Microsoft ADC Cybersecurity Skilling Program

Week 9 Lab Assignment

Student Name: Vincent Onchieku Collins

Student ID: ADC-CSS02-25052

Introduction

This lab series focuses on implementing robust security measures for data stored in Azure SQL Databases using Azure Key Vault. The exercises guide users through practical steps to configure, manage, and validate encryption of sensitive information by integrating Azure services. Starting with the provisioning of necessary Azure resources and enabling transparent data encryption, the lab progresses to configuring Always Encrypted with customer-managed keys stored in Azure Key Vault. Finally, it demonstrates how a data-driven application can interact with the encrypted database, securely retrieving and handling protected data using secure connection strings and application credentials. The hands-on experience gained through these tasks equips users with the skills required to secure SQL data in cloud environments using modern encryption and identity management techniques. The tasks you need to complete will include:

- Exercise 1: Deploy the base infrastructure of an Azure VM and an Azure SQL database from an ARM template
- Exercise 2: Configure the Key Vault resource with a key and a secret
- Exercise 3: Configure an Azure SQL database and a data-driven application
- Exercise 4: Demonstrate the use of Azure Key Vault in encrypting the Azure SQL database

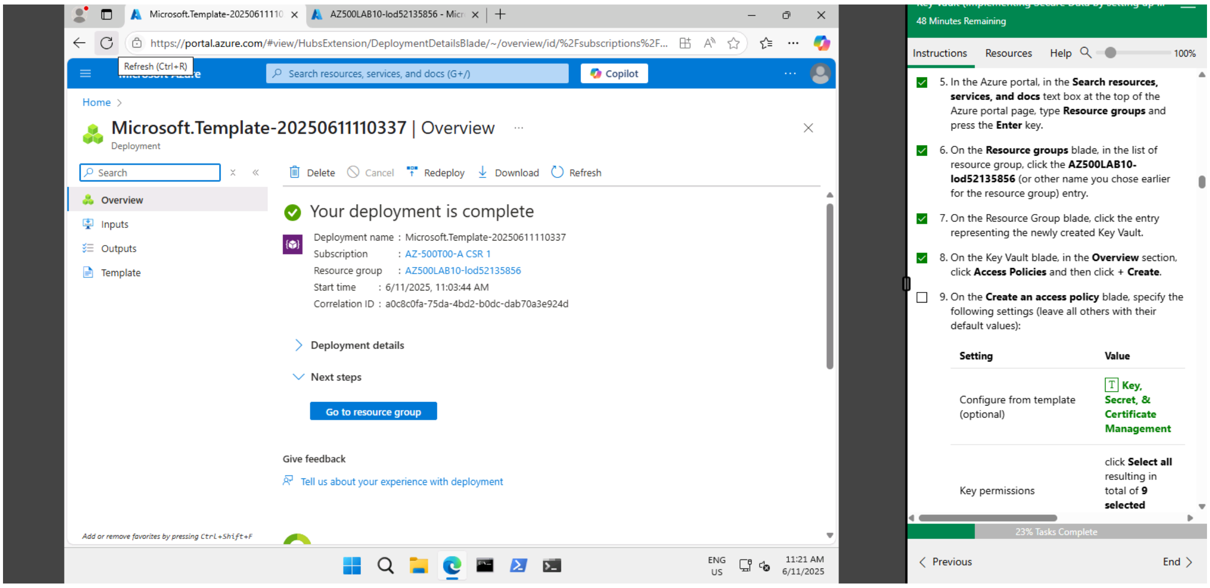
Tasks:

Exercise 1: Deploy the base infrastructure of an Azure VM and an Azure SQL database from an ARM template

In this first exercise, the main task is to automate the deployment of a virtual machine (VM) and Azure SQL Database using an Azure Resource Manager (ARM) template. The lab begins with signing in to the Azure portal, navigating to the custom deployment blade, and uploading the provided JSON template file (az-500-10_azuredeploy.json). This ARM template automatically provisions a virtual machine pre-installed with Visual Studio 2019 and SQL Server Management Studio 19 essential development tools needed for managing and interacting with Azure SQL Database. The VM and other resources are deployed in the East US region, as per lab instructions.

The custom deployment scope is then defined, which includes selecting the appropriate subscription, existing resource group (AZ500LAB10-lod52071852), and user credentials. After reviewing and confirming the settings, I clicked “Create” to initiate deployment. This automated method saves time and ensures consistency by predefining the resources and configurations required for the lab. The deployment process, however, takes around 20–25 minutes.

While the ARM template deploys the infrastructure in the background, this setup plays a critical foundational role. It ensures that all tools are ready for subsequent exercises involving Azure Key Vault and SQL database encryption. By using automation instead of manual setup, this exercise reinforces best practices in cloud infrastructure management, emphasizing repeatability, scalability, and compliance.



Microsoft.Template-2025061111(
AZ500LAB10-lod52135856 - Micr
https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2F...
Refresh (Ctrl+R)
Search resources, services, and docs (G+/)
Copilot
Home
Search
Overview
Inputs
Outputs
Template
Delete
Cancel
Redeploy
Download
Refresh
Add or remove favorites by pressing Ctrl+Shift+F
ENG
US
11:21 AM
6/11/2025

Microsoft.Template-20250611110337 | Overview

Deployment

Your deployment is complete

Deployment name : Microsoft.Template-20250611110337
Subscription : AZ-500T00-A CSR 1
Resource group : AZ500LAB10-lod52135856
Start time : 6/11/2025, 11:03:44 AM
Correlation ID : a0c8c0fa-75da-4bd2-b0dc-dab70a3e924d

Deployment details

Next steps

Go to resource group

Give feedback

Tell us about your experience with deployment

48 Minutes Remaining

Instructions Resources Help 100%

☑ 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-lod52135856** (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):

Setting	Value
Configure from template (optional)	T **Key, Secret, & Certificate Management**
Key permissions	click **Select all** resulting in total of **9 selected**

