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

Scenario

This module includes the following tasks:

- Azure confidential computing
- Azure Azure Key Vault

Exercise 1: Introduction to Azure Key Vault

Scenario

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

Task 1: Download SQL Server Management Studio

□ 1. To download the latest version of SQL Management Studio required for this lab visit the following link **https://aka.gd/2KplFiZ** and select download SQL Management Studio and install it using the defaults:

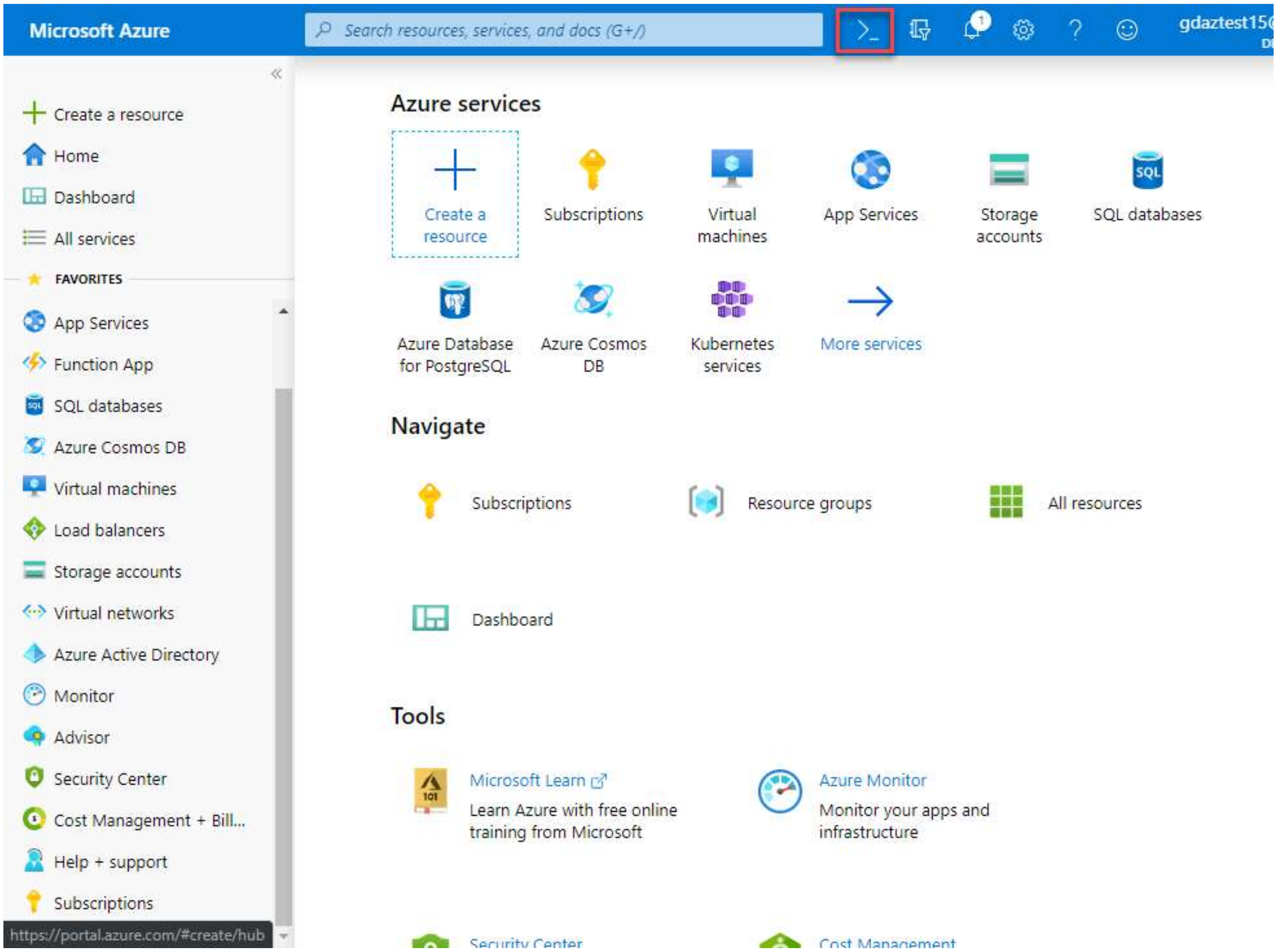
> **Note:** You do not need to wait for the SQL Management Studio to install before continuing.

