😊 Azure SQL – Always Encrypted Feature

What is Always Encrypted?

Always Encrypted is a **data security feature** in **Azure SQL Database and SQL Server** that ensures sensitive data remains encrypted **both at rest and in transit**. Unlike **Transparent Data Encryption (TDE)**, which protects data at rest, **Always Encrypted** ensures that even database administrators (DBAs) or unauthorized users **cannot see or access** the plaintext data.

The encryption and decryption happen **on the client-side** using **encryption keys stored outside the database**, ensuring that **only authorized applications or users can access decrypted data**.

Key Features of Always Encrypted

1. **Client-Side Encryption**
    - Data is **encrypted before being stored** in the database and is decrypted **only by authorized client applications**.
2. **Separation of Roles**
    - Even **DBAs or Azure administrators** cannot access sensitive data.
3. **Protects Data in Transit & at Rest**
    - Unlike TDE, which encrypts only at rest, Always Encrypted ensures **end-to-end encryption**.
4. **Column-Level Encryption**
    - Allows selective encryption of specific columns, such as **credit card numbers or Social Security Numbers (SSNs)**.
5. **Integration with Azure Key Vault**
    - Uses **Azure Key Vault** for secure **key management and rotation**.

Use Cases of Always Encrypted

1. Protecting Sensitive Data

- Encrypting **PII (Personally Identifiable Information)** such as SSNs, medical records, and credit card numbers.

2. Ensuring Regulatory Compliance

- Helps businesses comply with **GDPR, HIPAA, PCI DSS**, and other data security regulations.

3. Securing Financial Transactions

- Used in banking and fintech applications to **encrypt financial transactions and account details**.

4. Healthcare Data Protection

- Ensures **patient data privacy** by encrypting medical records.

5. Preventing Insider Threats

- Ensures that **even database administrators cannot see sensitive data**.

Comparison: Always Encrypted vs. Transparent Data Encryption (TDE)

Feature	**Always Encrypted**	**Transparent Data Encryption (TDE)**
Encryption Scope	Column-level	Entire database (storage-level)
Encryption Location	Client-side (before data enters the DB)	Server-side (storage-level)
Protection	Protects data at rest, in transit, and from DBAs	Protects data at rest only
Performance Impact	Higher (due to client-side encryption)	Minimal
Use Case	Protecting **highly sensitive data** (SSNs, credit cards)	General database encryption (preventing stolen backups from being readable)

Conclusion

Always Encrypted in Azure SQL Database ensures **end-to-end security** by keeping sensitive data encrypted **even from DBAs and Azure administrators**. It is ideal for **financial, healthcare, and regulatory-compliant applications**, offering **stronger data protection than TDE** while ensuring that **only authorized client applications can access decrypted data**.

