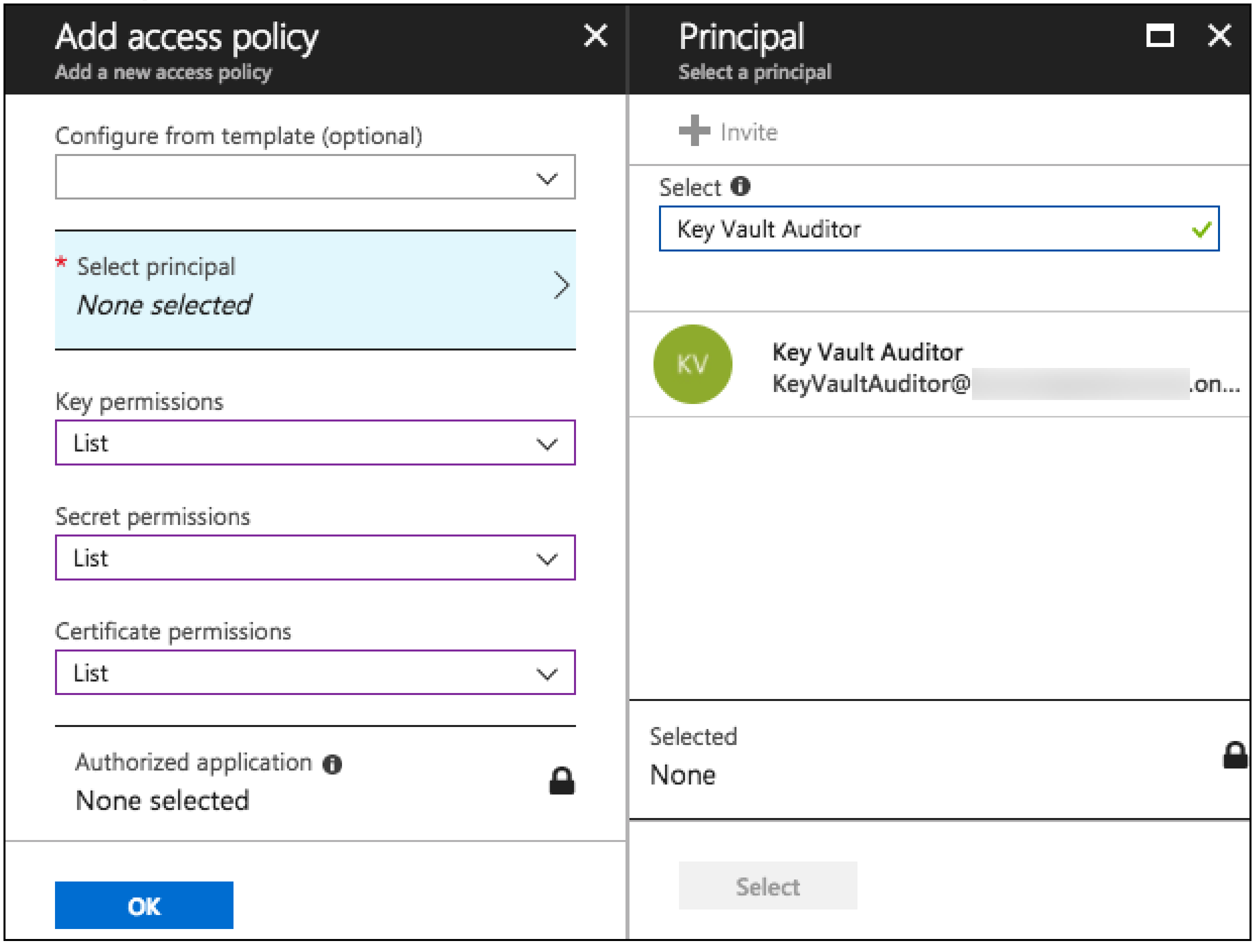
☑ Enable access to Azure Virtual Machines for deployment

☑ Enable access to Azure Resource Manager for template deployment

☑ Enable access to Azure Disk Encryption for volume encryption

4. Select **Save**.
5. Select **+Add new**.
6. On the Add access policy blade, enter the following:
    a. Select **Select principal**.
        i. Search for and select **Key Vault Auditor**.
        ii. Select **Select**.
    b. **Key permissions**: Check **List**.
    c. **Secret permissions**: Check **List**.

d. **Certificate permissions**: Check **List**.



Add access policy
Add a new access policy

Configure from template (optional)

* Select principal
None selected

Key permissions: List

Secret permissions: List

Certificate permissions: List

Authorized application
None selected

Principal
Select a principal

Select: Key Vault Auditor

Key Vault Auditor
KeyVaultAuditor@[redacted].on...

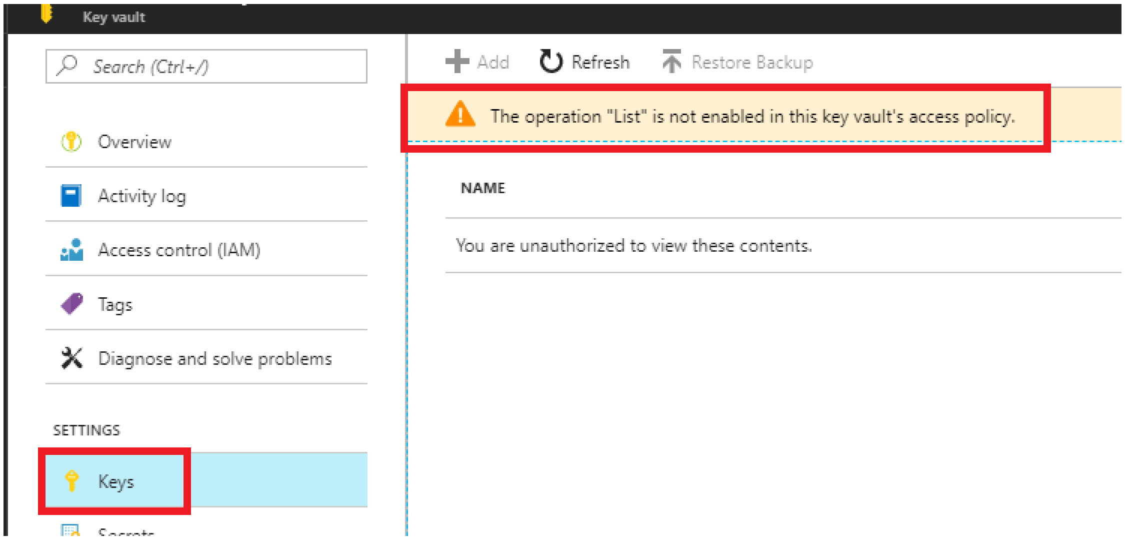
Selected
None

7. Select **OK**.

Task 4: Verify Azure Key Vault permissions

In this task, you will log in with the three different Azure AD user accounts you created previously and observe the impact of the IAM and Access policy permissions you set above.

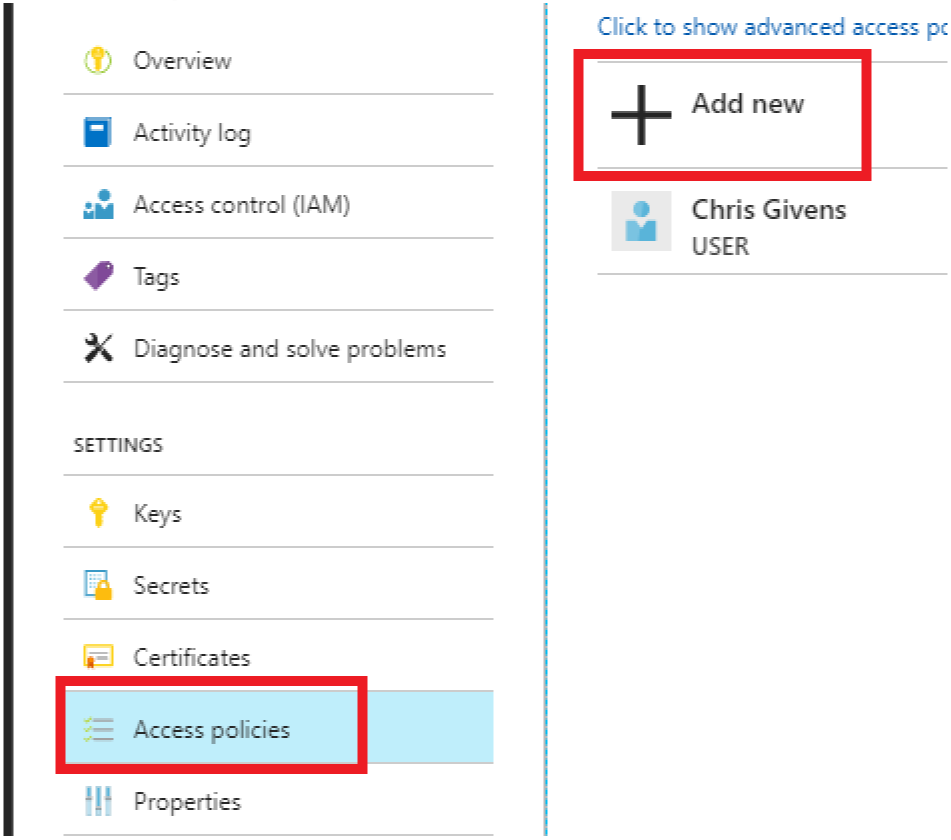
1. Return to your InPrivate or Incognito browser window, and login as the **KeyVaultAdmin**.
2. Search for and select **Key vaults**.
3. You should now see the key vault displayed, select it.
4. Select **Keys**, you should get a warning that the **List** operation is not assigned.
   a. NOTE: IAM permissions are different than Azure Key Vault access policies.



The operation "List" is not enabled in this key vault's access policy.

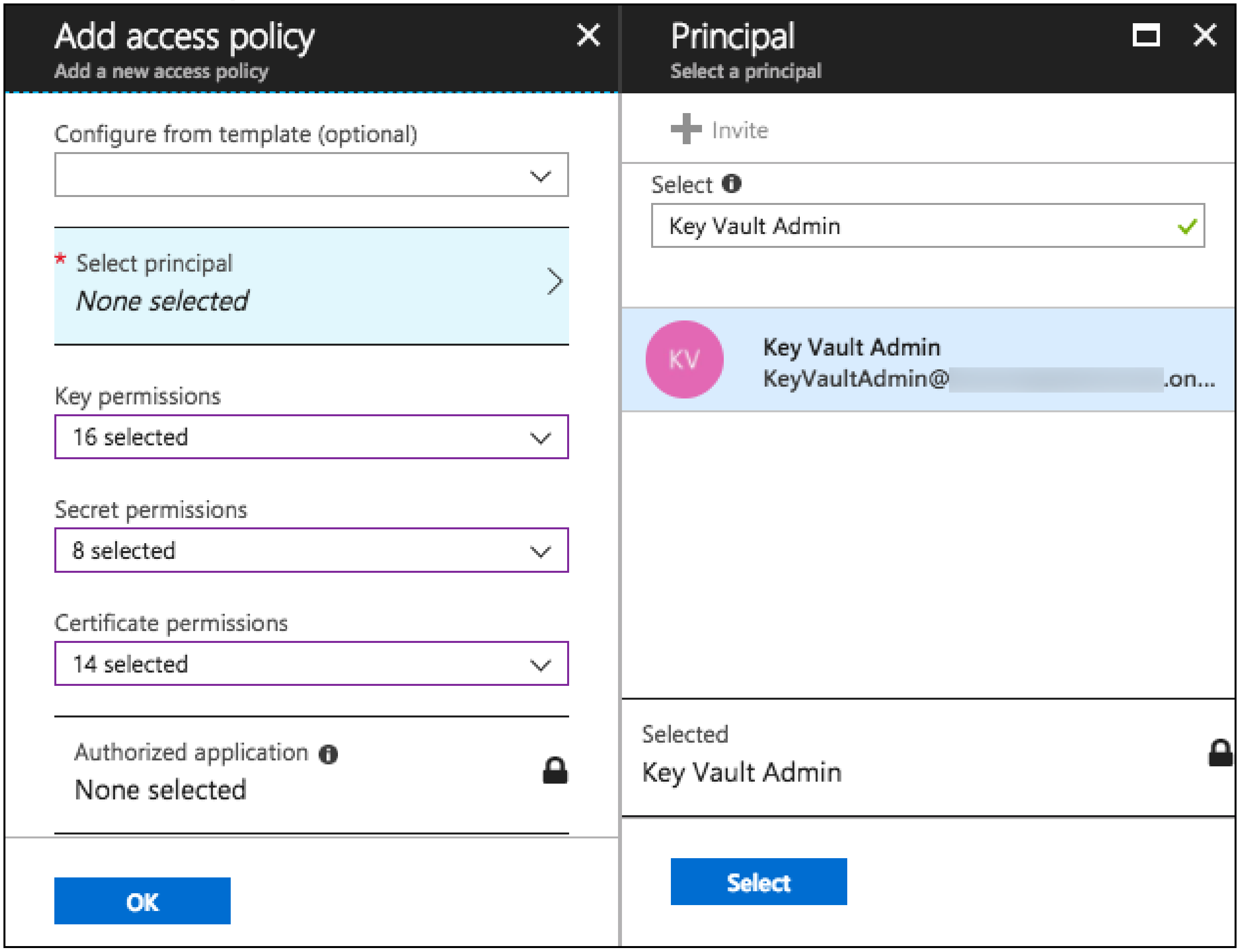
You are unauthorized to view these contents.

5. Select **Access policies**, then select **+Add new**.



6. On the Add access policy blade, enter the following:
   a. Select **Select principal**.
      i. Search for and select **Key Vault Admin**.
      ii. Select **Select**.
   b. For the **Key permissions**, check **Select all**.
   c. For the **Secret permissions**, check **Select all**.

d. For the **Certificate permissions**, check **Select all**.



Add access policy
Add a new access policy

Configure from template (optional)

* Select principal
None selected

Key permissions
16 selected

Secret permissions
8 selected

Certificate permissions
14 selected

Authorized application
None selected

Principal
Select a principal

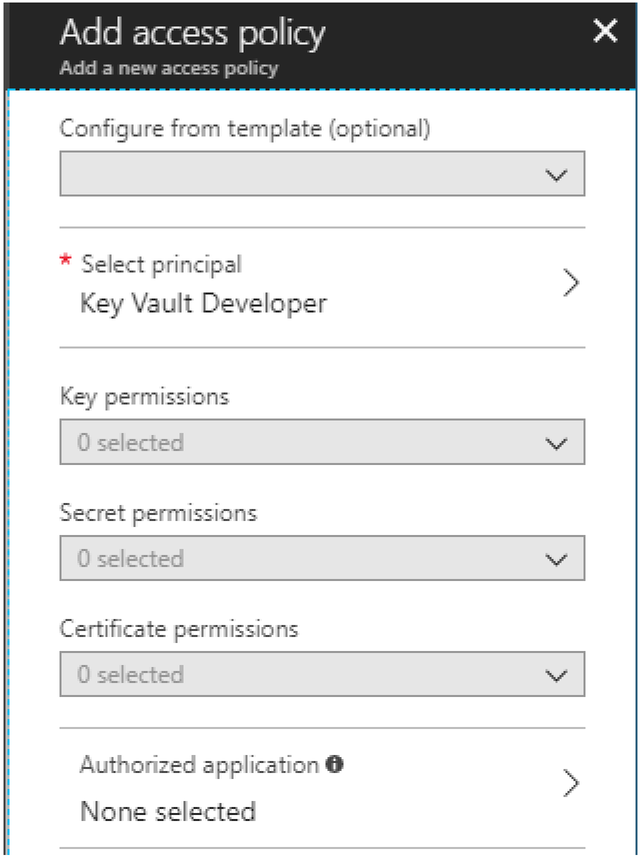
Select
Key Vault Admin ✓

Key Vault Admin
KeyVaultAdmin@[redacted].on...

Selected
Key Vault Admin

7. Select **OK**.
8. Select **Save**.
9. Select **Keys** again from the left-hand menu, and you should now see the error disappear.
10. In your InPrivate or Incognito browser window, logoff and login as the **KeyVaultDeveloper**.
    a. NOTE: You will need to reset the password for the account, as you did in Exercise 1, Task 3, Step 13.
    b. Update the password, when prompted.
11. Search for and select **Key vaults**.
12. You should not be able to see the key vault displayed.
13. Log out.
14. Login as the **KeyVaultAuditor**.
    a. NOTE: You will need to reset the password for the account, as you did in Exercise 1, Task 3, Step 13.
    b. Update the password, when prompted.
15. Search for and select **Key vaults**.
16. You should be able to see the key vault displayed, select it.
17. Select **Keys**, you should not get a warning.
18. Select **Access policies**.
19. Select **+Add new**.
20. Select **Select principal**.
    a. Search for and select **Key Vault Developer**.
    b. Select **Select**.

21. Notice the permission drop downs are greyed out! The Key Vault Auditor only has read permission therefore they cannot assign permissions to any other resources:



Add access policy
Add a new access policy

Configure from template (optional)

* Select principal
Key Vault Developer

Key permissions
0 selected

Secret permissions
0 selected

Certificate permissions
0 selected

Authorized application
None selected

22. Exit the Add access policy blade, discarding any changes.

Exercise 3: Azure deployments using Azure Key Vault

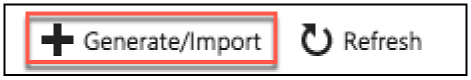
Duration: 45 minutes

Synopsis: In this exercise, attendees will utilize the Microsoft.Compute deployment access that was given in the previous exercise to gain access to an Azure Key Vault secret and certificate without saving them in the template(s).

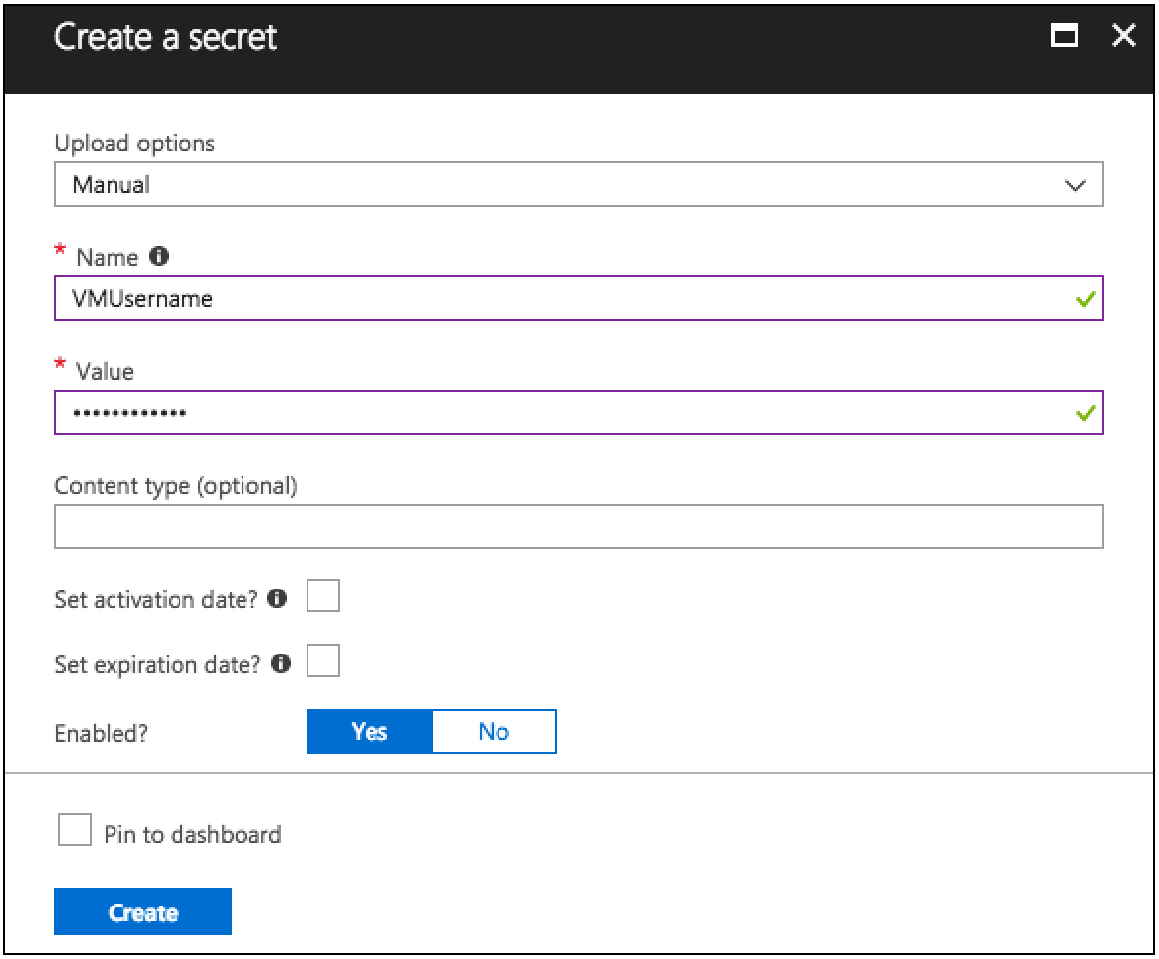
Task 1: Create new secrets

In this task, you will add two secrets to the key vault.

1. In your Incognito browser window, login as the **KeyVaultAdmin**.
2. Select **Key vaults**.
3. Select your key vault.
4. Select **Secrets**.
5. Select **+Generate/Import**.



