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

Yunis Mohamed

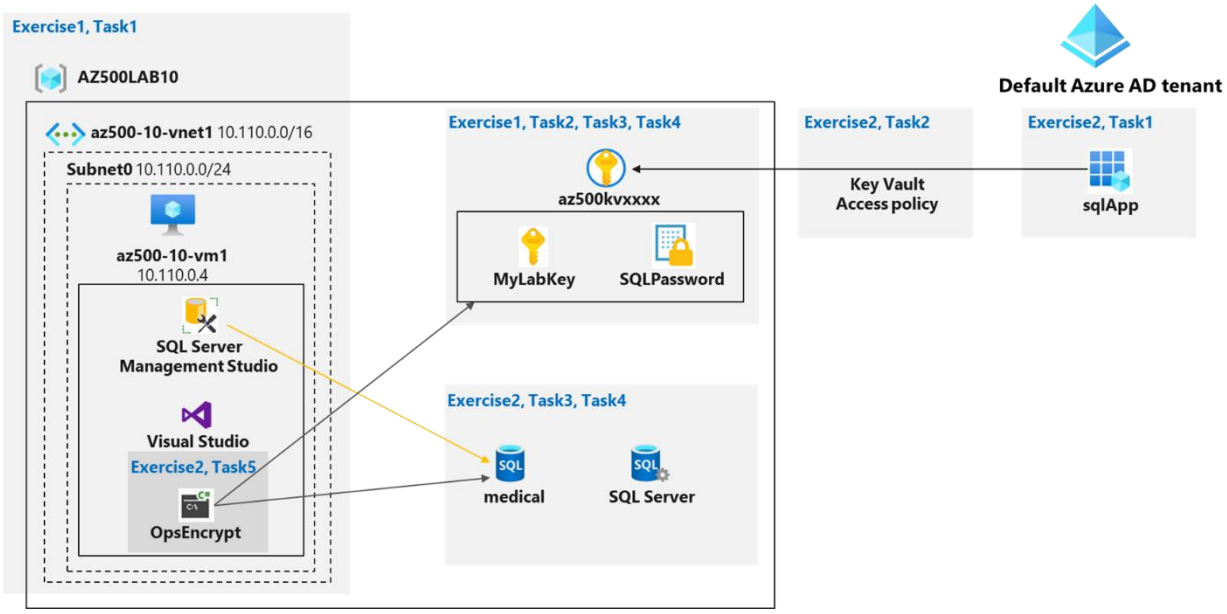
MICROSOFT AZURE Lab 10

Contents

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:

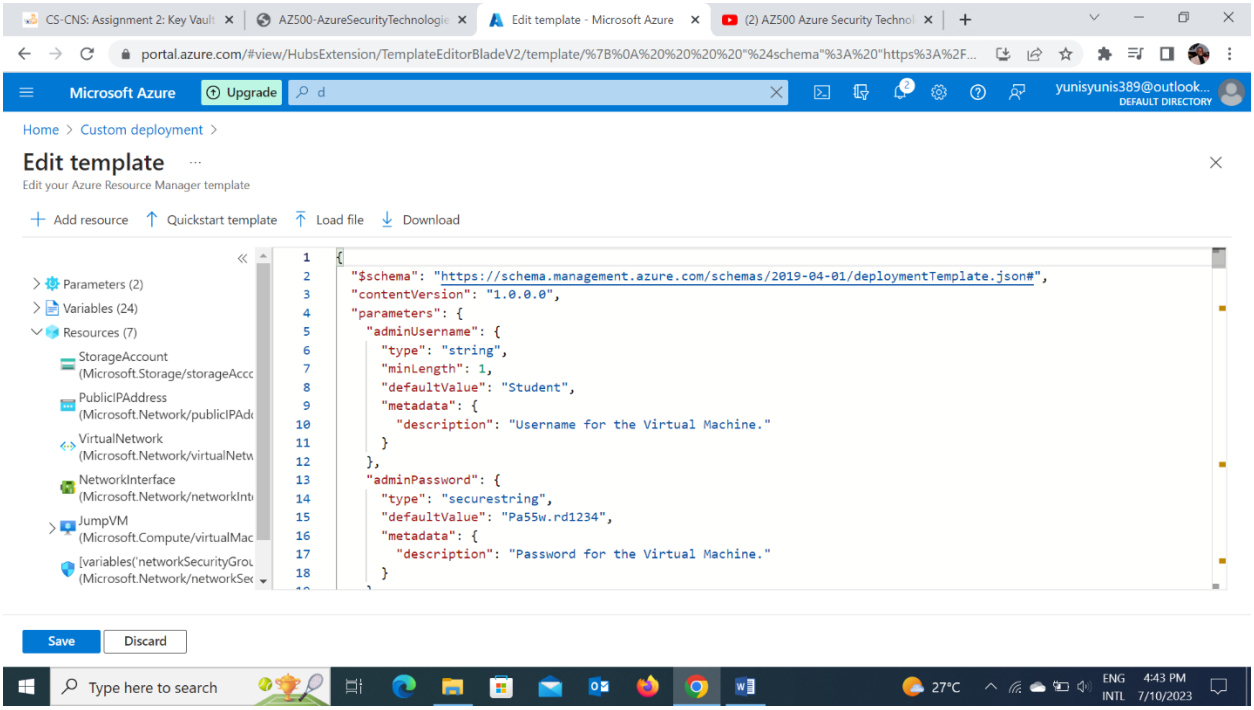
- Task 1: Deploy an Azure VM and an Azure SQL database

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

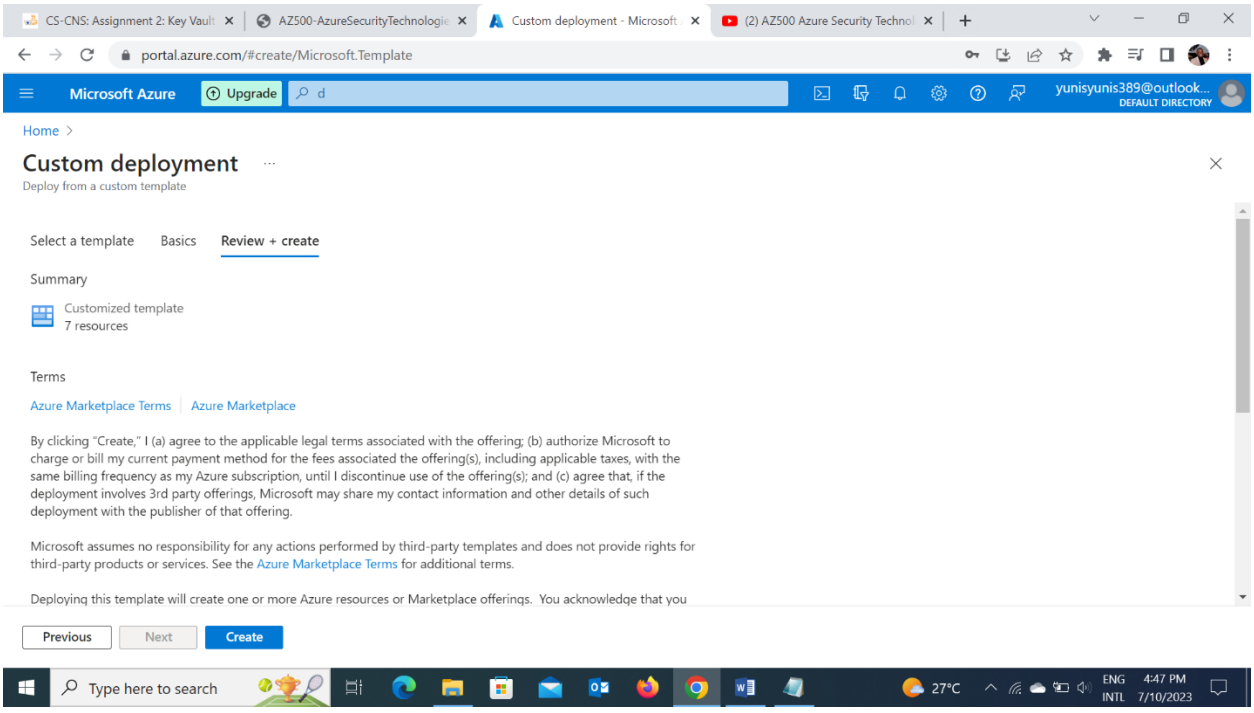
In this task, I will deploy an Azure VM, which will automatically install Visual Studio 2019 and SQL Server Management Studio 19 as part of the deployment.

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):



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



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

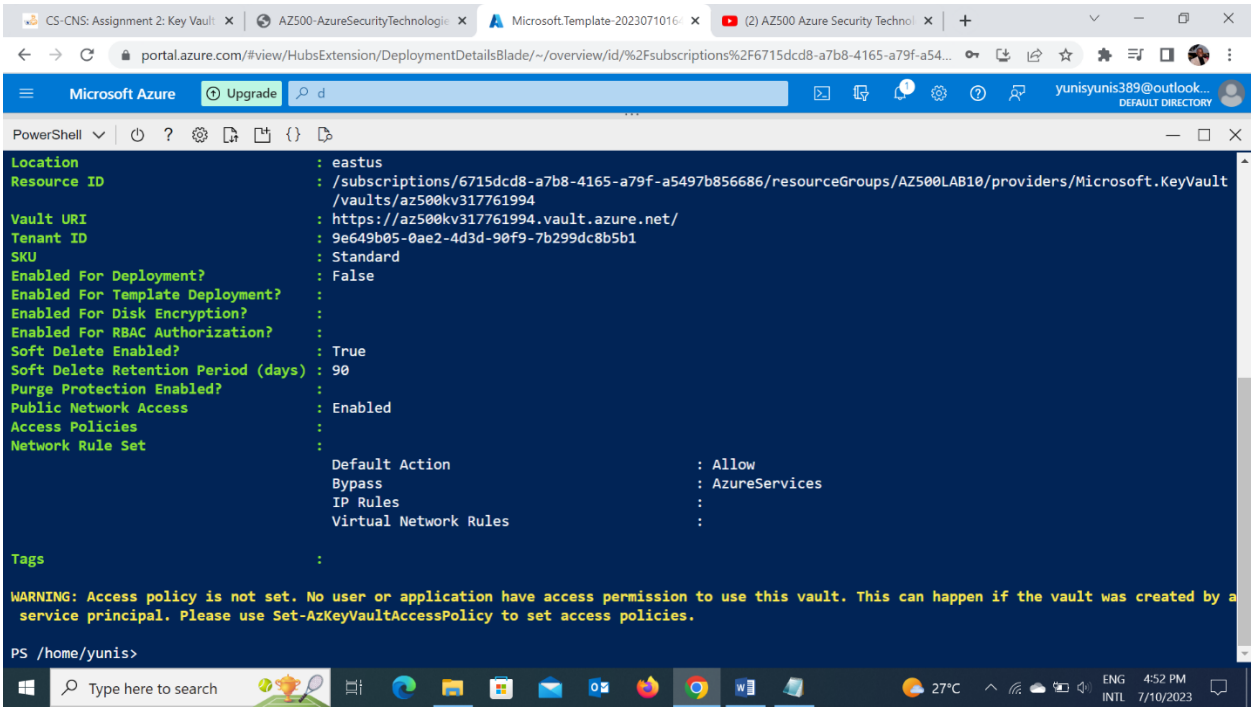
Task 1: Create and configure a Key Vault

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.