23% Tasks Complete

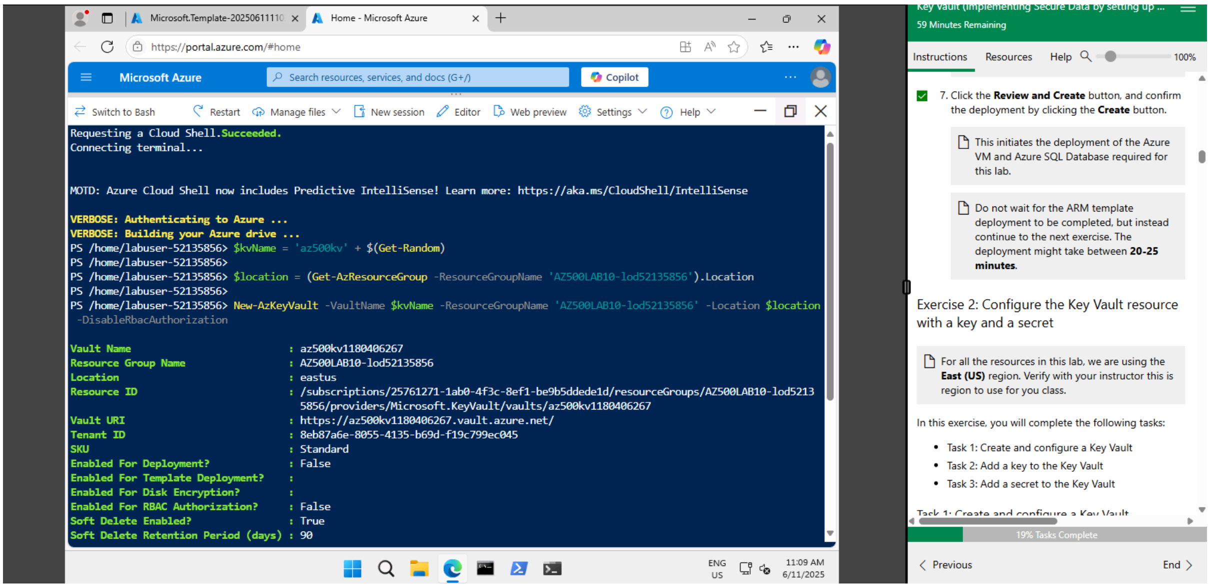
Previous End

Exercise 2: Configure the Key Vault Resource with a Key and a Secret

The second exercise focuses on creating and configuring an Azure Key Vault, which acts as a secure storage location for encryption keys and sensitive information like passwords. Is begin by launching Azure Cloud Shell and using PowerShell commands to create a Key Vault in their existing resource group. The vault name must be unique, and its location matches the rest of the resources (East US). After creating the vault, return to the Azure portal to configure access policies. These permissions grant their user account the ability to manage keys, secrets, and certificates—critical for the security configuration in later steps.

Next, I used PowerShell again to add a key to the Key Vault, specifically named MyLabKey. This key is software-protected and will be used to enable column-level encryption in the Azure SQL database. I verified the key's existence and retrieve its key identifier (URI), which is essential when binding the key to encrypted columns. Understanding the key structure and versioning prepares to manage key lifecycle operations in real-world scenarios.

Lastly, I add a secret to the Key Vault, representing a sample SQL password (SQLPassword). This is achieved by creating a secure string variable and setting it in the vault using PowerShell. The secret is also verified and its version examined in the Azure portal. This step demonstrates how secrets can be securely stored and retrieved by authorized applications, paving the way for secure authentication without hardcoding credentials. Overall, this exercise teaches foundational concepts in secure key and secret management using Azure Key Vault.



```
Requesting a Cloud Shell.Succeeded.
Connecting terminal...


MOTD: Azure Cloud Shell now includes Predictive IntelliSense! Learn more: https://aka.ms/CloudShell/IntelliSense

VERBOSE: Authenticating to Azure ...
VERBOSE: Building your Azure drive ...
PS /home/labuser-52135856> $kvName = 'az500kv' + $(Get-Random)
PS /home/labuser-52135856>
PS /home/labuser-52135856> $location = (Get-AzResourceGroup -ResourceGroupName 'AZ500LAB10-lod52135856').Location
PS /home/labuser-52135856>
PS /home/labuser-52135856> New-AzKeyVault -VaultName $kvName -ResourceGroupName 'AZ500LAB10-lod52135856' -Location $location
 -DisableRbacAuthorization

Vault Name                       : az500kv1180406267
Resource Group Name              : AZ500LAB10-lod52135856
Location                         : eastus
Resource ID                      : /subscriptions/25761271-1ab0-4f3c-8ef1-be9b5ddede1d/resourceGroups/AZ500LAB10-lod5213
                                   5856/providers/Microsoft.KeyVault/vaults/az500kv1180406267
Vault URI                        : https://az500kv1180406267.vault.azure.net/
Tenant ID                        : 8eb87a6e-8055-4135-b69d-f19c799ec045
SKU                              : Standard
Enabled For Deployment?          : False
Enabled For Template Deployment? :
Enabled For Disk Encryption?     :
Enabled For RBAC Authorization?  : False
Soft Delete Enabled?             : True
Soft Delete Retention Period (days) : 90
```

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

This initiates the deployment of the Azure VM and Azure SQL Database required for this lab.

Do not wait for the ARM template deployment to be completed, but instead continue to the next exercise. The deployment might take between **20-25 minutes**.

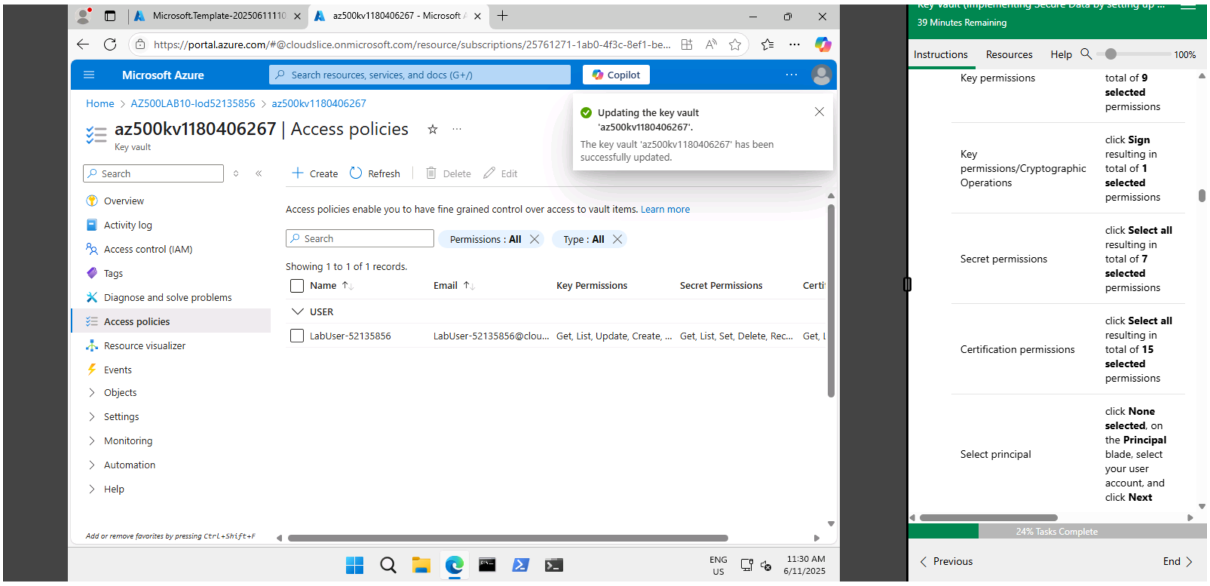
Exercise 2: Configure the Key Vault resource with a key and a secret

For all the resources in this lab, we are using the **East (US)** region. Verify with your instructor this is region to use for you class.

In this exercise, you 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


Microsoft.Template-20250611110
az500kv1180406267 - Microsoft
https://portal.azure.com/#@cloudslice.onmicrosoft.com/resource/subscriptions/25761271-1ab0-4f3c-8ef1-be...
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > AZ500LAB10-lod52135856 > az500kv1180406267
Key vault
Search
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
Access policies
Resource visualizer
Events
Objects
Settings
Monitoring
Automation
Help
Add or remove favorites by pressing Ctrl+Shift+F
Create
Refresh
Delete
Edit
Permissions : All
Type : All
ENG US
11:30 AM 6/11/2025

az500kv1180406267 | Access policies

Updating the key vault 'az500kv1180406267'.