6. On the Create a secret blade, enter the following:
   a. **Upload options**: Select **Manual**
   b. **Name**: Enter **VMUsername**
   c. **Value**: Enter **AzureKVAdmin**



Create a secret

Upload options: Manual

* Name: VMUsername ✓

* Value: ••••••••••••• ✓

Content type (optional):

Set activation date? □

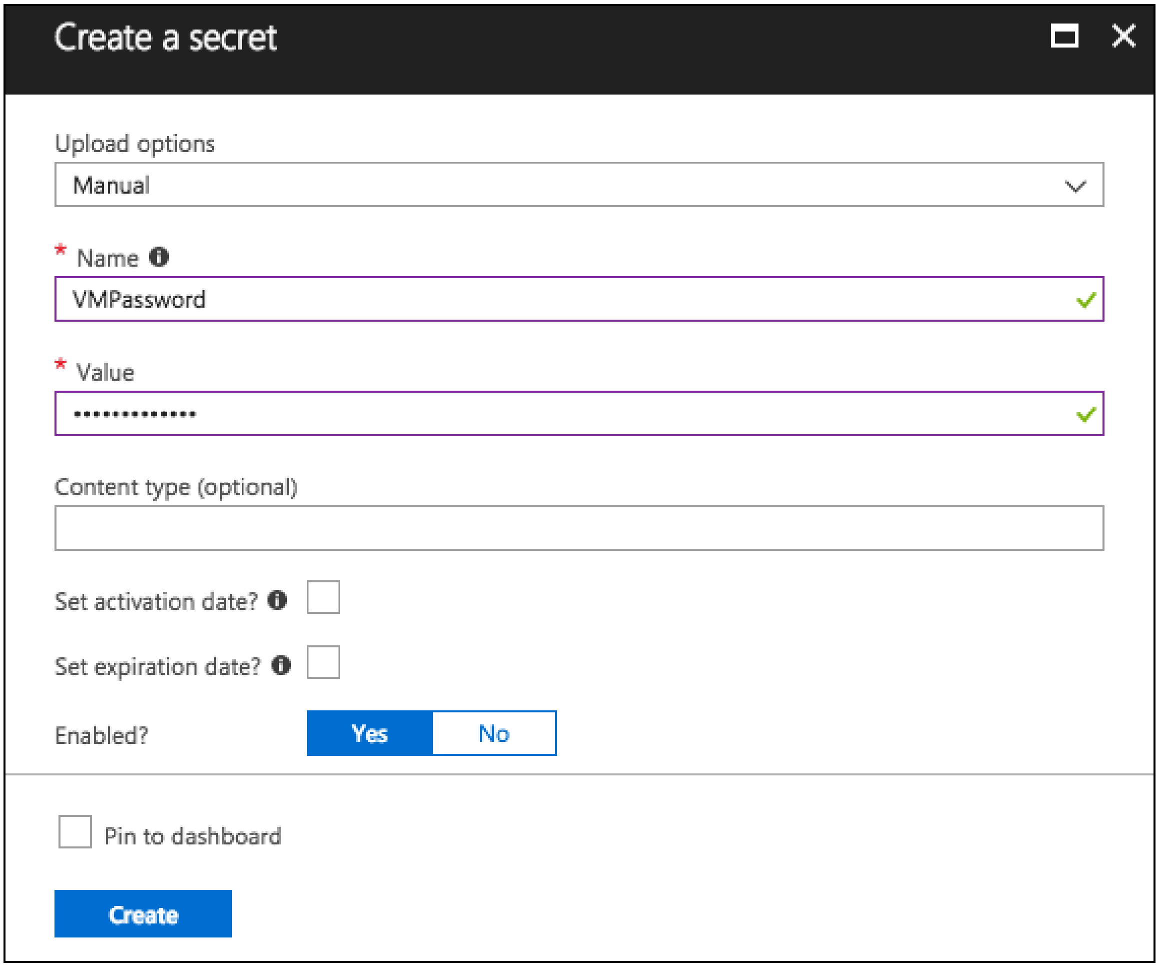
Set expiration date? □

Enabled? [✓ Yes] No

□ Pin to dashboard

7. Select **Create**.

8. Select **+Generate/Import** again.
9. On the Create a secret blade, enter the following:
    a. **Upload options**: Select **Manual**
    b. **Name**: Enter **VMPassword**
    c. **Value**: Enter **DevsC@ntSeeTh**



Create a secret

Upload options: Manual

* Name: VMPassword

* Value: ••••••••••••••

Content type (optional):

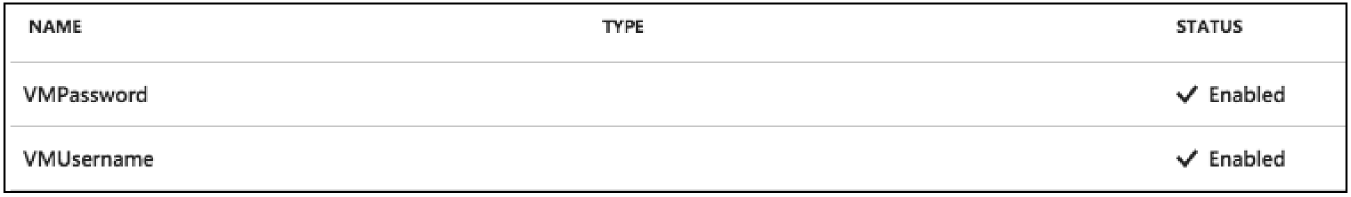
Set activation date? □

Set expiration date? □

Enabled? [✓ Yes] No

□ Pin to dashboard

10. Select **Create**.
11. You should now see two secrets in your Azure Key Vault:



NAME	TYPE	STATUS
VMPassword		✓ Enabled
VMUsername		✓ Enabled

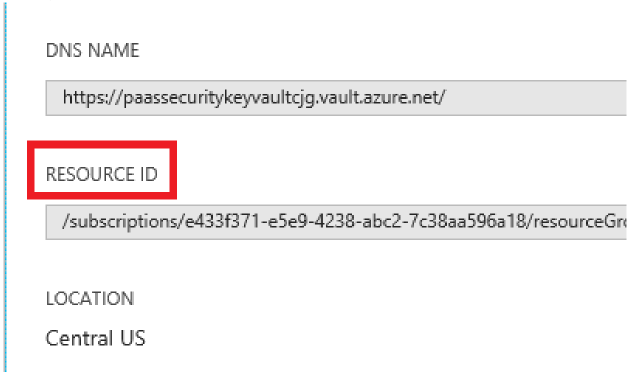
Task 2: Deploy an ARM template using Azure Key Vault resources

In this task, you will run another ARM template using PowerShell to create a SQL database which can use the key vault resources.

1. Open a **Windows PowerShell ISE** window.
2. Open the extracted **\AzureTemplate\deploy-securingpaas.ps1**.
    a. Review the file, note the following:
        i. Logs in the user

    ii. Starts an Azure RM Resource Group Deployment
    iii. Utilizes the azure-kv-sql-deploy.json and azure-kv-parameters.json files
  b. **Update the path to your extracted directory**.
  c. **Update the resource group to your resource group**.
  d. Save the file.

3. Open the extracted **\AzureTemplate\azure-kv-sql-deploy.json** file, review it.
   a. Notice that this file simply creates a virtual machine using the parameters passed in.
   b. **Update the SQL Server name parameter to something unique**.
   c. Save the file.
4. Open the extracted **\AzureTemplate\azure-kv-parameters.json** file.
   a. Notice how it makes a reference to your Azure Key Vault and secret to populate the parameters.
   b. **Update the Azure Key Vault resource id**.
      i. In the Azure portal, select **Key Vaults**.
      ii. Select your key vault.
      iii. Select **Properties**.
      iv. Copy the **RESOURCE ID**.



DNS NAME

https://paassecuritykeyvaultcjg.vault.azure.net/

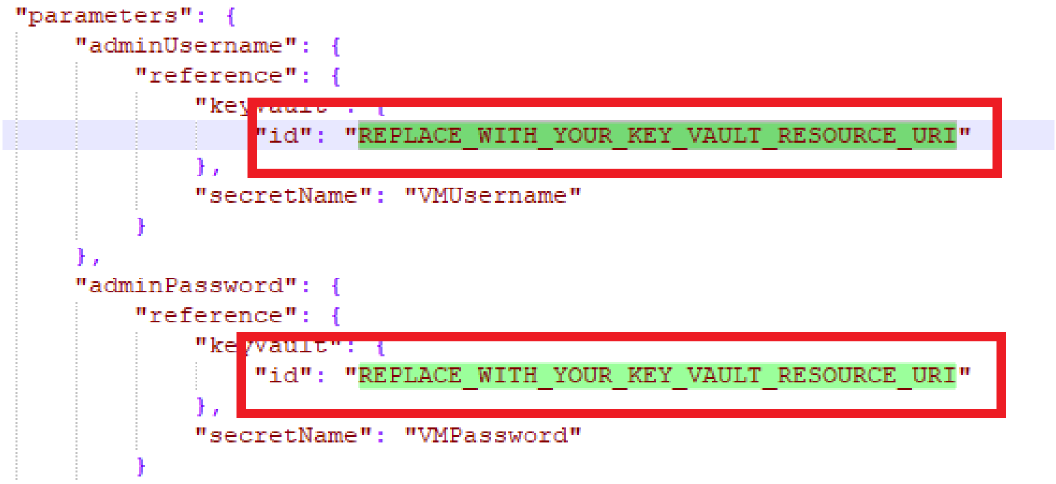
RESOURCE ID

/subscriptions/e433f371-e5e9-4238-abc2-7c38aa596a18/resourceGr

LOCATION

Central US

      v. Paste the **RESOURCE ID** in the parameters sections.



```
"parameters": {
    "adminUsername": {
        "reference": {
            "keyvault": {
                "id": "REPLACE_WITH_YOUR_KEY_VAULT_RESOURCE_URI"
            },
            "secretName": "VMUsername"
        }
    },
    "adminPassword": {
        "reference": {
            "keyvault": {
                "id": "REPLACE_WITH_YOUR_KEY_VAULT_RESOURCE_URI"
            },
            "secretName": "VMPassword"
        }
```

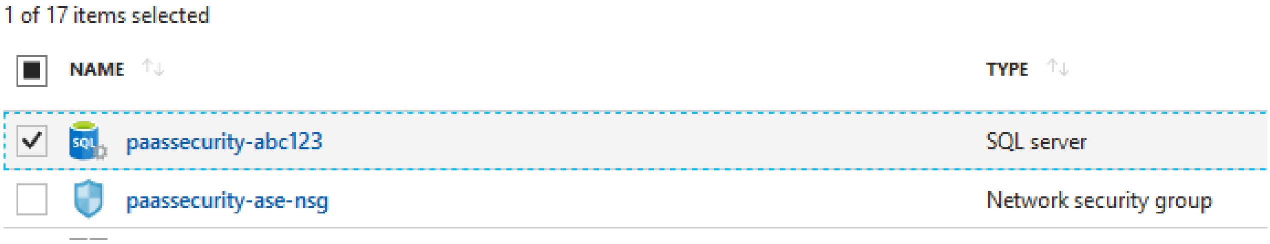
      vi. Save the file.

5. Execute the script in PowerShell by entering the following command: (NOTE: You need to be in the AzureTemplates directory)

```
.\deploy-securingpaas.ps1
```

6. Login as your subscription/resource group admin when prompted.

7. Switch to your Azure Portal, select **SQL Servers**. You should see a new SQL Server available that will be using the username and password values from your key vault:



1 of 17 items selected

■	NAME	TYPE
☑	paassecurity-abc123	SQL server
□	paassecurity-ase-nsg	Network security group

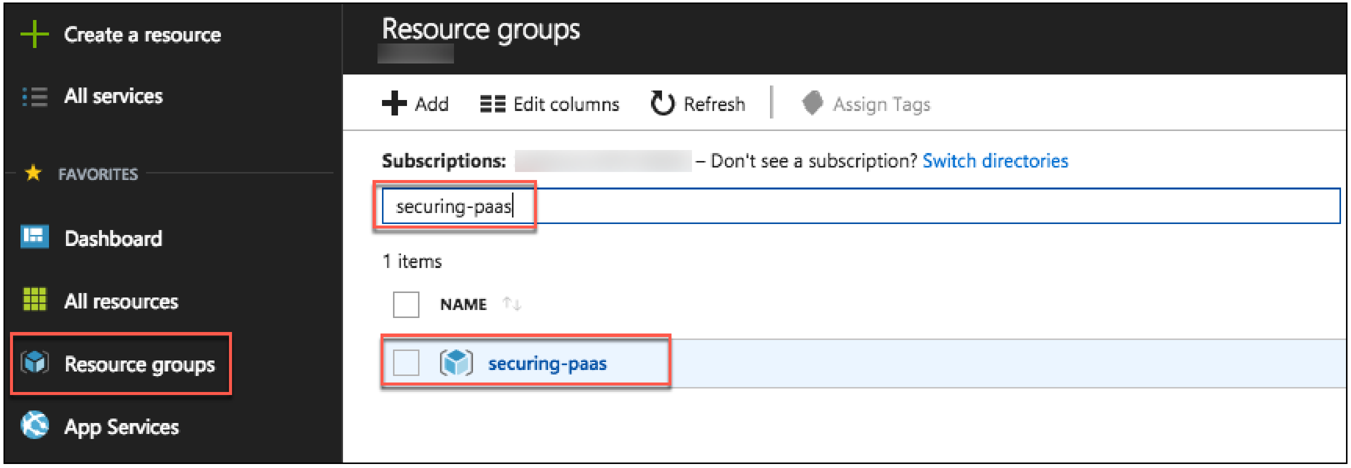
Exercise 4: Securing the web application and database

Duration: 45 minutes

Synopsis: In this exercise, attendees will utilize Azure SQL features to data mask database data and utilize Azure Key Vault to encrypt sensitive columns for users and applications that query the database.

Task 1: Setup the database

1. Return to the Azure portal window where you are logged in with your user account, not the Key Vault account.
2. Navigate to your resource group by selecting **Resource groups**, entering your resource group name in the Filter box, and selecting it from the list.



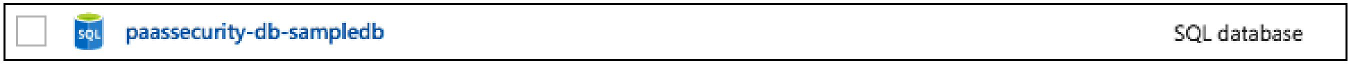
Subscriptions: [redacted] – Don't see a subscription? Switch directories

securing-paas

1 items

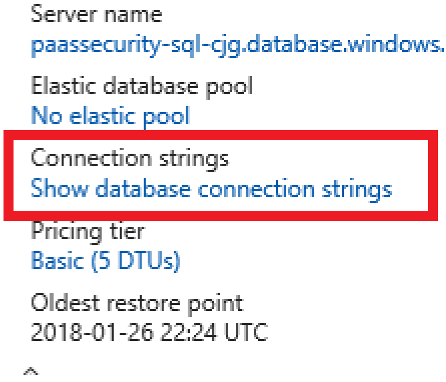
□	NAME
□	securing-paas

3. From the list of resources in your resource group, select the **sampledb** SQL database which was created by the ARM template you ran in the Before the hands-on lab exercise.



□ paassecurity-db-sampledb SQL database

4. In the summary section, select the **Show database connection strings**



Server name
paassecurity-sql-cjg.database.windows.net

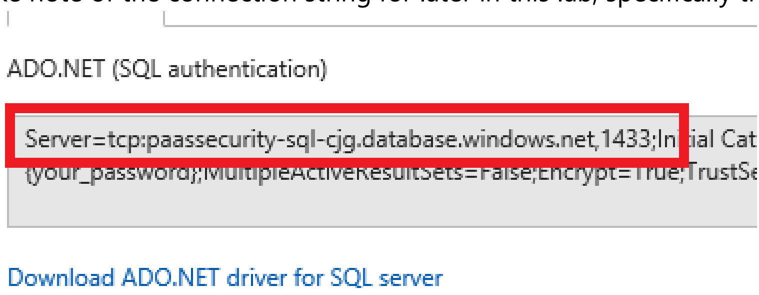
Elastic database pool
No elastic pool

Connection strings
Show database connection strings

Pricing tier
Basic (5 DTUs)

Oldest restore point
2018-01-26 22:24 UTC

5. Take note of the connection string for later in this lab, specifically the **Server** parameter:

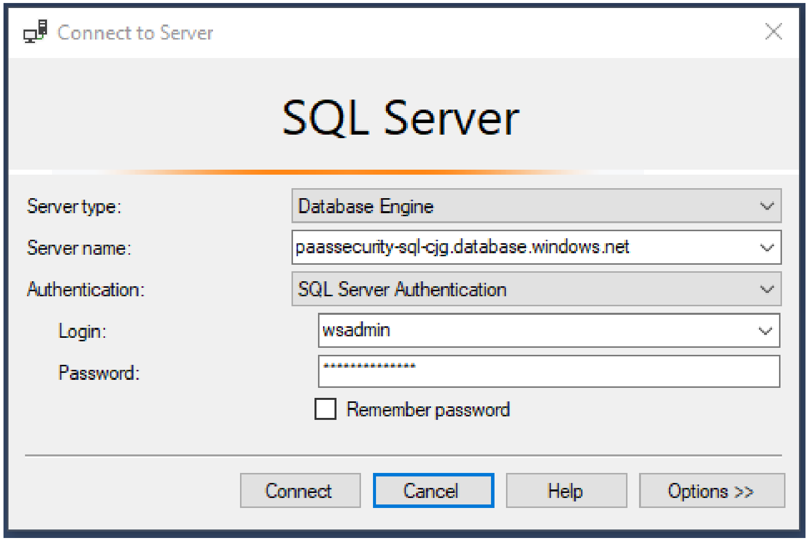


ADO.NET (SQL authentication)

```
Server=tcp:paassecurity-sql-cjg.database.windows.net,1433;Initial Cat
{your_password};MultipleActiveResultSets=False;Encrypt=True;TrustSe
```

Download ADO.NET driver for SQL server

6. Open **SQL Server Management Studio**.
7. In the Connect to Server dialog:
   a. **Server name:** Enter the database **server name** from above
   b. **Authentication**: Select SQL Server Authentication
   c. **Login**: Enter **wsadmin**
   d. **Password**: Enter **p@ssword1rocks**



Connect to Server

SQL Server

Server type: Database Engine

Server name: paassecurity-sql-cjg.database.windows.net

Authentication: SQL Server Authentication

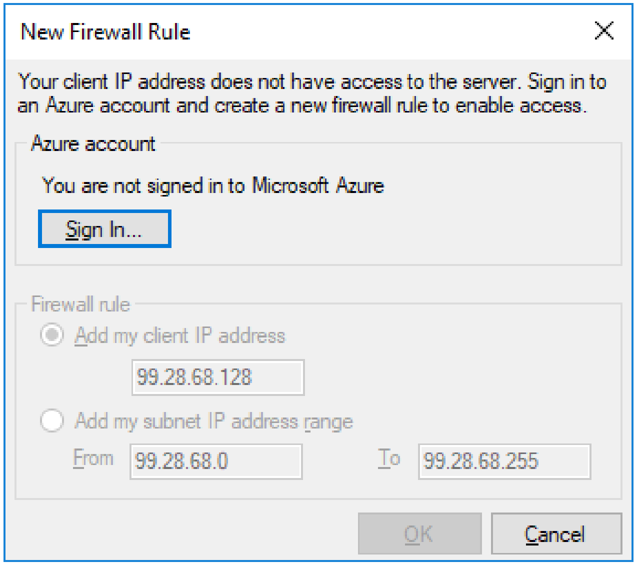
Login: wsadmin

Password: **************

☐ Remember password

   e. Select **Connect**

8. If presented with the New Firewall Rule dialog, select **Sign In**.



New Firewall Rule

Your client IP address does not have access to the server. Sign in to an Azure account and create a new firewall rule to enable access.

Azure account

You are not signed in to Microsoft Azure

Firewall rule

● Add my client IP address

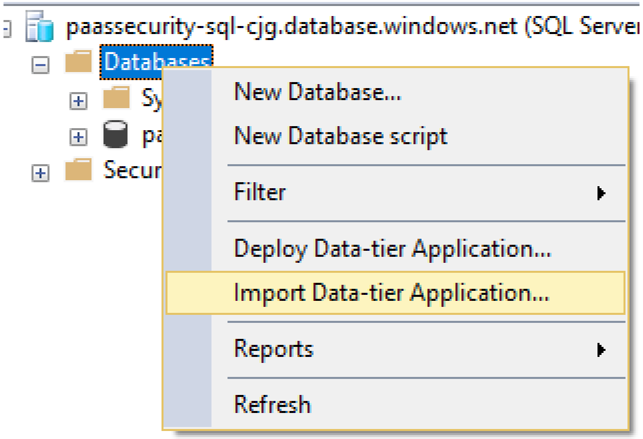
99.28.68.128

○ Add my subnet IP address range

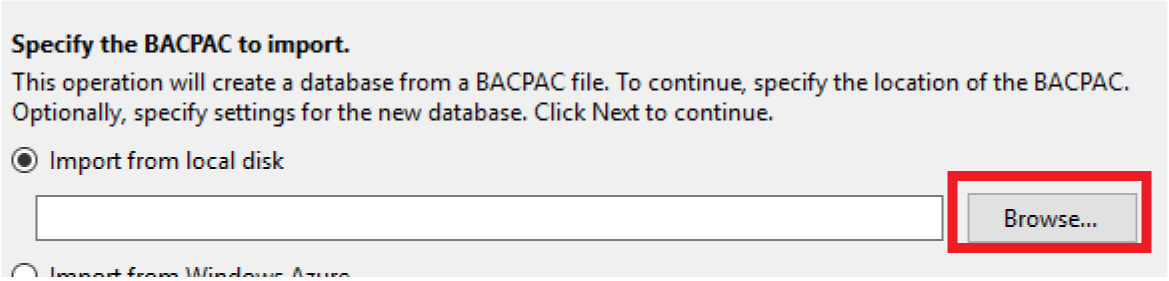
From 99.28.68.0 To 99.28.68.255

9. Sign in as your Azure tenant admin.

10. In the dialog, select **OK**, notice how your IP address will be added for connection.
11. Right-select **Databases**, select **Import Data-tier Application**.


paassecurity-sql-cjg.database.windows.net (SQL Server
Databases
Sy
pa
Secur
New Database...
New Database script
Filter
Deploy Data-tier Application...
Import Data-tier Application...
Reports
Refresh

12. In the Introduction dialog, select **Next**.
13. Select **Browse**.



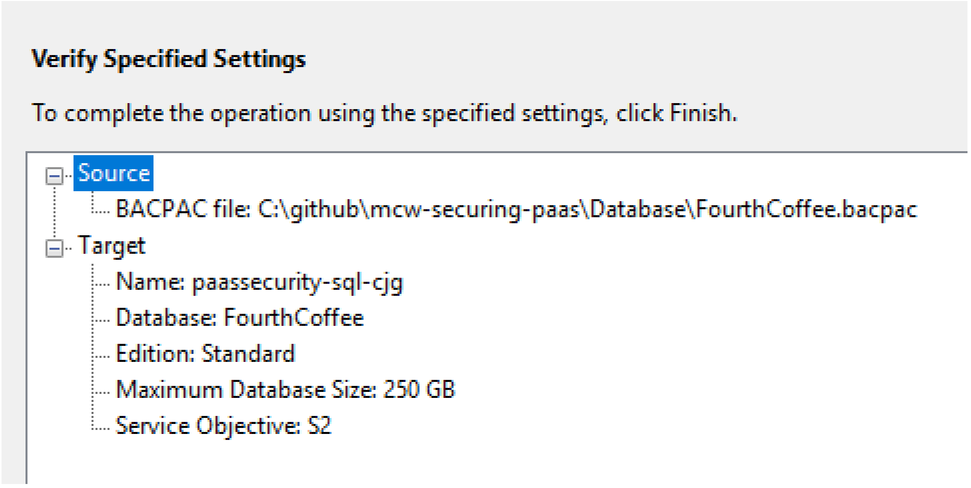
Specify the BACPAC to import.

This operation will create a database from a BACPAC file. To continue, specify the location of the BACPAC. Optionally, specify settings for the new database. Click Next to continue.

● Import from local disk

○ Import from Windows Azure

14. Navigate to the extracted /**Database** directory, select the **FourthCoffee.dacpac** file.
15. Select **Open**.
16. On the **Import Settings** dialog, select **Next**.
17. On the **Database Settings** dialog, select **Next**.
    a. NOTE: If you get an error, close and re-open SSMS and try the import again.

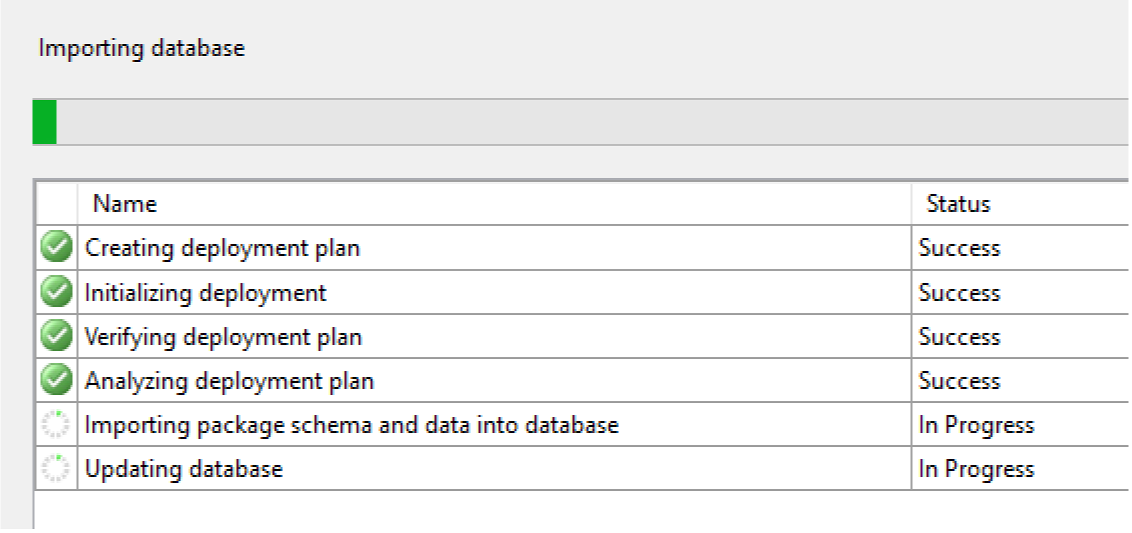


Verify Specified Settings

To complete the operation using the specified settings, click Finish.