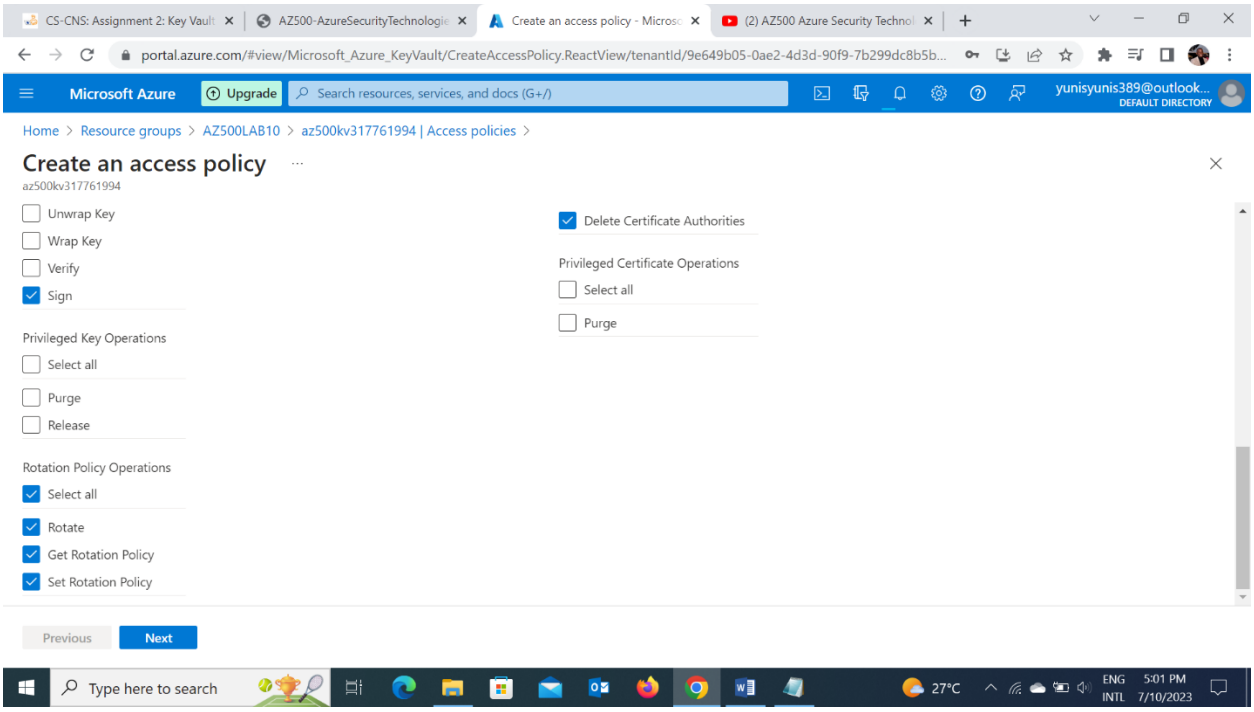
3. In the PowerShell session within the Cloud Shell pane, run the following to create an Azure Key Vault in the resource group **AZ500LAB10**. (If you chose another name for this lab's Resource Group out of Task 1, use that name for this task as well). The Key Vault name must be unique. Remember the name you have chosen. You will need it throughout this lab.

4. Close the Cloud Shell pane.



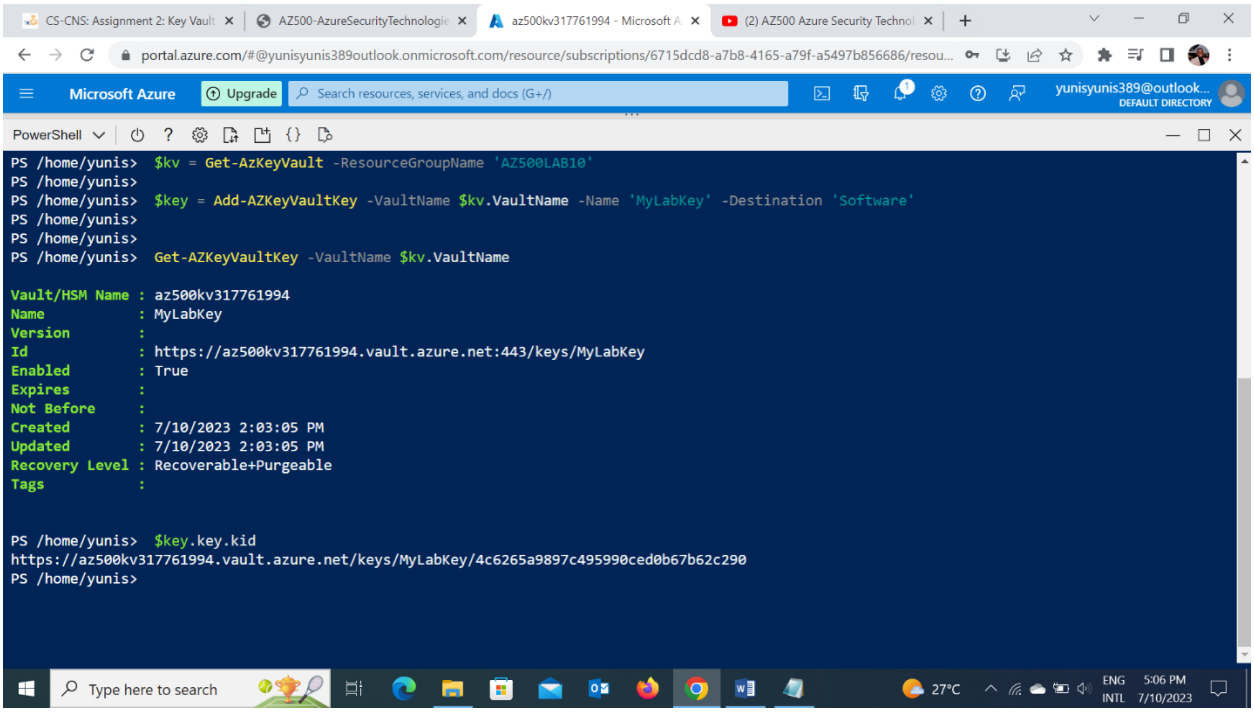
5. In the Azure portal, in the Search resources, services, and docs text box at the top of the Azure portal page, type Resource groups and press the Enter key.

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):

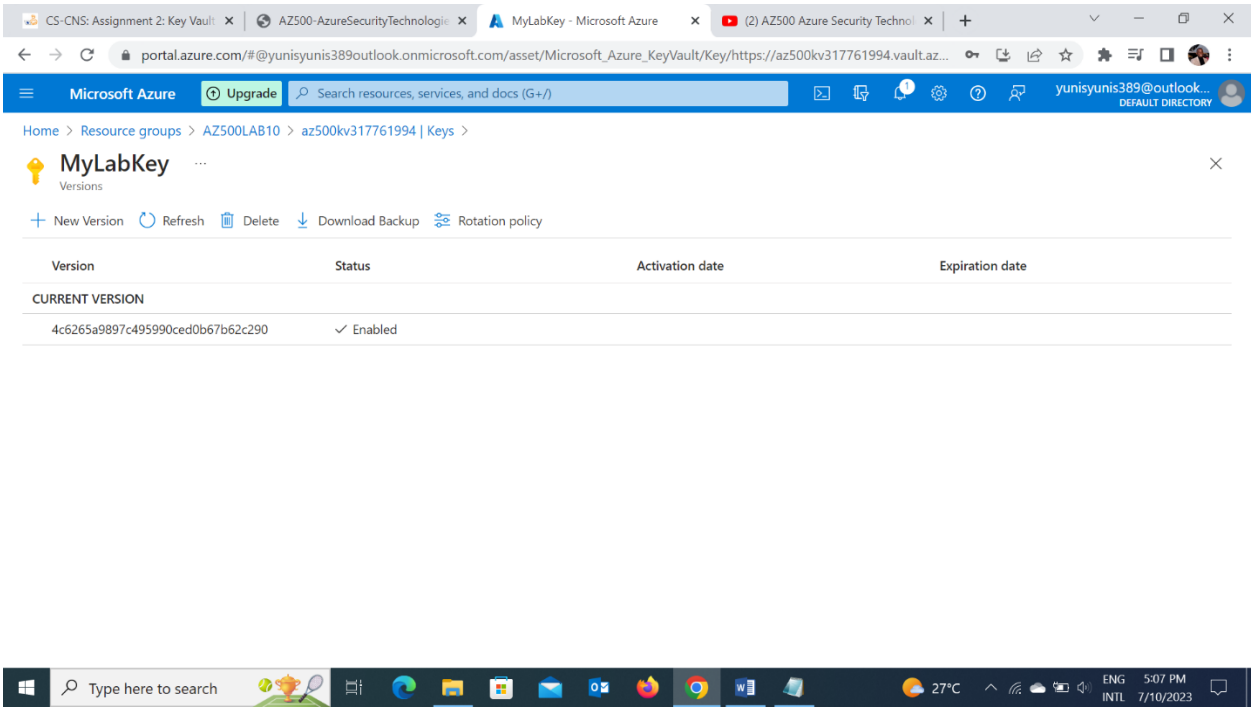


Task 2: Add a key to Key Vault

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.

