

## **Lab 10 (Key Vault Implementing Secure Data by setting up Always Encrypted) Report**

**Yunis Mohamed**

MICROSOFT AZURE Lab 10

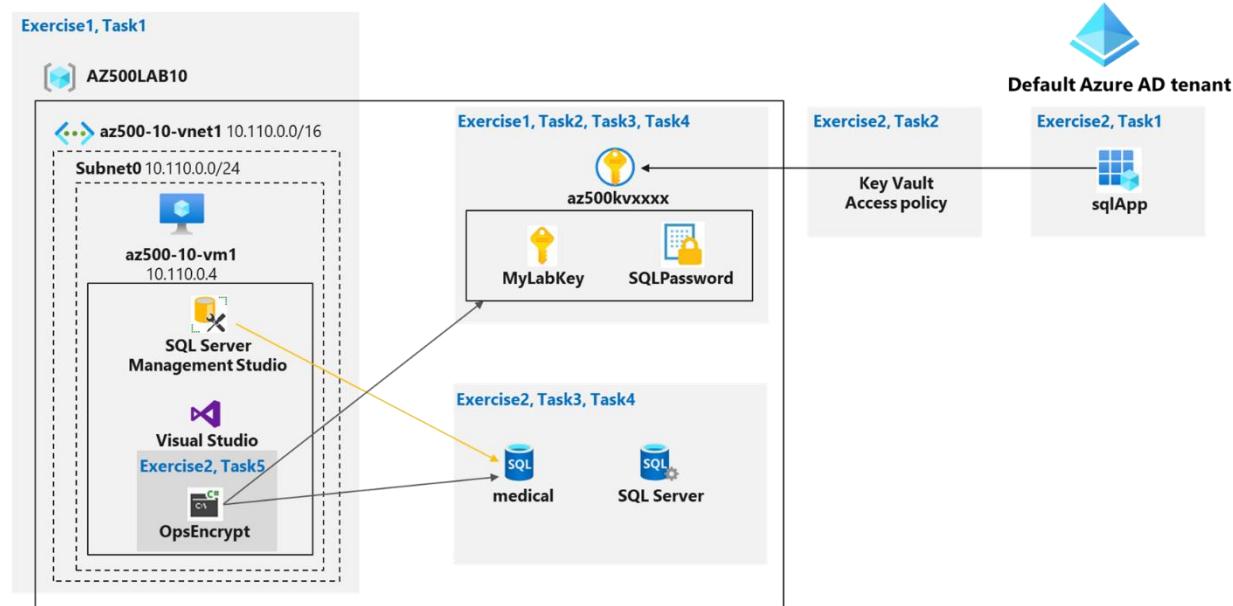
## Contents

<b>Introduction .....</b>	2
<b>Key Vault diagram.....</b>	2
<b>Exercise 1: Deploy the base infrastructure from an ARM template.....</b>	2
Task 1: Deploy an Azure VM and an Azure SQL database.....	2
<b>Exercise 2: Configure the Key Vault resource with a key and a secret.....</b>	4
Task 1: Create and configure a Key Vault .....	4
Task 2: Add a key to Key Vault .....	5
Task 3: Add a Secret to Key Vault .....	7
<b>Exercise 3: Configure an Azure SQL database and a data-driven application.....</b>	8
Task 1: Enable a client application to access the Azure SQL Database service. ....	8
Task 2: Create a policy allowing the application access to the Key Vault. ....	11
Task 3: Retrieve SQL Azure database ADO.NET Connection String .....	12
Task 4: Log on to the Azure VM running Visual Studio 2019 and SQL Management Studio 19 .....	12
Task 5: Create a table in the SQL Database and select data columns for encryption .....	13
<b>Exercise 4: Demonstrate the use of Azure Key Vault in encrypting the Azure SQL database.....</b>	17
Task 1: Run a data-driven application to demonstrate the use of Azure Key Vault in encrypting the Azure SQL database .....	17
<b>Conclusion.....</b>	21

# Introduction

This report offers a comprehensive overview and analysis of Lab 10, entitled "Key Vault: Implementing Secure Data by Setting up Always Encrypted," conducted within the Microsoft Azure environment. The lab's primary focus was to enhance data security through the effective utilization of Key Vault and the implementation of the Always Encrypted feature. Through an exploration of crucial concepts, techniques, and practical applications covered during the lab, this report overlays the importance of securing sensitive data and highlights the pivotal role played by Azure in accomplishing this objective. By examining the lab's objectives, methodology, and outcomes, this report aims to provide valuable insights into the practical implementation of secure data practices in Azure, specifically through the utilization of Key Vault and Always Encrypted.

## Key Vault diagram



## Exercise 1: Deploy the base infrastructure from an ARM template

In this exercise, I will complete the following tasks:

- Task 1: Deploy an Azure VM and an Azure SQL database

### Task 1: Deploy an Azure VM and an Azure SQL database

In this task, I will deploy an Azure VM, which will automatically install Visual Studio 2019 and SQL Server Management Studio 19 as part of the deployment.

1. Sign-in to the Azure portal <https://portal.azure.com/>.
2. In the Azure portal, in the Search resources, services, and docs text box at the top of the Azure portal page, type Deploy a custom template and press the Enter key.
3. On the Custom deployment blade, click the Build your own template in the editor option.
4. On the Edit template blade, click Load file, locate the \Allfiles\Labs\10\az-500-10\_azuredeploy.json file and click Open.
5. On the Edit template blade, click Save.

6. On the Custom deployment blade, under Deployment Scope ensure that the following settings are configured (leave any others with their default values):

The screenshot shows the Microsoft Azure 'Edit template' blade. At the top, there are four browser tabs: 'CS-CNS: Assignment 2: Key Vault', 'AZ500-AzureSecurityTechnologie', 'Edit template - Microsoft Azure', and '(2) AZ500 Azure Security Technol...'. The main content area has a blue header bar with the Microsoft Azure logo, an 'Upgrade' button, and a search icon. Below the header, the URL is 'portal.azure.com/#view/HubsExtension/TemplateEditorBladeV2/template/%7B%0A%20%20%20%20%20%24schema%3A%20"https%3A%2F...'. On the right side of the header, there are several icons: a user profile, a gear, a question mark, a help icon, a refresh icon, and a download icon. The main body of the page shows the title 'Edit template' and a sub-header 'Edit your Azure Resource Manager template'. There are buttons for '+ Add resource', 'Quickstart template', 'Load file', and 'Download'. To the left is a sidebar with sections for 'Parameters (2)', 'Variables (24)', and 'Resources (7)'. Under 'Resources', items listed include StorageAccount, PublicIPAddress, VirtualNetwork, NetworkInterface, JumpVM, and networkSecurityGroup. The main pane displays a JSON template:

```
1 {  
2   "$schema": "https://schema.management.azure.com/schemas/2019-04-01/deploymentTemplate.json#",  
3   "contentVersion": "1.0.0.0",  
4   "parameters": {  
5     "adminUsername": {  
6       "type": "string",  
7       "minLength": 1,  
8       "defaultValue": "Student",  
9       "metadata": {  
10         "description": "Username for the Virtual Machine."  
11       }  
12     },  
13     "adminPassword": {  
14       "type": "securestring",  
15       "defaultValue": "Pa55w.rd1234",  
16       "metadata": {  
17         "description": "Password for the Virtual Machine."  
18       }  
19     }  
20   }  
21 }
```

At the bottom, there are 'Save' and 'Discard' buttons. The taskbar at the very bottom includes icons for File Explorer, Edge, File, Mail, Task View, and Google Chrome, along with system status icons like battery level, signal strength, and volume.