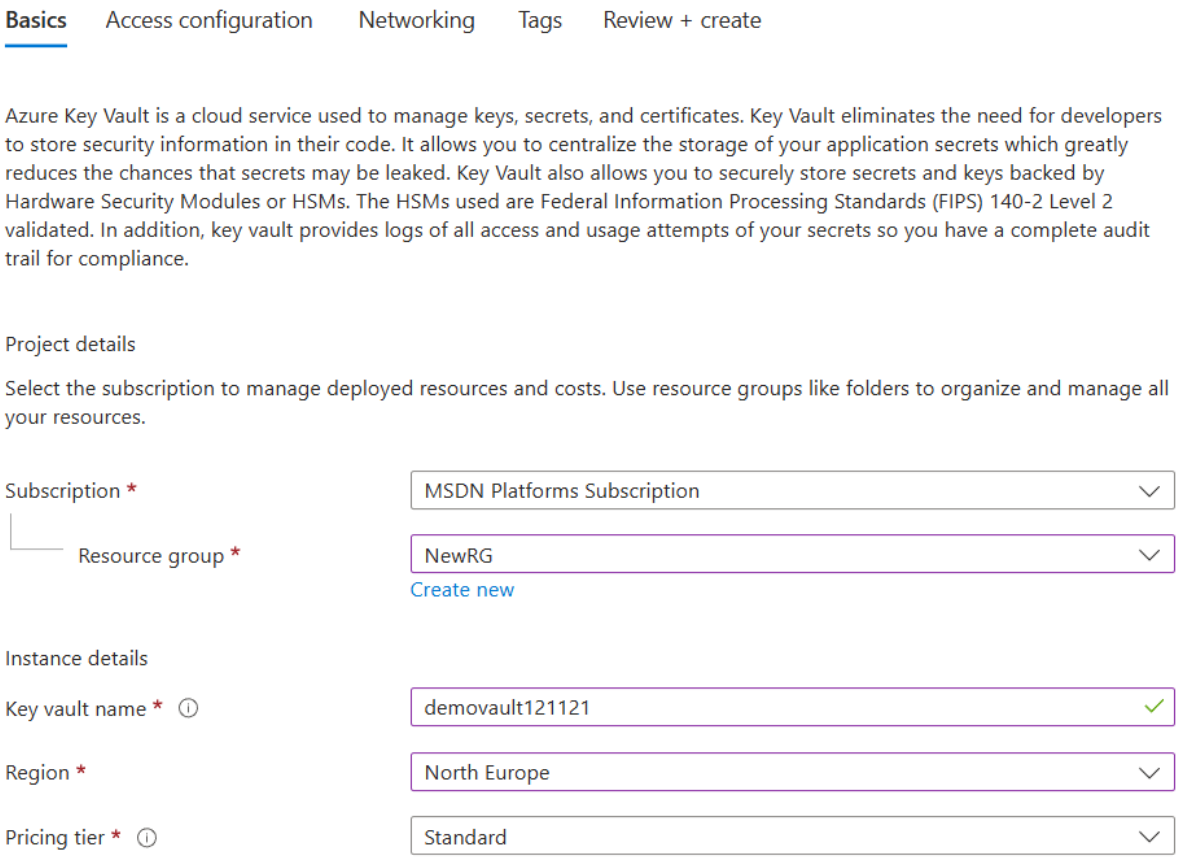
In this lab, we enable the Always Encrypted feature in Azure SQL Database using Azure Key Vault to secure data at rest and in transit. First, create an Azure Key Vault, configure access policies, and assign cryptographic permissions. Then, in SQL Server Management Studio (SSMS), select a table (e.g., Customer), choose Encrypt Columns, and specify Azure Key Vault for encryption. After encryption completes, the selected column (e.g., email) is encrypted, ensuring protection from unauthorized access.

End Goal: Implement Always Encrypted to enhance data security, confidentiality, and compliance, ensuring only authorized applications can decrypt sensitive data.