The key vault 'az500kv1180406267' has been successfully updated.

Access policies enable you to have fine grained control over access to vault items. Learn more

Showing 1 to 1 of 1 records.

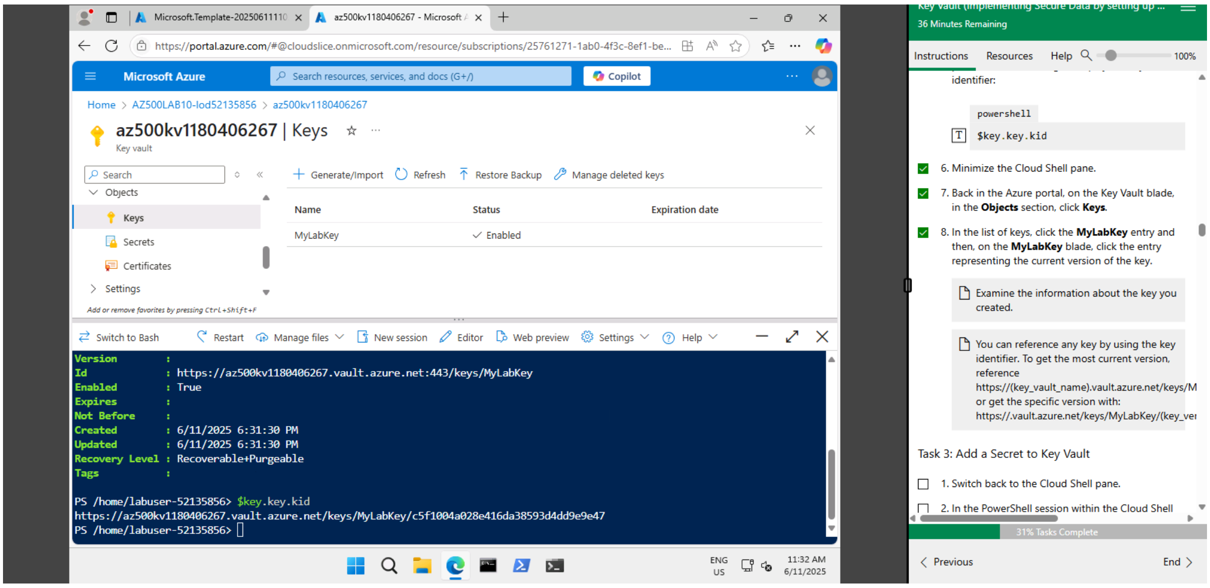
Name	Email	Key Permissions	Secret Permissions	Certi
USER				
LabUser-52135856	LabUser-52135856@clou...	Get, List, Update, Create, ...	Get, List, Set, Delete, Rec...	Get, L

Key Vault (Implementing Secure Data by setting up ...

39 Minutes Remaining

Key permissions	total of **9** **selected** permissions
Key permissions/Cryptographic Operations	click **Sign** resulting in total of **1** **selected** permissions
Secret permissions	click **Select all** resulting in total of **7** **selected** permissions
Certification permissions	click **Select all** resulting in total of **15** **selected** permissions
Select principal	click **None selected**, on the **Principal** blade, select your user account, and click **Next**

24% Tasks Complete


Microsoft.Template-20250611110
az500kv1180406267 - Microsoft
https://portal.azure.com/#@cloudslice.onmicrosoft.com/resource/subscriptions/25761271-1ab0-4f3c-8ef1-be...
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > AZ500LAB10-lod52135856 > az500kv1180406267
Key vault
Search
Objects
Keys
Secrets
Certificates
Settings
Add or remove favorites by pressing Ctrl+Shift+F
Generate/Import
Refresh
Restore Backup
Manage deleted keys
Switch to Bash
Restart
Manage files
New session
Editor
Web preview
Settings
Help
ENG US
11:32 AM 6/11/2025

az500kv1180406267 | Keys

Name	Status	Expiration date
MyLabKey	✓ Enabled	

```
Version        :
Id             : https://az500kv1180406267.vault.azure.net:443/keys/MyLabKey
Enabled        : True
Expires        :
Not Before     :
Created        : 6/11/2025 6:31:30 PM
Updated        : 6/11/2025 6:31:30 PM
Recovery Level : Recoverable+Purgeable
Tags           :

PS /home/labuser-52135856> $key.key.kid
https://az500kv1180406267.vault.azure.net/keys/MyLabKey/c5f1004a028e416da38593d4dd9e9e47
PS /home/labuser-52135856> 
```

Key Vault (Implementing Secure Data by setting up ...

36 Minutes Remaining

identifier:

```powershell
$key.key.kid
```

☑ 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.

> Examine the information about the key you created.

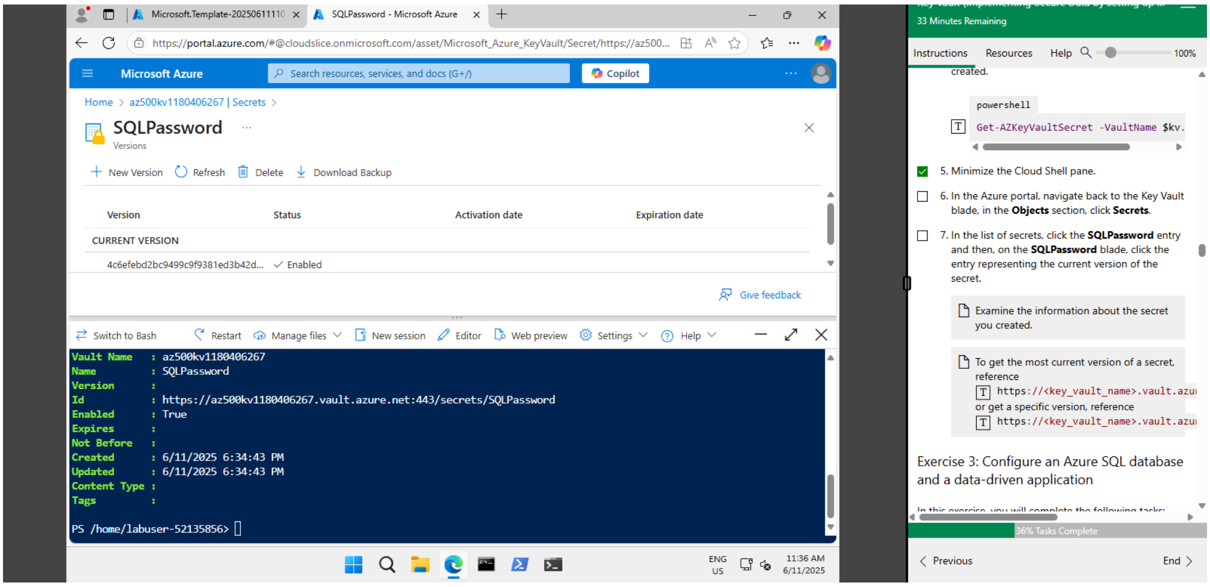
> You can reference any key by using the key identifier. To get the most current version, reference https://(key_vault_name).vault.azure.net/keys/M or get the specific version with: https://.vault.azure.net/keys/MyLabKey/(key_ver

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

31% Tasks Complete



Version	Status	Activation date	Expiration date
CURRENT VERSION			
4c6efebd2bc9499c9f9381ed3b42d...	✓ Enabled		

```
Vault Name   : az500kv1180406267
Name         : SQLPassword
Version      :
Id           : https://az500kv1180406267.vault.azure.net:443/secrets/SQLPassword
Enabled      : True
Expires      :
Not Before   :
Created      : 6/11/2025 6:34:43 PM
Updated      : 6/11/2025 6:34:43 PM
Content Type :
Tags         :

PS /home/labuser-52135856> 
```

created.

```powershell
Get-AZKeyVaultSecret -VaultName $kv.
```

☑ 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.