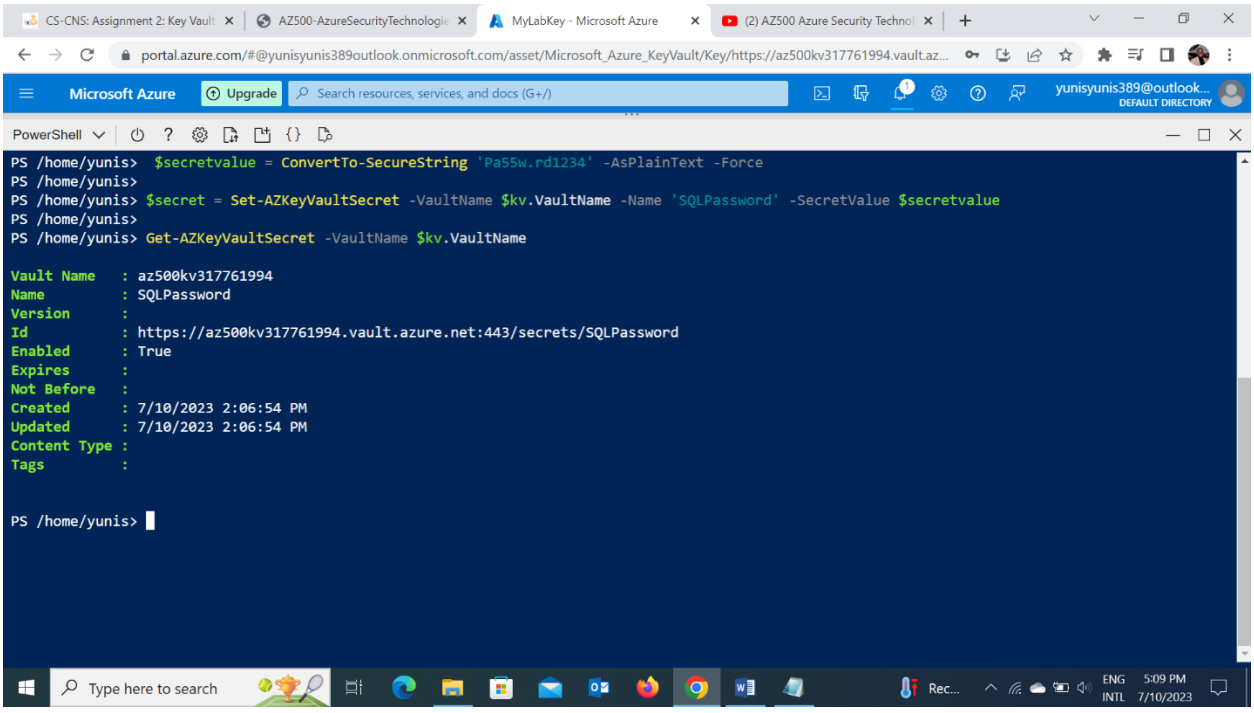


7. Back in the Azure portal, on the Key Vault blade, in the Objects section, click Keys.
8. In the list of keys, click the MyLabKey entry and then, on the MyLabKey blade, click the entry representing the current version of the key.

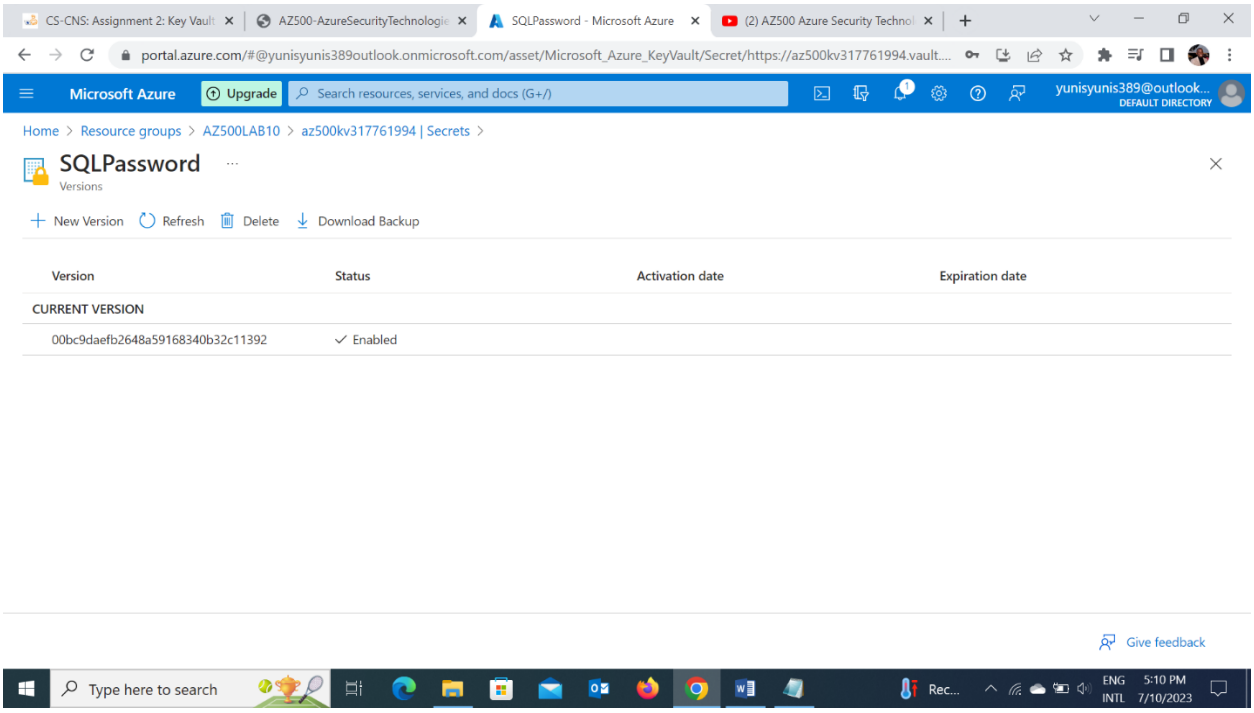


Task 3: Add a Secret to Key Vault

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.



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:

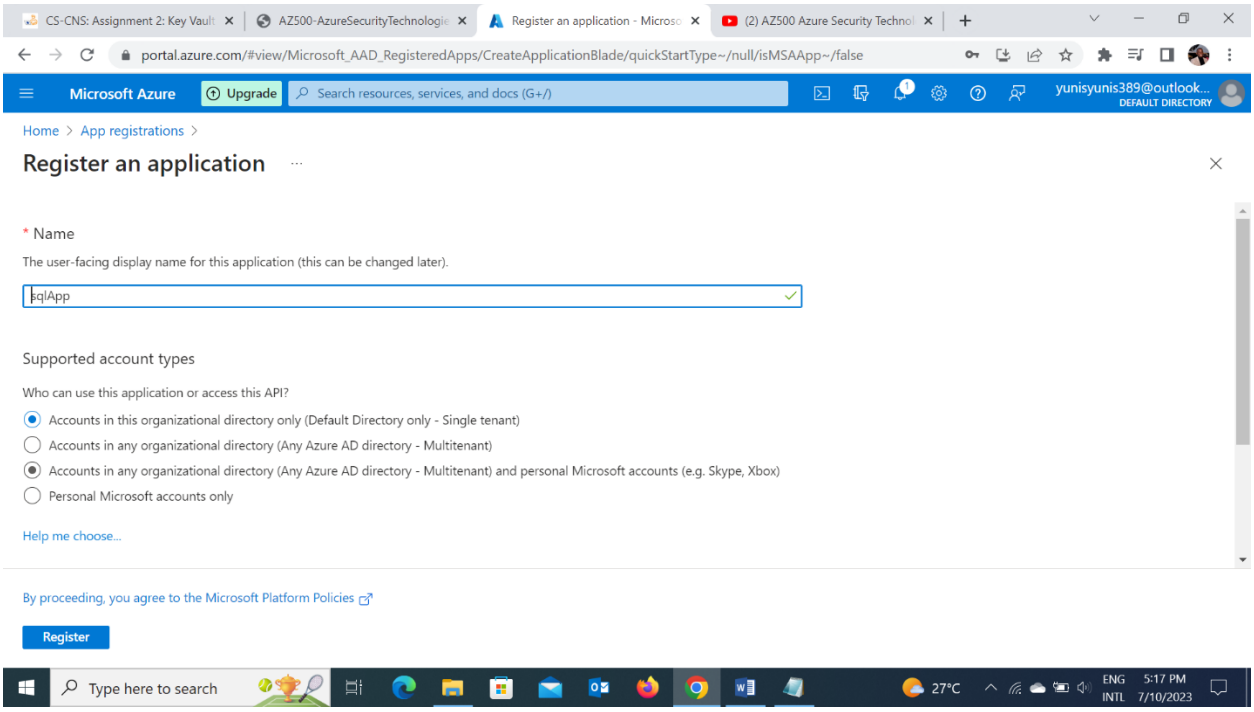
- Task 1: Enable a client application to access the Azure SQL Database service.
- Task 2: Create a policy allowing the application access to the Key Vault.
- Task 3: Retrieve SQL Azure database ADO.NET Connection String
- Task 4: Log on to the Azure VM running Visual Studio 2019 and SQL Management Studio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