7. Click the **Review and Create** button, and confirm the deployment by clicking the **Create** button.

CS-CNS: Assignment 2: Key Vault | AZ500-AzureSecurityTechnologie | Custom deployment - Microsoft | (2) AZ500 Azure Security Technol | +

portal.azure.com/#create/Microsoft.Template

Microsoft Azure    Upgrade    d

Home > Custom deployment

Deploy from a custom template

Select a template    Basics    Review + create

Summary

Customized template  
7 resources

Terms

Azure Marketplace Terms | Azure Marketplace

By clicking "Create," I (a) agree to the applicable legal terms associated with the offering; (b) authorize Microsoft to charge or bill my current payment method for the fees associated with the offering(s), including applicable taxes, with the same billing frequency as my Azure subscription, until I discontinue use of the offering(s); and (c) agree that, if the deployment involves 3rd party offerings, Microsoft may share my contact information and other details of such deployment with the publisher of that offering.

Microsoft assumes no responsibility for any actions performed by third-party templates and does not provide rights for third-party products or services. See the [Azure Marketplace Terms](#) for additional terms.

Deploying this template will create one or more Azure resources or Marketplace offerings. You acknowledge that you

Previous    Next    Create

Type here to search

27°C    ENG    447 PM    7/10/2023

## Exercise 2: Configure the Key Vault resource with a key and a secret

In this exercise, I will complete the following tasks:

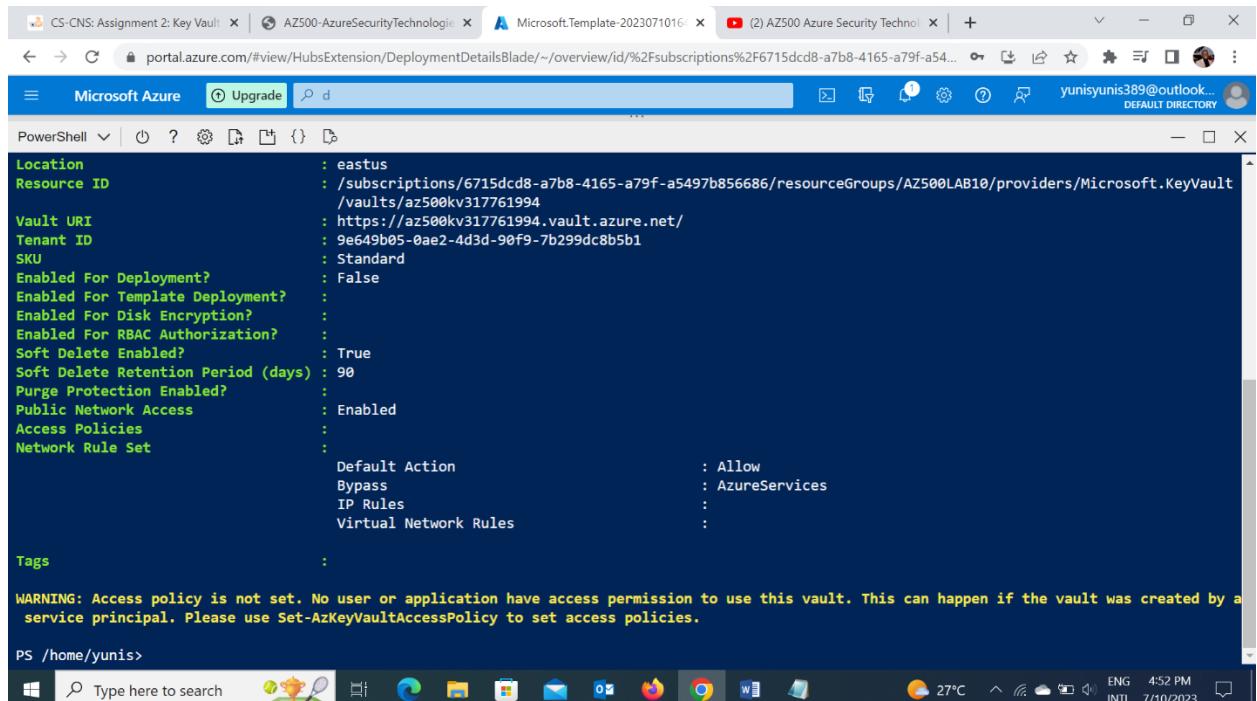
Task 1: Create and configure a Key Vault

Task 2: Add a key to the Key Vault

Task 3: Add a secret to the Key Vault

### Task 1: Create and configure a Key Vault

1. Open the Cloud Shell by clicking the first icon (next to the search bar) at the top right of the Azure portal. If prompted, select **PowerShell** and **Create storage**.
2. Ensure **PowerShell** is selected in the drop-down menu in the upper-left corner of the Cloud Shell pane.
3. In the PowerShell session within the Cloud Shell pane, run the following to create an Azure Key Vault in the resource group **AZ500LAB10**. (If you chose another name for this lab's Resource Group out of Task 1, use that name for this task as well). The Key Vault name must be unique. Remember the name you have chosen. You will need it throughout this lab.
4. Close the Cloud Shell pane.



The screenshot shows the Azure Cloud Shell interface. The PowerShell session displays the configuration of an Azure Key Vault. The output includes the location (eastus), resource ID, vault URI, tenant ID, SKU (Standard), and various deployment and access settings. A warning message at the bottom states: "WARNING: Access policy is not set. No user or application have access permission to use this vault. This can happen if the vault was created by a service principal. Please use Set-AzKeyVaultAccessPolicy to set access policies." The session ends with the command PS /home/yunis>.

```
Location : eastus
Resource ID : /subscriptions/6715dcdb-a7b8-4165-a79f-a5497b856686/resourceGroups/AZ500LAB10/providers/Microsoft.KeyVault/vaults/az500kv317761994
Vault URI : https://az500kv317761994.vault.azure.net/
Tenant ID : 9e649b05-0ae2-4d3d-90f9-7b299dc8b5b1
SKU : Standard
Enabled For Deployment? : False
Enabled For Template Deployment? :
Enabled For Disk Encryption? :
Enabled For RBAC Authorization? :
Soft Delete Enabled? : True
Soft Delete Retention Period (days) : 90
Purge Protection Enabled? :
Public Network Access : Enabled
Access Policies :
Network Rule Set :
    Default Action : Allow
    Bypass : AzureServices
    IP Rules :
    Virtual Network Rules :

Tags :

WARNING: Access policy is not set. No user or application have access permission to use this vault. This can happen if the vault was created by a service principal. Please use Set-AzKeyVaultAccessPolicy to set access policies.

PS /home/yunis>
```

5. In the Azure portal, in the Search resources, services, and docs text box at the top of the Azure portal page, type Resource groups and press the Enter key.

6. On the Resource groups blade, in the list of resource group, click the AZ500LAB10 (or other name you chose earlier for the resource group) entry.
7. On the Resource Group blade, click the entry representing the newly created Key Vault.
8. On the Key Vault blade, in the Overview section, click Access Policies and then click + Create.
9. On the Create an access policy blade, specify the following settings (leave all others with their default values):

The screenshot shows the 'Create an access policy' blade in the Azure portal. The blade is divided into several sections:

- Unwrap Key:** Contains checkboxes for Unwrap Key, Wrap Key, Verify, and Sign. 'Sign' is checked.
- Privileged Key Operations:** Contains checkboxes for Select all, Purge, and Release. None are checked.
- Rotation Policy Operations:** Contains checkboxes for Select all, Rotate, Get Rotation Policy, and Set Rotation Policy. All are checked.
- Certificate Operations:** Contains checkboxes for Delete Certificate Authorities, Select all, and Purge. 'Delete Certificate Authorities' is checked.

At the bottom of the blade are 'Previous' and 'Next' buttons. The browser address bar shows the URL for creating an access policy in the Azure portal.