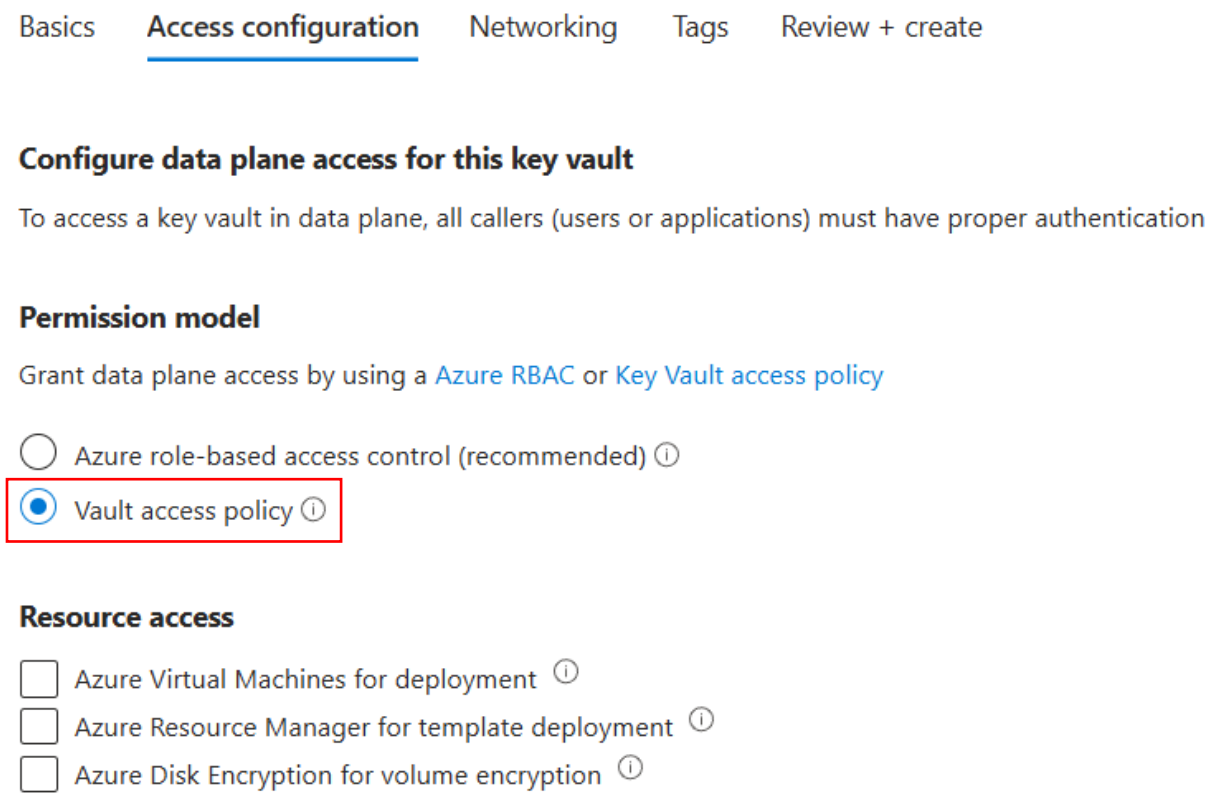
😄 To begin with the Lab

1. In this lab we will use the always-encrypted feature of Azure SQL Database by making use of Azure Key Vault. We use this feature to encrypt the data at rest or in motion.
2. Now to encrypt the data we should have an encrypted key and we have to store that key in the Azure key vault.

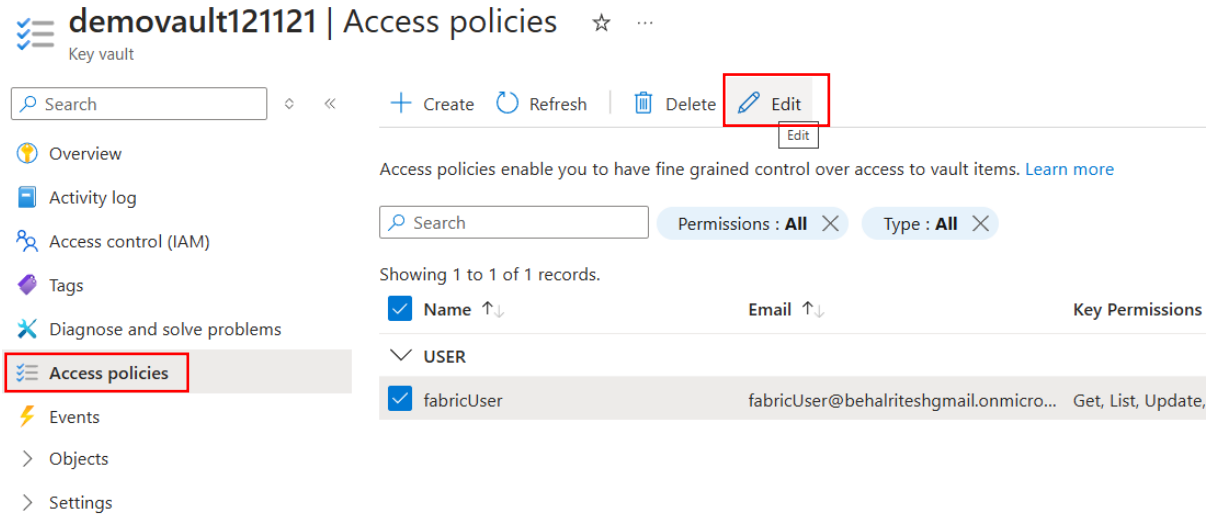
3. From the marketplace search for Azure key vault and move to the creation page.
4. Choose your resource group and give a unique key vault name.