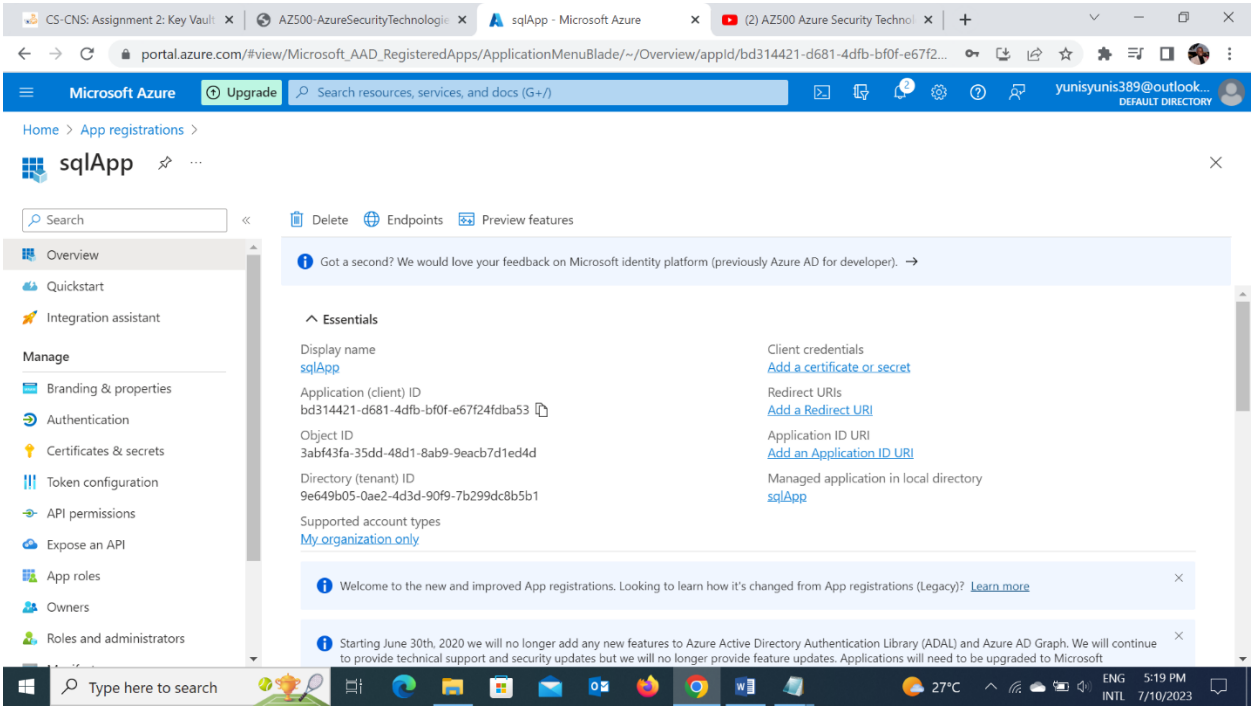
In this task, I will enable a client application to access the Azure SQL Database service. This will be done by setting up the required authentication and acquiring the Application ID and Secret that you will need to authenticate your application.

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**.



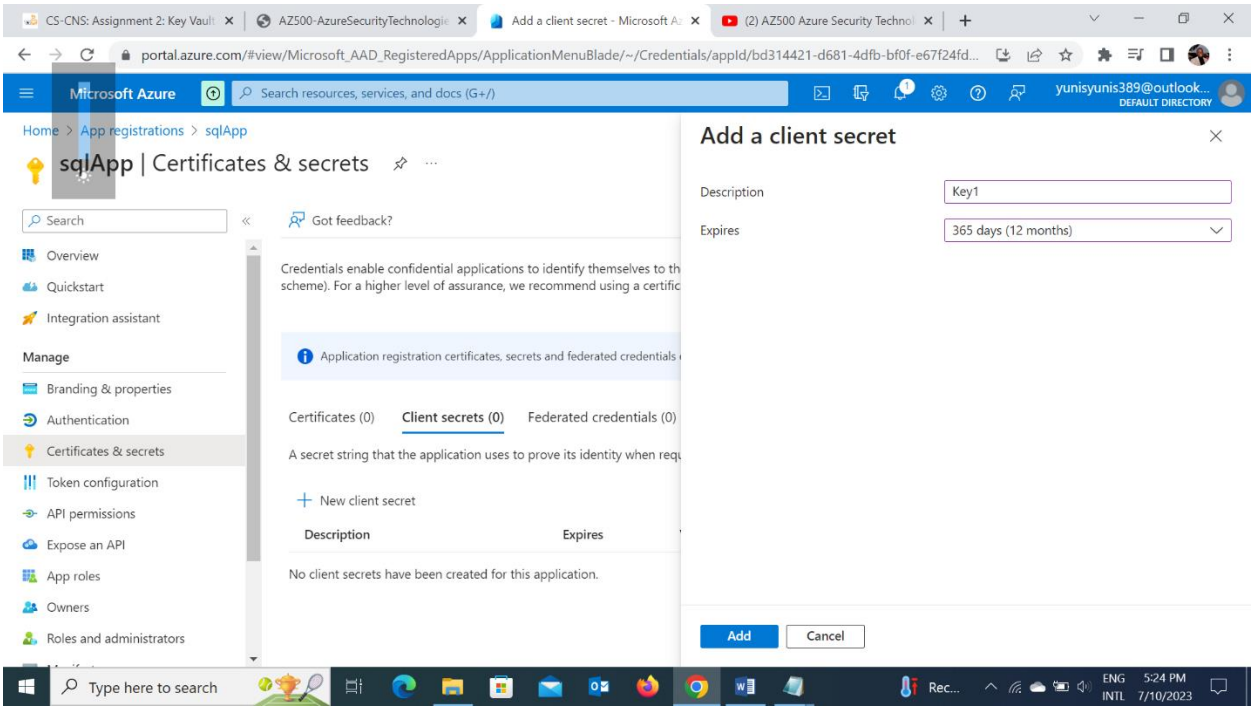
5. On the **sqlApp** blade, identify the value of **Application (client) ID**.



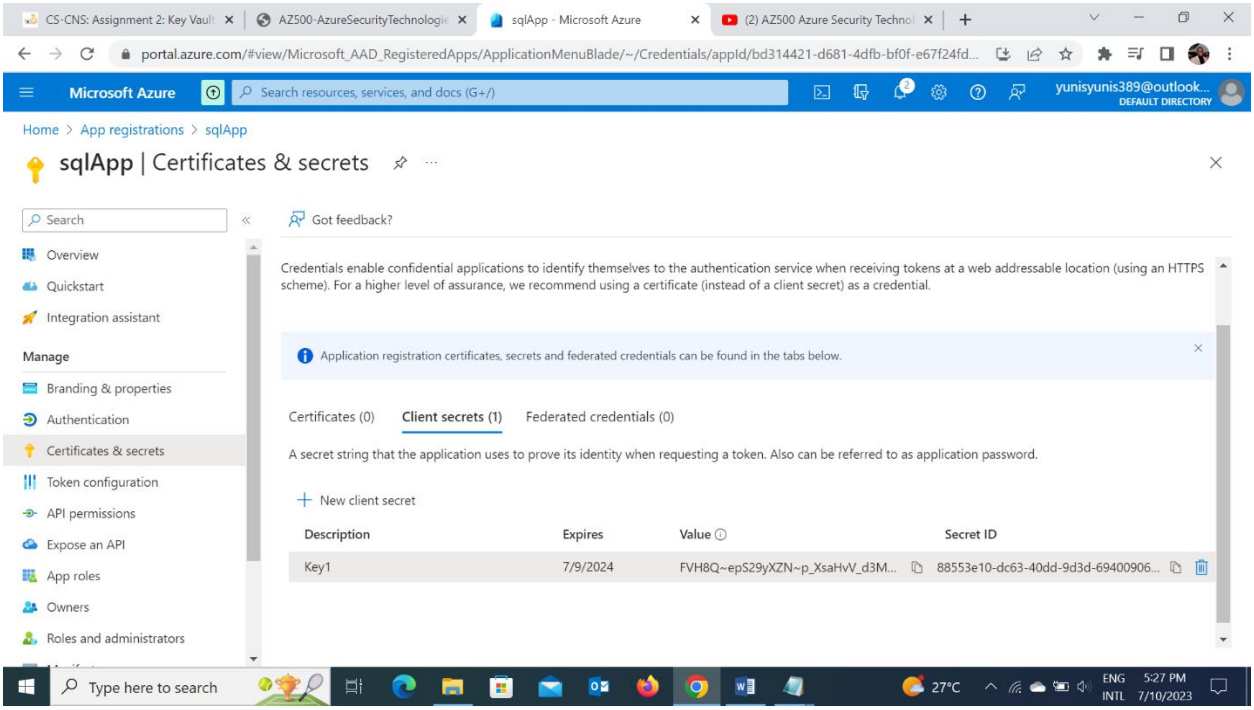
6. On the sqlApp blade, in the Manage section, click Certificates & secrets.

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.



10. On the **sqlApp Certificates & secrets** blade, identify the value of Key1.



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.

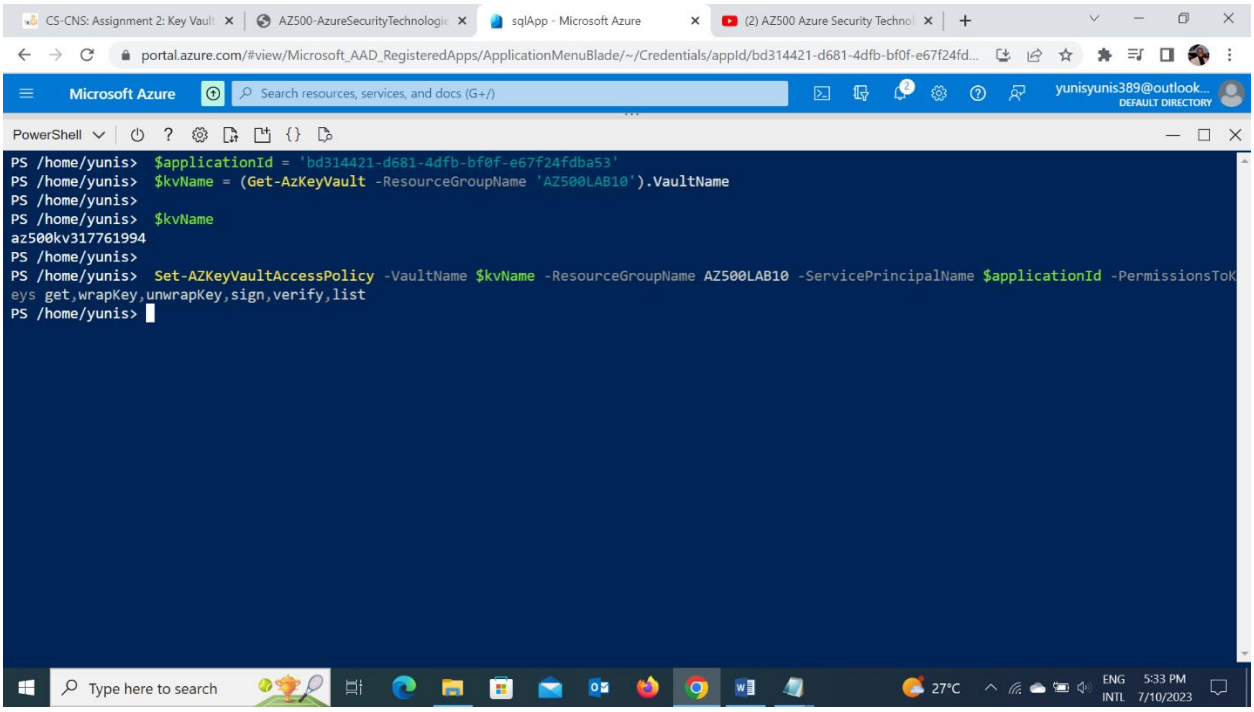
```
$kvName = (Get-AzKeyVault -ResourceGroupName 'AZ500LAB10').VaultName
```

```
$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
```