- Source
  - BACPAC file: C:\github\mcw-securing-paas\Database\FourthCoffee.bacpac
- Target
  - Name: paassecurity-sql-cjg
  - Database: FourthCoffee
  - Edition: Standard
  - Maximum Database Size: 250 GB
  - Service Objective: S2

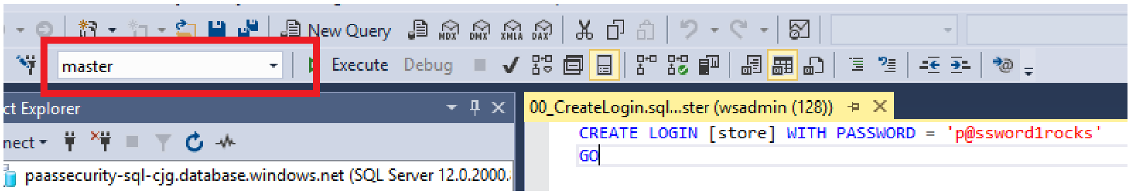
18. Select **Finish**, the database will deploy to Azure.



Importing database

	Name	Status
✓	Creating deployment plan	Success
✓	Initializing deployment	Success
✓	Verifying deployment plan	Success
✓	Analyzing deployment plan	Success
	Importing package schema and data into database	In Progress
	Updating database	In Progress

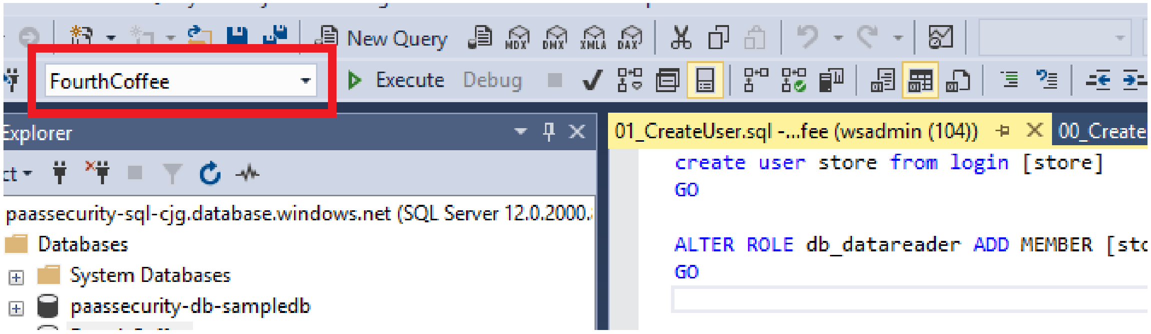
19. Once completed, select **Close**.
20. Ensure that the **master** database is selected.



New Query
master
Execute
Debug
ct Explorer
nect
paassecurity-sql-cjg.database.windows.net (SQL Server 12.0.2000.
00_CreateLogin.sql...ster (wsadmin (128))

```
CREATE LOGIN [store] WITH PASSWORD = 'p@ssword1rocks'
GO
```

21. In **SSMS**, select **File->Open->File**.
22. Browse to the extracted GitHub folder, select the **\Database\00_CreateLogin.ps1** file.
23. **Press F5** to run the script to create a login called **store**.
24. Ensure that the **FourthCoffee** database is selected.
25. Browse to the extracted folder, select the **\Database\01_CreateUser.ps1** file.
26. **Press F5** to run the script to create a non-admin user called **store**.



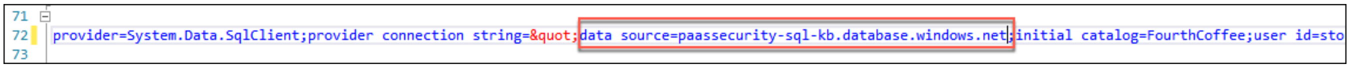
New Query
FourthCoffee
Execute
Debug
Explorer
ct
paassecurity-sql-cjg.database.windows.net (SQL Server 12.0.2000.
Databases
System Databases
paassecurity-db-sampledb
01_CreateUser.sql -...fee (wsadmin (104))
00_Create

```
create user store from login [store]
GO

ALTER ROLE db_datareader ADD MEMBER [stc
GO
```

Task 2: Test the web application solution

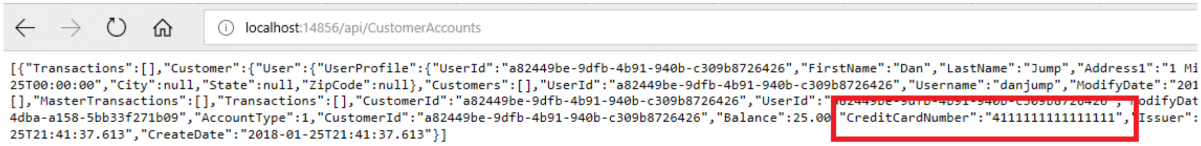
1. In the extracted directory, double-click the **/WebApp/FourthCoffeeAPI/FourthCoffeeAPI.sln** solution file to open the solution in Visual Studio 2017 Community edition.
    a. If prompted in the Visual Studio Version Selector, select Visual Studio 2017 as the program with which to open the solution.
    b. Login to Visual Studio when prompted.
2. In the **Solution Explorer**, navigate to and double select the **web.config** file to open it.
3. In the web.config, locate the database connection string (line 72), and update the "data source" property to point to the **FourthCoffee** database created in Task 2. You should only need to update the server name to point to your

Azure SQL Server.



```
provider=System.Data.SqlClient;provider connection string=&quot;data source=paassecurity-sql-kb.database.windows.net;initial catalog=FourthCoffee;user id=sto
```

4. Save the Web.config file.
5. Run the **FourthCoffeeAPI** solution, press **F5**.
6. In the browser window that opens, browse to [http://localhost:[PORT-NUMBER]/api/CustomerAccounts](http://localhost:[PORT-NUMBER]/api/CustomerAccounts), and you should get a JSON response that shows an unmasked credit card column:

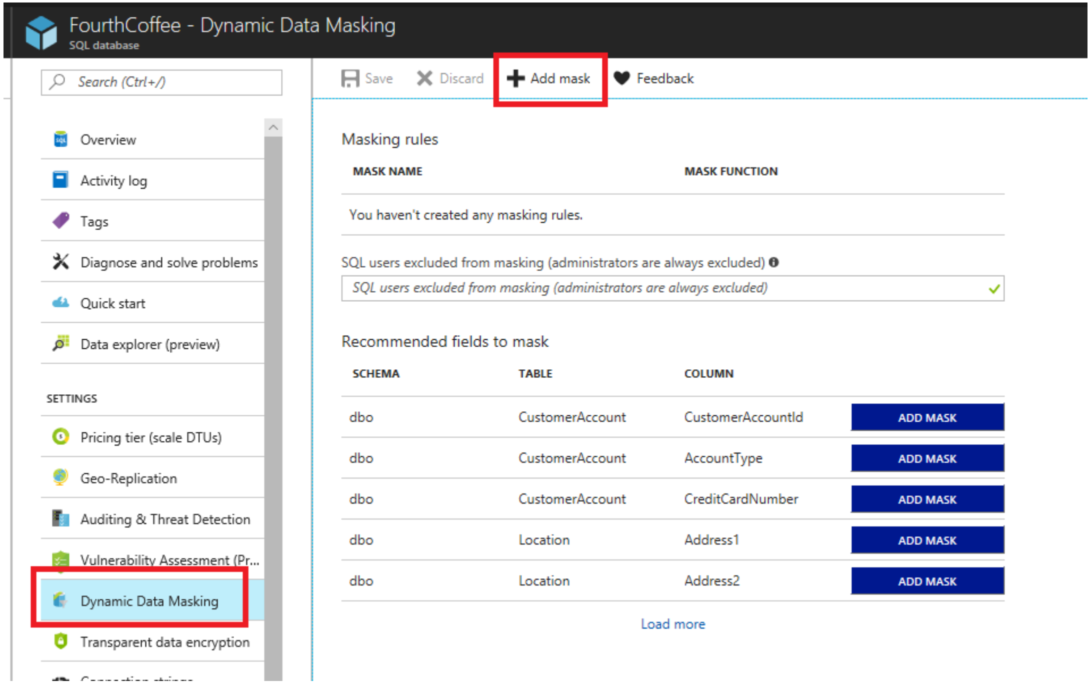


```
[{"Transactions":[],"Customer":{"User":{"UserProfile":{"UserId":"a82449be-9dfb-4b91-940b-c309b8726426","FirstName":"Dan","LastName":"Jump","Address1":"1 Mi
25T00:00:00","City":null,"State":null,"ZipCode":null},"Customers":[],"UserId":"a82449be-9dfb-4b91-940b-c309b8726426","Username":"danjump","ModifyDate":"201
[],"MasterTransactions":[],"Transactions":[],"CustomerId":"a82449be-9dfb-4b91-940b-c309b8726426","UserId":"a82449be-9dfb-4b91-940b-c309b8726426","ModifyDat
4dba-a158-5bb33f271b09","AccountType":1,"CustomerId":"a82449be-9dfb-4b91-940b-c309b8726426","Balance":25.00,"CreditCardNumber":"4111111111111111","Issuer":
25T21:41:37.613","CreateDate":"2018-01-25T21:41:37.613"}]
```

   a. NOTE: depending on your browser, you may need to download to view the JSON response.

Task 3: Utilize data masking

1. Switch to the Azure Portal.
2. Select **SQL databases**.
3. Select the **FourthCoffee** database.
4. In the menu, select **Dynamic Data Masking**, then select **+Add Mask**.



FourthCoffee - Dynamic Data Masking
SQL database
Search (Ctrl+/)
Overview
Activity log
Tags
Diagnose and solve problems
Quick start
Data explorer (preview)
SETTINGS
Pricing tier (scale DTUs)
Geo-Replication
Auditing & Threat Detection
Vulnerability Assessment (Pr...
Dynamic Data Masking
Transparent data encryption
Save
Discard
Add mask
Feedback

Masking rules

MASK NAME	MASK FUNCTION
You haven't created any masking rules.	

SQL users excluded from masking (administrators are always excluded)

SQL users excluded from masking (administrators are always excluded)

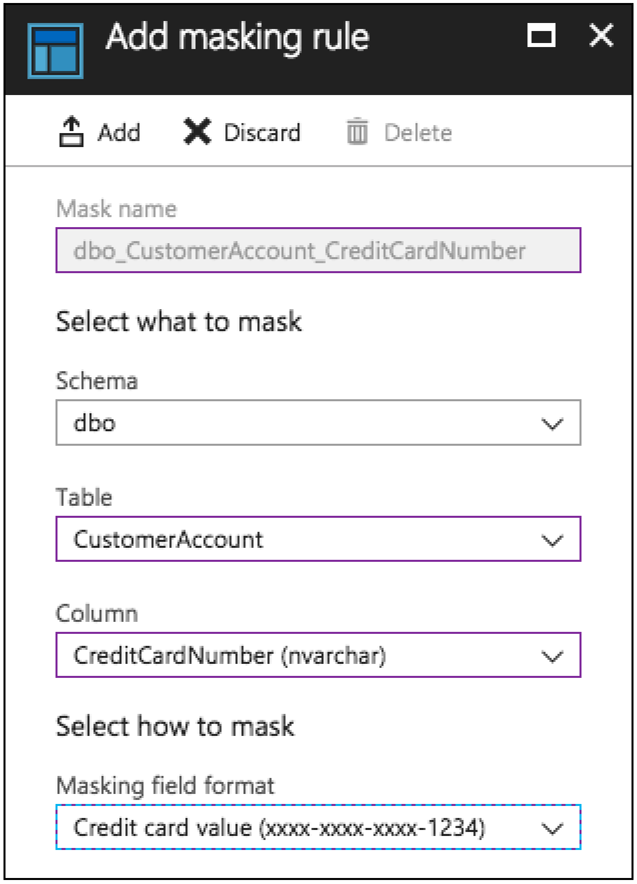
Recommended fields to mask

SCHEMA	TABLE	COLUMN	
dbo	CustomerAccount	CustomerAccountId	ADD MASK
dbo	CustomerAccount	AccountType	ADD MASK
dbo	CustomerAccount	CreditCardNumber	ADD MASK
dbo	Location	Address1	ADD MASK
dbo	Location	Address2	ADD MASK

Load more

5. In the Add masking rule blade, enter the following:
   a. **Schema**: Leave **dbo** selected
   b. **Table**: Select **CustomAccount**
   c. **Column**: Select **CreditCardNumber**

d. **Masking field format**: Select Credit card value (xxxx-xxxx-xxxx-1234)



Add masking rule

Mask name: dbo_CustomerAccount_CreditCardNumber

Select what to mask

Schema: dbo

Table: CustomerAccount

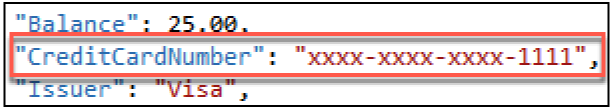
Column: CreditCardNumber (nvarchar)

Select how to mask

Masking field format: Credit card value (xxxx-xxxx-xxxx-1234)

e. Select **Add**.

6. Select **Save**.
7. Switch back to your **FourthCoffeeAPI** solution, refresh the page, you should see the **CreditCardNumber** column is now masked with **xxxx-xxxx-xxxx-1234**.
   a. NOTE: If you do not see this, then you are logged in as a user with dbo privileges



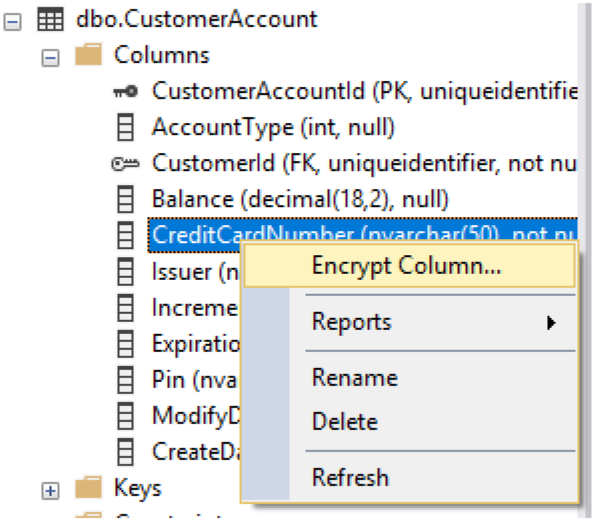
```
"Balance": 25.00,
"CreditCardNumber": "xxxx-xxxx-xxxx-1111",
"Issuer": "Visa",
```

8. Close **Visual Studio**.

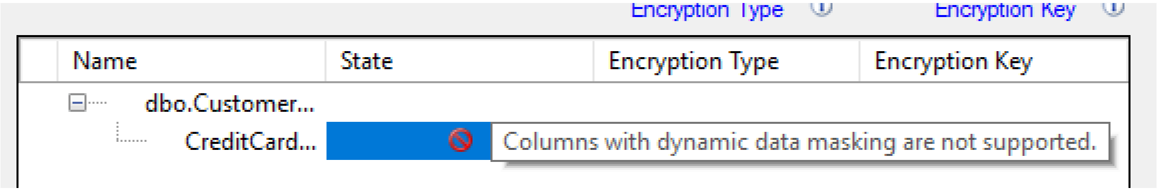
Task 4: Utilize column encryption with Azure Key Vault

1. Switch to **SQL Management Studio**.
2. In the extracted directory, navigate to the **Database** directory.
3. Open the **02_PermissionSetup.sql** file, copy and paste the TSQL to the Query Window.
4. Switch to the **FourthCoffee** database, execute the SQL statement.
5. In the **Object Explorer**, expand the **FourthCoffee** node.
6. Expand the **Tables** node.
7. Expand the **CustomerAccount** table node.
8. Expand the **Columns** node.

9. Right-click the **CreditCardNumber** column, select **Encrypt Column**.

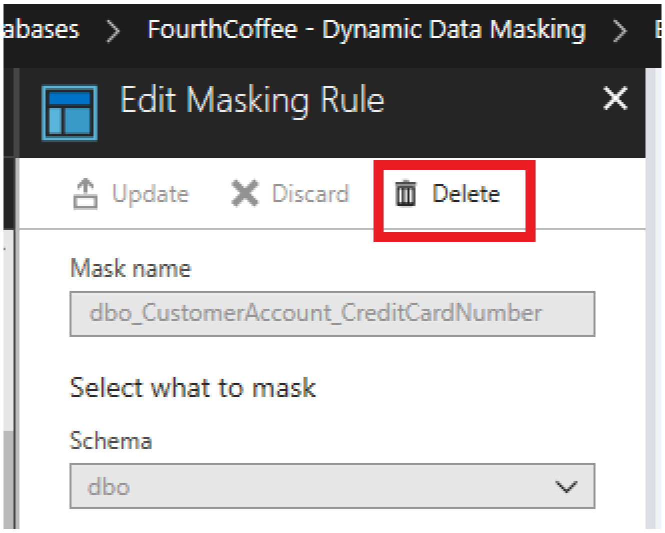

dbo.CustomerAccount
Columns
CustomerAccountId (PK, uniqueidentifie
AccountType (int, null)
CustomerId (FK, uniqueidentifier, not nu
Balance (decimal(18,2), null)
CreditCardNumber (nvarchar(50), not nu
Issuer (n
Increme
Expiratio
Pin (nva
ModifyD
CreateD
Keys
Encrypt Column...
Reports
Rename
Delete
Refresh

10. Select **Next** on the intro screen.
11. Notice that the State of the column is such that you cannot add encryption (data masking):



Name	State	Encryption Type	Encryption Key
dbo.Customer...			
CreditCard...	Columns with dynamic data masking are not supported.		

12. Select **Cancel**, then **Yes** to confirm.
13. Switch back to the Azure Portal, select the CustomerAccount.CreditCardNumber data masking.
14. Select **Delete**.



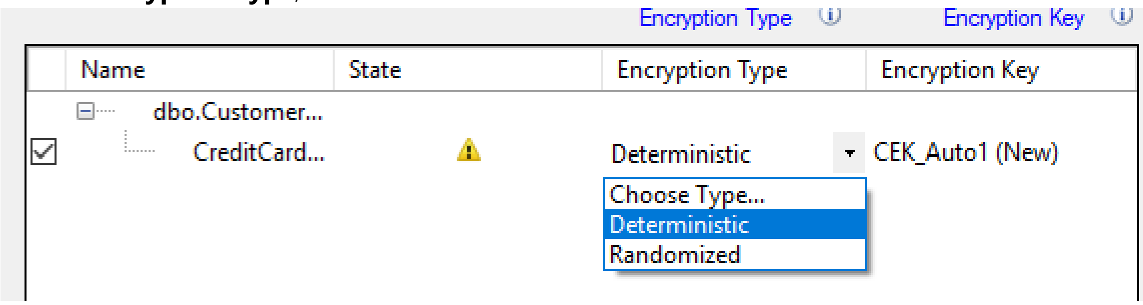
Mask name: dbo_CustomerAccount_CreditCardNumber

Select what to mask

Schema: dbo

15. Select **Save**.
16. Switch back to **SQL Management Studio**.
17. Right-click the **CreditCardNumber** column, select **Encrypt Column**.
18. Check the checkbox next to the **CreditCardNumber** column.

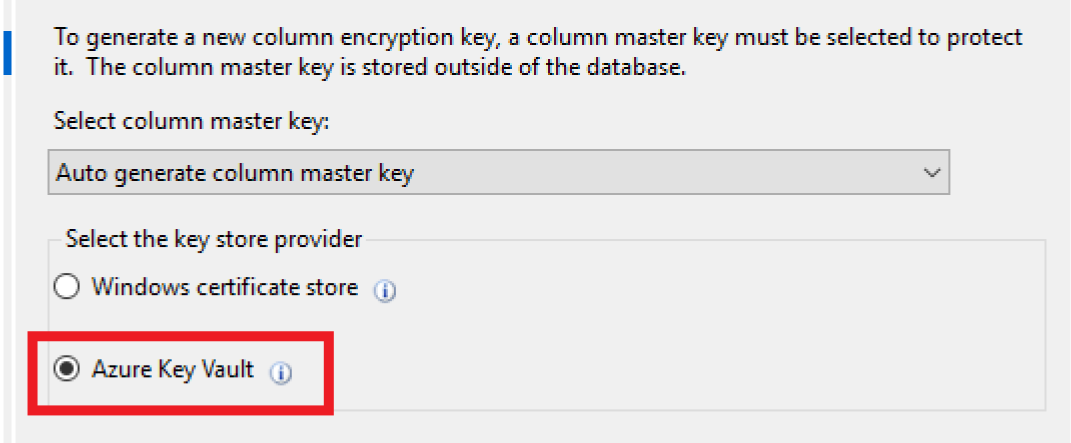
19. For the **Encryption Type**, select **Deterministic**.



Encryption Type ⓘ Encryption Key ⓘ

	Name	State	Encryption Type	Encryption Key
	⊟ dbo.Customer...			
☑	CreditCard...	⚠	Deterministic	CEK_Auto1 (New)

20. Select **Next**.
21. For the encryption, select **Azure Key Vault**, in the dialog.



To generate a new column encryption key, a column master key must be selected to protect it. The column master key is stored outside of the database.

Select column master key:

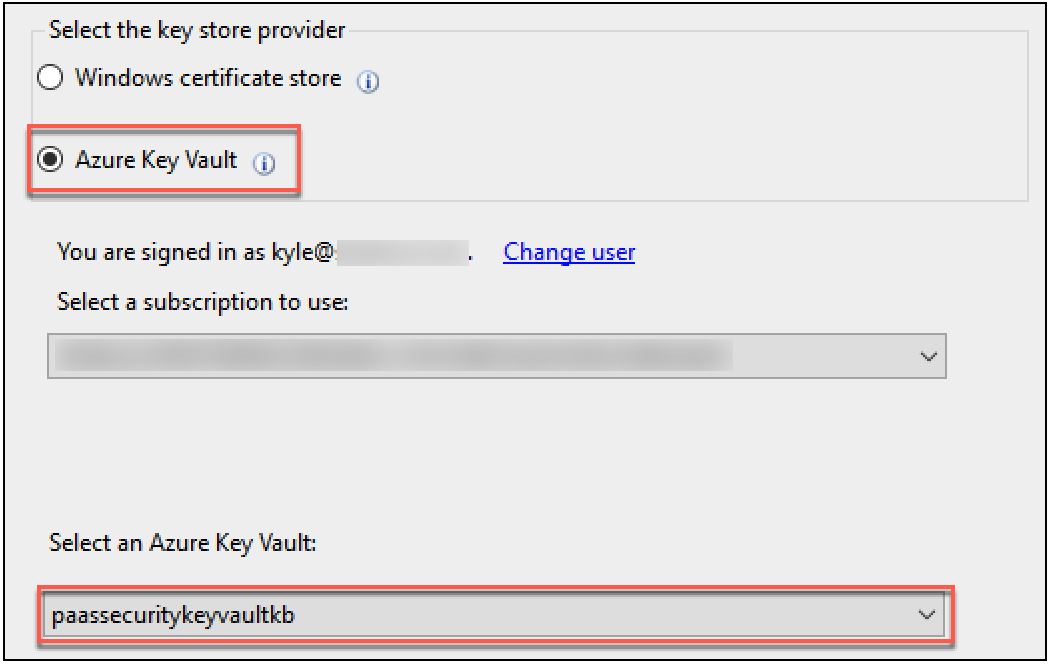
Auto generate column master key

Select the key store provider

○ Windows certificate store ⓘ

● Azure Key Vault ⓘ

22. Select **Sign In**.
23. Sign in with your Azure Portal credentials.
24. Select your Azure Key Vault.



Select the key store provider

○ Windows certificate store ⓘ

● Azure Key Vault ⓘ

You are signed in as kyle@[redacted]. Change user

Select a subscription to use:

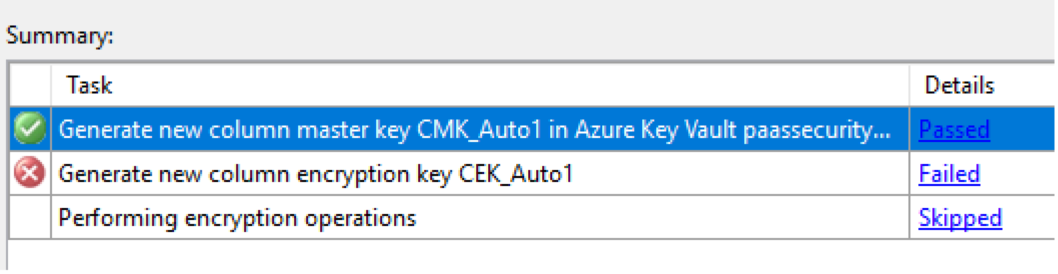
[redacted]

Select an Azure Key Vault:

paassecuritykeyvaultkb

25. Select **Next**.
26. On the **Run Settings**, leave **Proceed to finish now** selected, and select **Next**.
27. Select **Finish**, the configured will start. If prompted, login using your Azure Portal credentials.

a. NOTE: You may receive a "wrapKey" error, if so, ensure that your account has been assigned that permissions in the Azure Key Vault.