## Task 2: Add a key to Key Vault

1. In the Azure portal, open a PowerShell session in the Cloud Shell pane.
2. Ensure **PowerShell** is selected in the upper-left drop-down menu of the Cloud Shell pane.
3. In the PowerShell session within the Cloud Shell pane, run the following to add a software-protected key to the Key Vault:
4. In the PowerShell session within the Cloud Shell pane, run the following to verify the key was created:
5. In the PowerShell session within the Cloud Shell pane, run the following to display the key identifier:
6. Minimize the Cloud Shell pane.

```
PS /home/yunis> $kv = Get-AzKeyVault -ResourceGroupName 'AZ500LAB10'
PS /home/yunis>
PS /home/yunis> $key = Add-AZKeyVaultKey -VaultName $kv.VaultName -Name 'MyLabKey' -Destination 'Software'
PS /home/yunis>
PS /home/yunis> Get-AZKeyVaultKey -VaultName $kv.VaultName

Vault/HSM Name : az500kv317761994
Name : MyLabKey
Version :
Id : https://az500kv317761994.vault.azure.net:443/keys/MyLabKey
Enabled : True
Expires :
Not Before :
Created : 7/10/2023 2:03:05 PM
Updated : 7/10/2023 2:03:05 PM
Recovery Level : Recoverable+Purgeable
Tags :
```

PS /home/yunis> \$key.key.kid  
https://az500kv317761994.vault.azure.net/keys/MyLabKey/4c6265a9897c495990ced0b67b62c290  
PS /home/yunis>

7. Back in the Azure portal, on the Key Vault blade, in the Objects section, click Keys.
8. In the list of keys, click the MyLabKey entry and then, on the MyLabKey blade, click the entry representing the current version of the key.

Home > Resource groups > AZ500LAB10 > az500kv317761994 | Keys >

**MyLabKey** ...

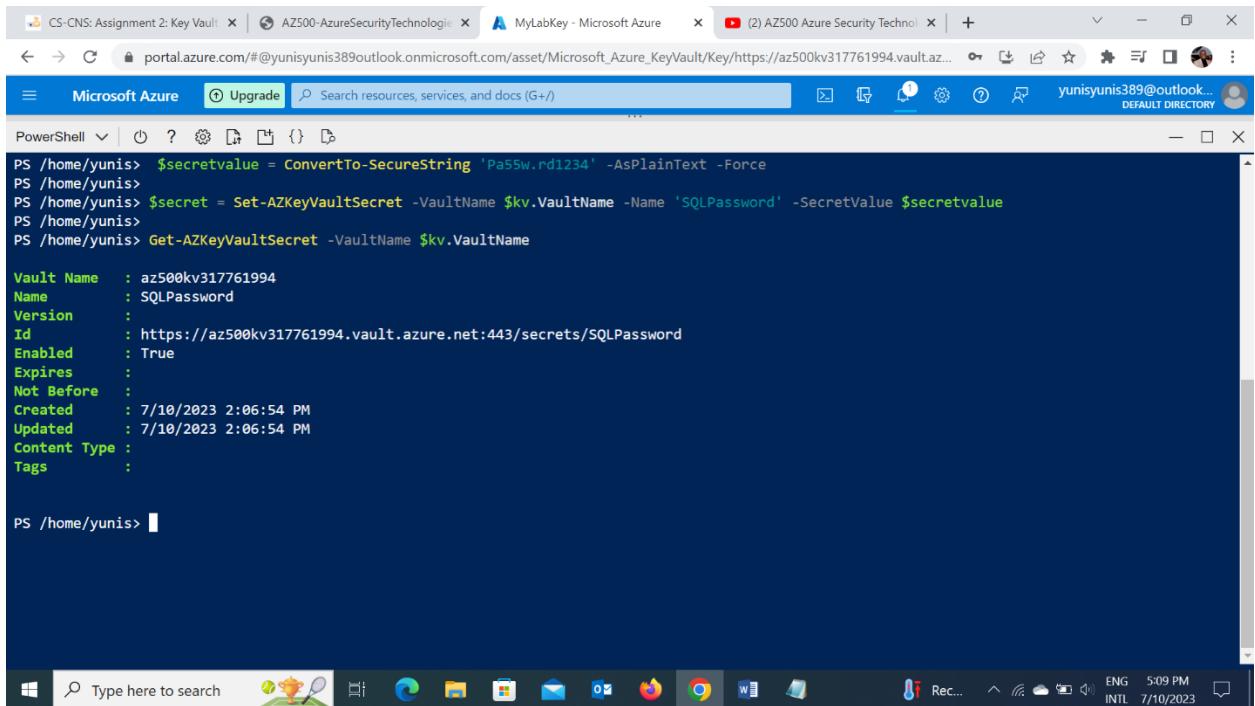
[New Version](#) [Refresh](#) [Delete](#) [Download Backup](#) [Rotation policy](#)

Version	Status	Activation date	Expiration date
<b>CURRENT VERSION</b>	✓ Enabled		
4c6265a9897c495990ced0b67b62c290			



### Task 3: Add a Secret to Key Vault

1. Switch back to the Cloud Shell pane.
2. In the PowerShell session within the Cloud Shell pane, run the following to create a variable with a secure string value:
3. In the PowerShell session within the Cloud Shell pane, run the following to add the secret to the vault:
4. In the PowerShell session within the Cloud Shell pane, run the following to verify the secret was created.
5. Minimize the Cloud Shell pane.



The screenshot shows the Microsoft Azure Cloud Shell interface. The title bar has four tabs: 'CS-CNS: Assignment 2: Key Vault', 'AZ500-AzureSecurityTechnologi...', 'MyLabKey - Microsoft Azure', and '(2) AZ500 Azure Security Techn...'. The main area is a terminal window titled 'PowerShell'. The user has run the following PowerShell commands:

```
PS /home/yunis> $secretvalue = ConvertTo-SecureString 'Pa55w.rd1234' -AsPlainText -Force
PS /home/yunis> $secret = Set-AZKeyVaultSecret -VaultName $kv.VaultName -Name 'SQLPassword' -SecretValue $secretvalue
PS /home/yunis> Get-AZKeyVaultSecret -VaultName $kv.VaultName

Vault Name : az500kv317761994
Name       : SQLPassword
Version    :
Id        : https://az500kv317761994.vault.azure.net:443/secrets/SQLPassword
Enabled   : True
Expires   :
Not Before :
Created   : 7/10/2023 2:06:54 PM
Updated   : 7/10/2023 2:06:54 PM
Content Type :
Tags      :

PS /home/yunis>
```

The terminal window is dark-themed. Below it is a taskbar with various icons and a search bar. The system tray shows the date and time as '7/10/2023 5:09 PM'.

6. In the Azure portal, navigate back to the Key Vault blade, in the Objects section, click Secrets.
7. In the list of secrets, click the SQLPassword entry and then, on the SQLPassword blade, click the entry representing the current version of the secret.

## Exercise 3: Configure an Azure SQL database and a data-driven application

In this exercise, I will complete the following tasks:

- Task 1: Enable a client application to access the Azure SQL Database service.
- Task 2: Create a policy allowing the application access to the Key Vault.
- Task 3: Retrieve SQL Azure database ADO.NET Connection String
- Task 4: Log on to the Azure VM running Visual Studio 2019 and SQL Management Studio 19
- Task 5: Create a table in the SQL Database and select data columns for encryption

### Task 1: Enable a client application to access the Azure SQL Database service.

In this task, I will enable a 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.

1. In the Azure portal, in the **Search resources, services, and docs** text box at the top of the Azure portal page, type **App Registrations** and press the **Enter** key.
2. On the **App Registrations** blade, click **+ New registration**.
3. On the **Register an application** blade, specify the following settings (leave all others with their default values):

4. On the **Register an application** blade, click **Register**.

The screenshot shows the 'Register an application' blade in the Microsoft Azure portal. The 'Name' field contains 'sqlApp'. Under 'Supported account types', the first option ('Accounts in this organizational directory only (Default Directory only - Single tenant)') is selected. At the bottom, there is a 'Register' button.

5. On the **sqlApp** blade, identify the value of **Application (client) ID**.

The screenshot shows the 'sqlApp' blade in the Microsoft Azure portal. The 'Overview' tab is selected, displaying the 'Application (client) ID' as 'bd314421-d681-4fdb-bf0f-e67f2...'. The 'Manage' sidebar is open, showing various configuration options. A message at the top right encourages feedback on the Microsoft identity platform.