Examine the information about the secret you created.

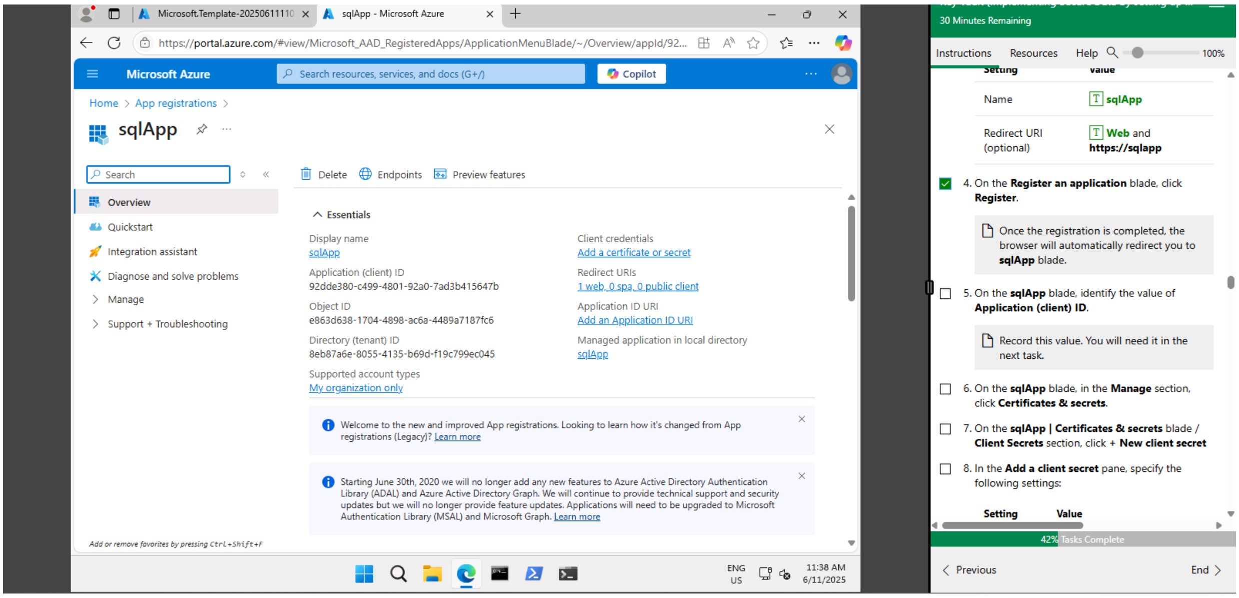
To get the most current version of a secret, reference `https://<key_vault_name>.vault.azu` or get a specific version, reference `https://<key_vault_name>.vault.azu`

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

Exercise 3: Configure an Azure SQL Database and a Data-Driven Application

In this exercise, the focus shifts to linking the Azure SQL Database with an application and enabling encryption using Azure Key Vault. First, I registered a new application (sqlApp) in Microsoft Entra ID (formerly Azure Active Directory), which will represent the data-driven app requiring database access. They record the Application (client) ID and generate a client secret for authentication. This registration ensures secure access to Azure services via identity-based control, instead of hardcoded credentials.

After app registration, I create a policy to allow the registered app to access the Key Vault. This involves assigning the necessary permissions (to use the stored keys and secrets) using access policies in the Key Vault's settings. These permissions enable the application to interact with the vault securely at runtime critical for managing encrypted database columns. Additionally, I retrieved the ADO.NET connection string for the Azure SQL Database, which will be used later when configuring the .NET application.


Microsoft.Template-20250611110
sqlApp - Microsoft Azure
https://portal.azure.com/#view/Microsoft_AAD_RegisteredApps/ApplicationMenuBlade/~/Overview/appId/92...
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > App registrations >
Search
Overview
Quickstart
Integration assistant
Diagnose and solve problems
Manage
Support + Troubleshooting
Delete
Endpoints
Preview features
Instructions
Resources
Help
100%
Previous
End
ENG US
11:38 AM 6/11/2025

sqlApp

Essentials

Display name: sqlApp

Application (client) ID: 92dde380-c499-4801-92a0-7ad3b415647b

Object ID: e863d638-1704-4898-ac6a-4489a7187fc6

Directory (tenant) ID: 8eb87a6e-8055-4135-b69d-f19c799ec045

Supported account types: My organization only

Client credentials: Add a certificate or secret

Redirect URIs: 1 web, 0 spa, 0 public client

Application ID URI: Add an Application ID URI

Managed application in local directory: sqlApp

Welcome to the new and improved App registrations. Looking to learn how it's changed from App registrations (Legacy)? Learn more

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory Authentication Library (ADAL) and Azure Active Directory Graph. We will continue to provide technical support and security updates but we will no longer provide feature updates. Applications will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph. Learn more

Add or remove favorites by pressing Ctrl+Shift+F

30 Minutes Remaining

Setting	Value
Name	sqlApp
Redirect URI (optional)	**Web** and **https://sqlapp**

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

Once the registration is completed, the browser will automatically redirect you to **sqlApp** blade.

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

Record this value. You will need it in the next task.