Summary:

	Task	Details
✓	Generate new column master key CMK_Auto1 in Azure Key Vault paassecurity...	Passed
✗	Generate new column encryption key CEK_Auto1	Failed
	Performing encryption operations	Skipped

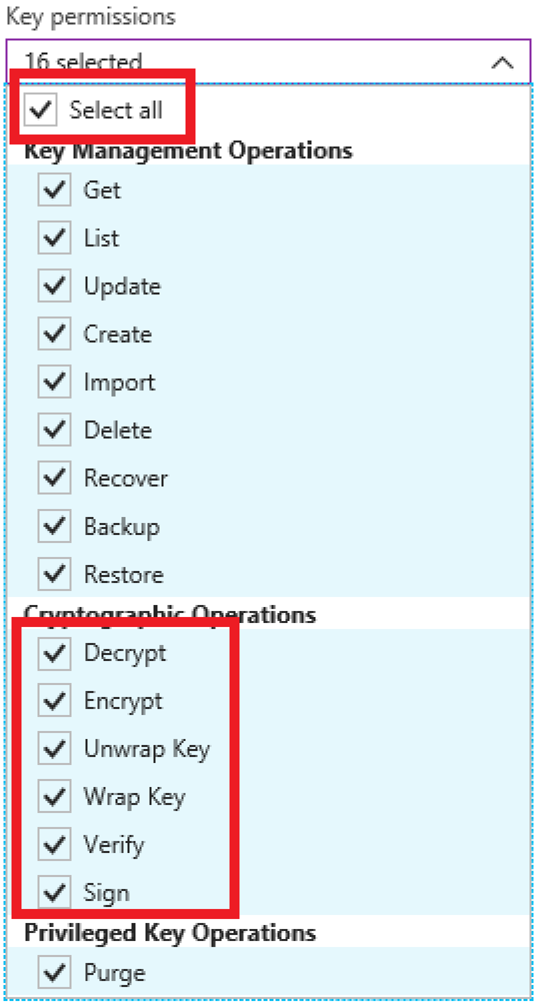
i. Select **Key vault**.

ii. Select your key vault.

iii. Select **Access policies**.

iv. Select your account.

v. Select **Key permissions**, select **Select all**.



Key permissions

16 selected

☑ Select all

Key Management Operations

☑ Get

☑ List

☑ Update

☑ Create

☑ Import

☑ Delete

☑ Recover

☑ Backup

☑ Restore

Cryptographic Operations

☑ Decrypt

☑ Encrypt

☑ Unwrap Key

☑ Wrap Key

☑ Verify

☑ Sign

Privileged Key Operations

☑ Purge

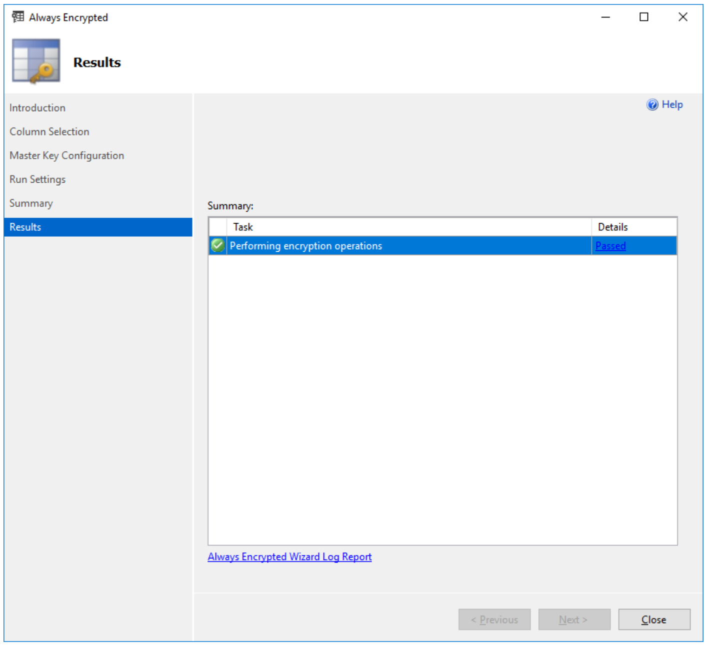
vi. Select **Secret permissions**, select **Select all**.

vii. Select **Certificate permissions**, select **Select all**.

viii. Select **OK**.

ix. Select **Save**.

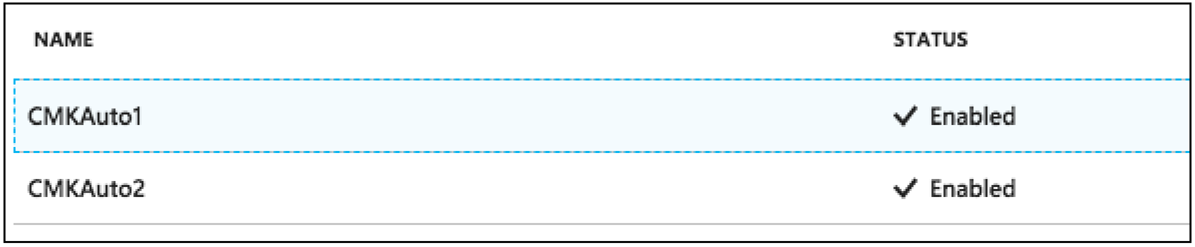
x. Retry the operation.



Summary:

Task	Details
✓ Performing encryption operations	Passed

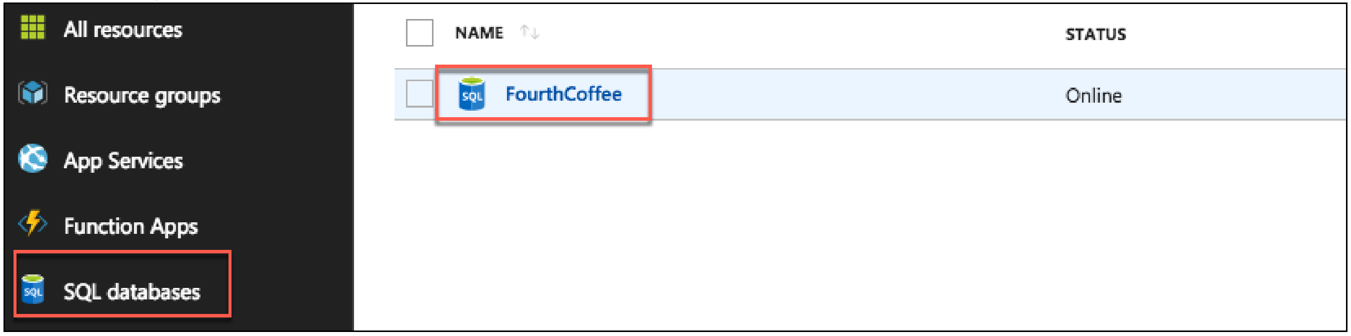
28. Select **Close**
29. Right-click the **CustomerAccount** table, select **Select top 1000 rows**.
30. You will notice the **CreditCardNumber** column is encrypted based on the new Azure Key Vault key.
31. Switch to the Azure Portal.
32. Select **Key Vaults**.
33. Select your Azure Key Vault, then select **Keys**. You should see the key created from the SQL Management Studio displayed:



NAME	STATUS
CMKAuto1	✓ Enabled
CMKAuto2	✓ Enabled

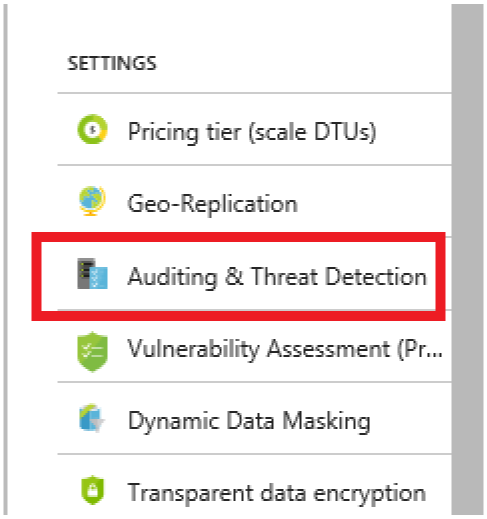
Task 5: Enable Azure SQL Auditing & Threat Detection

1. In the Azure portal, select **SQL Databases**, and select the **FourthCoffee** database.

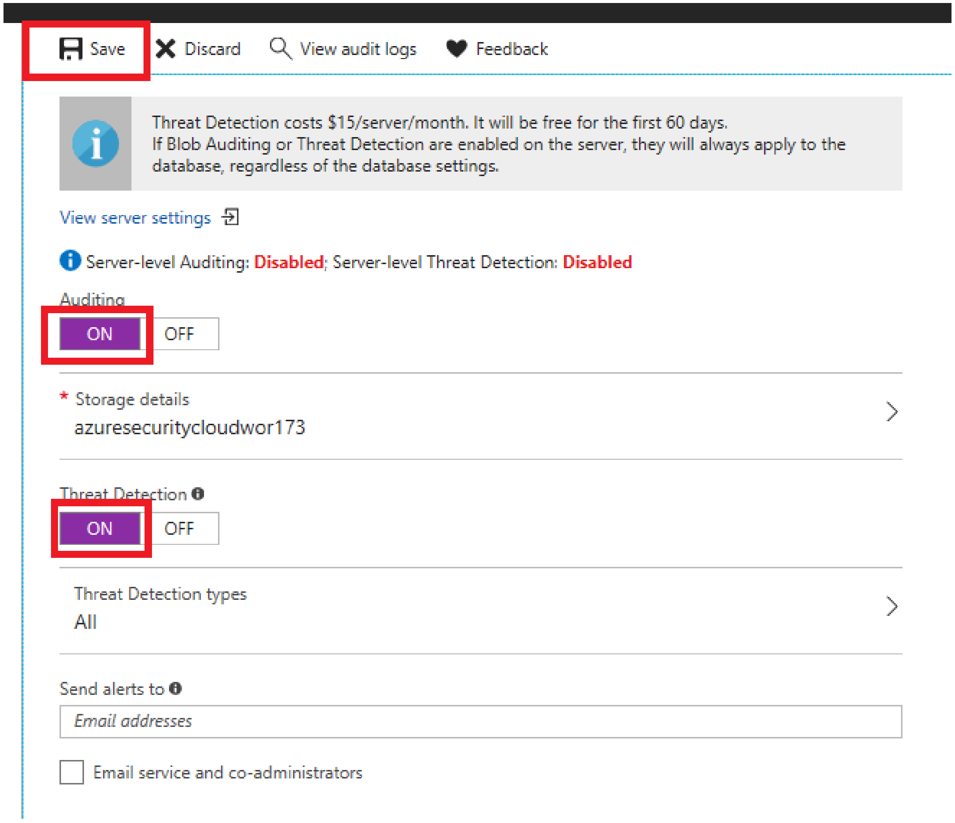


□	NAME	STATUS
□	FourthCoffee	Online

2. Select **Auditing & Threat Detection**.


SETTINGS
Pricing tier (scale DTUs)
Geo-Replication
Auditing & Threat Detection
Vulnerability Assessment (Pr...
Dynamic Data Masking
Transparent data encryption

3. For Auditing, toggle to **ON**.
4. Select **Storage details**.
5. Select **Storage account**, select your storage account.
6. Select **OK**.
7. For Threat Detection, toggle to **ON**.



Threat Detection costs $15/server/month. It will be free for the first 60 days.
If Blob Auditing or Threat Detection are enabled on the server, they will always apply to the database, regardless of the database settings.

View server settings

Server-level Auditing: Disabled; Server-level Threat Detection: Disabled

Auditing
ON OFF

* Storage details
azuresecuritycloudwor173

Threat Detection
ON OFF

Threat Detection types
All

Send alerts to
Email addresses

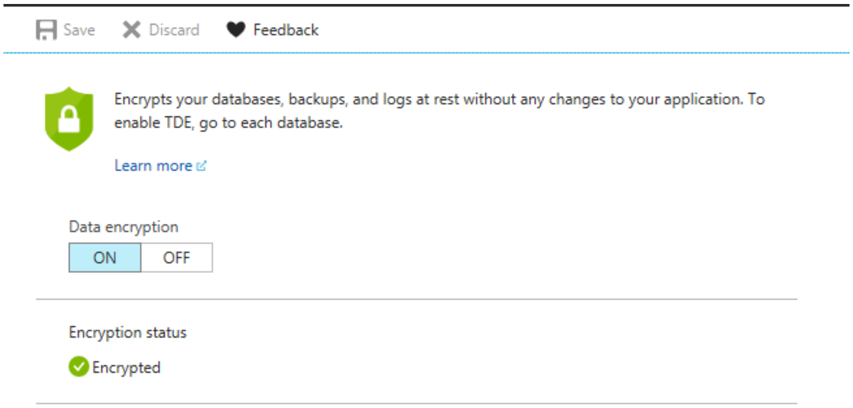
□ Email service and co-administrators

8. Enter your email address.
9. Select **Save**.

Task 6: Ensure SQL Azure Transparent Data Encryption (TDE) is enabled

1. Select **Transparent data encryption**.
2. For data encryption, ensure that the toggle is set to **ON**.

- NOTE: For newly created databases, this is automatically enabled.

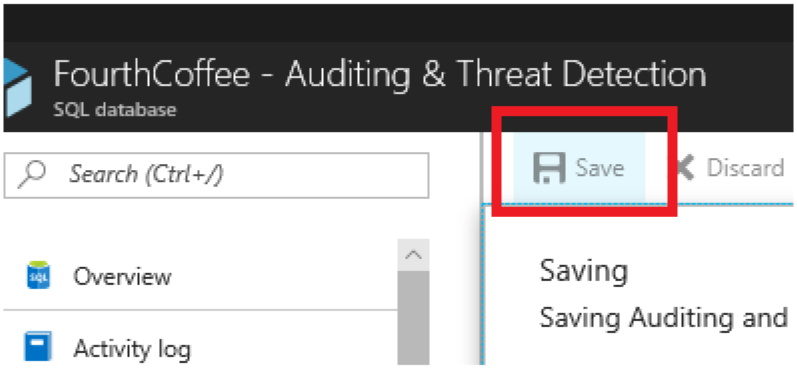


Encrypts your databases, backups, and logs at rest without any changes to your application. To enable TDE, go to each database.

Encryption status

Encrypted

3. Select **Save**.



FourthCoffee - Auditing & Threat Detection

SQL database

Saving

Saving Auditing and

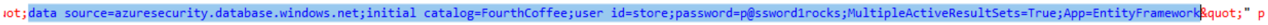
Exercise 5: Migrating web.config settings to Azure Key Vault

Duration: 30 minutes

Synopsis: In this exercise, attendees will learn how to migrate web application to utilize Azure Key Vault rather than storing valuable credentials (such as connection strings) in application configuration files.

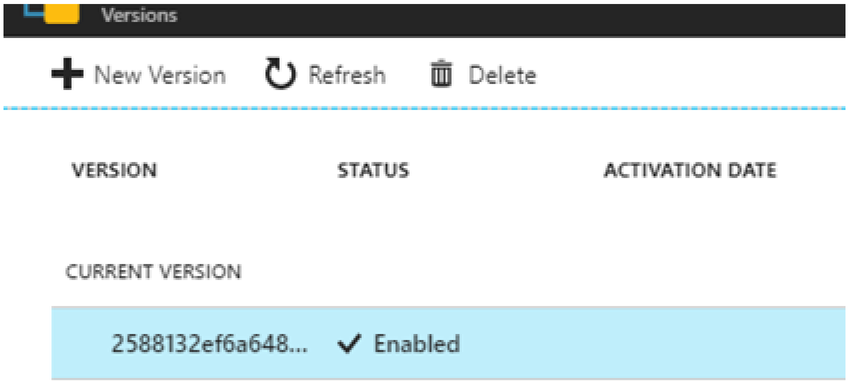
Task 1: Create an Azure Key Vault secret

1. From the extracted GitHub directory, open the **\WebApp\FourthCoffeeAPI_KeyVault\FourthCoffeeAPI.sln** solution.
2. Switch to your Azure Portal.
3. Select **Key Vaults**, then select your Azure Key Vault.
4. Select **Secrets**, then select **+Add**.
5. For the **Upload Options**, select **Manual**.
6. For the **Name**, enter **FourthCoffeeAPI**.
7. For the **Value,** copy the connection string information from the FourthCoffeeAPI solution web.config file on line 77:



```
ıot;data source=azuresecurity.database.windows.net;initial catalog=FourthCoffee;user id=store;password=p@ssword1rocks;MultipleActiveResultSets=True;App=EntityFramework&quot;" p
```

8. Select **Create**.
9. Select **Secrets**.
10. Select **FourthCoffeeAPI**.
11. Select the current version.



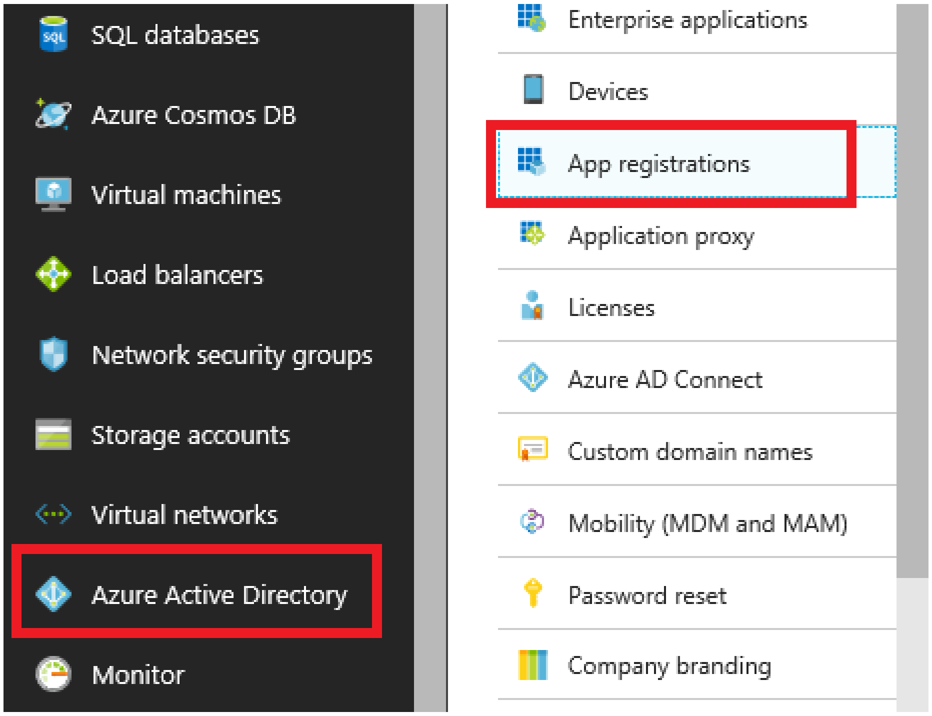
Versions

VERSION	STATUS	ACTIVATION DATE
CURRENT VERSION		
2588132ef6a648...	✓ Enabled	

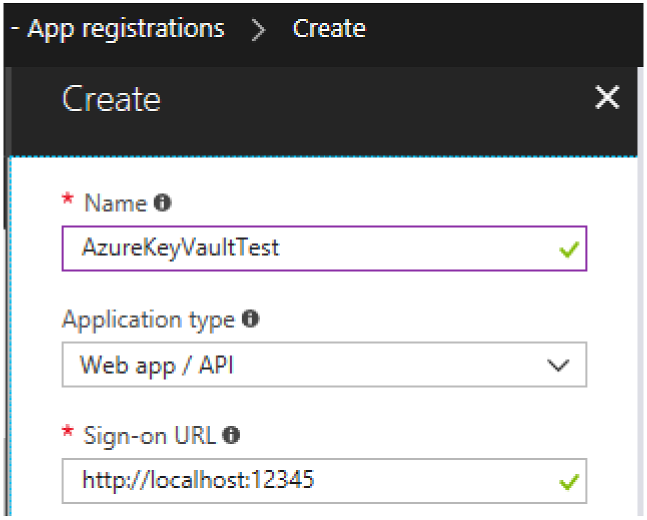
12. Copy and record the secret identifier URL for later use.

Task 2: Create an Azure Active Directory application

1. Select **Azure Active Directory**, then select **App Registrations**.



2. Select **+New application registration**.
3. For the name, enter **AzureKeyVaultTest**.
4. For the Sign-on URL, enter http://localhost:12345



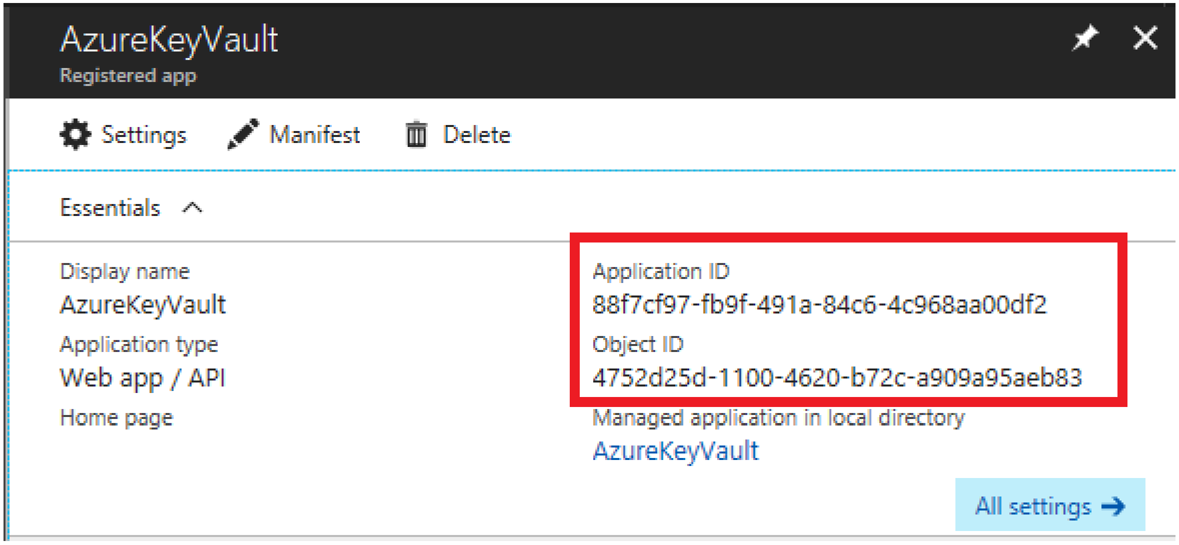
* Name: AzureKeyVaultTest ✓

Application type: Web app / API

* Sign-on URL: http://localhost:12345 ✓

5. Select **Create**.
6. Select the new **AzureKeyVaultTest** application.
7. Copy and record the **Application ID** for later use.

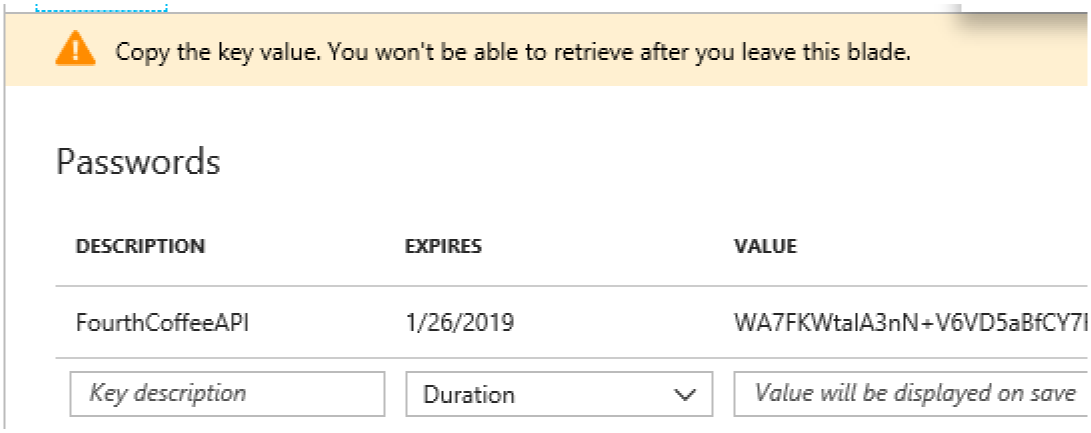
8. Copy and record the **Object ID** for later use.