Task 2: Use PowerShell to create a Key Vault

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

□ 1. Navigate to **portal.azure.com** and login using username **sheikhnasir7MKPQ@gdcs2.com** and password **yXzvOgL0HcKft7rG**.

□ 2. Select **Cloud Shell** from the Azure Portal tool bar.



□ 3. Select **PowerShell** on the Welcome screen.

□ 4. In the **You have no storage mounted** pane, click **Show advanced settings**, perform the following tasks:

- Leave the **Subscription** drop-down list entry set to its default value.
- In the **Cloud Shell region** drop-down list, select the Azure region matching or near the location where you intend to deploy resources in this exercise.
- In the **Resource group** section, select the Resource Group that has been created for you.
- In the **Storage account** section, ensure that the **Create new** option is selected and then, in the text box below, type a unique name consisting of a combination of between 3 and 24 characters and digits.
- In the **File share** section, ensure that the **Create new** option is selected and then, in the text box below, type cloudshell.
- Click the **Create storage** button.

□ 5. Wait for the **Cloud Shell** to finish its first-time setup procedures before you proceed to the next task.

□ 6. Create a key vault in a resource group by running the following commnnds.

> ⚠ **Note**: The VaultName must be unique therefore change **keyvault name** to somthing unique.

```
$ResourceGroup = (Get-AzResourceGroup).ResourceGroupName
New-AzKeyVault -VaultName 'keyvault name' -ResourceGroupName $ResourceGroup -Location 'eastus'
```

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

□ 7. In the Azure Portal open the **KeyVaultPSRG-GH81CIU1A8** Resource Group.

□ 8. Click on the Key Vault name to examine what you have created.

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

□ 9. Click **Access Policies** > **+ Add Access Policy**

□ 10. Select **Key, Secret and Certificate Management** from **Configure from template (optional)**

□ 11. Click **Select Principal** and search for and then click on your account, then click on **Select**

□ 12. Click **Add** and then **Save**

Task 3: Add a key and secret to Key Vault

□ 1. Return to the PowerShell window.

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

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

□ 3. Move back to **KeyVaultPS** in the Azure portal. Click **Keys** under Settings in the left navigation pane.

□ 4. Click **MyLabKey**

□ 5. Click the Current Version.

□ 6. Examine the information about the key you created.

> ⚠ **Note**: You can always reference this key by using its URI. To get the most current version, just reference **https://keyvaultps.vault.azure.net/keys/MyLabKey/** or if need be the exact version: **https://keyvaultps.vault.azure.net/keys/MyLabKey/da1a3a1efa5dxxxxxxxxxxxxxxd53c5959e**

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

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

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

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

Task 4: Add a Secret to Key Vault

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

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

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

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

□ 3. Move back to the Azure Portal on **KeyVaultPS** and click **Secrets**

□ 4. Click the Secret **SQLPassword**

□ 5. Click the current version

□ 6. Examine the Secret that you created

> **Note**: You can always reference this key by using its URI. To get the most current version just reference **https://keyvaultps.vault.azure.net/secrets/SQLPassword** or if need be the exact version: **https://keyvaultps.vault.azure.net/secrets/SQLPassword/c5aada85d3acxxxxxxxxxxxe8701efafcf3**

□ 7. Click the **Show secret value** button -- notice that the password appears.

□ 8. To view the Secret, use the Get-AzureKeyVaultSecret cmdlet. Be sure to change the placeholder text to your vault name.

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

Task 5: Enable a Client Application

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

□ 1. Open the Azure portal and navigate to Azure Active Directory.

□ 2. Click **App Registrations** under **Manage** in the left navigation pane.

□ 3. **Click + New registration**

□ 4. Provide a name such as **sqlApp** (Choose something unique) for your application. Under **Redirect URI (optional)**, select **Web**, and for the SIGN-ON URL type **https://sqlapp**

□ 5. Click **Register**.