6. On the **sqlApp** blade, in the **Manage** section, click **Certificates & secrets**.

7. On the **sqlApp Certificates & secrets** blade / Client Secrets section, click **+ New client secret**.

8. In the **Add a client secret** pane, specify the following settings:

9. Click Add to update the application credentials.

The screenshot shows the Microsoft Azure portal interface. The left sidebar is titled 'sqlApp | Certificates & secrets'. The main content area is titled 'Add a client secret'. The 'Description' field is set to 'Key1' and the 'Expires' field is set to '365 days (12 months)'. A note at the bottom of the dialog says: 'Certificates enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.' Below the dialog, the 'Client secrets' tab is selected in the navigation bar, showing '(0)' entries. A 'New client secret' button is visible. The status bar at the bottom right shows '5:24 PM 7/10/2023'.

10. On the \*\*sqlApp

Certificates & secrets\*\* blade, identify the value of Key1.

The screenshot shows the Microsoft Azure portal interface. The left sidebar is titled 'sqlApp | Certificates & secrets'. The main content area is titled 'Certificates & secrets'. The 'Client secrets' tab is selected, showing '(1)' entry. The table below lists the client secret: 

Description	Expires	Value	Secret ID
Key1	7/9/2024	FVH8Q~epS29yXZN~p_XsaHvV_d3M...	88553e10-dc63-40dd-9d3d-69400906...

 A note at the top of the blade says: 'Certificates enable confidential applications to identify themselves to the authentication service when receiving tokens at a web addressable location (using an HTTPS scheme). For a higher level of assurance, we recommend using a certificate (instead of a client secret) as a credential.' The status bar at the bottom right shows '5:27 PM 7/10/2023'.

## Task 2: Create a policy allowing the application access to the Key Vault.

In this task, I will grant the newly registered app permissions to access secrets stored in the Key Vault.

1. In the Azure portal, open a PowerShell session in the Cloud Shell pane.
2. Ensure **PowerShell** is selected in the upper-left drop-down menu of the Cloud Shell pane.
3. In the PowerShell session within the Cloud Shell pane, run the following to create a variable storing the **Application (client) ID** you recorded in the previous task (replace the <Azure\_AD\_Application\_ID> placeholder with the value of the **Application (client) ID**):
4. In the PowerShell session within the Cloud Shell pane, run the following to create a variable storing the Key Vault name.

```
$kvName = (Get-AzKeyVault -ResourceGroupName 'AZ500LAB10').VaultName
```

```
$kvName
```

5. In the PowerShell session within the Cloud Shell pane, run the following to grant permissions on the Key Vault to the application you registered in the previous task:

```
Set-AZKeyVaultAccessPolicy -VaultName $kvName -ResourceGroupName AZ500LAB10  
-ServicePrincipalName $applicationId -PermissionsToKeys  
get,wrapKey,unwrapKey,sign,verify,list
```

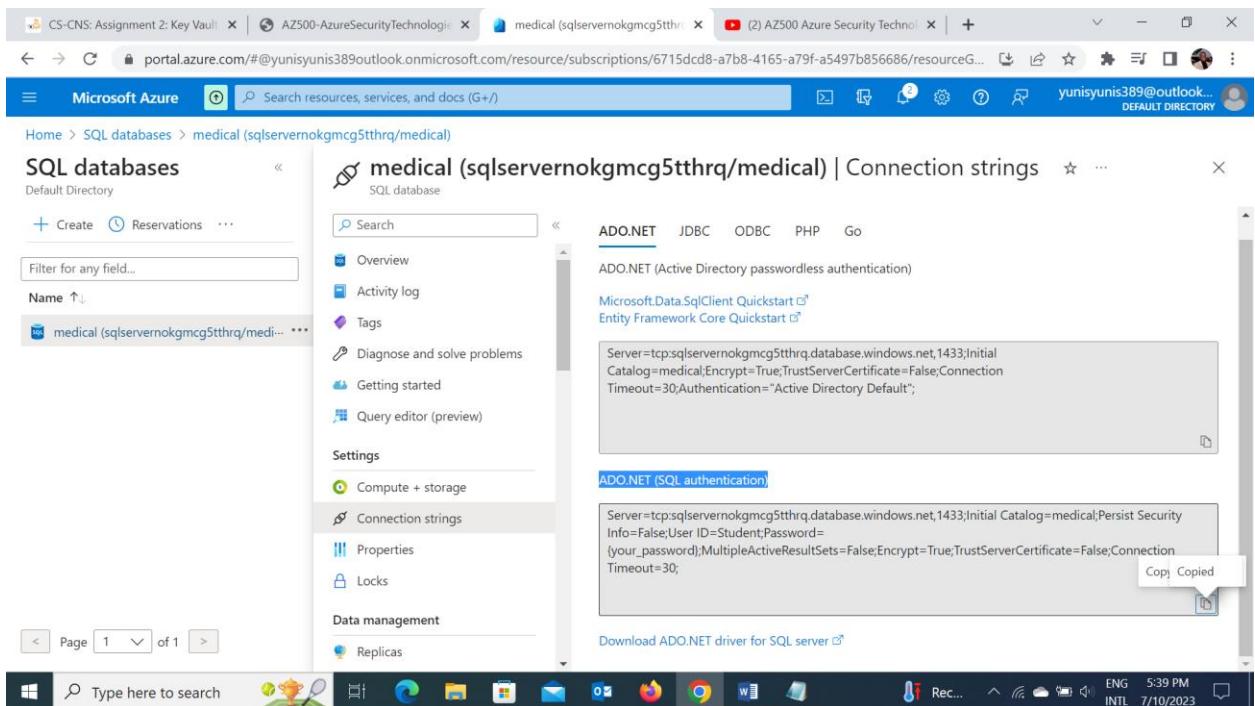
6. Close the Cloud Shell pane.

```
PS /home/yunis> $applicationId = 'bd314421-d681-4dfb-bf0f-e67f24fdb53'  
PS /home/yunis> $kvName = (Get-AzKeyVault -ResourceGroupName 'AZ500LAB10').VaultName  
PS /home/yunis>  
PS /home/yunis> $kvName  
az500kv317761994  
PS /home/yunis>  
PS /home/yunis> Set-AZKeyVaultAccessPolicy -VaultName $kvName -ResourceGroupName AZ500LAB10 -ServicePrincipalName $applicationId -PermissionsToKeys  
get,wrapKey,unwrapKey,sign,verify,list  
PS /home/yunis>
```

### Task 3: Retrieve SQL Azure database ADO.NET Connection String

The ARM-template deployment in Exercise 1 provisioned an Azure SQL Server instance and an Azure SQL database named **medical**. You will update the empty database resource with a new table structure and select data columns for encryption

1. In the Azure portal, in the **Search resources, services, and docs** text box at the top of the Azure portal page, type **SQL databases** and press the **Enter** key.
2. In the list of SQL databases, click the \*\*medical () \*\* entry.
3. On the SQL database blade, in the **Settings** section, click **Connection strings**.
4. Record the **ADO.NET (SQL authentication)** connection string. You will need it later.



### Task 4: Log on to the Azure VM running Visual Studio 2019 and SQL Management Studio 19

In this task, you log on to the Azure VM, which deployment you initiated in Exercise 1. This Azure VM hosts Visual Studio 2019 and SQL Server Management Studio 19.

1. In the Azure portal, in the **Search resources, services, and docs** text box at the top of the Azure portal page, type **virtual machines** and press the **Enter** key.
2. In the list of Virtual Machines shown, select the **az500-10-vm1** entry. On the **az500-10-vm1** blade, on the **Essentials** pane, take note of the **Public IP address**. You will use this later.

The screenshot shows the Microsoft Azure portal interface. The user is in the 'Virtual machines' section. A specific VM, 'az500-10-vm1', is selected. The 'Essentials' panel on the right provides detailed information about the VM, including its resource group (AZ500LAB10), operating system (Windows Server 2019 Datacenter), and public IP address (52.149.210.101). Other tabs like 'Overview', 'Activity log', and 'Tags' are also visible.