Display name
AzureKeyVault
Application type
Web app / API
Home page

Application ID
88f7cf97-fb9f-491a-84c6-4c968aa00df2
Object ID
4752d25d-1100-4620-b72c-a909a95aeb83
Managed application in local directory
AzureKeyVault

9. Select **Settings**.
10. Select **Keys**.
11. For Description, enter **FourthCoffeeAPI**.
12. For Expires, select **In 1 year**.
13. Select **Save**.



Copy the key value. You won't be able to retrieve after you leave this blade.

Passwords

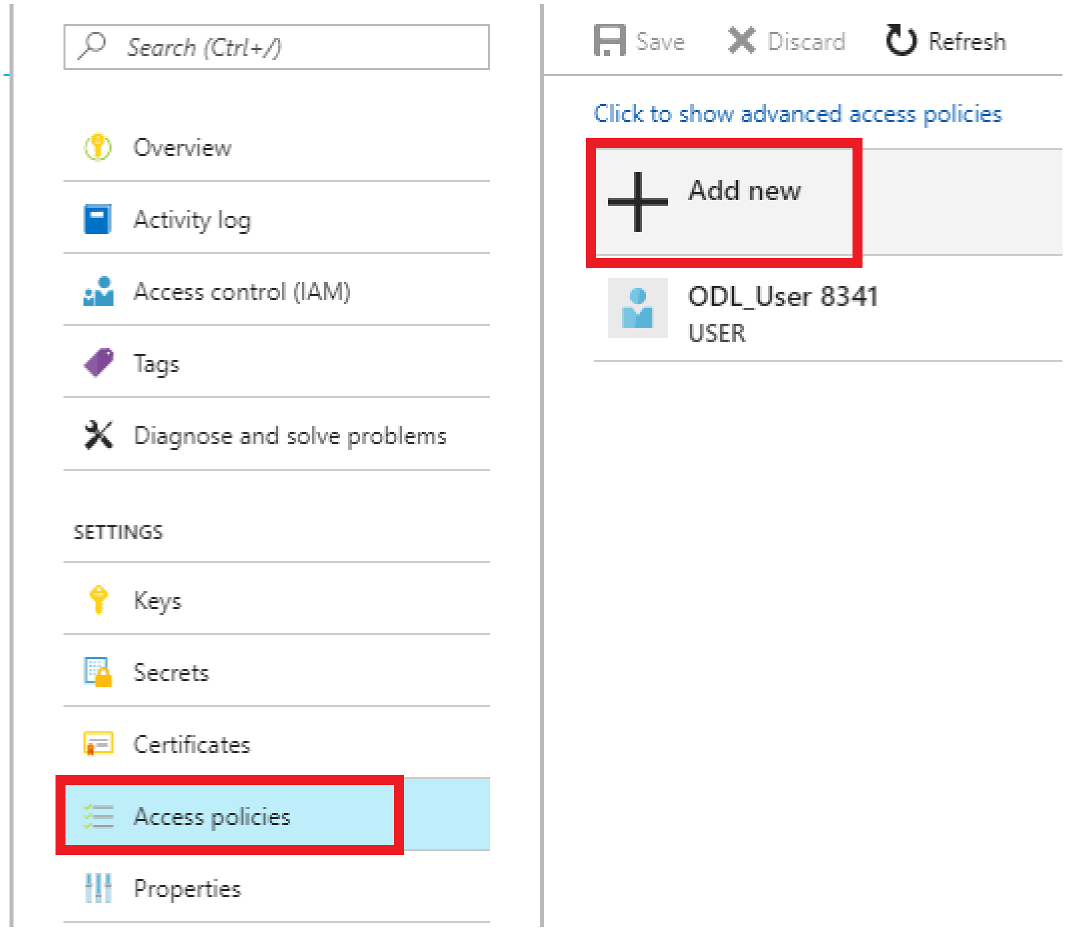
DESCRIPTION	EXPIRES	VALUE
FourthCoffeeAPI	1/26/2019	WA7FKWtalA3nN+V6VD5aBfCY7I
Key description	Duration	*Value will be displayed on save*

14. Copy and record the key value for later use.

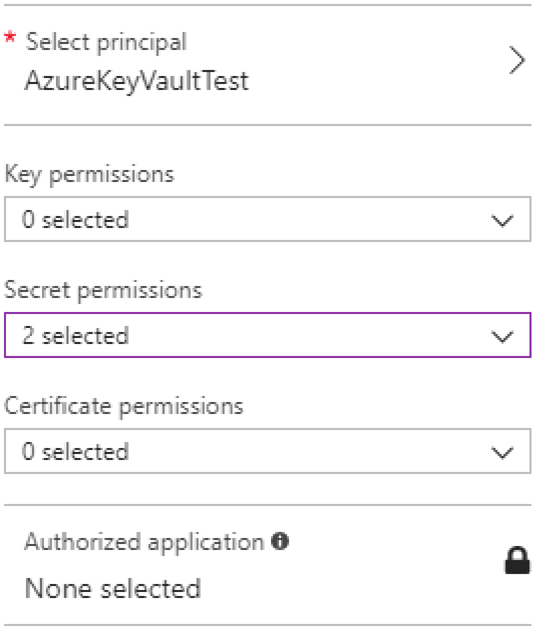
Task 3: Assign the new Application Azure Key Vault permissions

1. Switch back to Azure Portal and select your Azure Key Vault.
2. Select **Access Policies**.

3. Select **+Add New**.



4. Select **Select principal**, enter **AzureKeyVaultTest**.
5. Select the application service principal, click **Select**.
6. Select the **Secret permissions** drop down, check the **Get** and **List** permissions.


* Select principal: AzureKeyVaultTest

Key permissions: 0 selected

Secret permissions: 2 selected

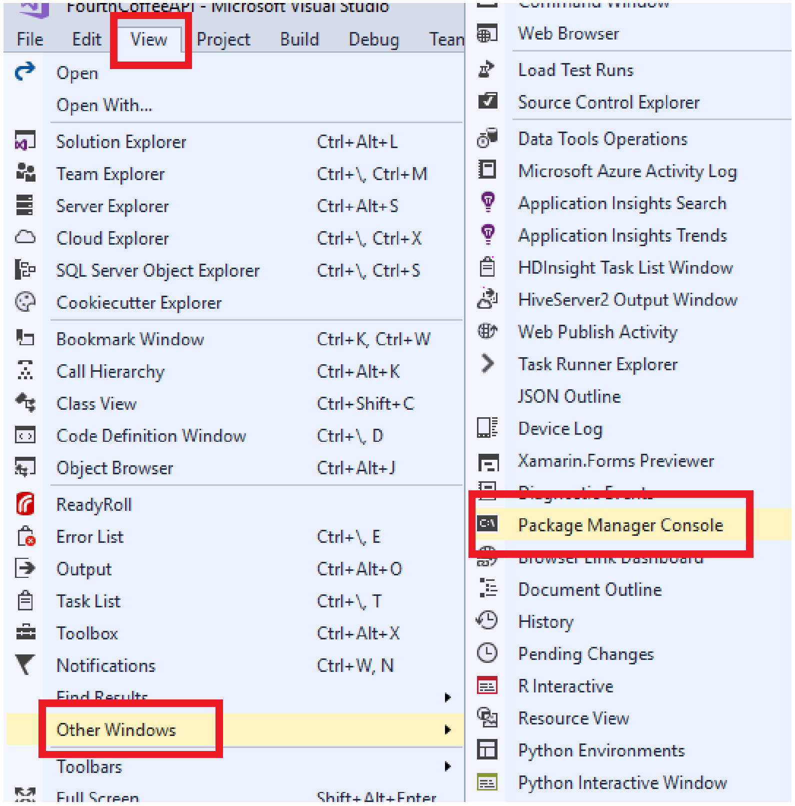
Certificate permissions: 0 selected

Authorized application: None selected

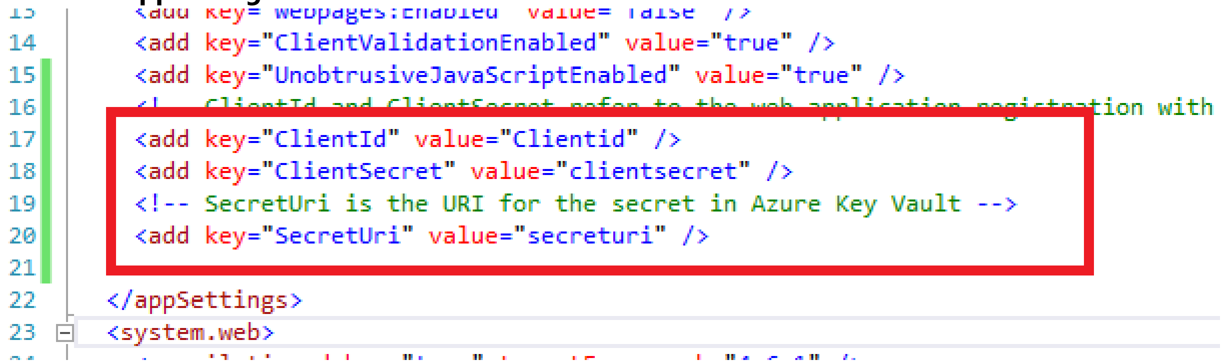
7. Select **OK**.
8. Select **Save**.

Task 4: Install NuGet packages

1. Switch to **Visual Studio**.
2. In the menu, select **View->Other Windows->Package Manager Console**.



3. In the new window that opens, run the following commands (NOTE that these already exist in the project but are provided as a reference).
   a. `Install-Package Microsoft.IdentityModel.Clients.ActiveDirectory -Version 2.16.204221202`
   b. `Install-Package Microsoft.Azure.KeyVault`
4. From **Solution Explorer**, double-select the **web.config** file to open it.
5. Notice the **appSettings** section has some token values:

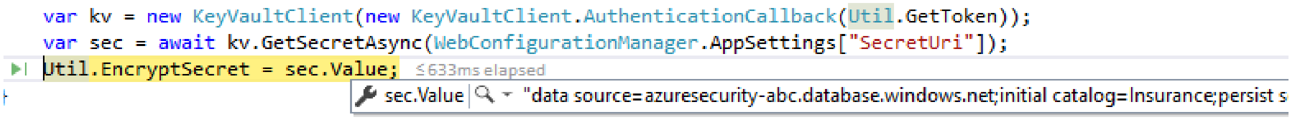


```
    <add key="webpages:Enabled" value="false" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
    <!-- ClientId and ClientSecret refer to the web application registration with
    <add key="ClientId" value="ClientId" />
    <add key="ClientSecret" value="clientsecret" />
    <!-- SecretUri is the URI for the secret in Azure Key Vault -->
    <add key="SecretUri" value="secreturi" />

  </appSettings>
  <system.web>
```

6. Replace the values as follows:
   a. **ClientId**: Replace with the Application ID value copied in Task 2, Step 7. and **C**
   b. **CllientSecret**: Replace with the FourthCoffeeAPI Key values from copied in Task 2, Step 14.
   c. Replace the **SecretUri**: Replace with the Azure Key Vault secret key Uri from Task 1, Step 12.
7. Save Web.config.

Task 5: Test the solution

1. In the **web.config**, delete the **connectionString** from the file at line 78.
2. Save the **web.config** file.
3. Open the **global.asax.cs** file, place a break point at line 31.
    - NOTE: This code makes a call to get an accessToken as the application you setup above, then make a call to the Azure Key Vault using that accessToken.
4. Run the solution, press **F5**.
5. You should see that you execute a call to Azure Key Vault and get back the secret (which in this case is the connection string to the Azure Database).



```
var kv = new KeyVaultClient(new KeyVaultClient.AuthenticationCallback(Util.GetToken));
var sec = await kv.GetSecretAsync(WebConfigurationManager.AppSettings["SecretUri"]);
Util.EncryptSecret = sec.Value;  ≤633ms elapsed
                    sec.Value  "data source=azuresecurity-abc.database.windows.net;initial catalog=Insurance;persist s
```

6. Press **F5**, and navigate to http://localhost:[PORT-NUMBERportno]/api/CustomerAccounts, you should see your data displayed.

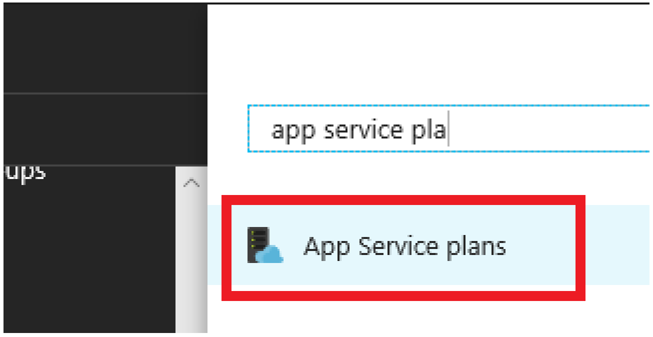
Exercise 6: Securing PaaS web applications with App Service Environment and Web Application Firewall

Duration: 45 minutes

Synopsis: In this exercise, attendees will deploy a cloud web application with a web application gateway and firewall enabled.

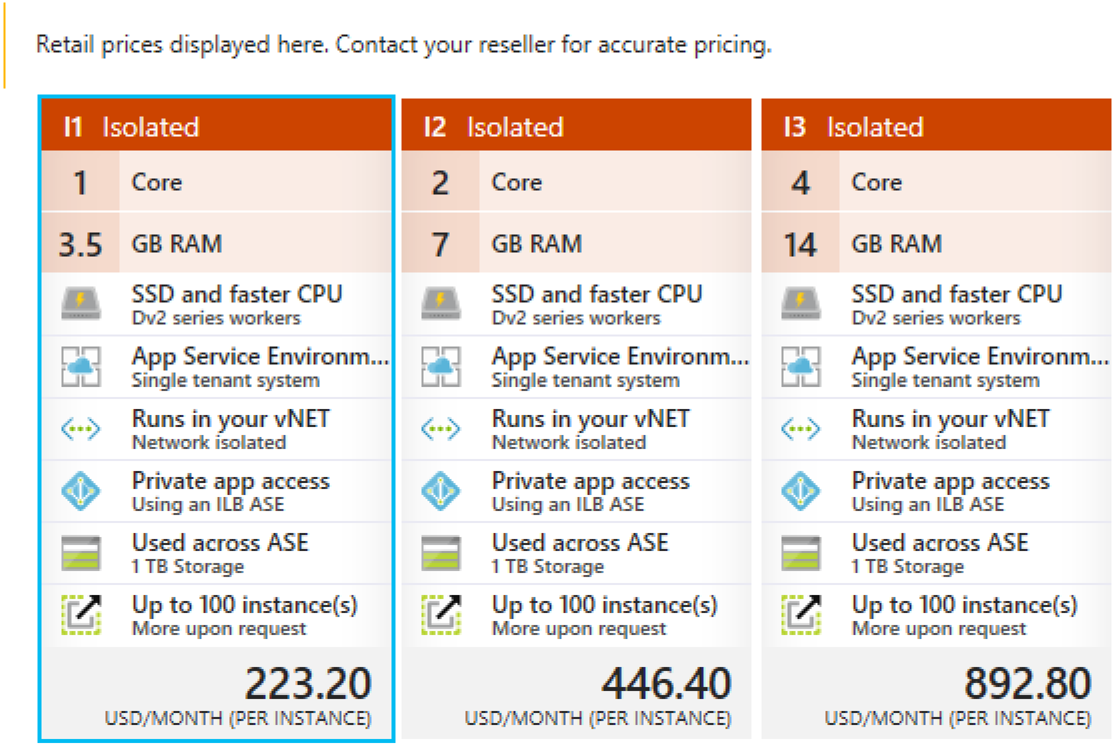
Task 1: Deploy web application to App Service Environment

1. Search for and select **App Service Plans**.



app service pla

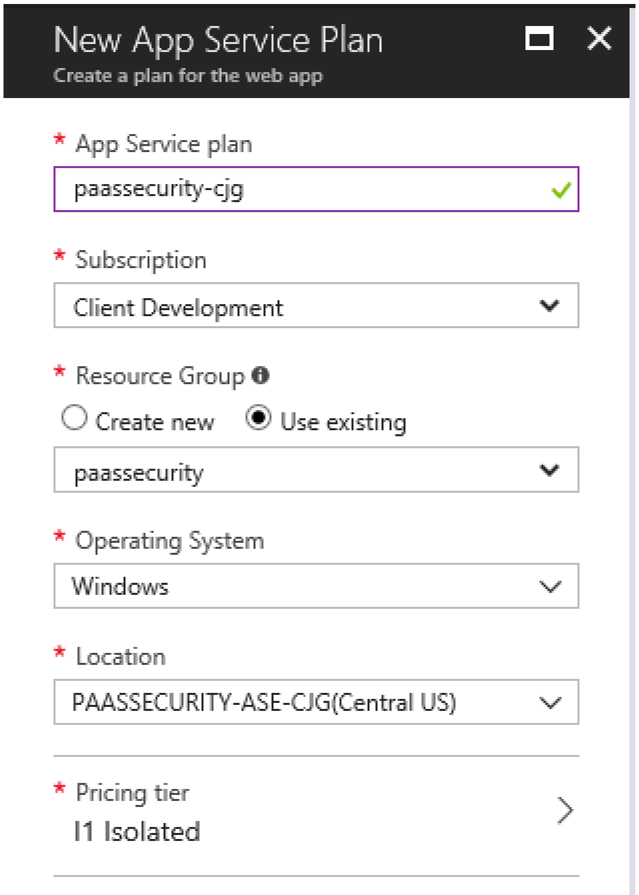
2. Select **+Add**.
3. For the name, enter **paassecurity-[your initials]**.
4. Select your resource group.
5. Select **Location**, then select your **paassecurity-ase-[your initials]** App Service Environment.
6. Select **Pricing tier**, for the pricing tier select **I1 Isolated**.



Retail prices displayed here. Contact your reseller for accurate pricing.

I1 Isolated	I2 Isolated	I3 Isolated
1 Core	2 Core	4 Core
3.5 GB RAM	7 GB RAM	14 GB RAM
SSD and faster CPU Dv2 series workers	SSD and faster CPU Dv2 series workers	SSD and faster CPU Dv2 series workers
App Service Environm... Single tenant system	App Service Environm... Single tenant system	App Service Environm... Single tenant system
Runs in your vNET Network isolated	Runs in your vNET Network isolated	Runs in your vNET Network isolated
Private app access Using an ILB ASE	Private app access Using an ILB ASE	Private app access Using an ILB ASE
Used across ASE 1 TB Storage	Used across ASE 1 TB Storage	Used across ASE 1 TB Storage
Up to 100 instance(s) More upon request	Up to 100 instance(s) More upon request	Up to 100 instance(s) More upon request
223.20 USD/MONTH (PER INSTANCE)	446.40 USD/MONTH (PER INSTANCE)	892.80 USD/MONTH (PER INSTANCE)

7. Select **Select**.



New App Service Plan

Create a plan for the web app

* App Service plan

paassecurity-cjg ✓

* Subscription

Client Development

* Resource Group

○ Create new ● Use existing

paassecurity

* Operating System

Windows

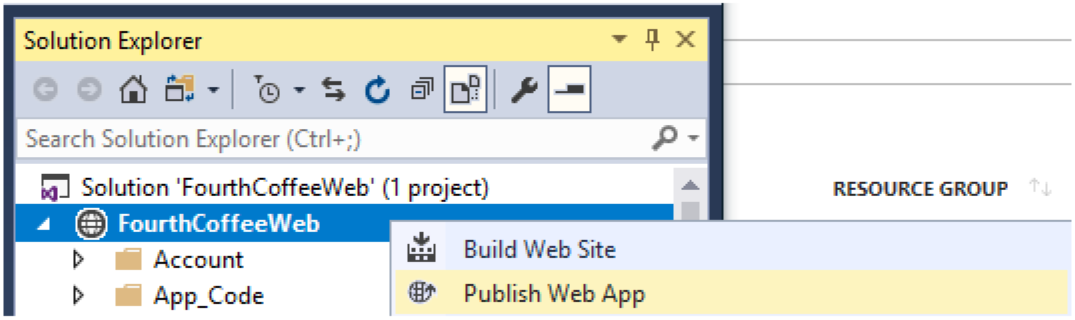
* Location

PAASSECURITY-ASE-CJG(Central US)

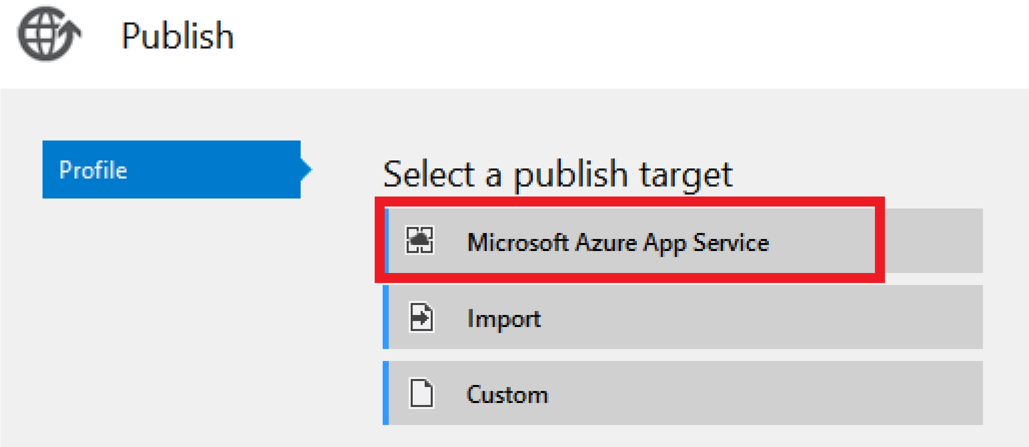
* Pricing tier

I1 Isolated

8. Select **Create**.
9. Switch to your jump VM that is running inside your Azure subscription.
    a. NOTE: You cannot publish from outside the Azure Virtual Network to an internal ASE.
10. Open the extracted folder **\WebApp\FourthCoffeeWeb.sln**
    a. NOTE: You will need to provide an authorized MSDN Visual Studio licensed user.
    b. Select **Sign in**, enter your username, select **Next**.
    c. Enter your password.
11. Select **Sign In**.
12. Right-click the project, select **Publish Web App**.

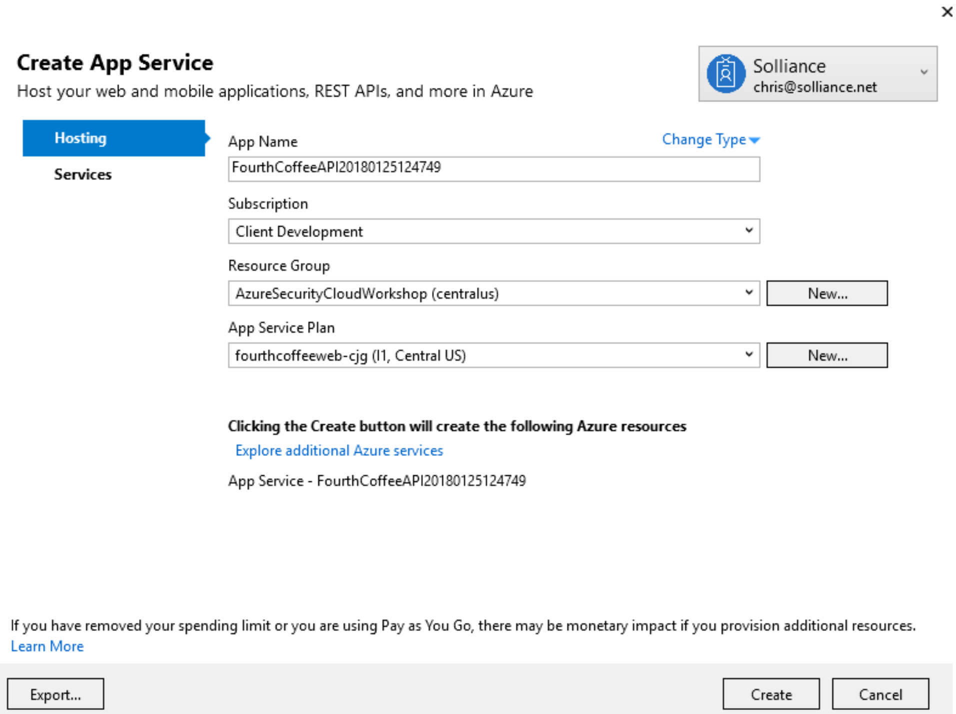


13. Select **Microsoft Azure App Service**.



Publish

14. If prompted, select **Reenter your credentials** such that they match the Azure Subscription you are deploying too.
15. Select **New**.
16. Select your subscription.
17. Select your resource group.
18. For App Service Plan, select the **fourthcoffeeweb-[your initials]**.