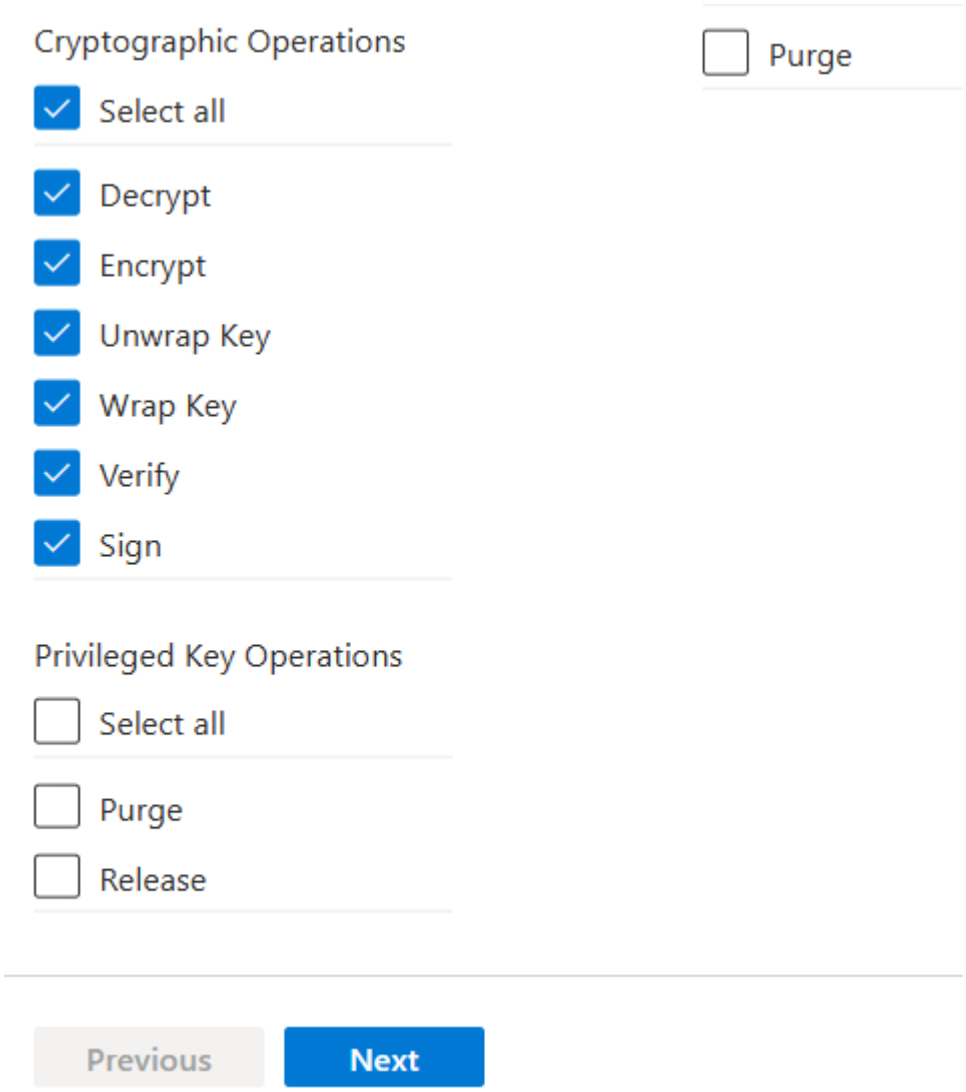
5. Then on the next option that is Access configuration choose vault access policy and create your key vault.