□ 6. Once the App Registration is complete click on **sqlApp** if it does not automatically appear.

□ 7. Copy your Application (client) ID as you will need it later.

□ 8. Click **Certificates & secrets**

□ 9. Click **+ New client secret**

□ 10. In the **Description** section, enter **Key1** for the description. Select **1 year** from the **Expires** list, then click **Add**

□ 11. Copy the Key1 value as you will need it later. If you close and reopen the blade, the value will show as hidden.

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

□ 1. In the **Azure portal** open your **Resource Group**

□ 2. Select the **Azure Key** vault

□ 3. Click **Access Policies**

□ 4. Select the account associated with your Azure subscription

□ 5. In the **Key Permissions** drop down select **Select All** to highlight all permissions

□ 6. Select **Save**

> **Important**! You must click save otherwise the permissions will not be committed

□ 7. Run the following commands in Cloud Shell (Powershell) to set the sqlApp key permissions replacing the placeholder text with **your account details**

```
$applicationId = '[Azure_AD_Application_ID]'
$ResourceGroup = (Get-AzResourceGroup).ResourceGroupName
$location = 'eastus'
$vaultName = '[KeyVault_Name]'
```

```
Set-AZKeyVaultAccessPolicy -VaultName $vaultName -ResourceGroupName $ResourceGroup -ServicePrincipalName  $applicationId -PermissionsTok
```

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

Scenario

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

□ 1. From the Azure Portal click **+ Create a resource> Databases > SQL Database**

□ 2. Provide the following details on the SQL Database blade and click **Create**.

- Resource Group: select your Resource Group.
- Database Name: **medical**
- Server: **Create new**
  - Server name: **[Unique Server Name]**
  - Server Admin Login: **demouser**
  - Password: **Pa55w.rd1234**
  - Location: **[same location as KeyVaultPS]**
  - Then click **OK**
- Pricing Tier: Standard S0

□ 3. Once everything above is configured, select **Review + create,** then **Create**

□ 4. Once the SQL Database is deployed, open it in the Azure Portal to locate and then copy the **ADO.NET Connection String**.

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

Task 8: Create a Table in the SQL Database

□ 1. Use the Azure portal to locate the Server name where the Medical Database is located and copy the name.

□ 2. On this same blade click Set Server firewall.

□ 3. Next click + **Add client IP** and then click **Save**.

□ 4. Open SQL Server Management Studio. Connect to the Server using these properties for the **Connect to Server** dialog.

- Server Type: **Database Engine**
- Server Name: **[found on the Database Overview Blade]**
- Authentication: **SQL Server Authentication**
- Login: **demouser**
- Password: **Pa55w.rd1234**

Task 9: Create and Encrypt a Table

□ 1. In SQL Server Management Studio expand **Databases > Right-click medical > New Query**.

□ 2. Paste the following code into the query window and click Execute

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

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

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

    [FirstName] [nvarchar](50) NULL,

    [LastName] [nvarchar](50) NULL,

    [MiddleName] [nvarchar](50) NULL,

    [StreetAddress] [nvarchar](50) NULL,

    [City] [nvarchar](50) NULL,

    [ZipCode] [char](5) NULL,

    [State] [char](2) NULL,

    [BirthDate] [date] NOT NULL

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

☐ 3. After the table is created successfully, expand **medical > tables > right-click dbo.Patients** and select **Encrypt Columns**.

☐ 4. Click **Next**.

☐ 5. On the Column Selection Screen check **SSN** and **Birthdate**. Then set the Encryption Type for SSN to **Deterministic** and for Birthdate **Randomized**. Click **Next**.

☐ 6. On the Master Key Configuration page on the Select the Key store provider, click **Azure Key Vault.** Click **Sign in** and authenticate. Select your Azure Key Vault. Click **Next**.

☐ 7. On the Run Settings screen click **Next** and then **Finish** to Proceed with the encrypting.

☐ 8. When the encryption process is complete, click **Close** and expand **medical > security > Always Encrypted Keys** and note that now there are keys found.

✓ **Results**: You have now completed this Lab.