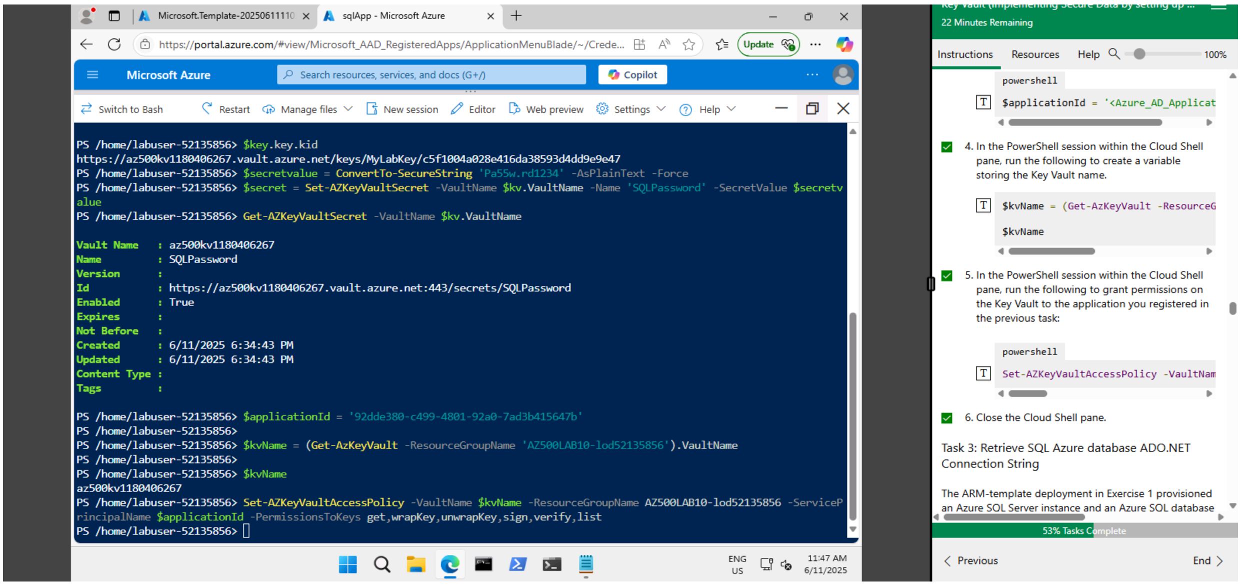
□ 6. On the **sqlApp** blade, in the **Manage** section, click **Certificates & secrets**.

□ 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:

Setting	Value

42% Tasks Complete



```
PS /home/labuser-52135856> $key.key.kid
https://az500kv1180406267.vault.azure.net/keys/MyLabKey/c5f1004a028e416da38593d4dd9e9e47
PS /home/labuser-52135856> $secretvalue = ConvertTo-SecureString 'Pa55w.rd1234' -AsPlainText -Force
PS /home/labuser-52135856> $secret = Set-AZKeyVaultSecret -VaultName $kv.VaultName -Name 'SQLPassword' -SecretValue $secretv
alue
PS /home/labuser-52135856> Get-AZKeyVaultSecret -VaultName $kv.VaultName

Vault Name   : az500kv1180406267
Name         : SQLPassword
Version      :
Id           : https://az500kv1180406267.vault.azure.net:443/secrets/SQLPassword
Enabled      : True
Expires      :
Not Before   :
Created      : 6/11/2025 6:34:43 PM
Updated      : 6/11/2025 6:34:43 PM
Content Type :
Tags         :

PS /home/labuser-52135856> $applicationId = '92dde380-c499-4801-92a0-7ad3b415647b'
PS /home/labuser-52135856>
PS /home/labuser-52135856> $kvName = (Get-AzKeyVault -ResourceGroupName 'AZ500LAB10-lod52135856').VaultName
PS /home/labuser-52135856>
PS /home/labuser-52135856> $kvName
az500kv1180406267
PS /home/labuser-52135856> Set-AZKeyVaultAccessPolicy -VaultName $kvName -ResourceGroupName AZ500LAB10-lod52135856 -ServiceP
rincipalName $applicationId -PermissionsToKeys get,wrapKey,unwrapKey,sign,verify,list
PS /home/labuser-52135856>
```

22 Minutes Remaining

```
powershell
$applicationId = '<Azure_AD_Applicat
```

☑ 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 -ResourceG
$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:

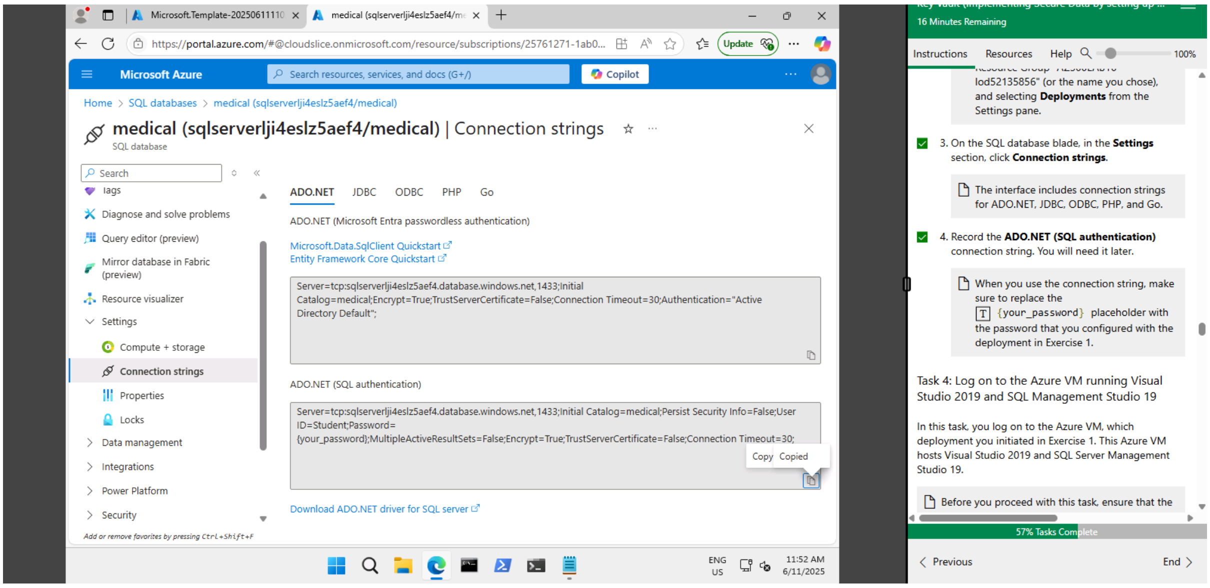
```
powershell
Set-AZKeyVaultAccessPolicy -VaultNam
```

☑ 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

53% Tasks Complete


Microsoft.Template-20250611110
medical (sqlserverlji4eslz5aef4/m
https://portal.azure.com/#@cloudslice.onmicrosoft.com/resource/subscriptions/25761271-1ab0...
Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
Home > SQL databases > medical (sqlserverlji4eslz5aef4/medical)
SQL database
Search
Tags
Diagnose and solve problems
Query editor (preview)
Mirror database in Fabric (preview)
Resource visualizer
Settings
Compute + storage
Connection strings
Properties
Locks
Data management
Integrations
Power Platform
Security
Add or remove favorites by pressing Ctrl+Shift+F
ADO.NET
JDBC
ODBC
PHP
Go
Microsoft.Data.SqlClient Quickstart
Entity Framework Core Quickstart
Copy
Copied
Download ADO.NET driver for SQL server
ENG US
11:52 AM 6/11/2025
Instructions
Resources
Help
100%
57% Tasks Complete
Previous
End

medical (sqlserverlji4eslz5aef4/medical) | Connection strings

ADO.NET (Microsoft Entra passwordless authentication)

```
Server=tcp:sqlserverlji4eslz5aef4.database.windows.net,1433;Initial
Catalog=medical;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;Authentication="Active
Directory Default";
```

ADO.NET (SQL authentication)

```
Server=tcp:sqlserverlji4eslz5aef4.database.windows.net,1433;Initial Catalog=medical;Persist Security Info=False;User
ID=Student;Password=
{your_password};MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;
```

16 Minutes Remaining

lod52135856" (or the name you chose), and selecting **Deployments** from the Settings pane.