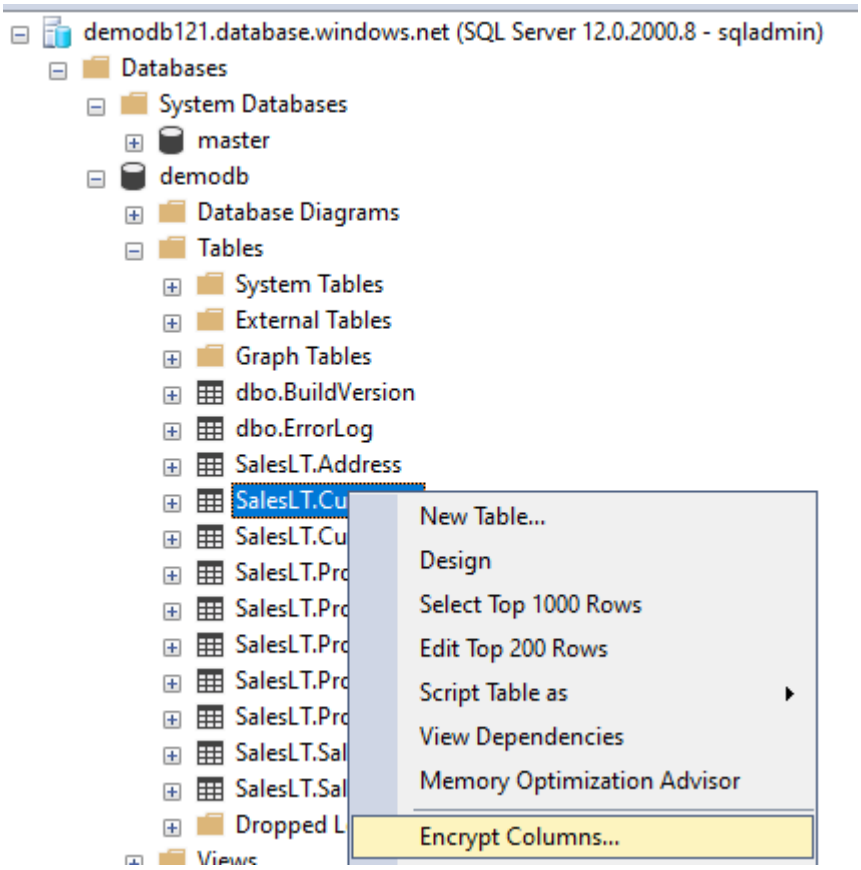
6. Then inside your key vault choose access policies then choose the user and click on edit.