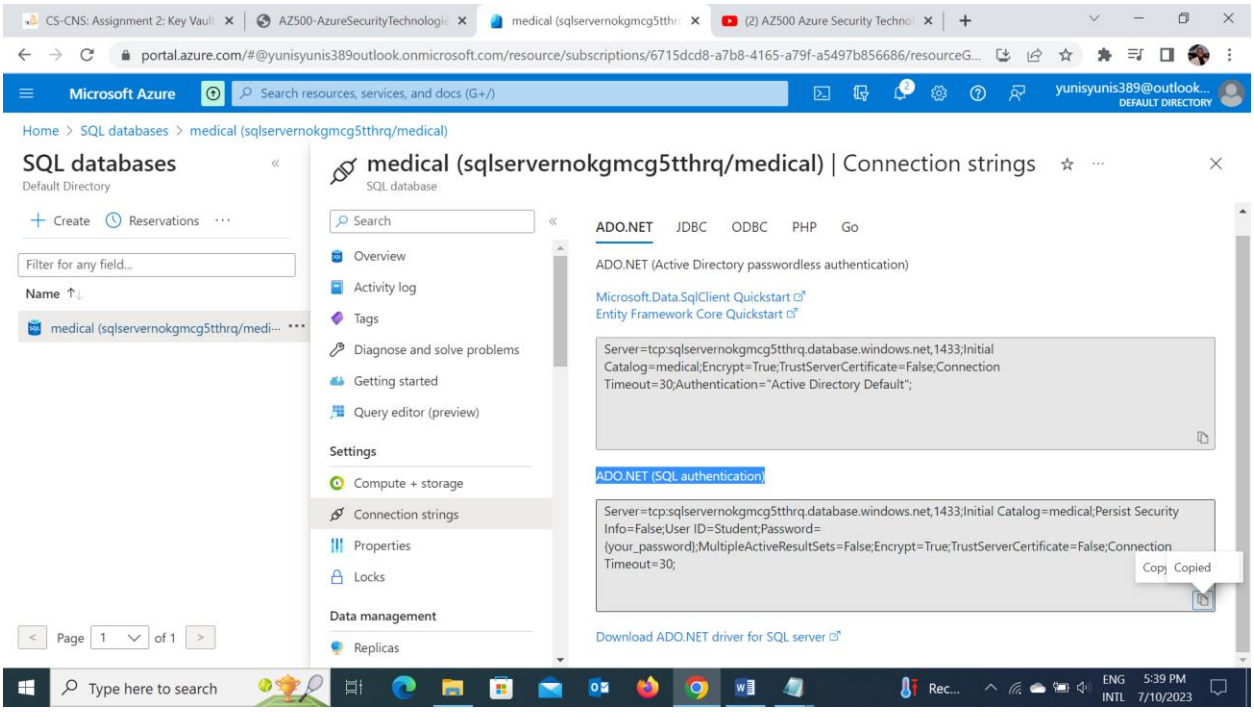
6. Close the Cloud Shell pane.



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

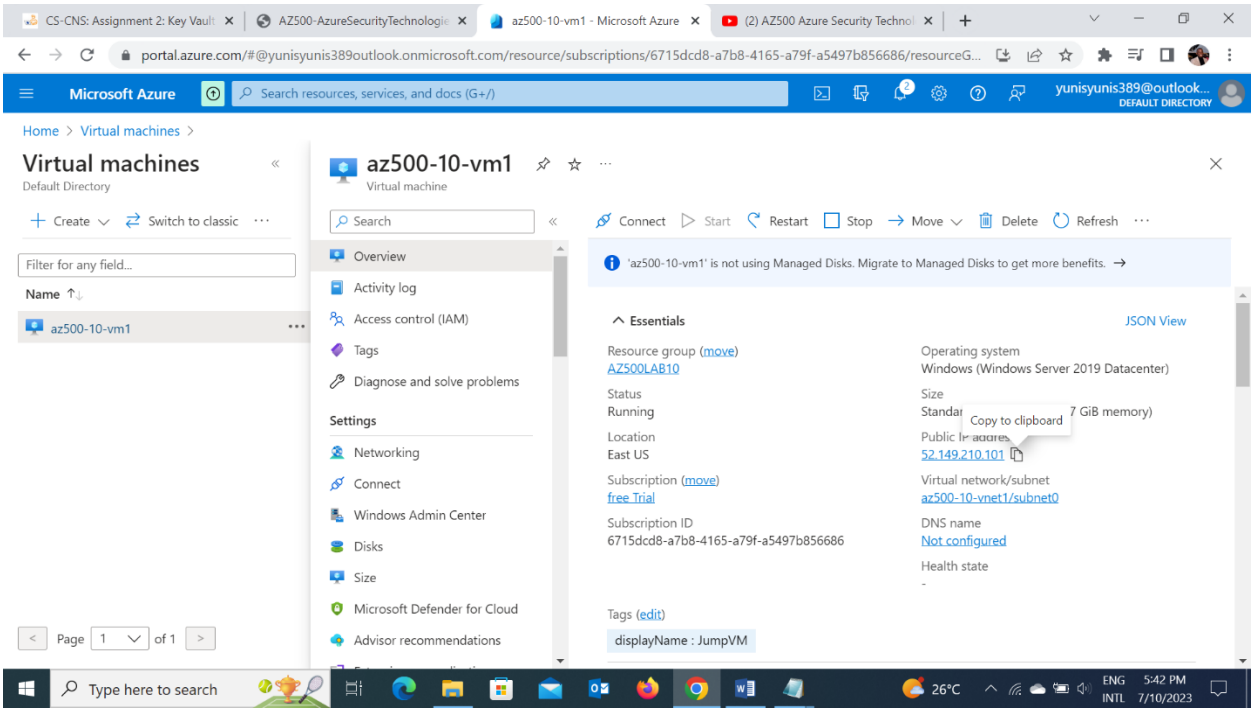
1. In the Azure portal, in the **Search resources, services, and docs** text box at the top of the Azure portal page, type **SQL databases** and press the **Enter** key.
2. In the list of SQL databases, click the **medical () ** entry.
3. On the SQL database blade, in the **Settings** section, click **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.

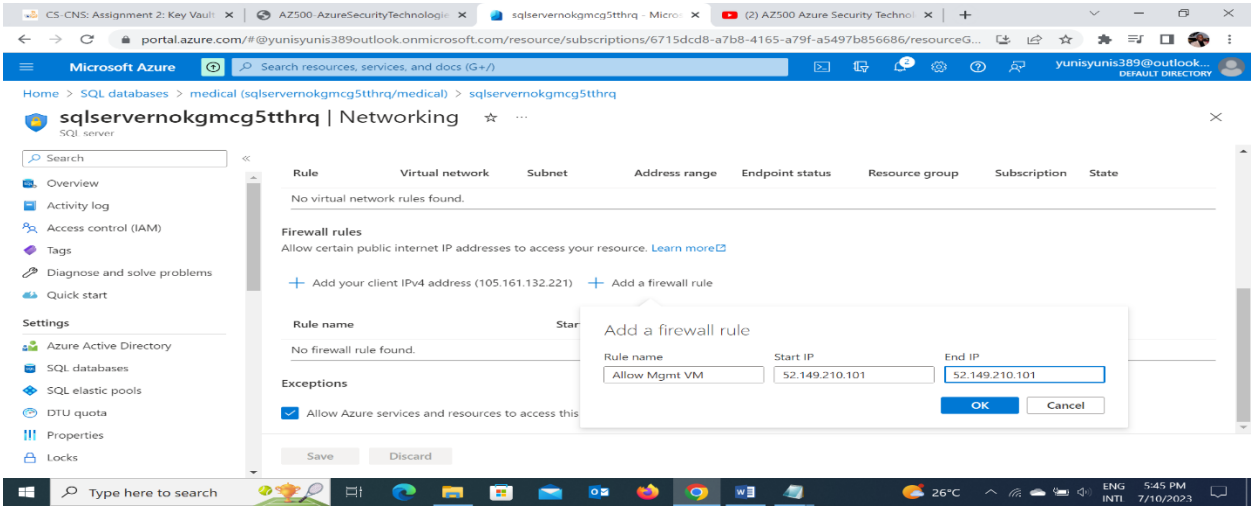
1. In the Azure portal, in the **Search resources, services, and docs** text box at the top of the Azure portal page, type **virtual machines** and press the **Enter** key.
2. In the list of Virtual Machines shown, select the **az500-10-vm1** entry. On the **az500-10-vm1** blade, on the **Essentials**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



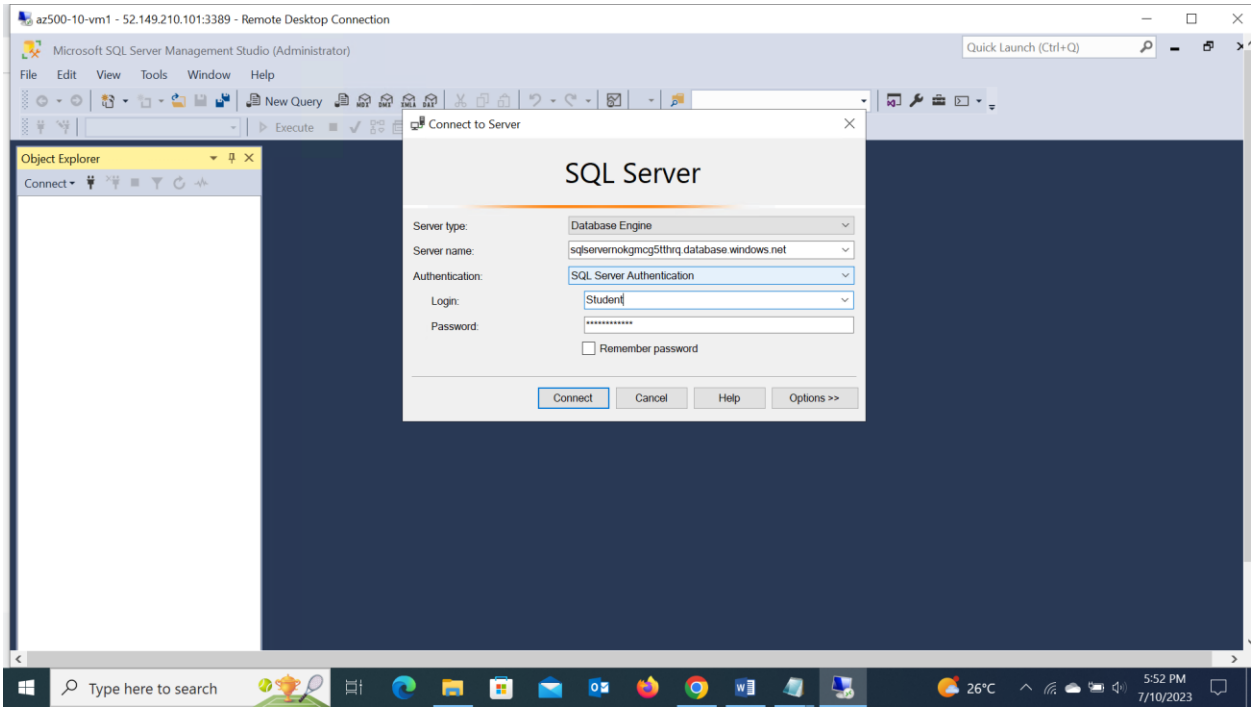
Task 5: Create a table in the SQL Database and select data columns for encryption

n 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.