☑ 3. On the SQL database blade, in the **Settings** section, click **Connection strings**.

> The interface includes connection strings for ADO.NET, JDBC, ODBC, PHP, and Go.

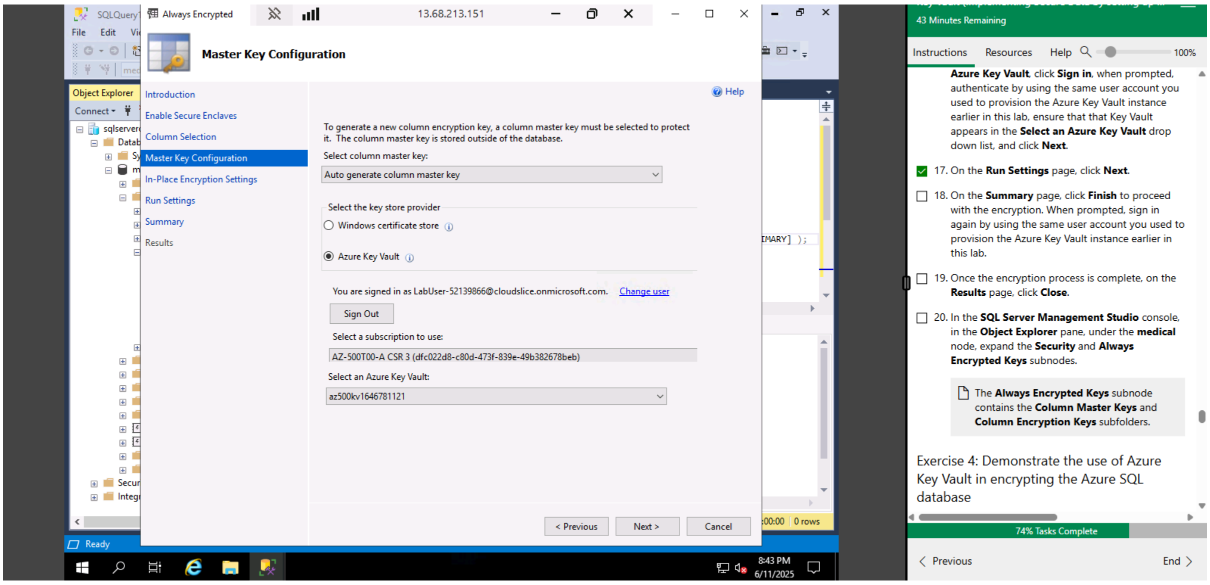
☑ 4. Record the **ADO.NET (SQL authentication)** connection string. You will need it later.

> When you use the connection string, make sure to replace the `{your_password}` placeholder with the password that you configured with the deployment in Exercise 1.

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.

> Before you proceed with this task, ensure that the



Master Key Configuration

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

You are signed in as LabUser-52139866@cloudslice.onmicrosoft.com. Change user

Select a subscription to use: AZ-500T00-A CSR 3 (dfc022d8-c80d-473f-839e-49b382678beb)

Select an Azure Key Vault: az500kv1646781121

43 Minutes Remaining

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**.

☑ 17. On the **Run Settings** page, click **Next**.

☐ 18. 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.

☐ 19. Once the encryption process is complete, on the **Results** page, click **Close**.

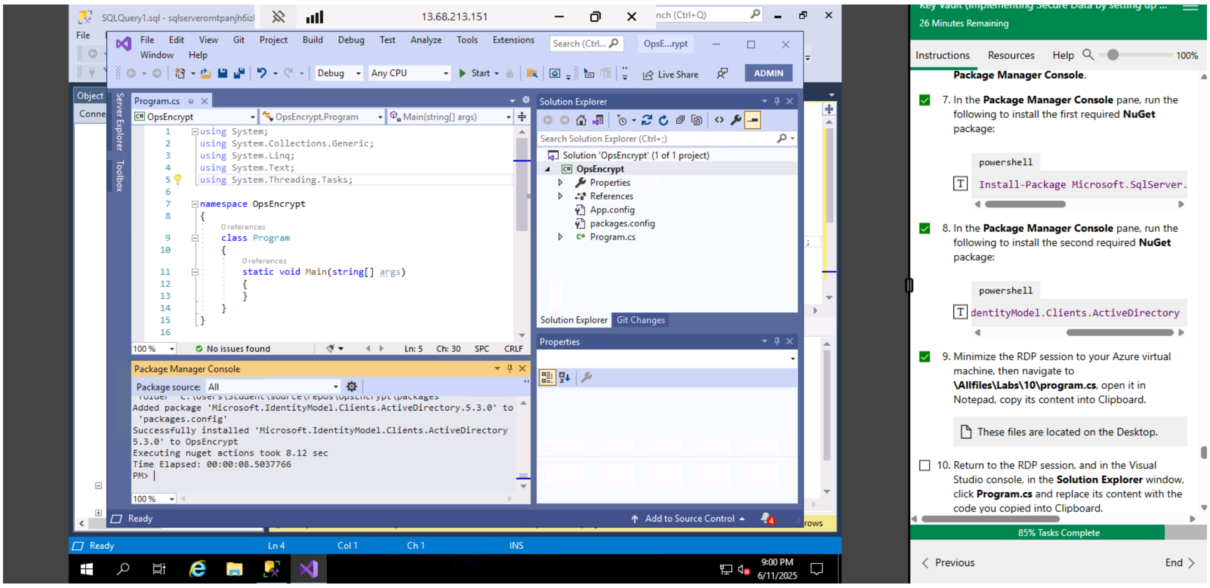
☐ 20. In the **SQL Server Management Studio** console, in the **Object Explorer** pane, under the **medical** node, expand the **Security** and **Always Encrypted Keys** subnodes.

> The **Always Encrypted Keys** subnode contains the **Column Master Keys** and **Column Encryption Keys** subfolders.

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

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

Finally, I remotely access the Azure VM that was provisioned earlier. Inside the VM, using SQL Server Management Studio, they create a table in the Azure SQL Database and select specific columns for Always Encrypted configuration (e.g., columns storing sensitive data like SSNs or credit card numbers). This process integrates the Key Vault key created in Exercise 2, ensuring that data in these columns remains encrypted at all times—at rest, in transit, and even during query operations. This hands-on implementation of Always Encrypted completes the secure data handling workflow in a cloud-based application environmentIn the fourth exercise, the task involved demonstrating how to use Azure Key Vault to encrypt sensitive data stored in an Azure SQL Database using a data-driven console application. This was done by creating a C# .NET Framework console app in Visual Studio 2019 on an Azure virtual machine, installing necessary NuGet packages for Azure Key Vault and Active Directory integration, and using a sample program (program.cs) that was configured with a valid ADO.NET connection string, application client ID, and key value from Azure. Once configured, the application was run to securely load data into encrypted columns in the database. Verification of encryption was done using SQL Management Studio by querying the encrypted table and observing that sensitive data (like SSNs) appeared encrypted. When queried through the application with a valid SSN, the same encrypted data was retrieved in decrypted form, showing secure access via Key Vault. The exercise concluded with resource cleanup using PowerShell in Azure Cloud Shell to avoid unnecessary charges.



```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace OpsEncrypt
{
    class Program
    {
        static void Main(string[] args)
        {
        }
    }
}
```

```
Added package 'Microsoft.IdentityModel.Clients.ActiveDirectory.5.3.0' to
 'packages.config'
Successfully installed 'Microsoft.IdentityModel.Clients.ActiveDirectory
5.3.0' to OpsEncrypt
Executing nuget actions took 8.12 sec
Time Elapsed: 00:00:08.5037766
PM> 
```

26 Minutes Remaining

Instructions Resources Help 100%

Package Manager Console.