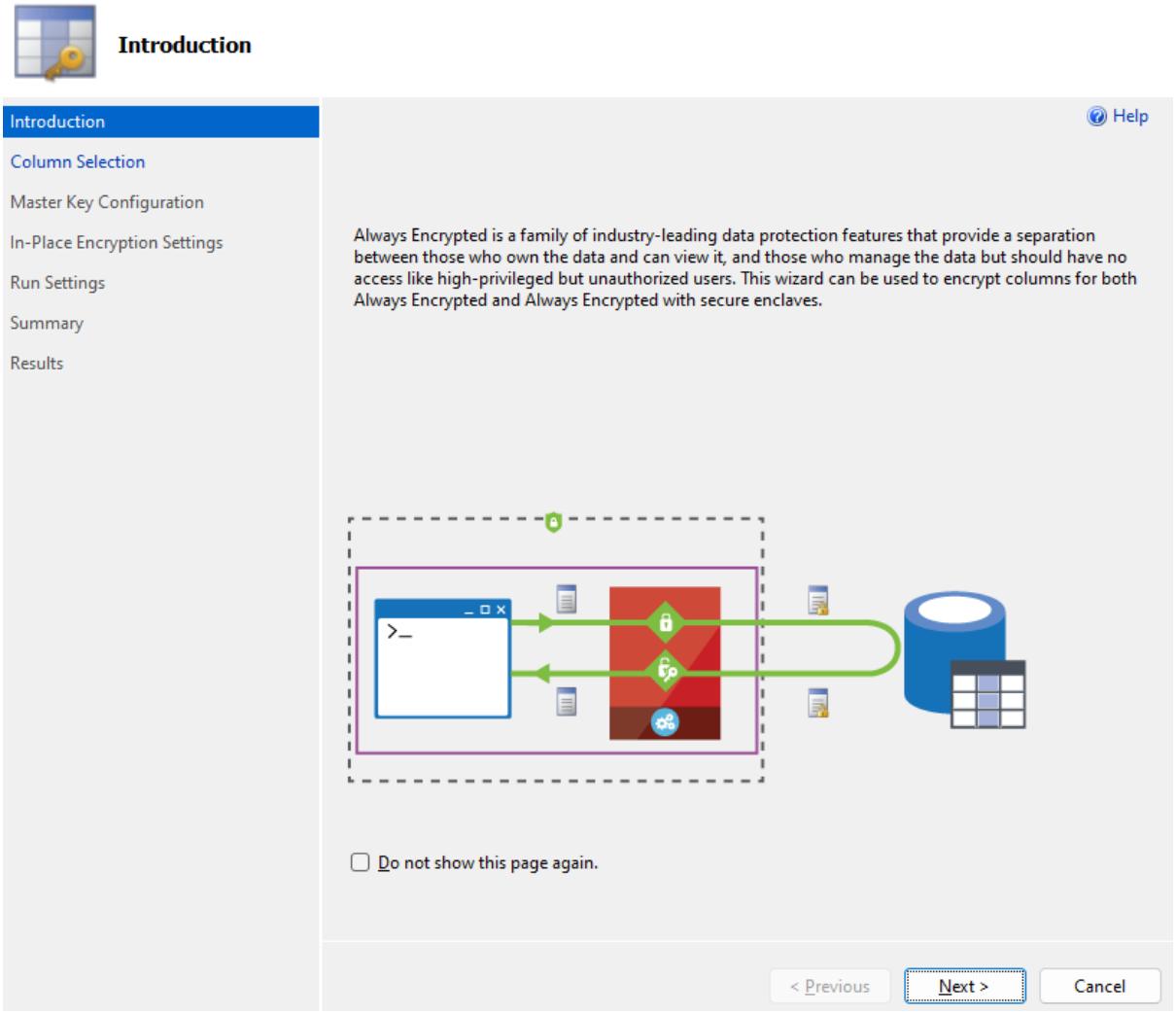
7. Then in the permissions you have to assign yourself the Cryptographic operations and click on save.



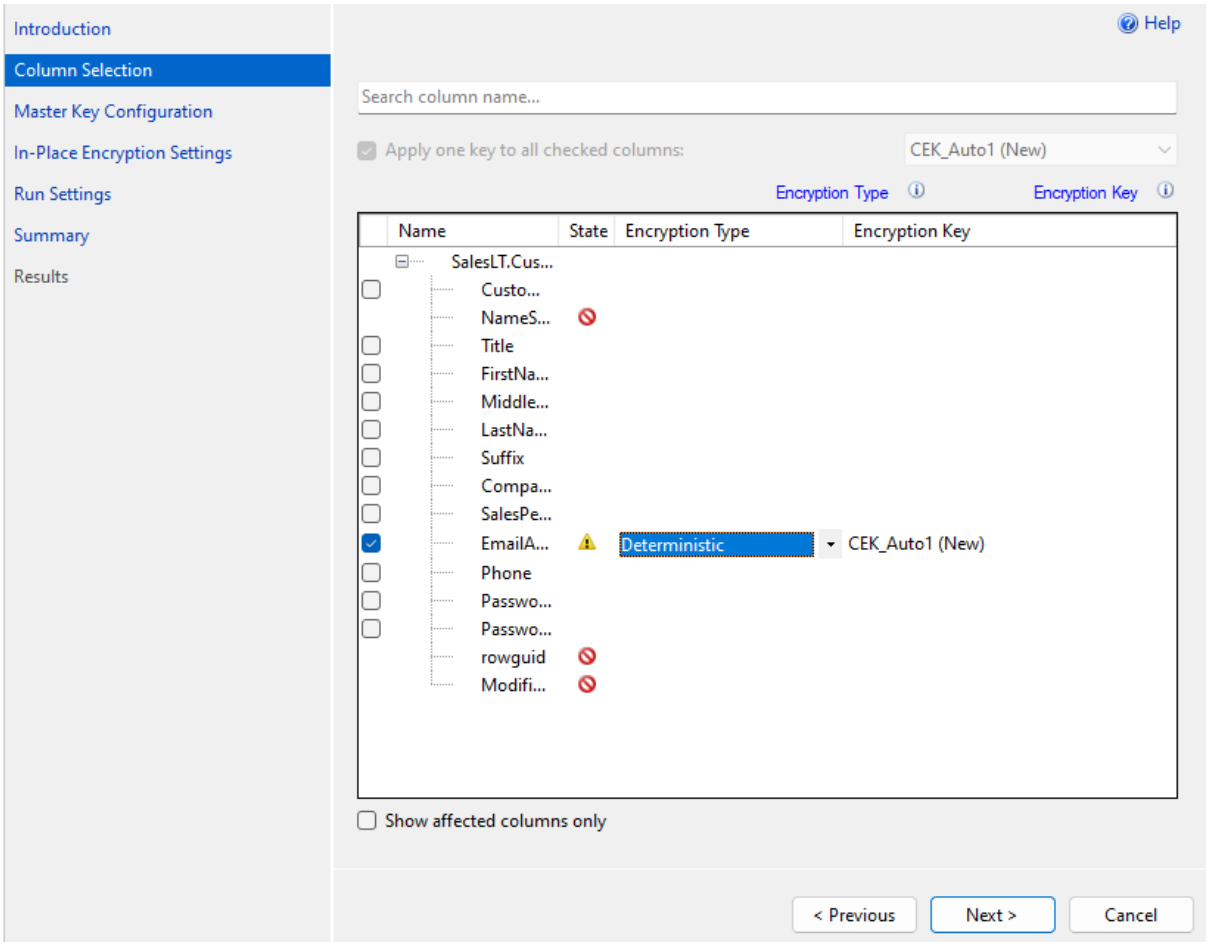
8. Now open your SQL Server Management Studio as the admin user expands your tables then choose the Customer table right click on it and choose Encrypt Columns.