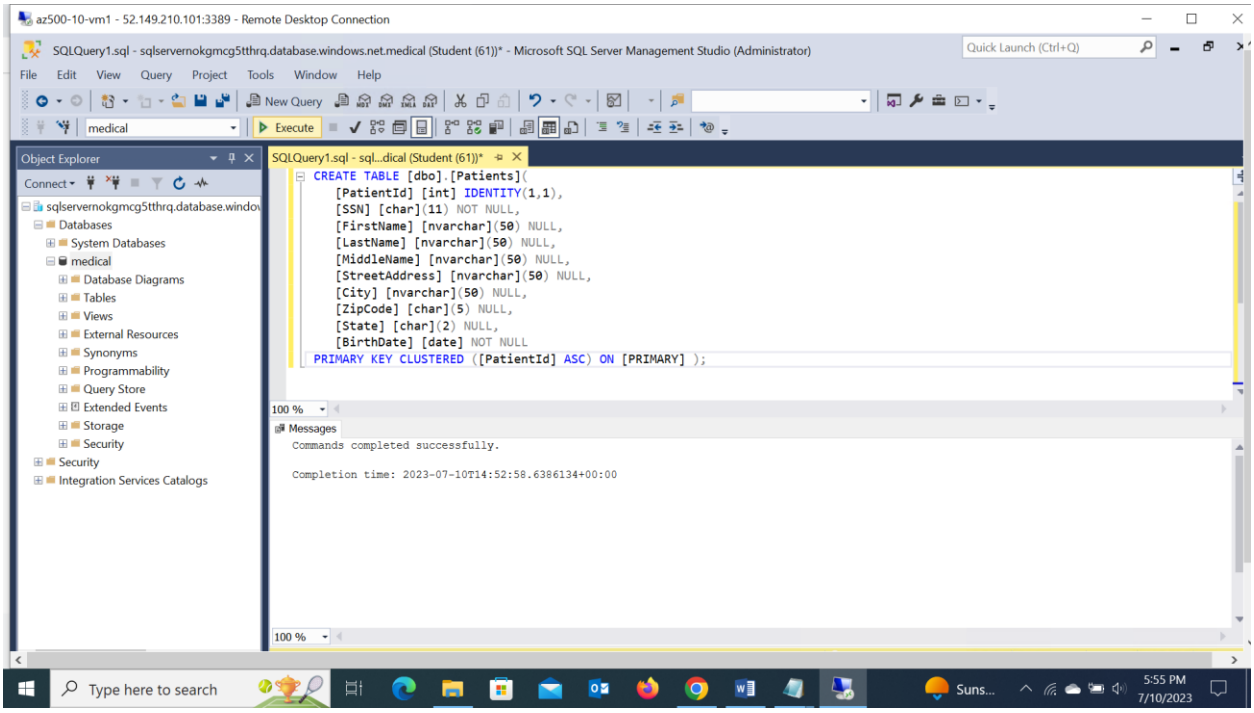
1. In the Azure portal, navigate to the blade of the **medical** SQL database, in the **Essentials** section, identify the **Server name** (copy to clipboard), and then, in the toolbar, click **Set server firewall**.
2. On the **Firewall settings** blade, scroll down to Rule Name, click **+ Add a firewall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



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 credntials:
6. Click Start, in the Start menu, expand the Microsoft SQL Server Tools 19 folder, and click the Micosoft 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.



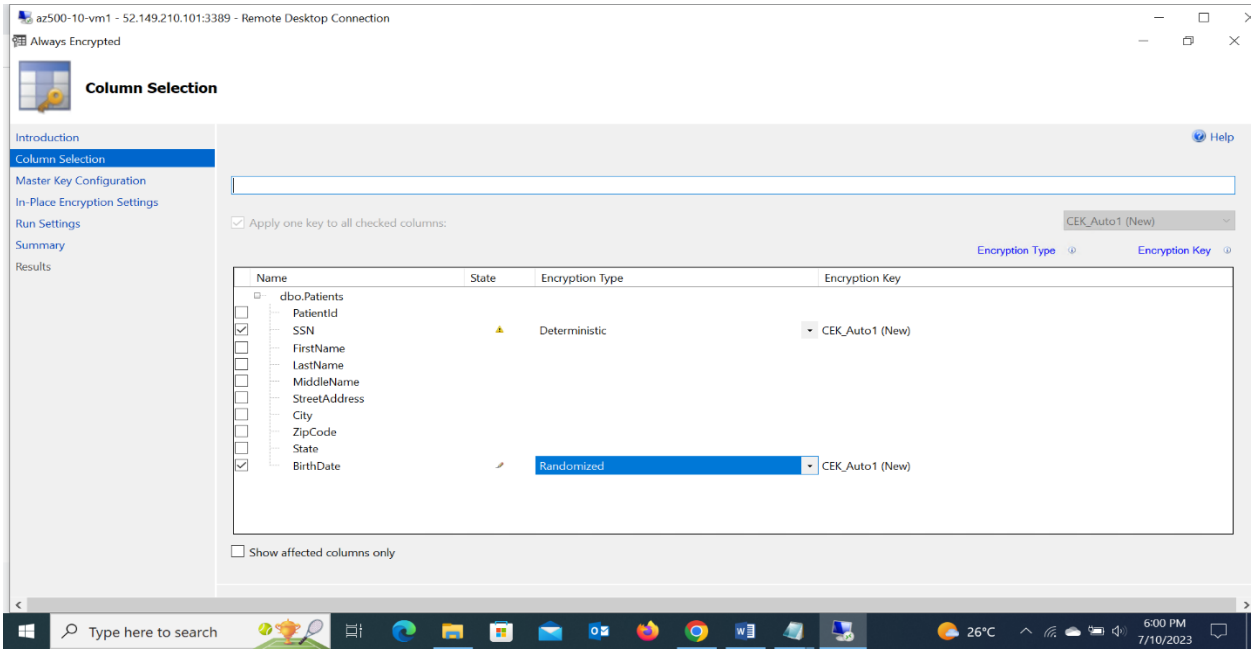
9. Within the SQL Server Management Studio console, in the Object Explorer pane, expand the Databases folder.
10. In the Object Explorer pane, right-click the medical database and click New Query.
11. Paste the following code into the query window and click Execute. This will create a Patients table.



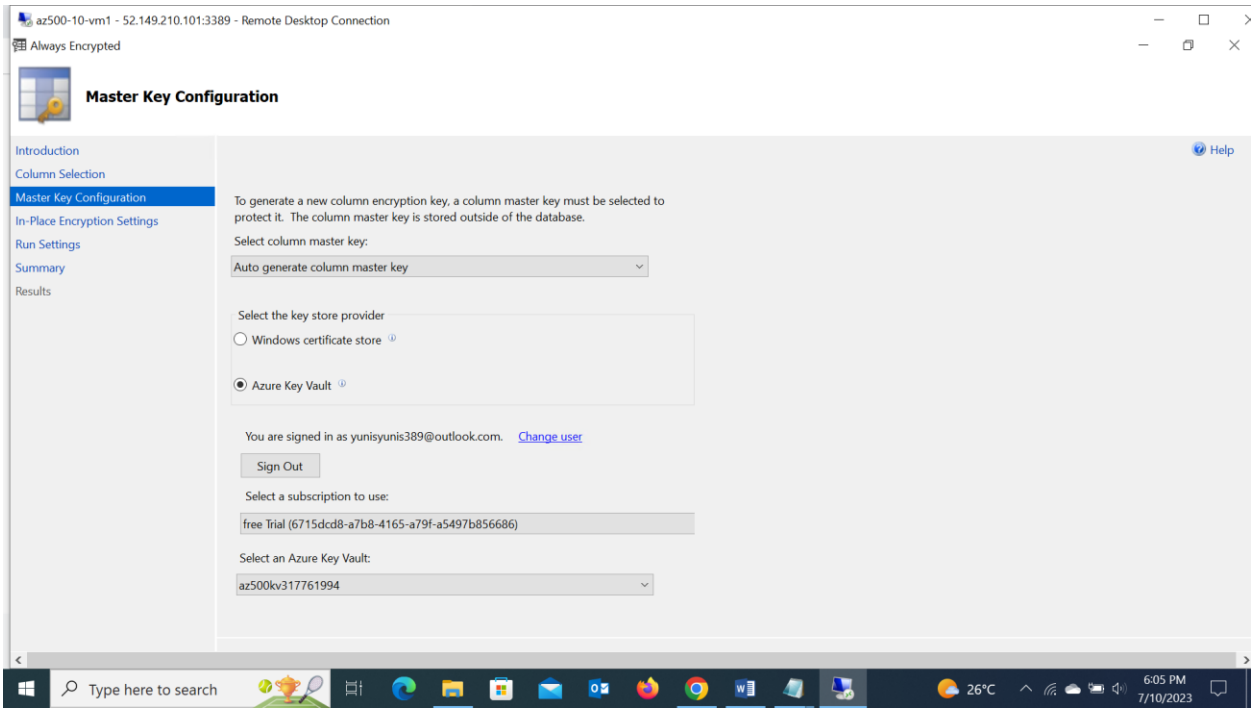
12. After the table is created successfully, in the **Object Explorer** pane, expand the **medical** database node, the **tables** node, right-click the **dbo.Patients** node, and click **Encrypt Columns**.