Create App Service

Host your web and mobile applications, REST APIs, and more in Azure

Solliance
chris@solliance.net

App Name: FourthCoffeeAPI20180125124749

Subscription: Client Development

Resource Group: AzureSecurityCloudWorkshop (centralus)

App Service Plan: fourthcoffeeweb-cjg (I1, Central US)

Clicking the Create button will create the following Azure resources

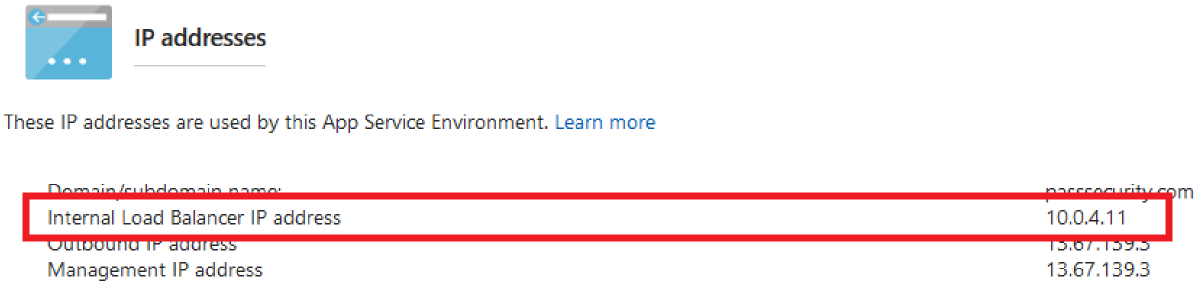
Explore additional Azure services

App Service - FourthCoffeeAPI20180125124749

If you have removed your spending limit or you are using Pay as You Go, there may be monetary impact if you provision additional resources.
Learn More

19. Select **Create**.
    - NOTE: In some versions of Visual Studio, you may need to do this twice.
20. Take note of the URL that your app will be published too:
    a. Switch to the Azure Portal.
    b. Select **App Service Environment**.
    c. Select your **App Service Environment**.
    d. Select **IP Addresses**.

e. Take note of your App Service Environment Internal Load Balancer IP Address.



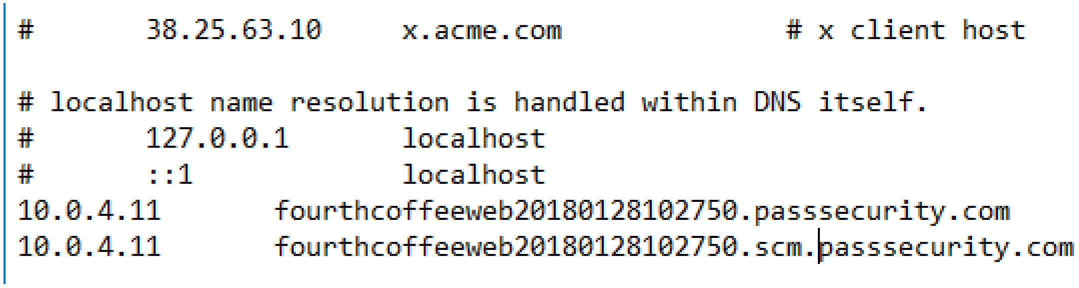
IP addresses

These IP addresses are used by this App Service Environment. Learn more

Domain/subdomain name:	passsecurity.com
Internal Load Balancer IP address	10.0.4.11
Outbound IP address	13.67.139.3
Management IP address	13.67.139.3

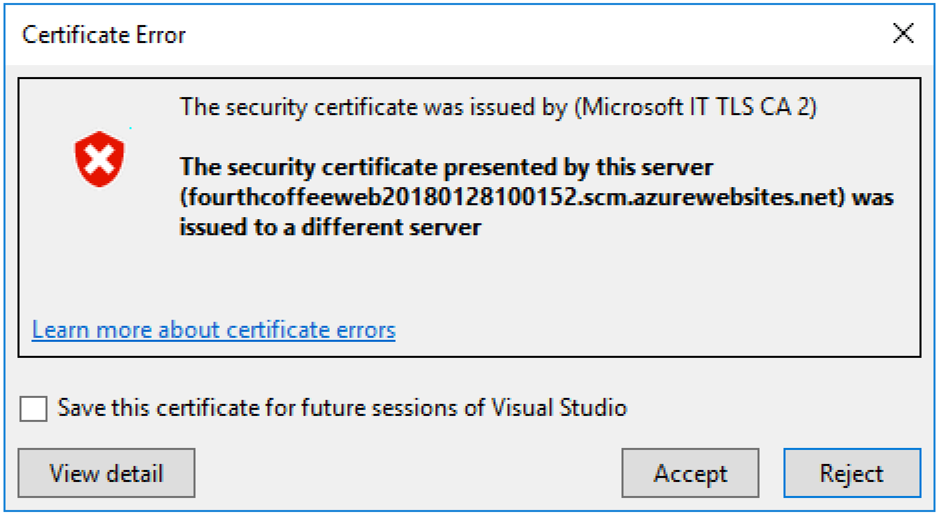
f. On your jump VM, open **Notepad as an administrator**.

g. Open the c:\windows\system32\drivers\etc\hosts file, add the following:



```
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
10.0.4.11       fourthcoffeeweb20180128102750.passsecurity.com
10.0.4.11       fourthcoffeeweb20180128102750.scm.passsecurity.com
```

21. Select **Validate Connection**, if prompted, select **Accept** for the certificate:



Certificate Error

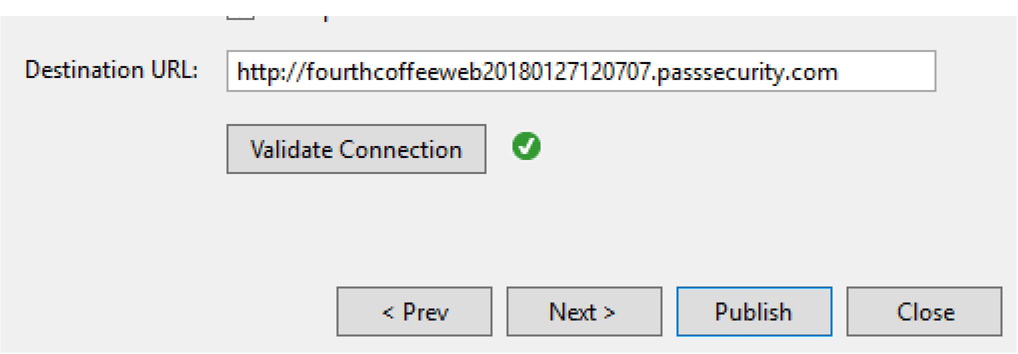
The security certificate was issued by (Microsoft IT TLS CA 2)

The security certificate presented by this server (fourthcoffeeweb20180128100152.scm.azurewebsites.net) was issued to a different server

Learn more about certificate errors

□ Save this certificate for future sessions of Visual Studio

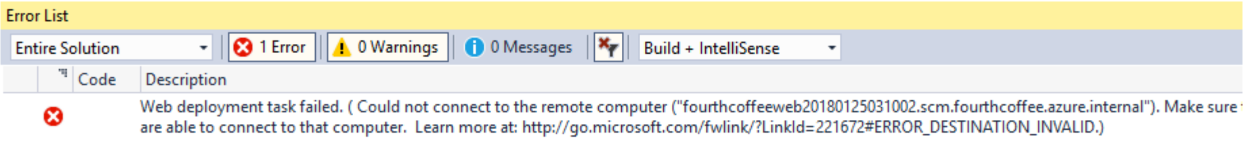
22. The connection should validate with a green checkmark:



Destination URL: http://fourthcoffeeweb20180127120707.passsecurity.com

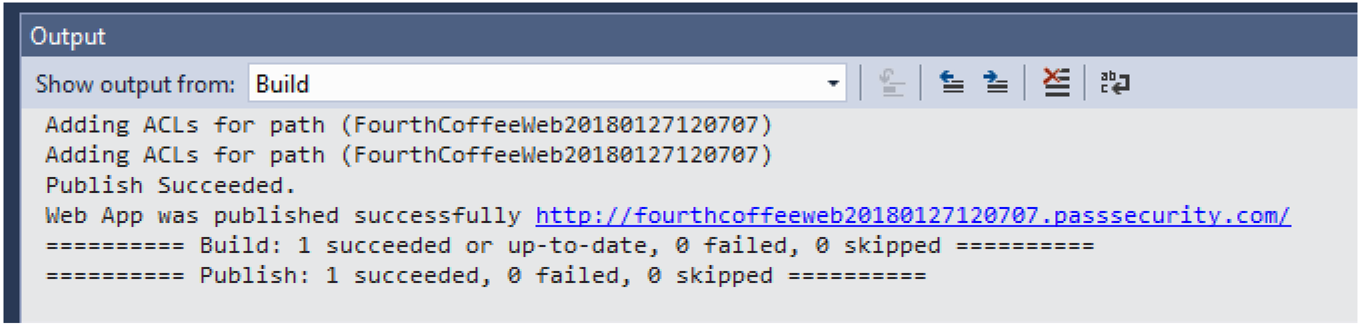
23. Record the destination URL for later in this exercise.

24. Select **Publish.**
    - NOTE: If you get an error, you may be trying to publish outside of the Azure Virtual Network or you did not setup a DNS/Hosts entry for your custom internal domain.



Web deployment task failed. (Could not connect to the remote computer ("fourthcoffeeweb20180125031002.scm.fourthcoffee.azure.internal"). Make sure are able to connect to that computer. Learn more at: http://go.microsoft.com/fwlink/?LinkId=221672#ERROR_DESTINATION_INVALID.)

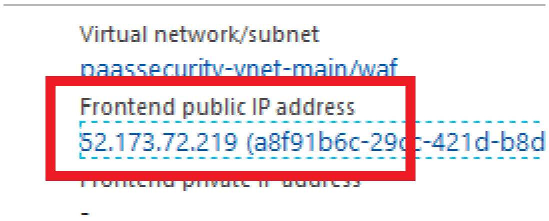
25. Your web application should be published successfully:



```
Adding ACLs for path (FourthCoffeeWeb20180127120707)
Adding ACLs for path (FourthCoffeeWeb20180127120707)
Publish Succeeded.
Web App was published successfully http://fourthcoffeeweb20180127120707.passsecurity.com/
========== Build: 1 succeeded or up-to-date, 0 failed, 0 skipped ==========
========== Publish: 1 succeeded, 0 failed, 0 skipped ==========
```

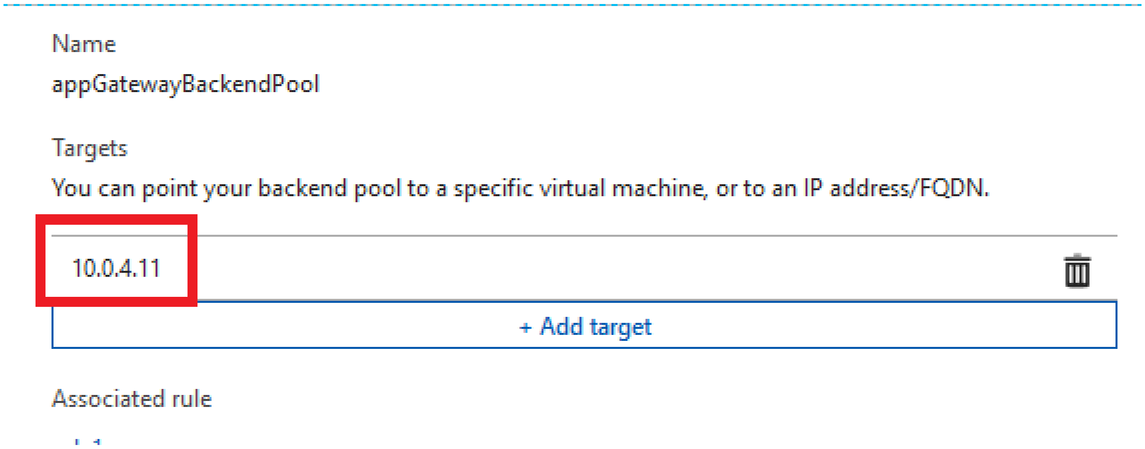
Task 2: Configure the Web Application Firewall

1. Select **Application Gateway**.
2. Select your application gateway.
3. Select **Overview**, record the public IP address of the application gateway for later use:



Virtual network/subnet
paassecurity-vnet-main/waf
Frontend public IP address
52.173.72.219 (a8f91b6c-29c-421d-b8d
Frontend private IP address
-

4. Select **Backend pools**.
5. Select the single backend pool displayed, ensure the IP address is that of the ASE internal load balancer IP. If it is not, delete the one displayed and add the correct IP:



Name
appGatewayBackendPool

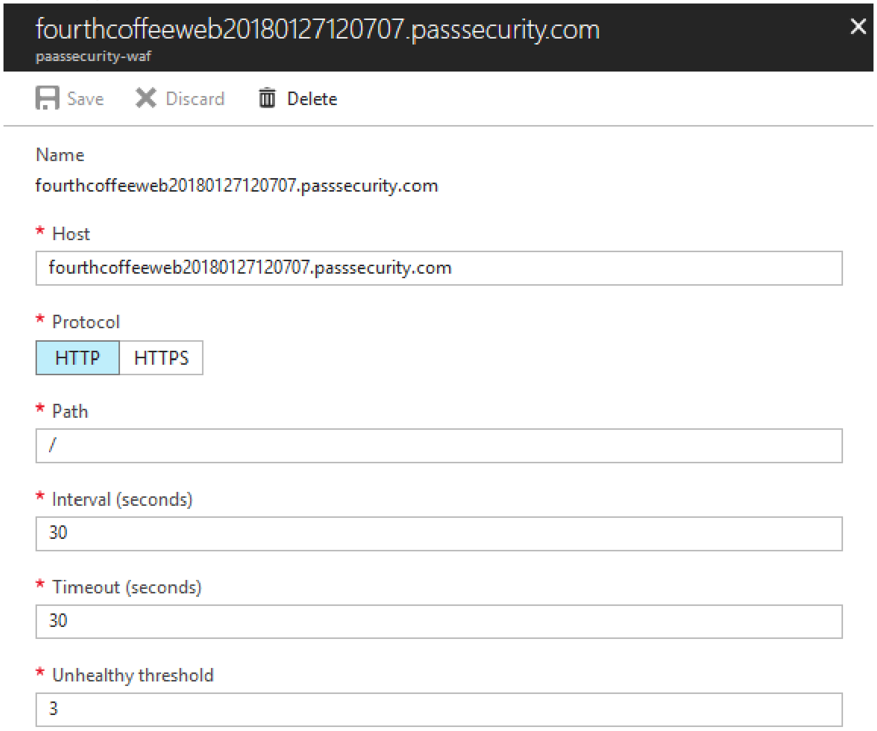
Targets
You can point your backend pool to a specific virtual machine, or to an IP address/FQDN.

10.0.4.11

Associated rule

6. Select **Health probes**.
7. Select **+Add**.
8. Copy the web app DNS address into the name and host address.

9. For the path, enter **/**



fourthcoffeeweb20180127120707.passsecurity.com
paassecurity-waf

Name
fourthcoffeeweb20180127120707.passsecurity.com

* Host
fourthcoffeeweb20180127120707.passsecurity.com

* Protocol
HTTP HTTPS

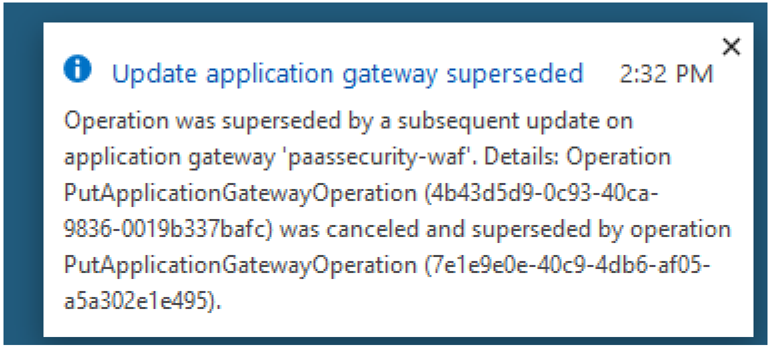
* Path
/

* Interval (seconds)
30

* Timeout (seconds)
30

* Unhealthy threshold
3

10. Select **OK** and then wait for the web application gateway to finish updating.
    a. NOTE: If you do not wait, future actions may result in the following error:

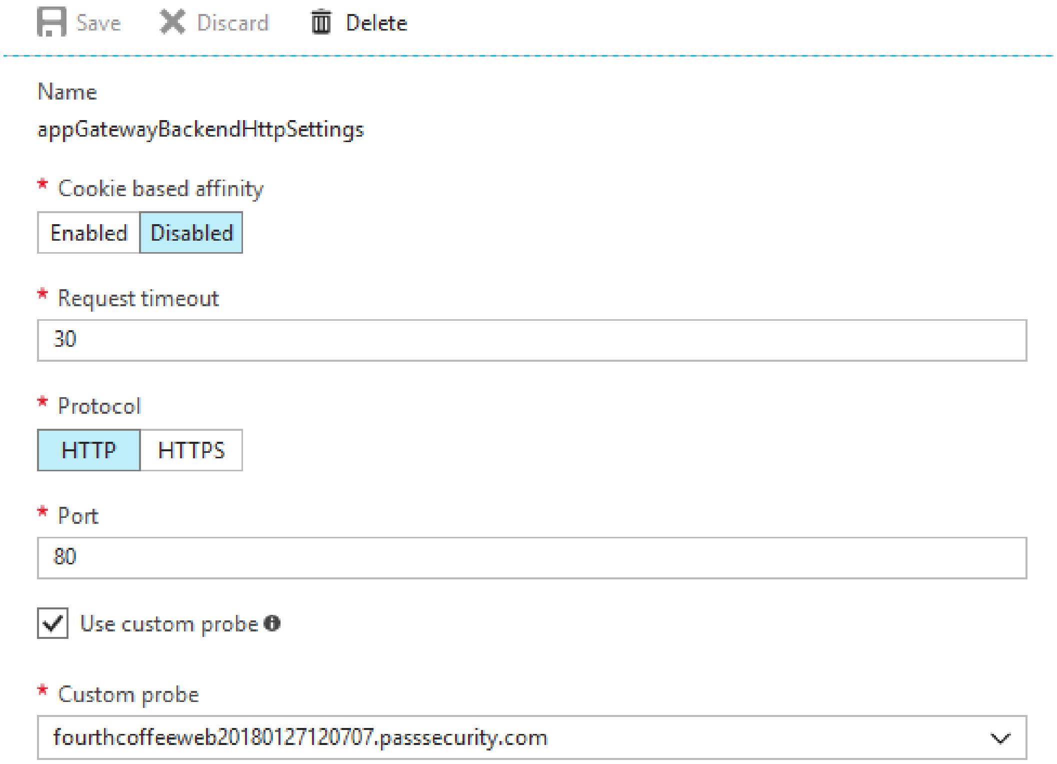


Update application gateway superseded 2:32 PM

Operation was superseded by a subsequent update on application gateway 'paassecurity-waf'. Details: Operation PutApplicationGatewayOperation (4b43d5d9-0c93-40ca-9836-0019b337bafc) was canceled and superseded by operation PutApplicationGatewayOperation (7e1e9e0e-40c9-4db6-af05-a5a302e1e495).

11. Select **HTTP settings**.
12. Select the only **backendHttpSetting**.
13. Check the **Use custom probe** checkbox.

14. Select the custom probe you just added.
    - NOTE: This will make it such that the WAF knows about the host header of the incoming requests and where to route them



Name
appGatewayBackendHttpSettings

* Cookie based affinity
Enabled [✓ Disabled]

* Request timeout
30

* Protocol
[✓ HTTP] HTTPS

* Port
80

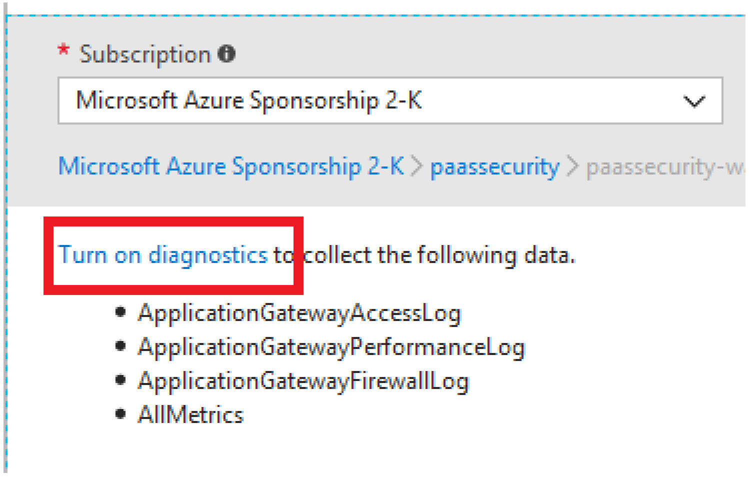
☑ Use custom probe

* Custom probe
fourthcoffeeweb20180127120707.passsecurity.com

15. Select **Save**, wait for the application gateway to finish updating.

Task 3: Enable Application Gateway logging

1. Select **Diagnostic Logs**.
2. Select **Turn on diagnostics**.


* Subscription
Microsoft Azure Sponsorship 2-K

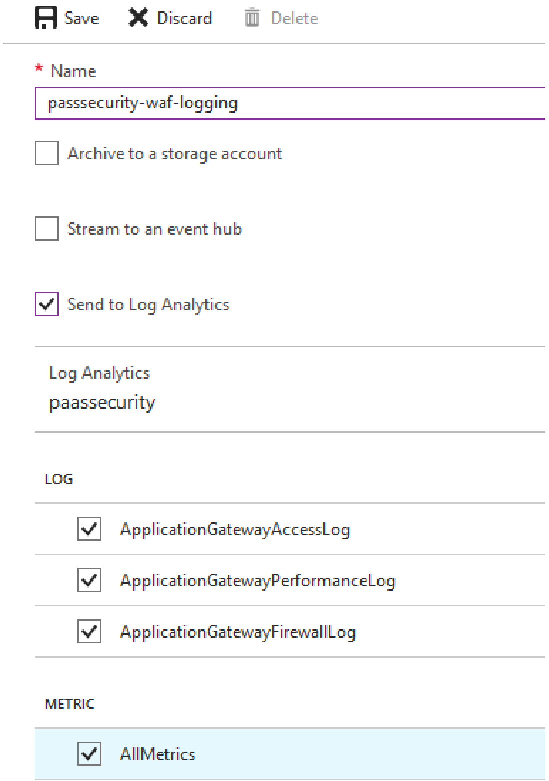
Microsoft Azure Sponsorship 2-K > paassecurity > paassecurity-w

Turn on diagnostics to collect the following data.

- ApplicationGatewayAccessLog
- ApplicationGatewayPerformanceLog
- ApplicationGatewayFirewallLog
- AllMetrics

3. For the name, enter **paassecurity-waf-logging**.
4. Check the **Send to Log Analytics** checkbox.
5. For **Log Analytics**, select your default workspace.
    - NOTE: If you do not have a workspace create one.