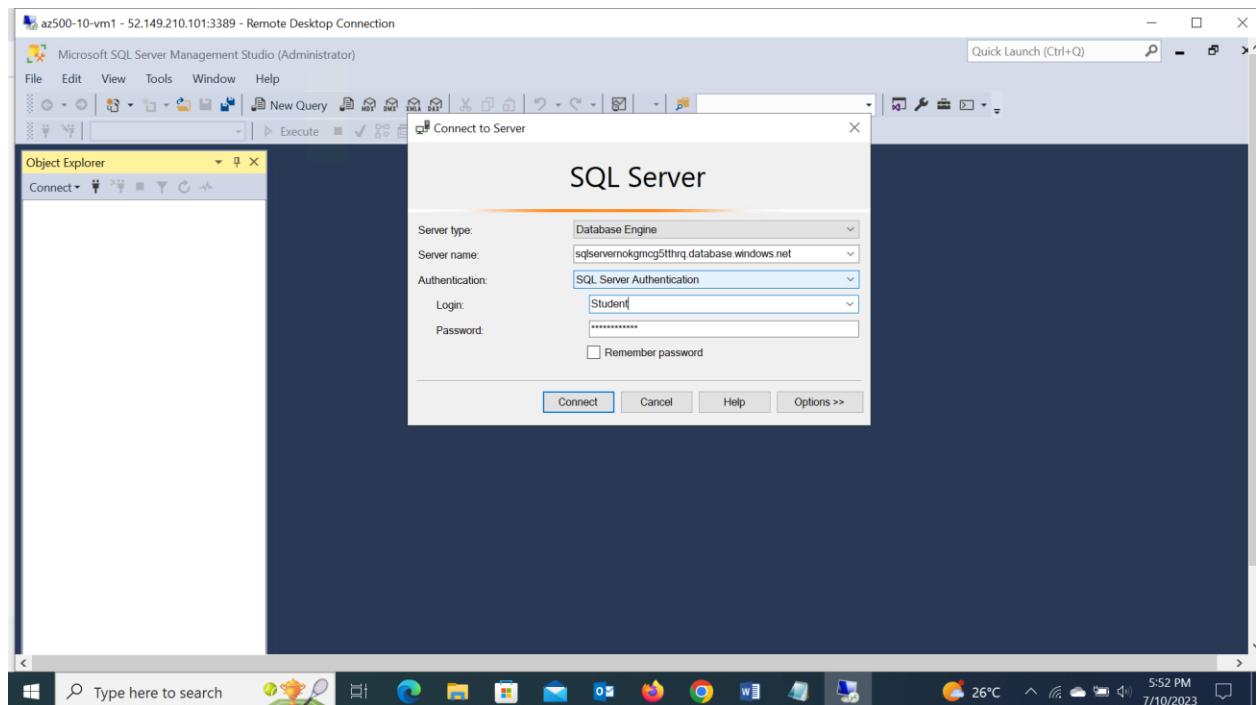
**Task 5: Create a table in the SQL Database and select data columns for encryption**

In this task, I will connect to the SQL Database with SQL Server Management Studio and create a table. You will then encrypt two data columns using an auto generated key from the Azure Key Vault.

1. In the Azure portal, navigate to the blade of the **medical** SQL database, in the **Essentials** section, identify the **Server name** (copy to clipboard), and then, in the toolbar, click **Set server firewall**.
2. On the **Firewall settings** blade, scroll down to Rule Name, click **+ Add a firewall rule**, and specify the following settings:
3. Click **Save** to save the change and close the confirmation pane.

The screenshot shows the 'Firewall rules' blade for the 'medical' SQL database. A new rule is being added, with the 'Rule name' set to 'Allow Mgmt VM' and the 'Start IP' and 'End IP' both set to '52.149.210.101'. The 'Exceptions' section includes a checked option for 'Allow Azure services and resources to access this'. The 'Add a firewall rule' dialog is open in the foreground.

4. Navigate back to the az500-10-vm1 blade, click Overview, next click Connect and, in the drop down menu, click RDP.
5. Click Download RDP File and use it to connect to the az500-10-vm1 Azure VM via Remote Desktop. When prompted to authenticate, provide the following credentials:
6. Click Start, in the Start menu, expand the Microsoft SQL Server Tools 19 folder, and click the Microsoft SQL Server Management Studio menu item.
7. In the Connect to Server dialog box, specify the following settings:
8. In the Connect to Server dialog box, click Connect.



9. Within the SQL Server Management Studio console, in the Object Explorer pane, expand the Databases folder.
10. In the Object Explorer pane, right-click the medical database and click New Query.
11. Paste the following code into the query window and click Execute. This will create a Patients table.

```

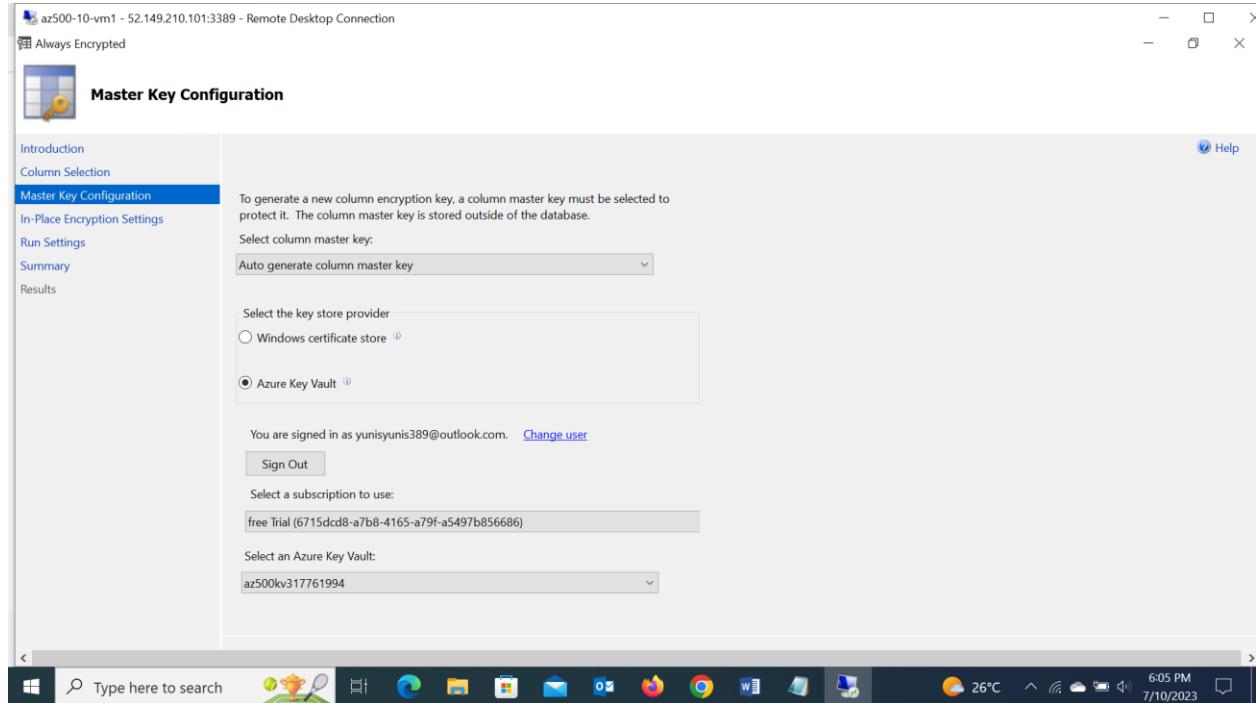
CREATE TABLE [dbo].[Patients](
    [PatientId] [int] IDENTITY(1,1),
    [SSN] [char](11) NOT NULL,
    [FirstName] [nvarchar](50) NULL,
    [LastName] [nvarchar](50) NULL,
    [MiddleName] [nvarchar](50) NULL,
    [StreetAddress] [nvarchar](50) NULL,
    [City] [nvarchar](50) NULL,
    [ZipCode] [char](5) NULL,
    [State] [char](2) NULL,
    [BirthDate] [date] NOT NULL
PRIMARY KEY CLUSTERED ([PatientId] ASC) ON [PRIMARY]
)

```

12. After the table is created successfully, in the **Object Explorer** pane, expand the **medical** database node, the **Tables** node, right-click the **dbo.Patients** node, and click **Encrypt Columns**.
13. On the Introduction page, click Next.
14. On the Column Selection page, select the SSN and Birthdate columns, set the Encryption Type of the SSN column to Deterministic and of the Birthdate column to Randomized, and click Next.