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.



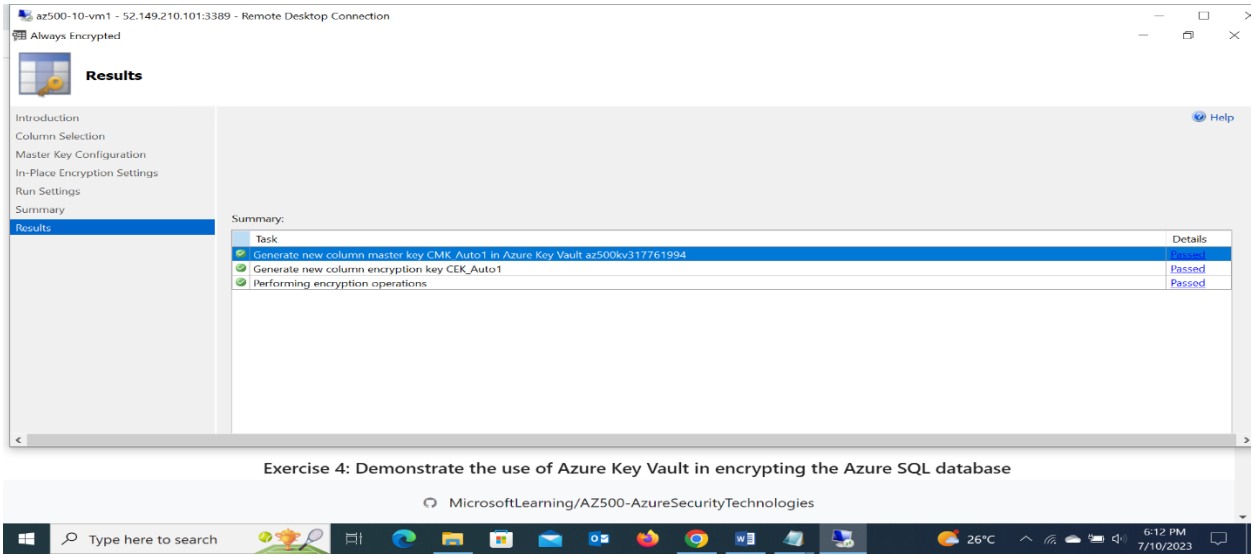
15. On the Master Key Configuration page, select Azure Key Vault, click Sign in, when prompted, authenticate by using the same user account you used to provision the Azure Key Vault instance earlier in this lab, ensure that that Key Vault appears in the Select an Azure Key Vault drop down list, and click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



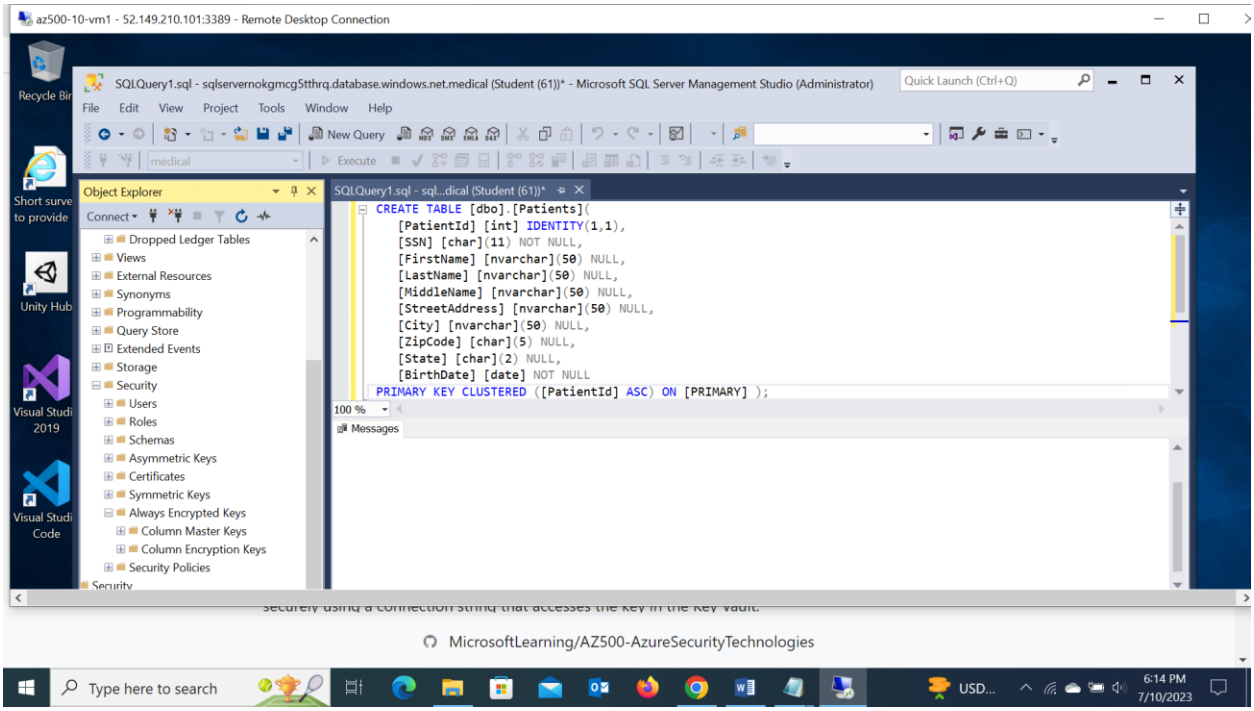
16. On the Run Settings page, click Next.

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:

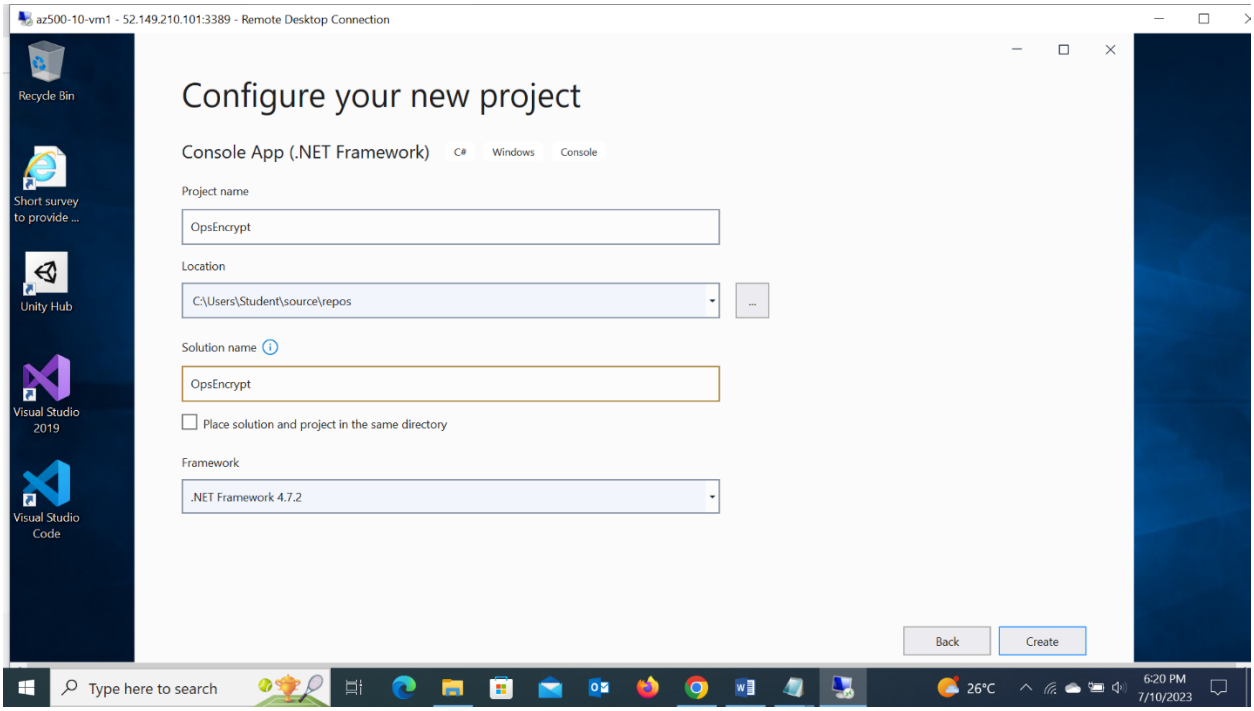
- Task 1: Run a data-driven application to demonstrate the use of Azure Key Vault in encrypting the Azure SQL database.

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 credntials 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**.