6. Check all **LOG** checkboxes.



* Name

passsecurity-waf-logging

□ Archive to a storage account

□ Stream to an event hub

☑ Send to Log Analytics

Log Analytics
paassecurity

LOG

☑ ApplicationGatewayAccessLog

☑ ApplicationGatewayPerformanceLog

☑ ApplicationGatewayFirewallLog

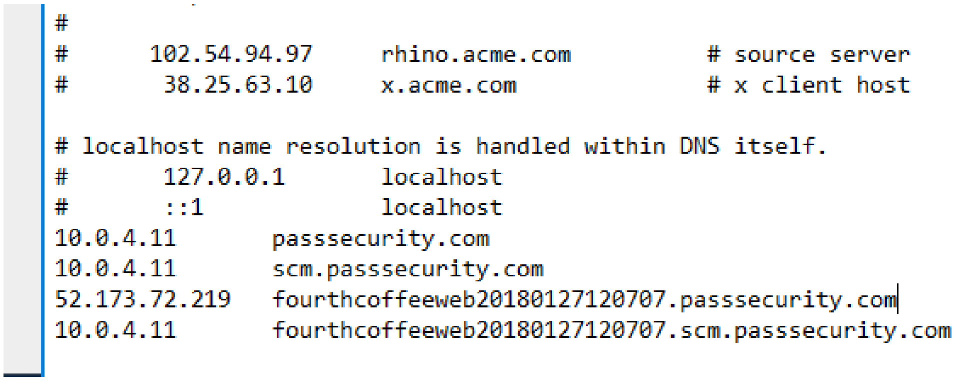
METRIC

☑ AllMetrics

7. Select **Save**.

Task 4: Attack a ASE Web Application with Detection Only

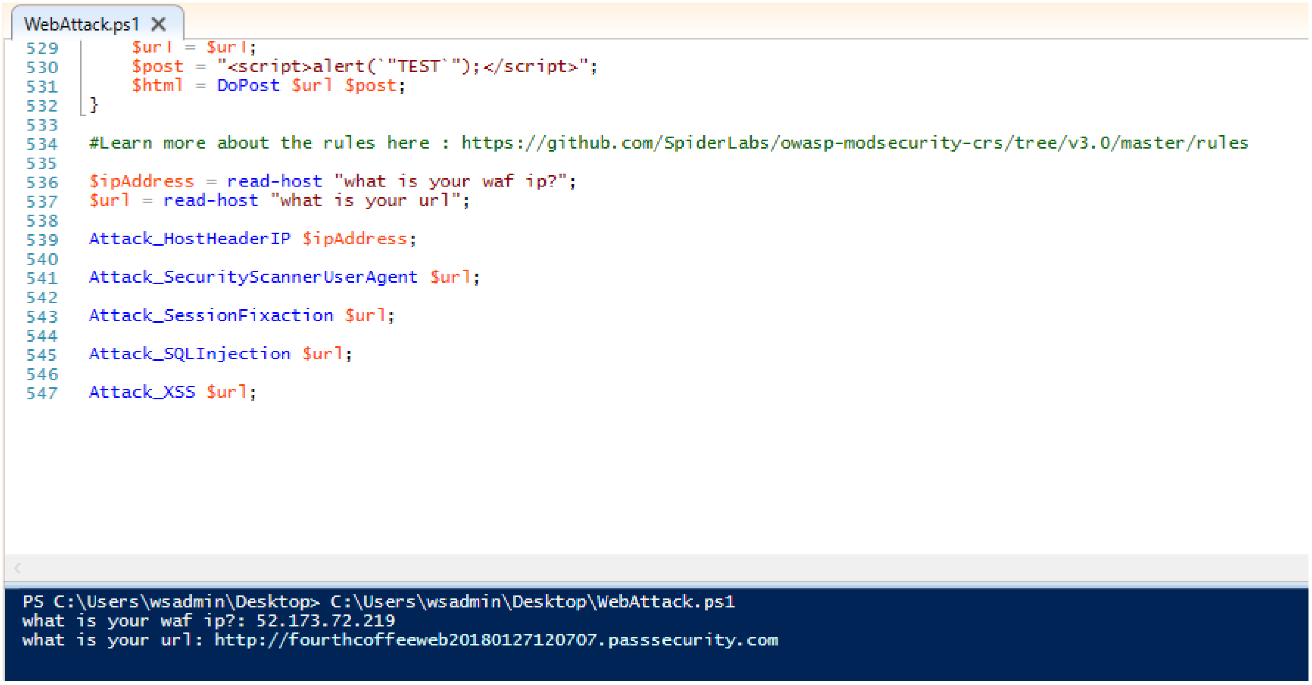
1. Switch to your jump VM.
2. Edit the c:\windows\system32\drivers\etc\hosts file to update the web app URL to point to the WAF public IP Address:



```
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1             localhost
10.0.4.11       passsecurity.com
10.0.4.11       scm.passsecurity.com
52.173.72.219   fourthcoffeeweb20180127120707.passsecurity.com
10.0.4.11       fourthcoffeeweb20180127120707.scm.passsecurity.com
```

3. Save the file.
4. Open a browser window, ensure that the web site opens successfully.
5. Launch Fiddler on your jump VM, so you can observe the network traffic resulting from the following step.
6. Open a **Windows PowerShell ISE** window.
7. From the extracted folder, open the **/Scripts/WebAttack.ps1**.

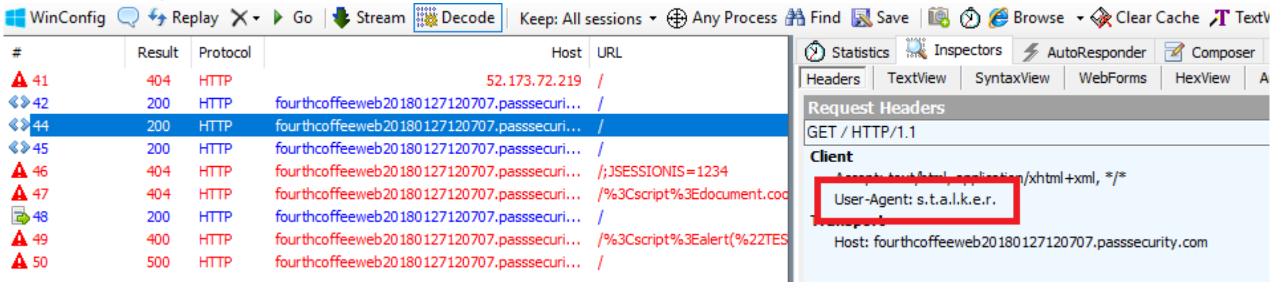
8. Run the script, when prompted, enter the following information:
    a. Web application IP address;
    b. Web application DNS.



```
        $url = $url;
        $post = "<script>alert(`"TEST`");</script>";
        $html = DoPost $url $post;
    }

    #Learn more about the rules here : https://github.com/SpiderLabs/owasp-modsecurity-crs/tree/v3.0/master/rules

    $ipAddress = read-host "what is your waf ip?";
    $url = read-host "what is your url";

    Attack_HostHeaderIP $ipAddress;

    Attack_SecurityScannerUserAgent $url;

    Attack_SessionFixaction $url;

    Attack_SQLInjection $url;

    Attack_XSS $url;
```

```
PS C:\Users\wsadmin\Desktop> C:\Users\wsadmin\Desktop\WebAttack.ps1
what is your waf ip?: 52.173.72.219
what is your url: http://fourthcoffeeweb20180127120707.passsecurity.com
```

9. The script will execute a series of attacks on the Azure web application. Although they won't technically be successful, they will make it to the web application. In Fiddler, you will be able to see the traffic is allowed, even known bad agents:



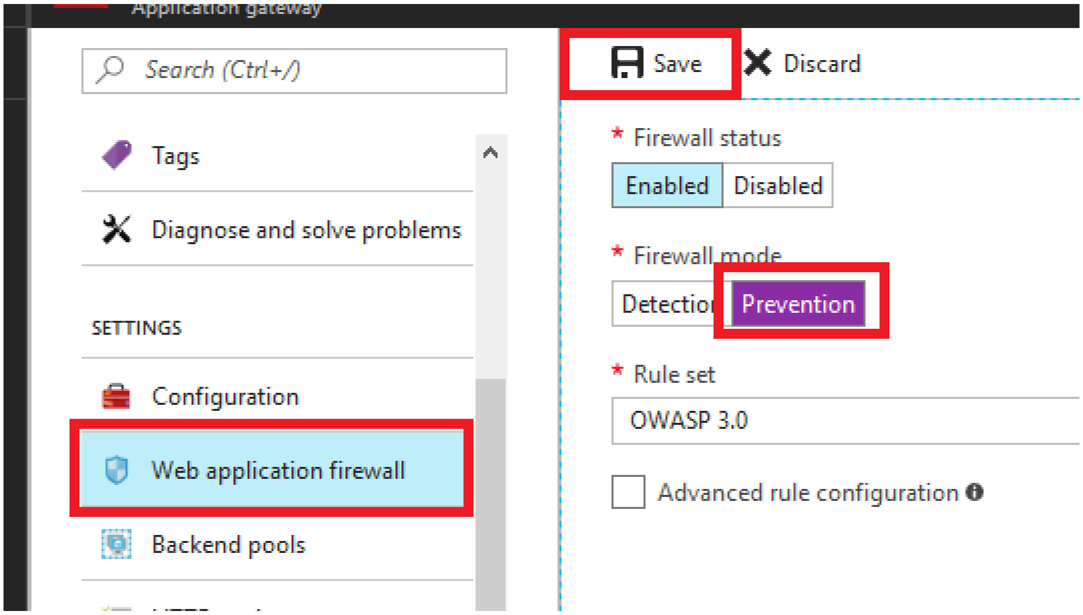
#	Result	Protocol	Host	URL
41	404	HTTP	52.173.72.219	/
42	200	HTTP	fourthcoffeeweb20180127120707.passsecuri...	/
44	200	HTTP	fourthcoffeeweb20180127120707.passsecuri...	/
45	200	HTTP	fourthcoffeeweb20180127120707.passsecuri...	/
46	404	HTTP	fourthcoffeeweb20180127120707.passsecuri...	/;JSESSIONIS=1234
47	404	HTTP	fourthcoffeeweb20180127120707.passsecuri...	/%3Cscript%3Edocument.coo
48	200	HTTP	fourthcoffeeweb20180127120707.passsecuri...	/
49	400	HTTP	fourthcoffeeweb20180127120707.passsecuri...	/%3Cscript%3Ealert(%22TES
50	500	HTTP	fourthcoffeeweb20180127120707.passsecuri...	/

```
Request Headers
GET / HTTP/1.1
Client
    Accept: text/html, application/xhtml+xml, */*
    User-Agent: s.t.a.l.k.e.r.
Transport
    Host: fourthcoffeeweb20180127120707.passsecurity.com
```

Task 5: Enable Web Application Firewall Prevention

1. In the Azure portal, select **Application gateway**.
2. Select your application gateway.
3. Select **Web application firewall**.

4. For **Firewall mode**, select the **Prevention**.



* Firewall status

Enabled | Disabled

* Firewall mode

Detection | Prevention

* Rule set

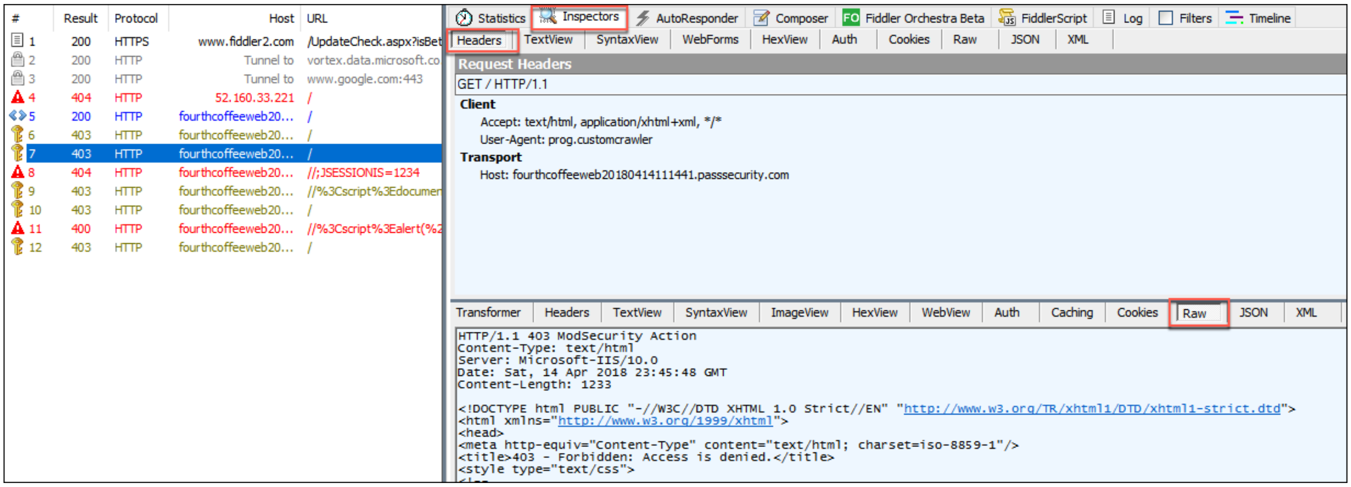
OWASP 3.0

□ Advanced rule configuration

5. Select **Save**, wait for the application gateway to be updated.

Task 6: Reattack an ASE Web Application with Prevention enabled

1. Switch back to the **Windows PowerShell ISE** window.
2. Run the WebAttack script, then when prompted, enter the following information:
   a. Web application gateway IP address
   b. Web application gateway DNS
3. In Fiddler, you should see that your attack is being prevented from making it to the web server. This will also generate logs that we will use to create attack assessment reports in later exercises. Again, with fiddler available you can see the denied traffic by selecting the Inspectors tab, then Headers in the top section, and Raw in the bottom section:



#	Result	Protocol	Host	URL
1	200	HTTPS	www.fiddler2.com	/UpdateCheck.aspx?isBet
2	200	HTTP	Tunnel to	vortex.data.microsoft.co
3	200	HTTP	Tunnel to	www.google.com:443
4	404	HTTP	52.160.33.221	/
5	200	HTTP	fourthcoffeeweb20...	/
6	403	HTTP	fourthcoffeeweb20...	/
7	403	HTTP	fourthcoffeeweb20...	/
8	404	HTTP	fourthcoffeeweb20...	//;JSESSIONIS=1234
9	403	HTTP	fourthcoffeeweb20...	//%3Cscript%3Edocumen
10	403	HTTP	fourthcoffeeweb20...	/
11	400	HTTP	fourthcoffeeweb20...	//%3Cscript%3Ealert(%2
12	403	HTTP	fourthcoffeeweb20...	/

Request Headers

GET / HTTP/1.1

Client
Accept: text/html, application/xhtml+xml, */*
User-Agent: prog.customcrawler

Transport
Host: fourthcoffeeweb20180414111441.passsecurity.com

```
HTTP/1.1 403 ModSecurity Action
Content-Type: text/html
Server: Microsoft-IIS/10.0
Date: Sat, 14 Apr 2018 23:45:48 GMT
Content-Length: 1233

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"/>
<title>403 - Forbidden: Access is denied.</title>
<style type="text/css">
```

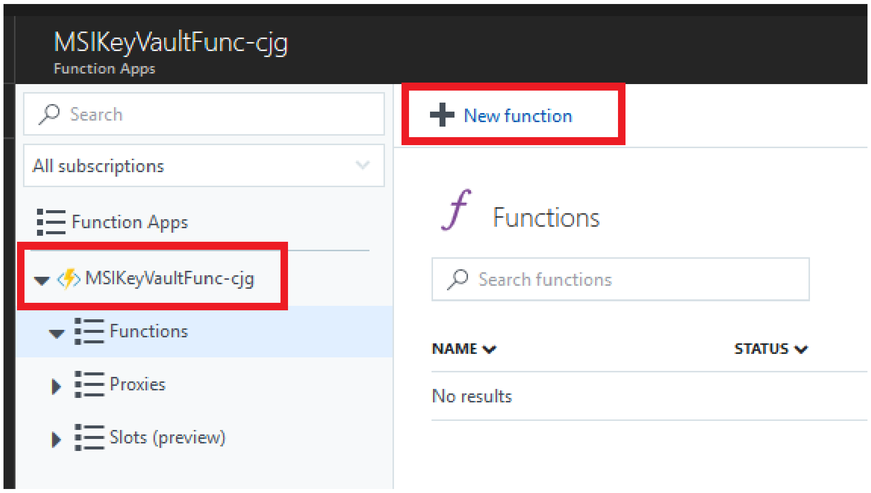
Exercise 7: Securing Azure Functions with Managed Service Identities

Duration: 30 minutes

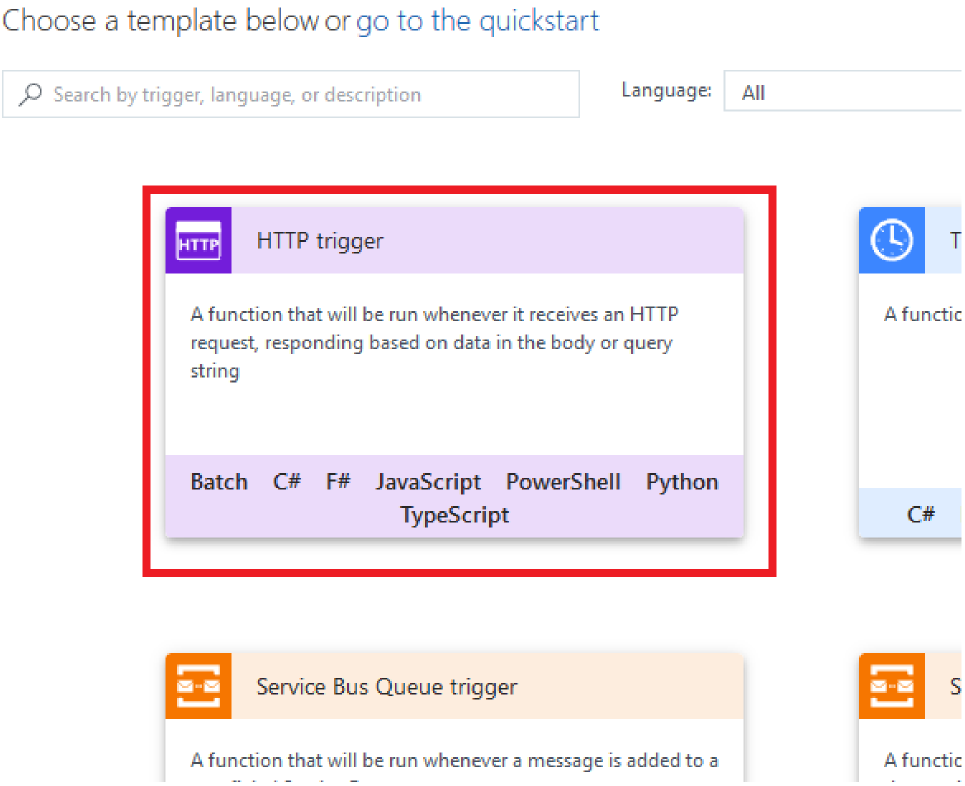
Synopsis: In this exercise, attendees will learn how to use Azure Functions that access Azure Key Vault as a Managed Service Identity.

Task 1: Create an Azure Function

1. Open the Azure Function App creation page (https://portal.azure.com/#create/Microsoft.FunctionApp)
2. For the name, enter **MSIKeyVaultFunc-[Your Initials]**.
3. Select your resource group.
4. Ensure that your location matches what you have been using.
5. For **Storage**, select the storage account.
6. Select **Create**.
7. Select **Function Apps**.
8. Once provisioning completes, select your new function app.
9. Select the **Functions** node.
10. Select **New function**.



11. Select **HTTP trigger**.



Choose a template below or go to the quickstart

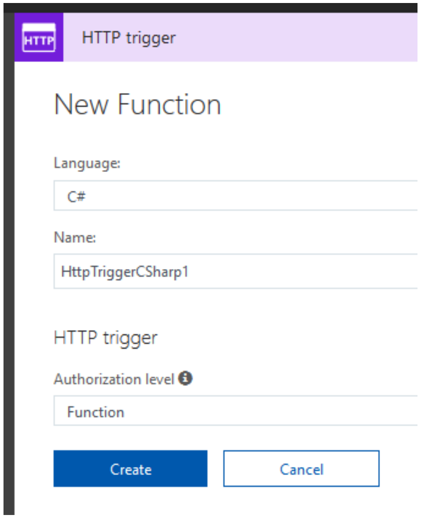
Search by trigger, language, or description
Language: All
HTTP
HTTP trigger

A function that will be run whenever it receives an HTTP request, responding based on data in the body or query string

Batch C# F# JavaScript PowerShell Python TypeScript
T
A functic
C#
Service Bus Queue trigger

A function that will be run whenever a message is added to a

12. For the **language** select **C#**.
13. Keep the name **HttpTriggerCSharp1**.



New Function

Language: C#

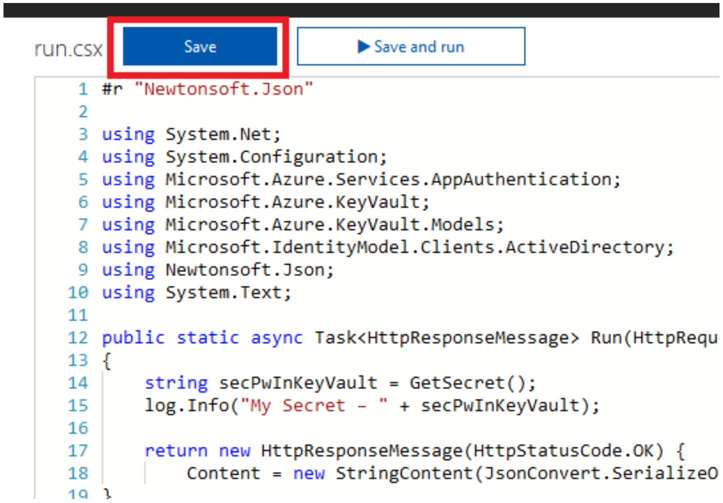
Name: HttpTriggerCSharp1

HTTP trigger

Authorization level: Function

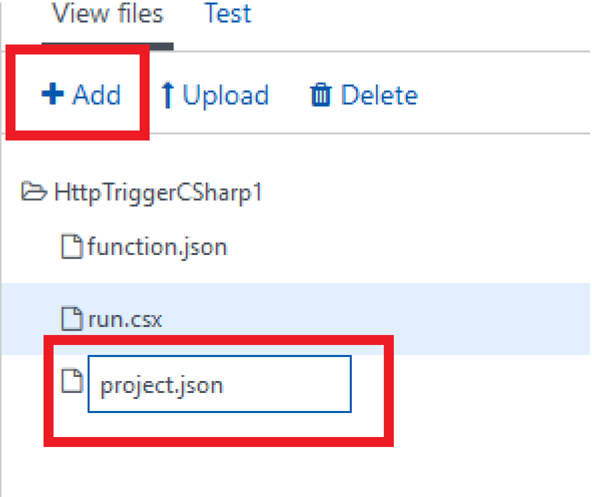
14. Select **Create**.
15. Open the extracted folder file **\AzureFunction\run.csx**.

16. Copy the contents into the window.



```
#r "Newtonsoft.Json"

using System.Net;
using System.Configuration;
using Microsoft.Azure.Services.AppAuthentication;
using Microsoft.Azure.KeyVault;
using Microsoft.Azure.KeyVault.Models;
using Microsoft.IdentityModel.Clients.ActiveDirectory;
using Newtonsoft.Json;
using System.Text;

public static async Task<HttpResponseMessage> Run(HttpRequ
{
    string secPwInKeyVault = GetSecret();
    log.Info("My Secret - " + secPwInKeyVault);

    return new HttpResponseMessage(HttpStatusCode.OK) {
        Content = new StringContent(JsonConvert.SerializeO
}
```

17. Select **Save**.
18. Select **View Files**.
    - NOTE: You may need to scroll to the right to see the View Files tab.
19. Select **+Add**.
20. For the name, enter **project.json**.