Name	State	Encryption Type	Encryption Key
dbo.Patients			
PatientId			
SSN		Deterministic	CEK_Auto1 (New)
FirstName			
LastName			
MiddleName			
StreetAddress			
City			
ZipCode			
State			
BirthDate		Randomized	CEK_Auto1 (New)

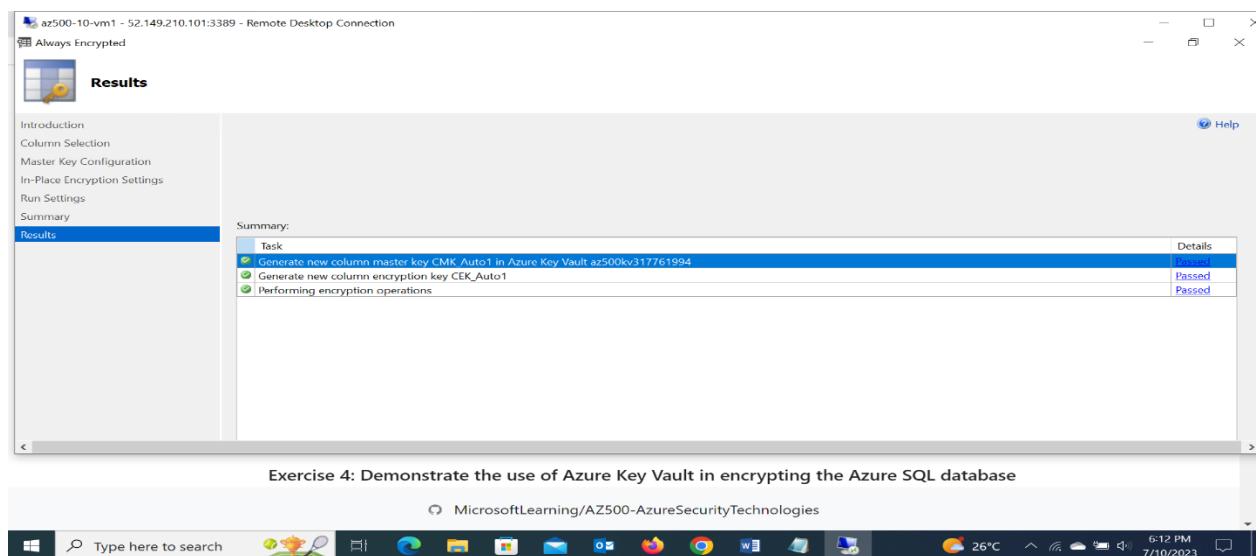
15. On the Master Key Configuration page, select Azure Key Vault, click Sign in, when prompted, authenticate by using the same user account you used to provision the Azure Key Vault instance earlier in this lab, ensure that that Key Vault appears in the Select an Azure Key Vault drop down list, and click Next.



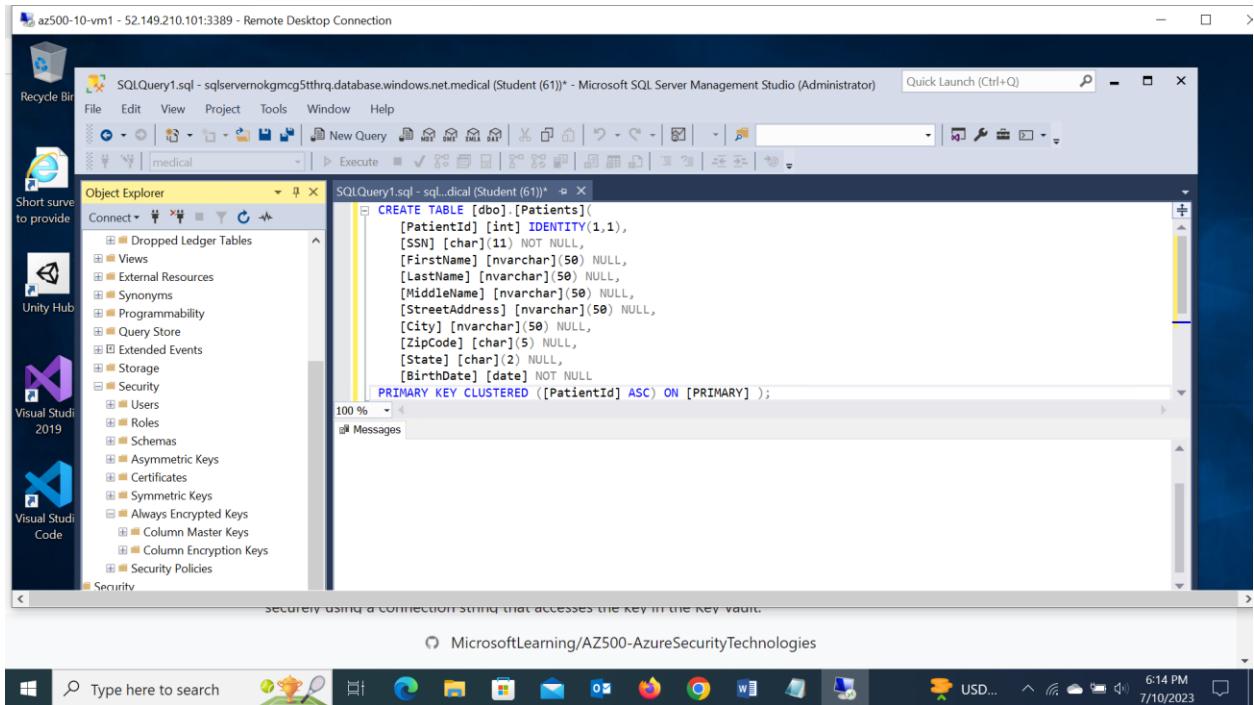
16. On the Run Settings page, click Next.

17. On the Summary page, click Finish to proceed with the encryption. When prompted, sign in again by using the same user account you used to provision the Azure Key Vault instance earlier in this lab.

18. Once the encryption process is complete, on the Results page, click Close.



19. In the SQL Server Management Studio console, in the Object Explorer pane, under the medical node, expand the Security and Always Encrypted Keys subnodes.



## Exercise 4: Demonstrate the use of Azure Key Vault in encrypting the Azure SQL database

In this exercise, I will complete the following tasks:

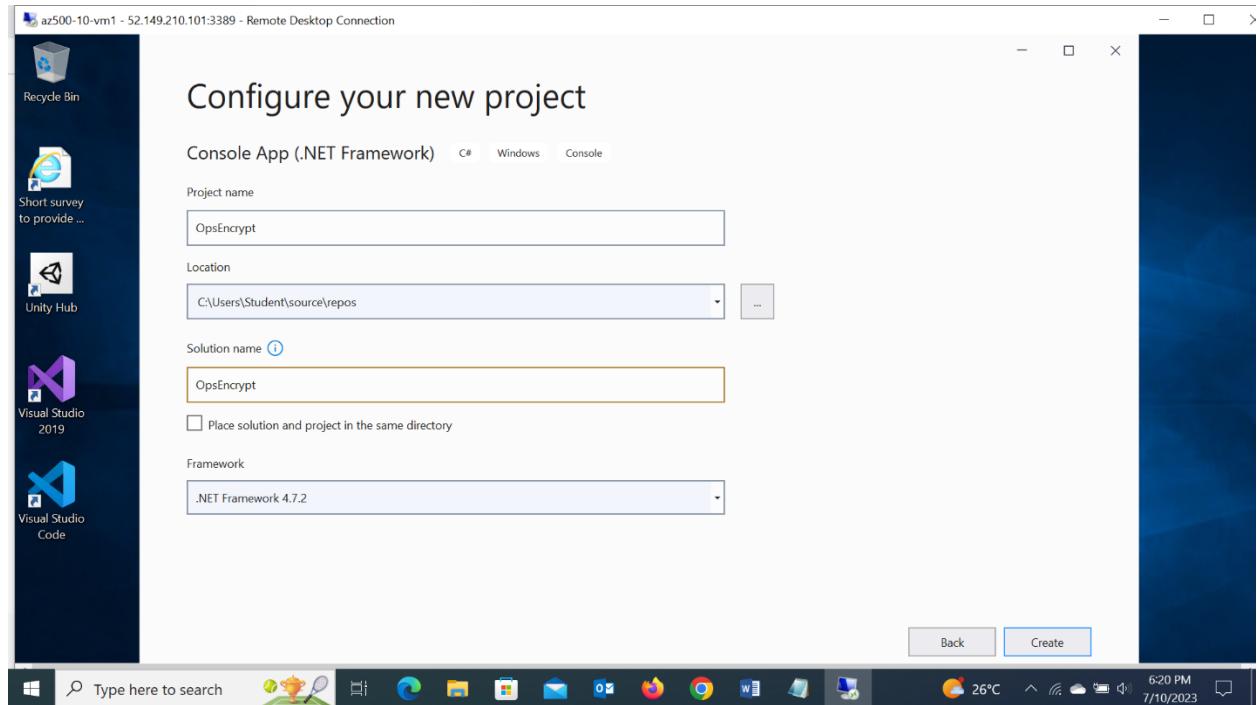
- Task 1: Run a data-driven application to demonstrate the use of Azure Key Vault in encrypting the Azure SQL database.

### Task 1: Run a data-driven application to demonstrate the use of Azure Key Vault in encrypting the Azure SQL database

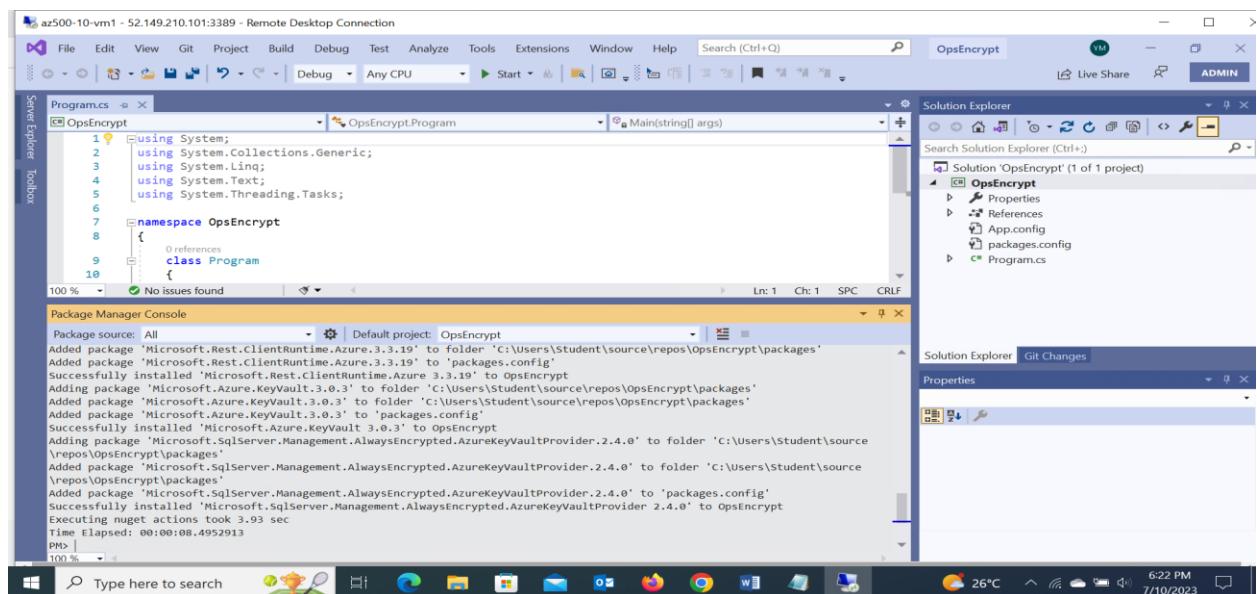
You will create a Console application using Visual Studio to load data into the encrypted columns and then access that data securely using a connection string that accesses the key in the Key Vault.

1. From the RDP session to the **az500-10-vm1**, launch **Visual Studio 2019** from the **Start menu**.
2. Switch to the window displaying Visual Studio 2019 welcome message, click the **Sign in** button and, when prompted, provide the credentials you used to authenticate to the Azure subscription you are using in this lab.
3. On the **Get started** page, click **Create a new project**.
4. In the list of project templates, search for **Console App (.NET Framework)**, in the list of results, click **Console App (.NET Framework)** for C#, and click **Next**.

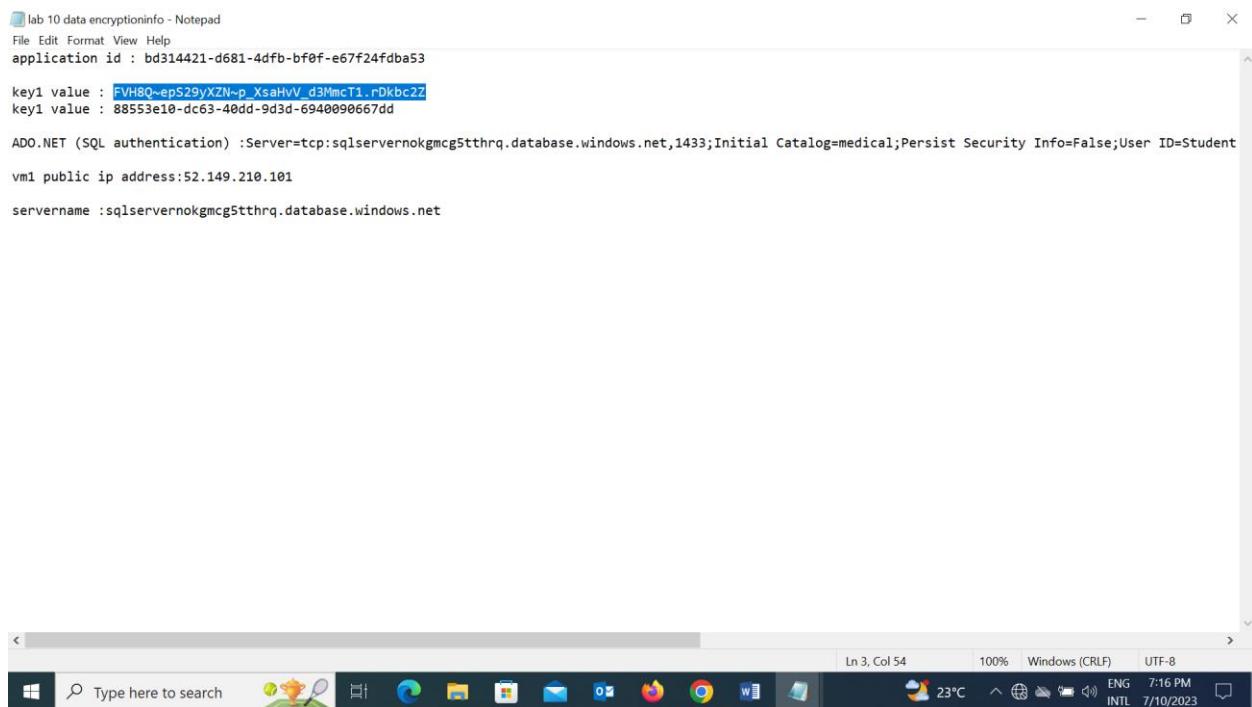
- On the **Configure your new project** page, specify the following settings (leave other settings with their default values), then click **Create**:



- In the Visual Studio console, click the Tools menu, in the drop down menu, click NuGet Package Manager, and, in the cascading menu, click Package Manager Console.
- In the Package Manager Console pane, run the following to install the first required NuGet package:
- In the **Package Manager Console** pane, run the following to install the second required **NuGet** package:



9. Minimize the RDP session to your Azure virtual machine, then navigate to \Allfiles\Labs\10\program.cs, open it in Notepad, and copy its content into Clipboard.
10. Return to the RDP session, and in the Visual Studio console, in the Solution Explorer window, click Program.cs and replace its content with the code you copied into Clipboard.
11. In the Visual Studio window, in the Program.cs pane, in line 15, replace the <connection string noted earlier> placeholder with the Azure SQL database ADO.NET connection string you recorded earlier in the lab. In the connection string, replace the {your\_password} placeholder, with the password that you specified in the deployment in Exercise 1. If you saved the string on the lab computer, you may need to leave the RDP session to copy the ADO string, then return to the Azure virtual machine to paste it in.
12. In the Visual Studio window, in the Program.cs pane, in line 16, replace the <client id noted earlier> placeholder with the value of Application (client) ID of the registered app you recorded earlier in the lab.
13. In the Visual Studio window, in the Program.cs pane, in line 17, replace the <key value noted earlier> placeholder with the the value of Key1 of the registered app you recorded earlier in the lab.
14. Minimize the RDP session to your Azure virtual machine, then navigate to \Allfiles\Labs\10\program.cs, open it in Notepad, and copy its content into Clipboard.



```
lab 10 data encryptioninfo - Notepad
File Edit Format View Help
application id : bd314421-d681-4dfb-bf0f-e67f24fdbba53

key1 value : FVH8Q~ep529yXZN~p_XsaHvV_d3MmcT1.rDkbc22
key1 value : 88553e10-dc63-40dd-9d3d-6940090667dd

ADO.NET (SQL authentication) :Server=tcp:sqlservernokgmcg5tthrq.database.windows.net,1433;Initial Catalog=medical;Persist Security Info=False;User ID=Student
vm1 public ip address:52.149.210.101
servername :sqlservernokgmcg5tthrq.database.windows.net
```

15. Return to the RDP session, and in the Visual Studio console, in the Solution Explorer window, click Program.cs and replace its content with the code you copied into Clipboard.
16. In the Visual Studio window, in the Program.cs pane, in line 15, replace the <connection string noted earlier> placeholder with the Azure SQL database ADO.NET connection string you recorded earlier in the lab. In the connection string, replace the {your\_password} placeholder, with the password that you specified in the deployment in Exercise 1. If you saved the string on the lab computer, you may need to leave the RDP session to copy the ADO string, then return to the Azure virtual machine to paste it in.
17. In the Visual Studio window, in the Program.cs pane, in line 16, replace the <client id noted earlier> placeholder with the value of Application (client) ID of the registered app you recorded earlier in the lab.
18. In the Visual Studio window, in the Program.cs pane, in line 17, replace the <key value noted earlier> placeholder with the the value of Key1 of the registered app you recorded earlier in the lab.

```

az500-10-vm1 - 52.149.210.101:3389 - Remote Desktop Connection
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Search (Ctrl+Q) OpsEncrypt VM Live Share ADMIN
Program.cs* AlwaysEncryptedConsoleAKVApp.Program clientSecret
Server Explorer Toolbox
Solution Explorer
Search Solution Explorer (Ctrl+.)
System.Runtime.Serialization
System.Security.Cryptography.Algorithms
System.Security.Cryptography.Encoding
System.Security.Cryptography.Primitives
System.Security.Cryptography.X509Certificate
System.Windows.Forms
System.Xml
System.Xml.Linq
App.config
packages.config
Program.cs
Solution Explorer Git Changes Properties
Package Manager Console
Type here to search 7/10/2023 6:40 PM

```

```

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Data;
7  using System.Data.SqlClient;
8  using Microsoft.IdentityModel.Clients.ActiveDirectory;
9  using Microsoft.SqlServer.Management.AlwaysEncrypted.AzureKeyVaultProvider;
10 namespace AlwaysEncryptedConsoleAKVApp
11 {
12     // Update this line with your Medical database connection string from the Azure portal.
13     static string connectionString = @"ADO.NET (SQL authentication) :Server=tcp:sqlservernokgmcg5tthr
14     static string clientId = @"bd314421-d681-4dfb-bf0f-e67f24fdb53";
15     static string clientSecret = "FVH8Q-epS29yXZN-p_XsaHv_V_d3MmcT1.rDkbc2Z";
16
17     static void Main(string[] args)
18     {
19         InitializeAzureKeyVaultProvider();
20         Console.WriteLine("Signed in as: " + _clientCredential.ClientId);
21         Console.WriteLine("Original connection string copied from the Azure portal:");
22         Console.WriteLine(connectionString);
23         // Create a SqlConnectionStringBuilder
24         SqlConnectionStringBuilder connStringBuilder =
25             new SqlConnectionStringBuilder(connectionString);
26         // Enable Always Encrypted for the connection

```

19. Switch back to the console application where you are prompted to enter a valid SSN. This will query the encrypted column for the data. At the Command Prompt, type the following and press the Enter key:

```
Signed in as: bd314421-d681-4dfb-bf0f-e67f24fdbba5
Original connection string copied from the Azure portal:
Server=tcp:sqlservernokgmrg5tthrq.database.windows.net,1433;Initial Catalog=medical;Persist Security Info=False;User ID=student;Password=Pa55w.rd1234;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;
sqlservr
Data updated connection string with Always Encrypted enabled:
Data Source=tcp:sqlservernokgmrg5tthrq.database.windows.net,1433;Initial Catalog=medical;Persist Security Info=False;User ID=student;Password=Pa55w.rd1234;MultipleActiveResultSets=False;Connect Timeout=30;Encrypt=True;TrustServerCertificate=False;Column Encryption Setting=Enabled
Enter server password:
Pa55w.rd1234
Adding sample patient data to the database...
All the records currently in the Patients table:
Orlando Gee   SSN: 999-99-0001   Birthdate: 1/4/1964 12:00:00 AM
Keith Harris   SSN: 999-99-0002   Birthdate: 6/20/1977 12:00:00 AM
Donna Carreras SSN: 999-99-0003   Birthdate: 2/9/1973 12:00:00 AM
Janet Gates    SSN: 999-99-0004   Birthdate: 8/31/1985 12:00:00 AM
Lucy Harrington SSN: 999-99-0005   Birthdate: 5/6/1993 12:00:00 AM
Now lets locate records by searching the encrypted SSN column.
Please enter a valid SSN (ex. 999-99-0003):
999-99-0003
Patient found with SSN = 999-99-0003
Donna Carreras SSN: 999-99-0003   Birthdate: 2/9/1973 12:00:00 AM
Press Enter to exit...
```

20. To terminate the console app, press the Enter key

## Conclusion

This lab has provided me with a comprehensive understanding of how to enhance data security by utilizing the Key Vault service and implementing Always Encrypted technology. Throughout the lab, I was able to explore the step-by-step process of setting up a Key Vault in Azure and configuring it to securely store cryptographic keys. By using the Key Vault, I learned how to centralize key management and establish a secure repository for encryption keys, reducing the risk of unauthorized access to sensitive data.

The lab also emphasized the integration of Azure Active Directory (Azure AD) with Key Vault, enabling fine-grained access control and ensuring that only authorized users and applications can interact with the cryptographic keys. By utilizing the Azure AD's role-based access control (RBAC) and assigning appropriate permissions, I was able to enforce the principle of least privilege and maintain strict control over key management.

In conclusion, this lab has provided me with a practical and learning experience in bolstering data security using Key Vault and Always Encrypted. The lab has effectively equipped me with the essential knowledge and practical skills required to protect sensitive data and address the potential dangers linked to unauthorized access.