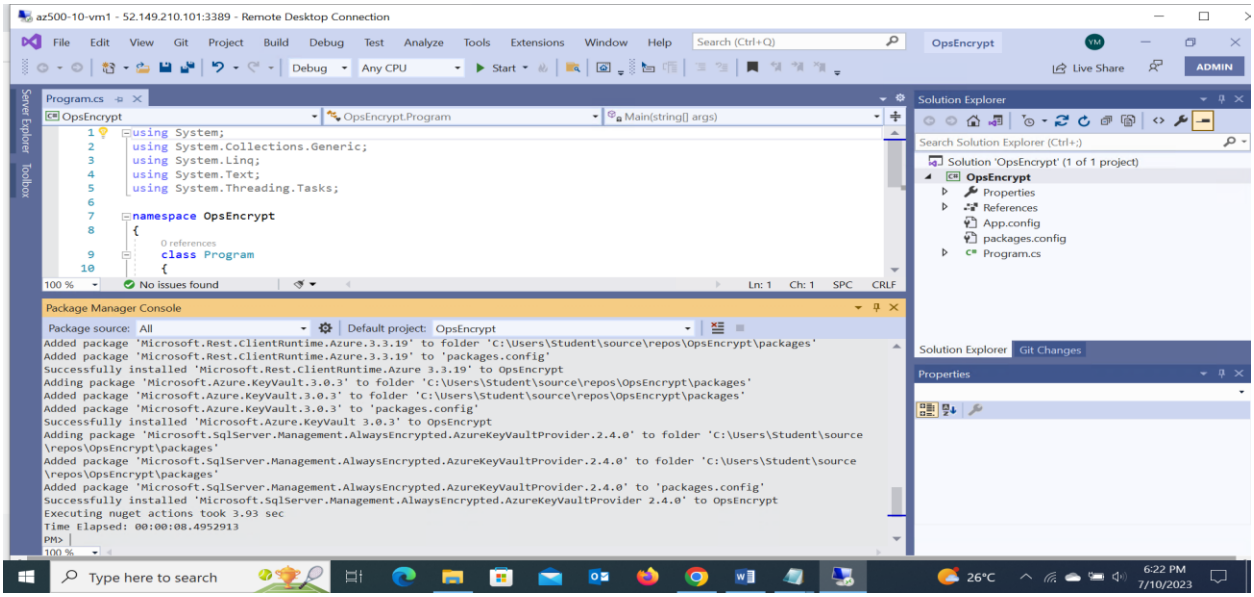
5. On the **Configure your new project** page, specify the following settings (leave other settings with their default values), then click **Create**:



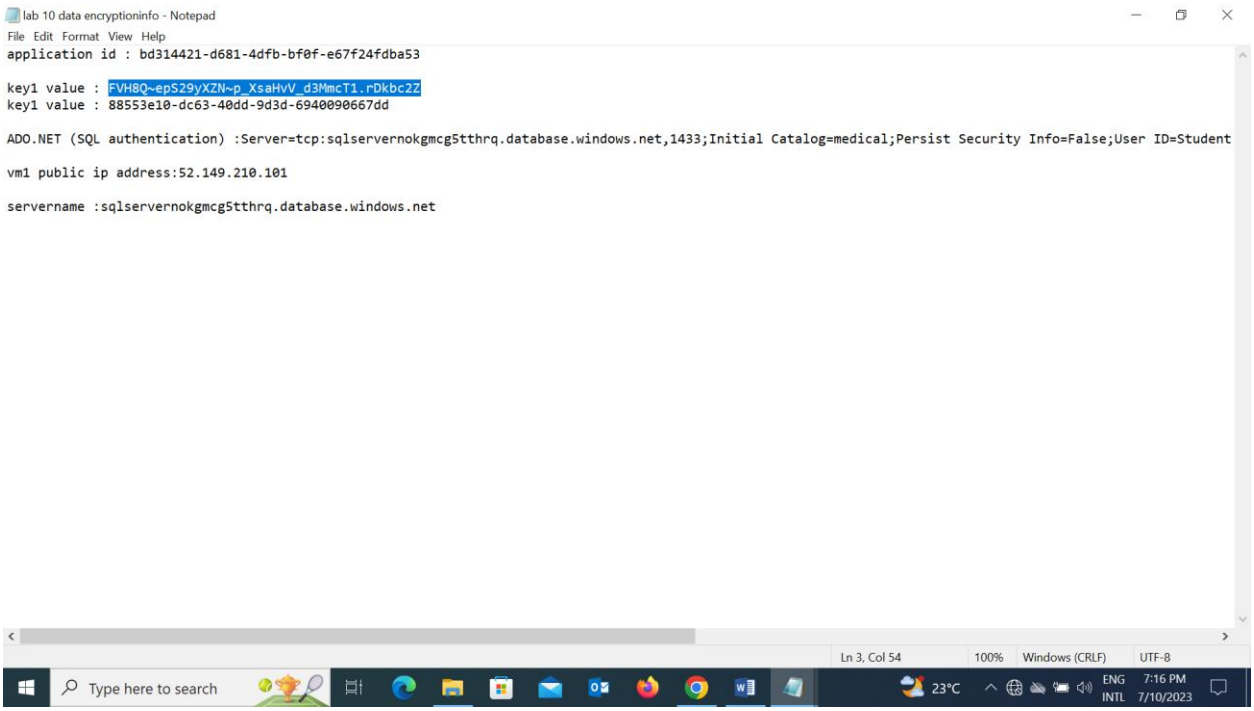
6. In the Visual Studio console, click the Tools menu, in the drop down menu, click NuGet Package Manager, and, in the cascading menu, click Package Manager Console.

7. In the Package Manager Console pane, run the following to install the first required NuGet package:

8. 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} placehodler, 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.

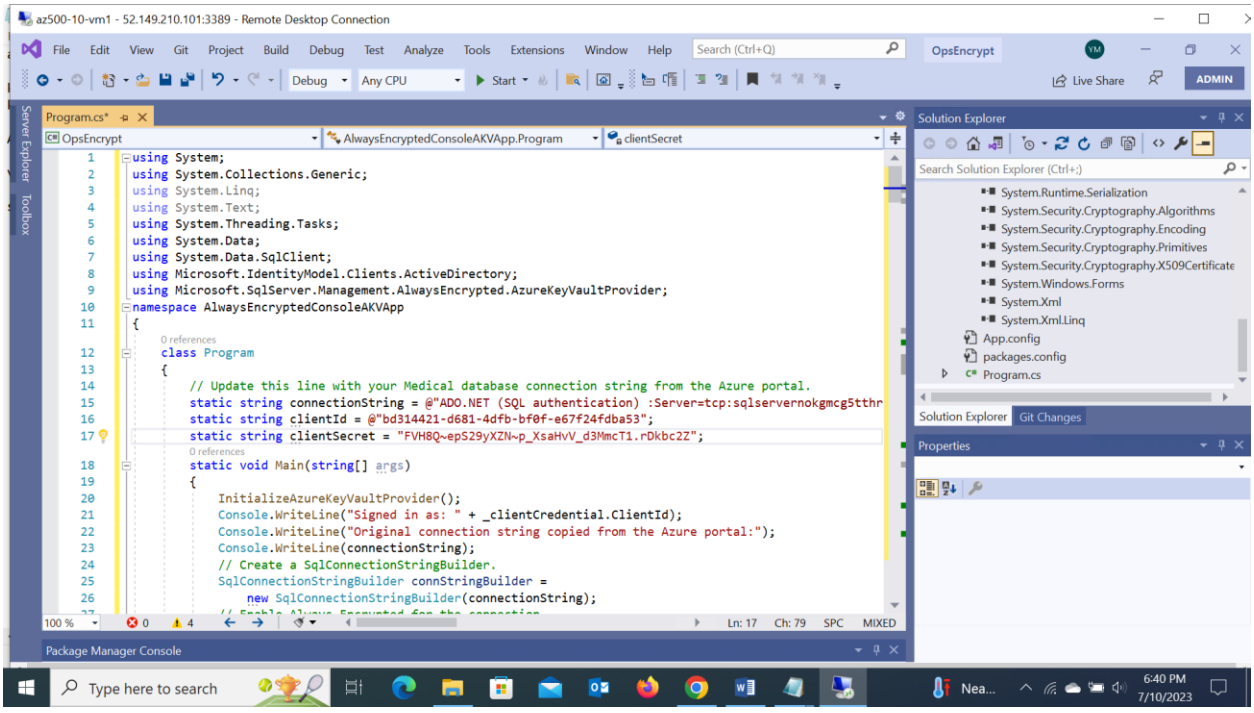


15. Return to the RDP session, and in the Visual Studio console, in the Solution Explorer window, click Program.cs and replace its content with the code you copied into Clipboard.

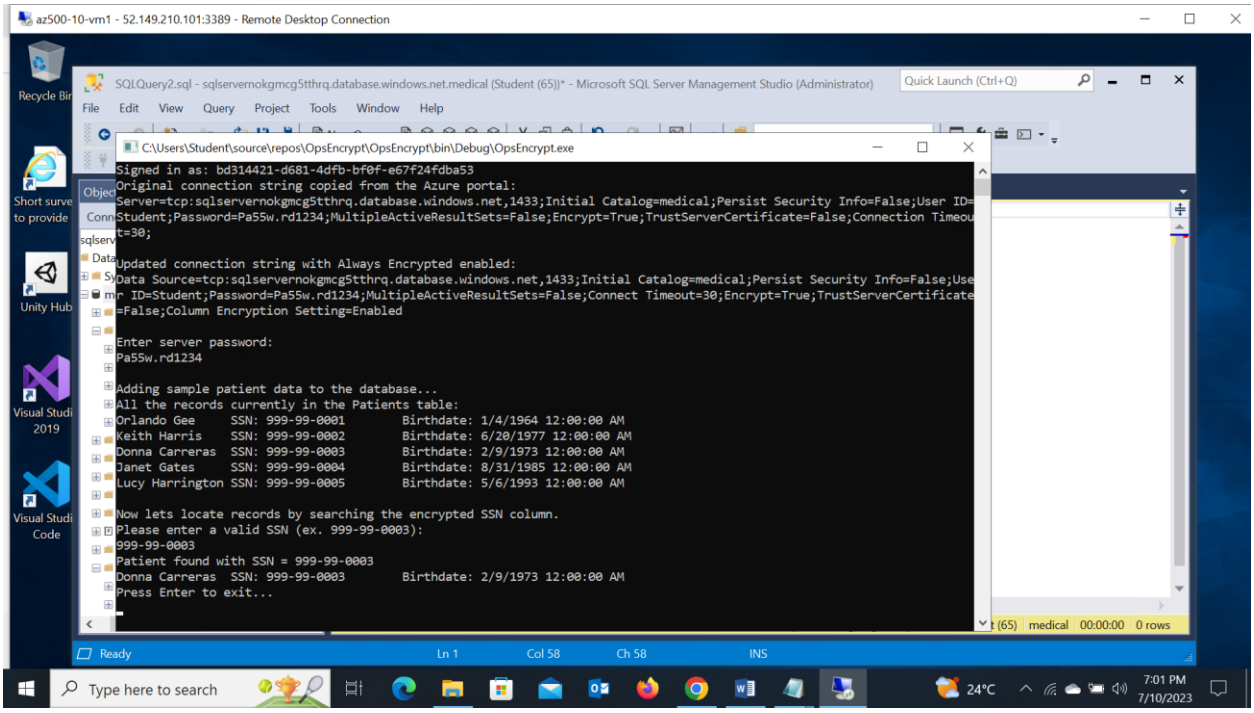
16. In the Visual Studio window, in the Program.cs pane, in line 15, replace the <connection string noted earlier> placeholder with the Azure SQL database ADO.NET connection string you recorded earlier in the lab. In the connection string, replace the {your_password} placehodler, with the password that 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.



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:



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.