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

```powershell
Install-Package Microsoft.SqlServer.
```

☑ 8. In the **Package Manager Console** pane, run the following to install the second required **NuGet** package:

```powershell
dentityModel.Clients.ActiveDirectory
```

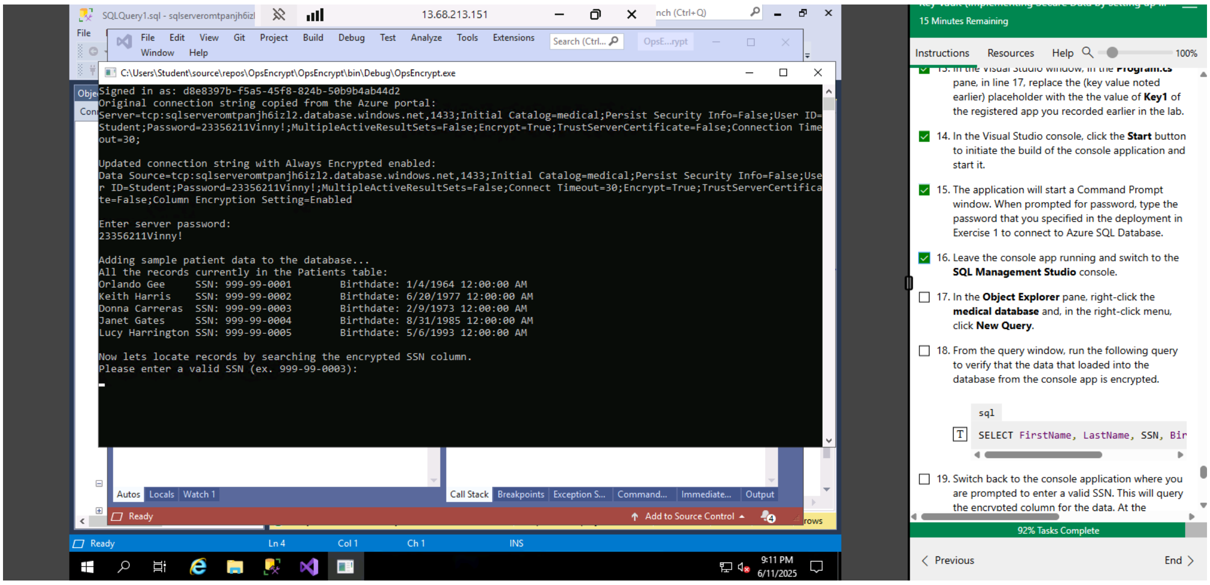
☑ 9. Minimize the RDP session to your Azure virtual machine, then navigate to **\Allfiles\Labs\10\program.cs**, open it in Notepad, copy its content into Clipboard.

These files are located on the Desktop.

☐ 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.

85% Tasks Complete

< Previous End >



SQLQuery1.sql - sqlserveromtpanjh6izl
13.68.213.151
File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
OpsE...rypt
Autos Locals Watch 1
Call Stack Breakpoints Exception S... Command... Immediate... Output
Ready
Add to Source Control
Ln 4 Col 1 Ch 1 INS
9:11 PM
6/11/2025

C:\Users\Student\source\repos\OpsEncrypt\OpsEncrypt\bin\Debug\OpsEncrypt.exe

```
Signed in as: d8e8397b-f5a5-45f8-824b-50b9b4ab44d2
Original connection string copied from the Azure portal:
Server=tcp:sqlserveromtpanjh6izl2.database.windows.net,1433;Initial Catalog=medical;Persist Security Info=False;User ID=
Student;Password=23356211Vinny!;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Time
out=30;

Updated connection string with Always Encrypted enabled:
Data Source=tcp:sqlserveromtpanjh6izl2.database.windows.net,1433;Initial Catalog=medical;Persist Security Info=False;Use
r ID=Student;Password=23356211Vinny!;MultipleActiveResultSets=False;Connect Timeout=30;Encrypt=True;TrustServerCertifica
te=False;Column Encryption Setting=Enabled

Enter server password:
23356211Vinny!

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):
```

15 Minutes Remaining

Instructions Resources Help 100%

☑ 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. In the Visual Studio console, click the **Start** button to initiate the build of the console application and start it.

☑ 15. The application will start a Command Prompt window. When prompted for password, type the password that you specified in the deployment in Exercise 1 to connect to Azure SQL Database.

☑ 16. Leave the console app running and switch to the **SQL Management Studio** console.

□ 17. In the **Object Explorer** pane, right-click the **medical database** and, in the right-click menu, click **New Query**.

□ 18. From the query window, run the following query to verify that the data that loaded into the database from the console app is encrypted.

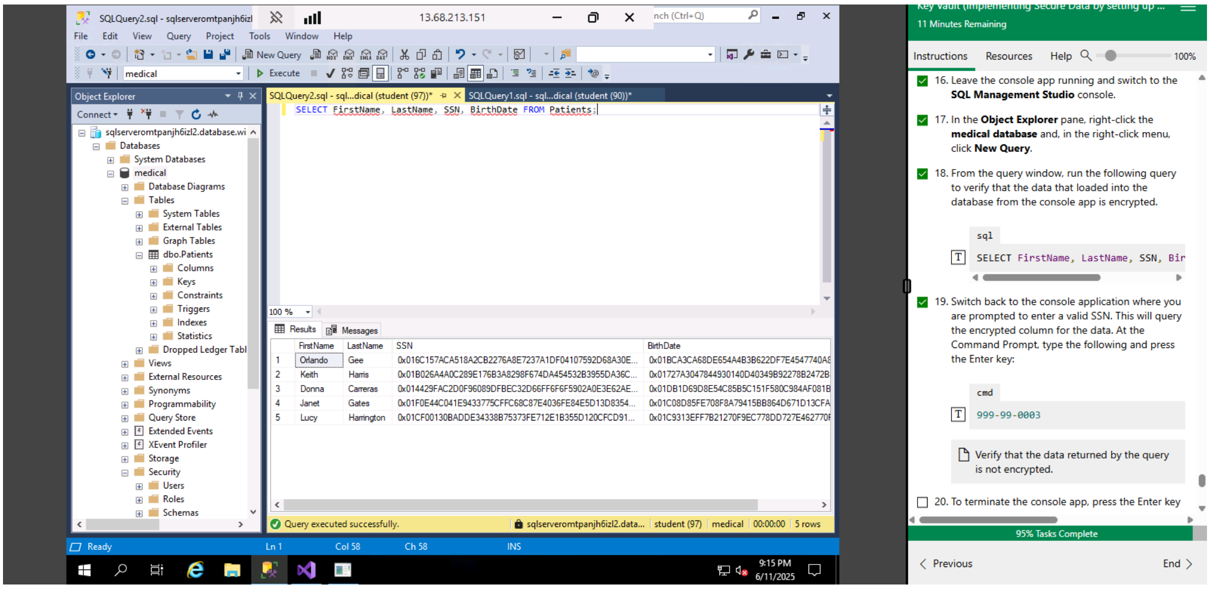
sql

```
SELECT FirstName, LastName, SSN, Bir
```