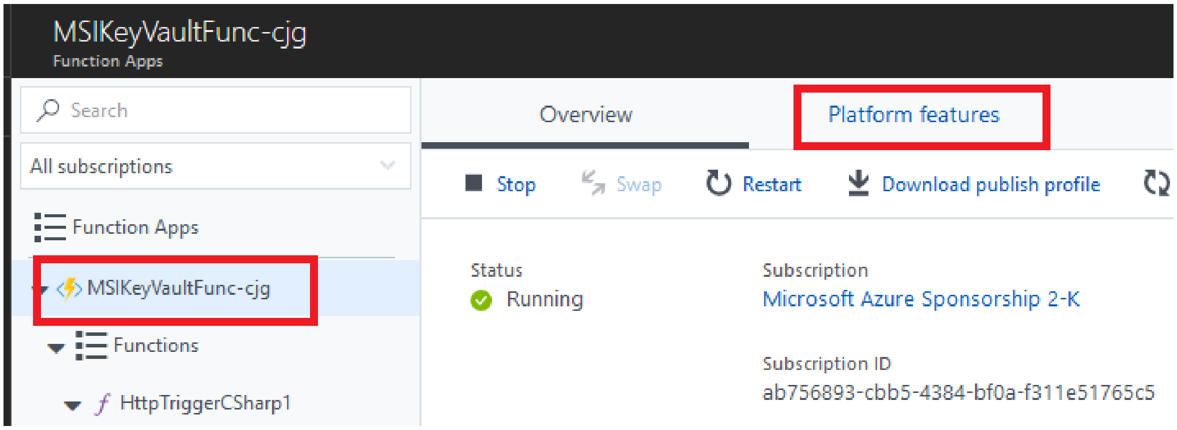


21. Press **Enter**.
22. Open the **\AzureFunction\project.json** file, copy the contents to the online version.
23. Select **Save**.

Task 2: Create a Managed Service Identity

1. Select the **MSIKeyVaultFunc-[your initials]** function node.

2. Select the **Platform features** tab.

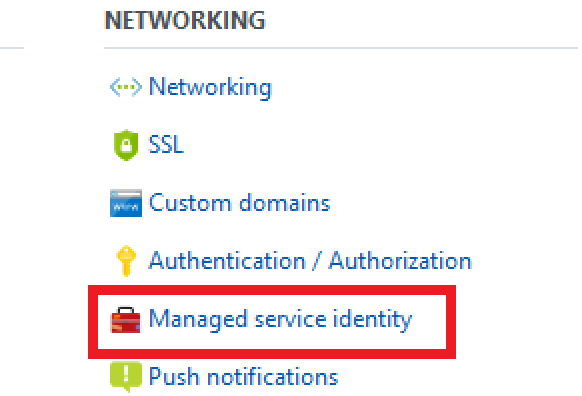


Status
Running

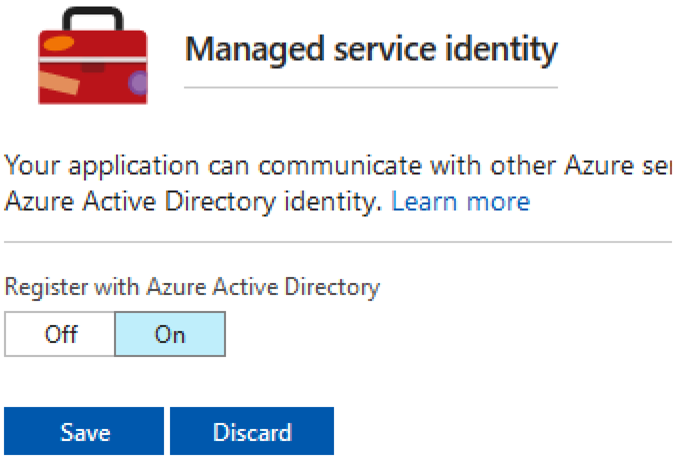
Subscription
Microsoft Azure Sponsorship 2-K

Subscription ID
ab756893-cbb5-4384-bf0a-f311e51765c5

3. Under **Networking**, select **Managed service identity**.



4. For the **Register with Azure Active Directory** setting, toggle it to **On**.



Your application can communicate with other Azure sei
Azure Active Directory identity. Learn more

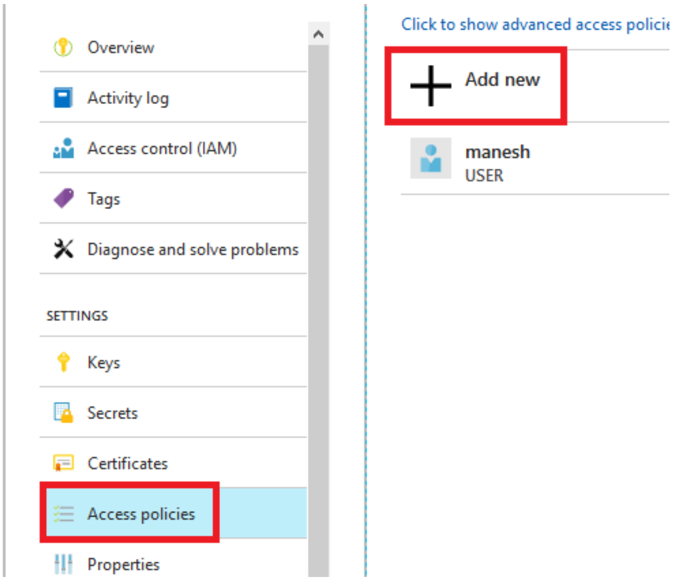
Register with Azure Active Directory
Off [✓ On]

5. Select **Save**.

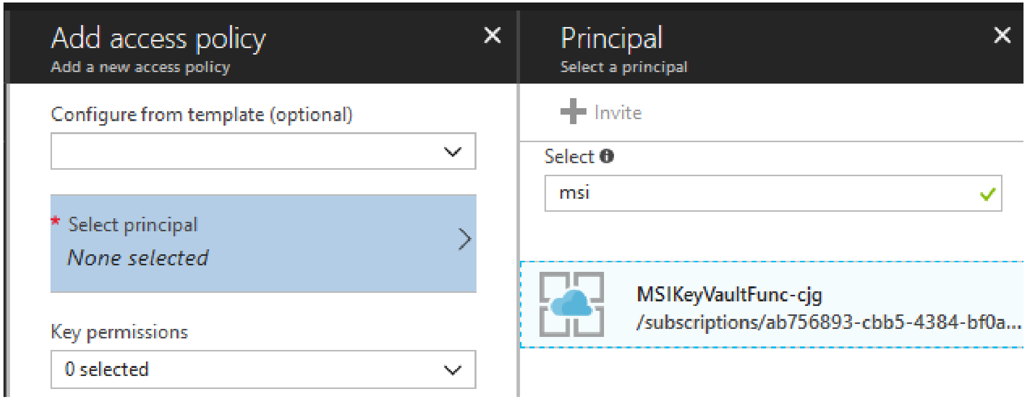
Task 3: Assign Managed Service Identity Azure Key Vault permissions

1. Select **Key vaults**.
2. Select your key vault.

3. Select **Access policies**.
4. Select **+Add new**.



5. Select the **Select principal**.
6. Search for the **MSIKeyVaultFunc** application, select it.



Configure from template (optional)

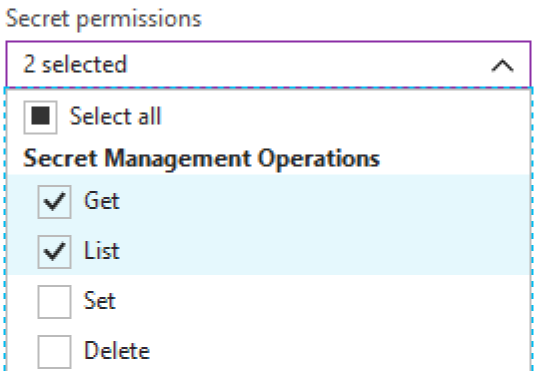
* Select principal
None selected

Key permissions
0 selected

Select
msi

MSIKeyVaultFunc-cjg
/subscriptions/ab756893-cbb5-4384-bf0a...

7. Select **Select**.
8. Select the **Secret permissions** drop down, check the **Get** and **List** permissions.



Secret permissions
2 selected

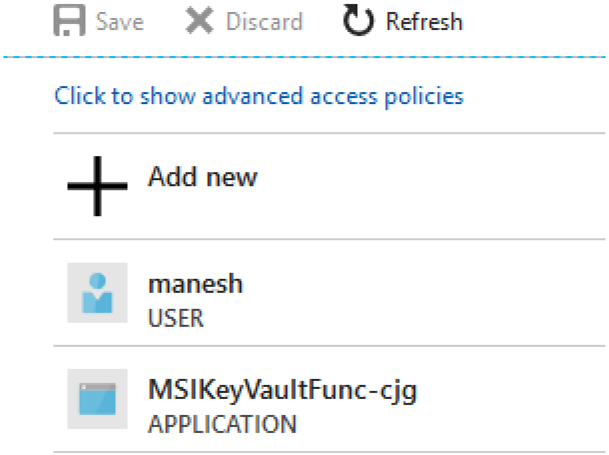
■ Select all

Secret Management Operations

☑ Get
☑ List
☐ Set
☐ Delete

9. Select **OK**.

10. Select **Save**, you should now see the application listed:

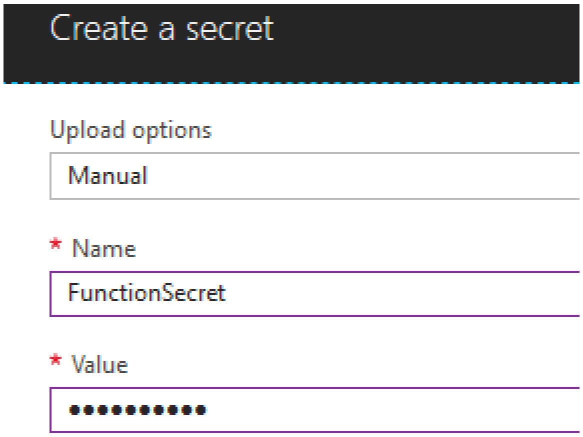


manesh
USER

MSIKeyVaultFunc-cjg
APPLICATION

Task 4: Test your Azure Function

1. Select **Key vaults**.
2. Select **Secrets**.
3. Select **+Generate/Import**.
   - NOTE: If you can't add a new Secret, you will need to assign yourself permission to do so via Access policies.
4. In **Upload options**, select **Manual**.
5. For the **name**, enter **FunctionSecret**.
6. For the **value**, enter **HelloWorld**.

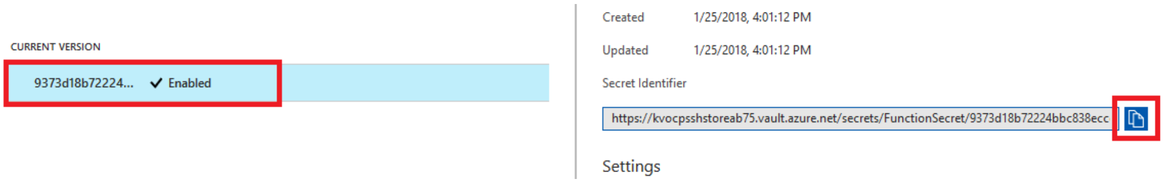


Upload options
Manual

* Name
FunctionSecret

* Value
●●●●●●●●●●

7. Select **Create**.
8. Select **FunctionSecret**.
9. Select the current version, then select and record the Secret Identifier URL

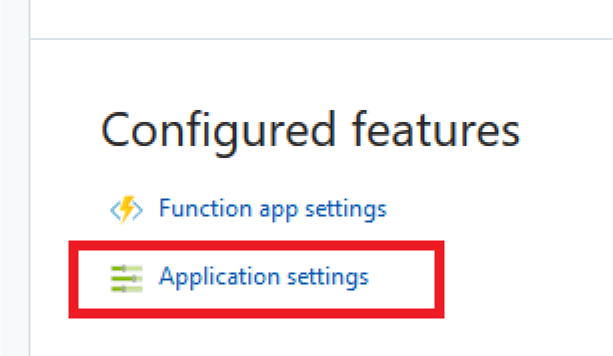


CURRENT VERSION
9373d18b72224... ✓ Enabled

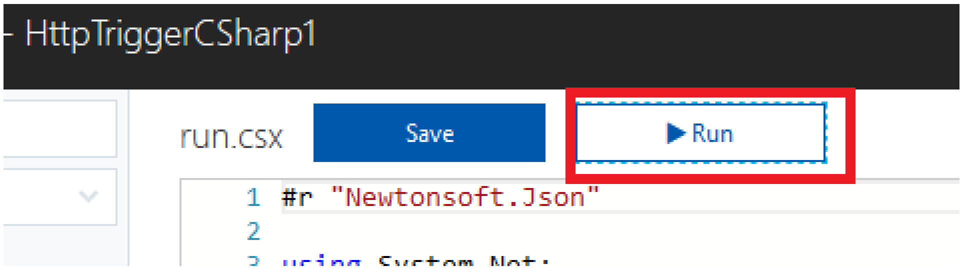
Created 1/25/2018, 4:01:12 PM
Updated 1/25/2018, 4:01:12 PM
Secret Identifier
https://kvocpsshstoreab75.vault.azure.net/secrets/FunctionSecret/9373d18b72224bbc838ecc

10. Select **Function Apps**.
11. Select **MSIKeyVaultFunc-[your initials]**.

12. Select **Application Settings**.

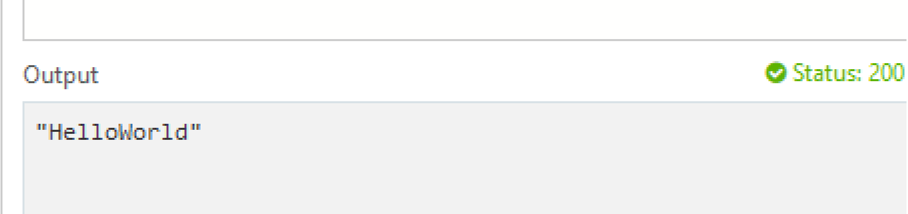


13. Under **Application Settings**, select **+Add new setting**.
14. For the **name**, enter **KeyVaultUri**.
15. For the **value**, copy the Secret Identifier URL you copied in this task.
16. Scroll to the top, select **Save**.
17. Select the **HttpTriggerCSharp1** function.
18. Select **Run.**



```
#r "Newtonsoft.Json"

using System.Net;
```

19. In the Output window you should see your Key Vault Secret displayed.



```
"HelloWorld"
```

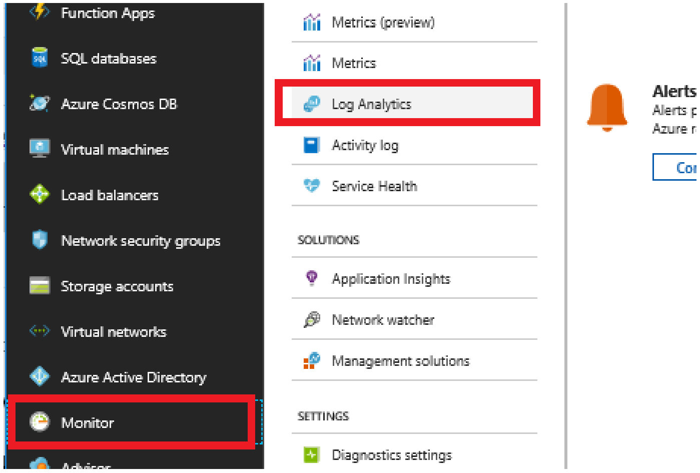
Exercise 8: Creating PaaS Audit and Compliance Power BI Reports

Duration: 20 minutes

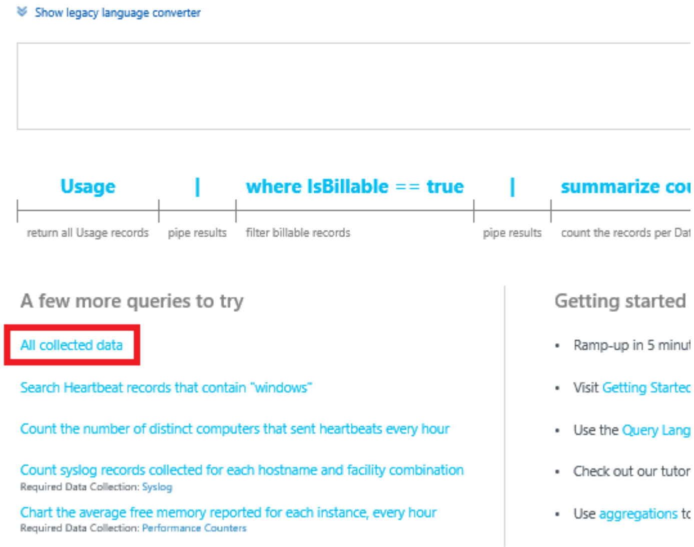
Synopsis: In this exercise, attendees will learn to utilize the Log Analytics feature of Azure to create Power BI Reports.

Task 1: Export a Power Query formula from Log Analytics

1. Select **Monitor**, then select **Log Analytics**.

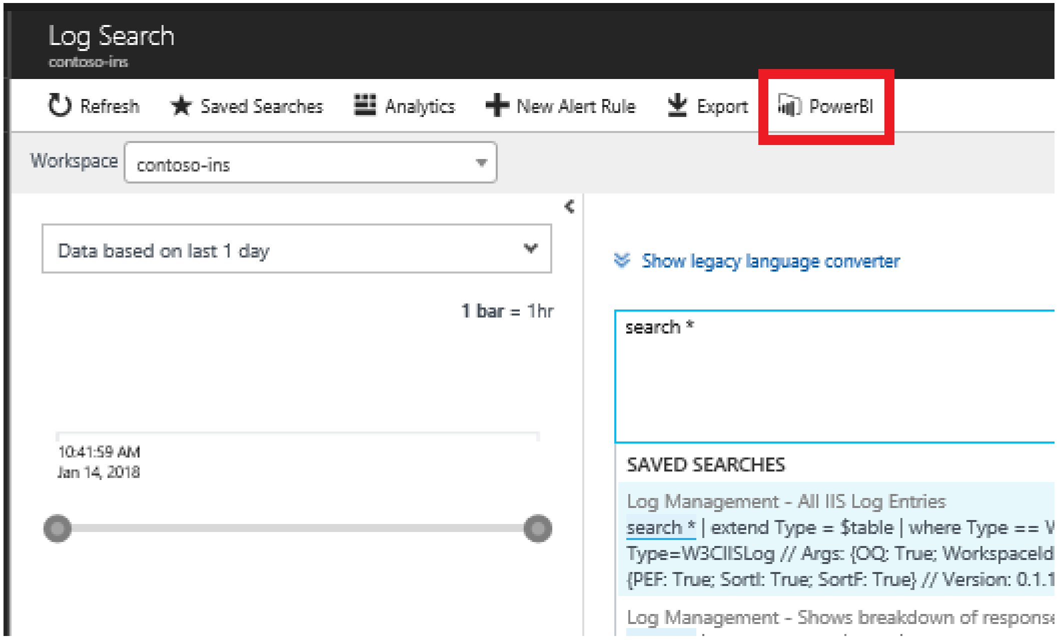


2. Select **All collected data**.


Show legacy language converter
Usage
|
where IsBillable == true
|
summarize cou
return all Usage records
pipe results
filter billable records
pipe results
count the records per Dat
A few more queries to try
All collected data
Search Heartbeat records that contain "windows"
Count the number of distinct computers that sent heartbeats every hour
Count syslog records collected for each hostname and facility combination
Required Data Collection: Syslog
Chart the average free memory reported for each instance, every hour
Required Data Collection: Performance Counters
Getting started
Ramp-up in 5 minut
Visit Getting Startec
Use the Query Lang
Check out our tutor
Use aggregations tc

3. Update the search textbox to be **search * | where Type == "SecurityDetection**."
   - NOTE: If you wanted to see things that were specific from your IP address you can add **| where ExtendedProperties contains "X.X.X.X"** to the query.
4. Select the **Run** button.

5. In the **Log Search** dialog, select the **Power BI** link.

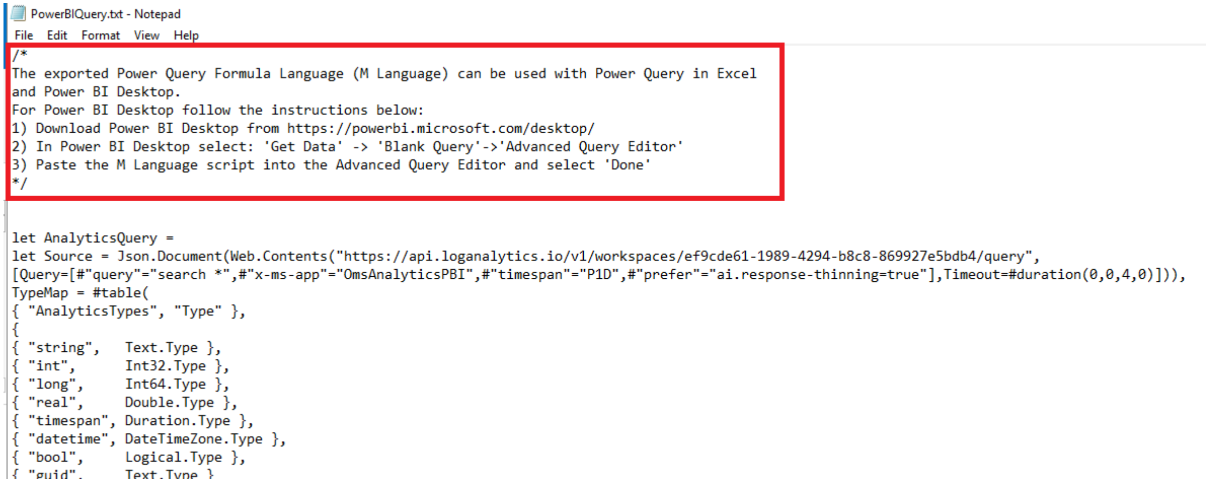


```
search *
```

Log Management - All IIS Log Entries
search * | extend Type = $table | where Type == V
Type=W3CIISLog // Args: {OQ: True; WorkspaceId
{PEF: True; SortI: True; SortF: True} // Version: 0.1.1

Log Management - Shows breakdown of respons

6. Select **Open**, a text document with the Power Query M Language will be displayed.
7. Follow the instructions in the document to execute the query in Power BI.



```
/*
The exported Power Query Formula Language (M Language) can be used with Power Query in Excel
and Power BI Desktop.
For Power BI Desktop follow the instructions below:
1) Download Power BI Desktop from https://powerbi.microsoft.com/desktop/
2) In Power BI Desktop select: 'Get Data' -> 'Blank Query'->'Advanced Query Editor'
3) Paste the M Language script into the Advanced Query Editor and select 'Done'
*/


let AnalyticsQuery =
let Source = Json.Document(Web.Contents("https://api.loganalytics.io/v1/workspaces/ef9cde61-1989-4294-b8c8-869927e5bdb4/query",
[Query=[#"query"="search *",#"x-ms-app"="OmsAnalyticsPBI",#"timespan"="P1D",#"prefer"="ai.response-thinning=true"],Timeout=#duration(0,0,4,0)])),
TypeMap = #table(
{ "AnalyticsTypes", "Type" },
{
{ "string",   Text.Type },
{ "int",      Int32.Type },
{ "long",     Int64.Type },
{ "real",     Double.Type },
{ "timespan", Duration.Type },
{ "datetime", DateTimeZone.Type },
{ "bool",     Logical.Type },
{ "guid",     Text.Type }
```

8. Close **Power BI**.

After the hands-on lab

Duration: 10 minutes

In this exercise, attendees will deprovision any Azure resources that were created in support of the lab.

Task 1: Delete resource group

1. Using the Azure portal, navigate to the Resource group you used throughout this hands-on lab by selecting **Resource groups** in the left menu.
2. Search for the name of your resource group and select it from the list.
3. Select **Delete** in the command bar and confirm the deletion by re-typing the Resource group name and selecting **Delete**.

Task 2: Delete Azure AD objects

1. Navigate to Azure Active Directory in the Azure portal.
2. Delete the groups you created.
   a. Key Vault Mgmt Admins
   b. Key Vault Key Admins
3. Delete the users you created.
   a. Key Vault Admin
   b. Key Vault Auditor
   c. Key Vault Developer
4. Delete the App you registered.
   a. Select App registrations
   b. Select View all applications
   c. Select and delete the AzureKeyVaultTest app.

Task 3: Delete lab environment (optional)

1. If you are using a hosted platform, make sure you shut it down/delete it.

You should follow all steps provided *after* attending the Hands-on lab.