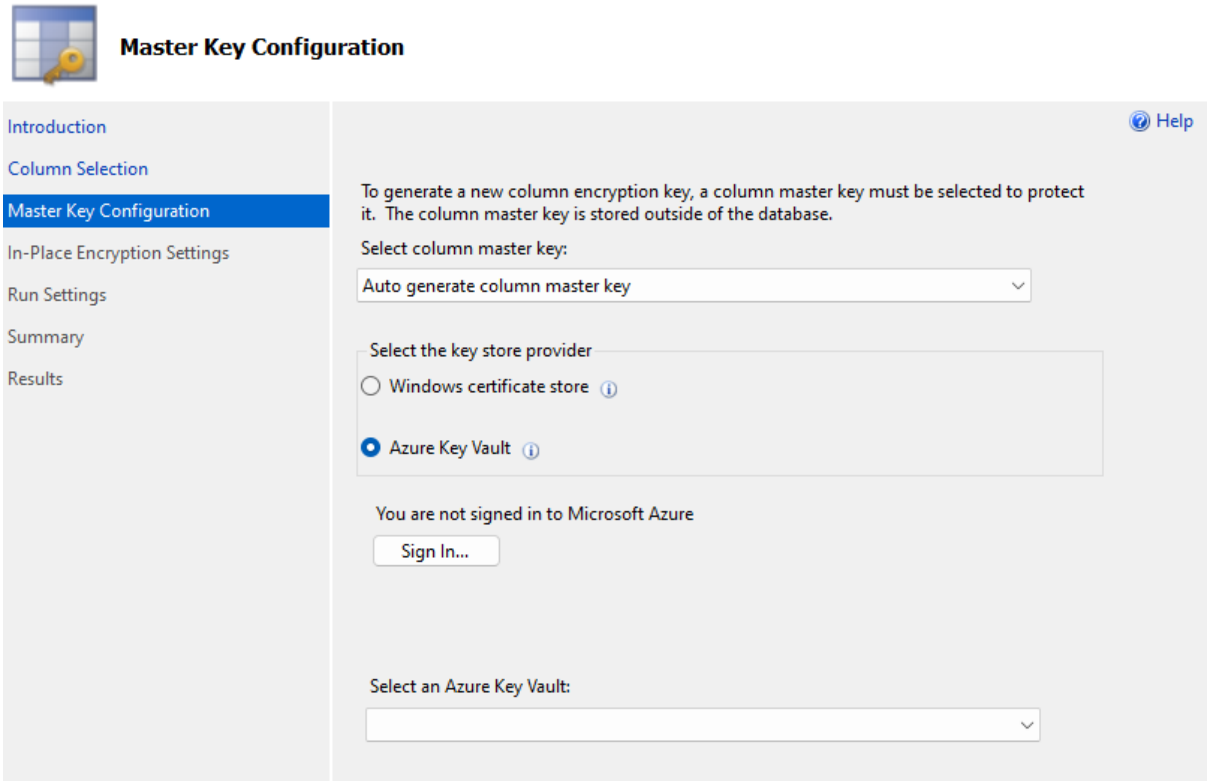
9. You will see that a wizard has been opened. Click on next.