□ 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

92% Tasks Complete

Previous End



```
SELECT FirstName, LastName, SSN, BirthDate FROM Patients;
```

	FirstName	LastName	SSN	BirthDate
1	Orlando	Gee	0x016C157ACA518A2CB2276A8E7237A1DF04107592D68A30E...	0x01BCA3CA68DE654A4B3B622DF7E4547740A8
2	Keith	Harris	0x01B026A4A0C289E176B3A8298F674DA454532B3955DA36C...	0x01727A3047844930140D40349B92278B2472B
3	Donna	Carreras	0x014429FAC2D0F96089DFBEC32D66FF6F6F5902A0E3E62AE...	0x01DB1D69D8E54C85B5C151F580C984AF081B
4	Janet	Gates	0x01F0E44C041E9433775CFFC68C87E4036FE84E5D13D8354...	0x01C08D85FE708F8A79415BB864D671D13CFA
5	Lucy	Harrington	0x01CF00130BADDE34338B75373FE712E1B355D120CFCD91...	0x01C9313EFF7B21270F9EC778DD727E462770F

Query executed successfully. sqlserveromtpanjh6izl2.data... student (97) medical 00:00:00 5 rows

Key Vault (Implementing Secure Data by setting up ...

11 Minutes Remaining

Instructions Resources Help 100%

☑ 16. Leave the console app running and switch to the **SQL Management Studio** console.

☑ 17. In the **Object Explorer** pane, right-click the **medical database** and, in the right-click menu, click **New Query**.

☑ 18. From the query window, run the following query to verify that the data that loaded into the database from the console app is encrypted.

sql

```
SELECT FirstName, LastName, SSN, Bir
```

☑ 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:

cmd

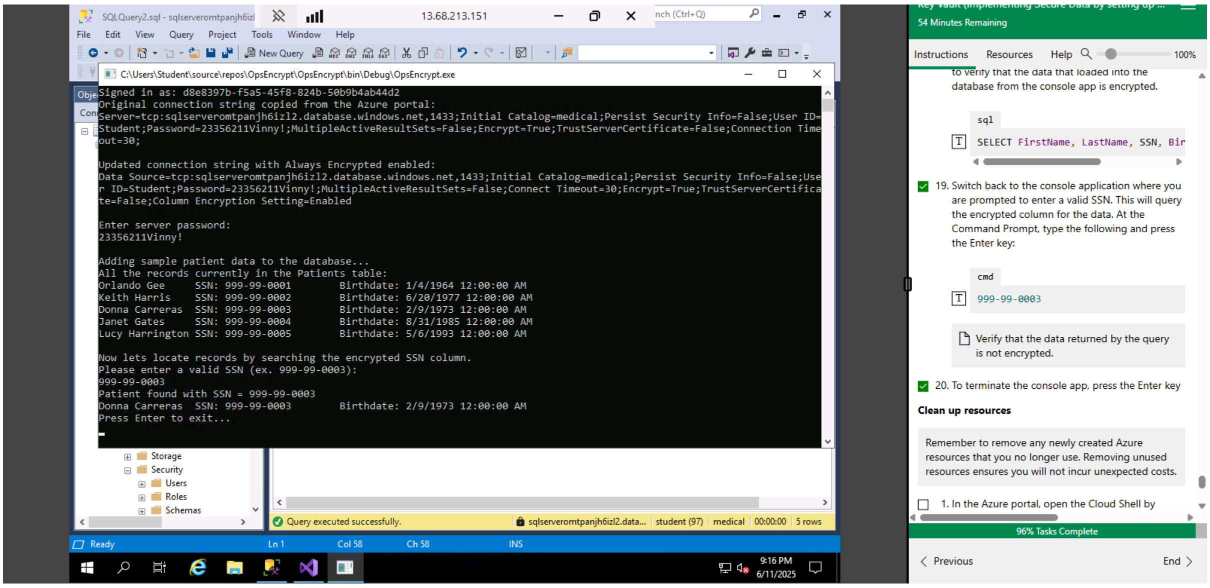
```
999-99-0003
```

Verify that the data returned by the query is not encrypted.

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

95% Tasks Complete

Previous End


SQLQuery2.sql - sqlserveromtpanjh6izl
13.68.213.151
File Edit View Query Project Tools Window Help
New Query
C:\Users\Student\source\repos\OpsEncrypt\OpsEncrypt\bin\Debug\OpsEncrypt.exe
Storage
Security
Users
Roles
Schemas
Query executed successfully.
sqlserveromtpanjh6izl2.data...
student (97)
medical
00:00:00
5 rows
Ready
Ln 1
Col 58
Ch 58
INS
9:16 PM
6/11/2025
Key Vault (Implementing Secure Data by setting up ...
54 Minutes Remaining
Instructions
Resources
Help
100%
96% Tasks Complete
Previous
End

```
Signed in as: d8e8397b-f5a5-45f8-824b-50b9b4ab44d2
Original connection string copied from the Azure portal:
Server=tcp:sqlserveromtpanjh6izl2.database.windows.net,1433;Initial Catalog=medical;Persist Security Info=False;User ID=
Student;Password=23356211Vinny!;MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Time
out=30;

Updated connection string with Always Encrypted enabled:
Data Source=tcp:sqlserveromtpanjh6izl2.database.windows.net,1433;Initial Catalog=medical;Persist Security Info=False;Use
r ID=Student;Password=23356211Vinny!;MultipleActiveResultSets=False;Connect Timeout=30;Encrypt=True;TrustServerCertifica
te=False;Column Encryption Setting=Enabled

Enter server password:
23356211Vinny!

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...
```

to verify that the data that loaded into the database from the console app is encrypted.

sql

```
SELECT FirstName, LastName, SSN, Bir
```

☑ 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:

cmd

```
999-99-0003
```

Verify that the data returned by the query is not encrypted.

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

Clean up resources

Remember to remove any newly created Azure resources that you no longer use. Removing unused resources ensures you will not incur unexpected costs.

☐ 1. In the Azure portal, open the Cloud Shell by

Conclusion

Through these exercises, a complete workflow for securing sensitive data in Azure SQL databases using Azure Key Vault was successfully implemented and tested. From initial setup and configuration of the Azure environment to enabling Always Encrypted and securely accessing encrypted columns via a console application, the lab provided real-world insights into enterprise-grade data protection strategies. This reinforced the importance of key management, secure application access, and the role of encryption in safeguarding data against unauthorized access. Additionally, the cleanup of unused resources ensured cost-effective and responsible Azure usage. Overall, the exercises enhanced understanding of integrating Azure Key Vault with SQL databases and prepared users to apply these skills in production environments.