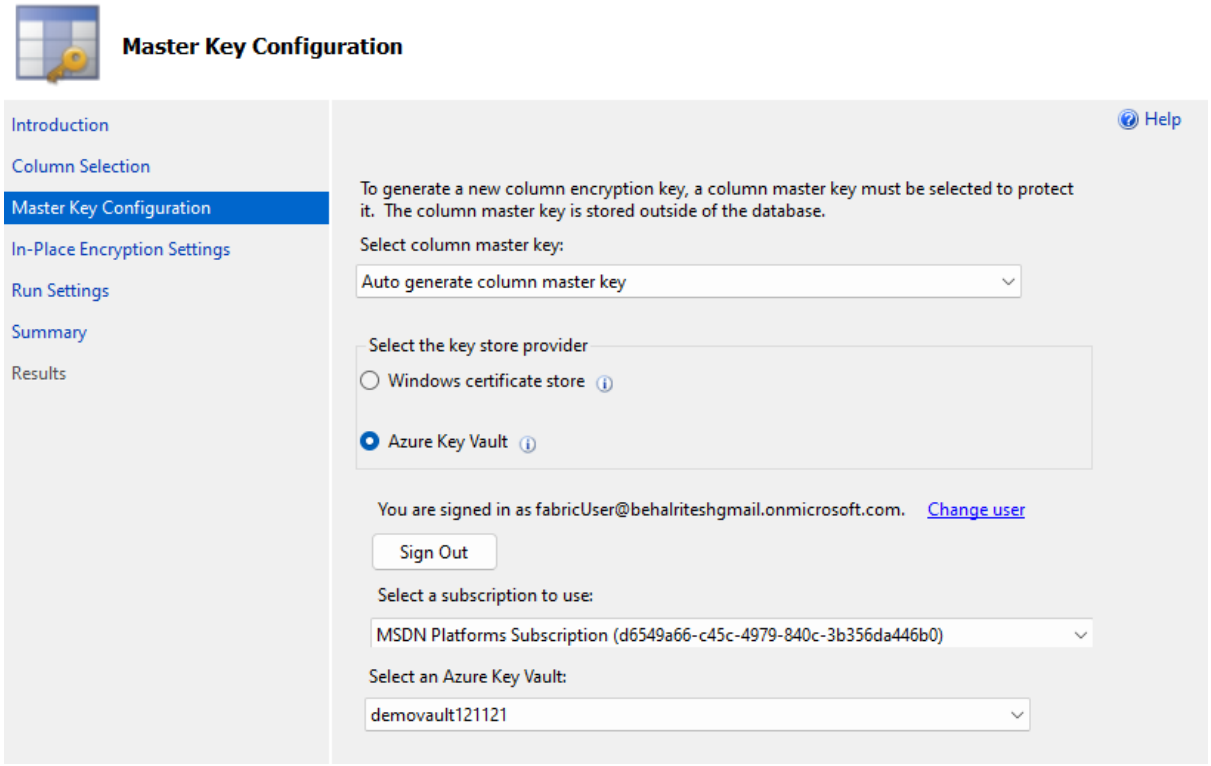
10. Now choose which column you want to encrypt and what is the encryption type. Click on next.



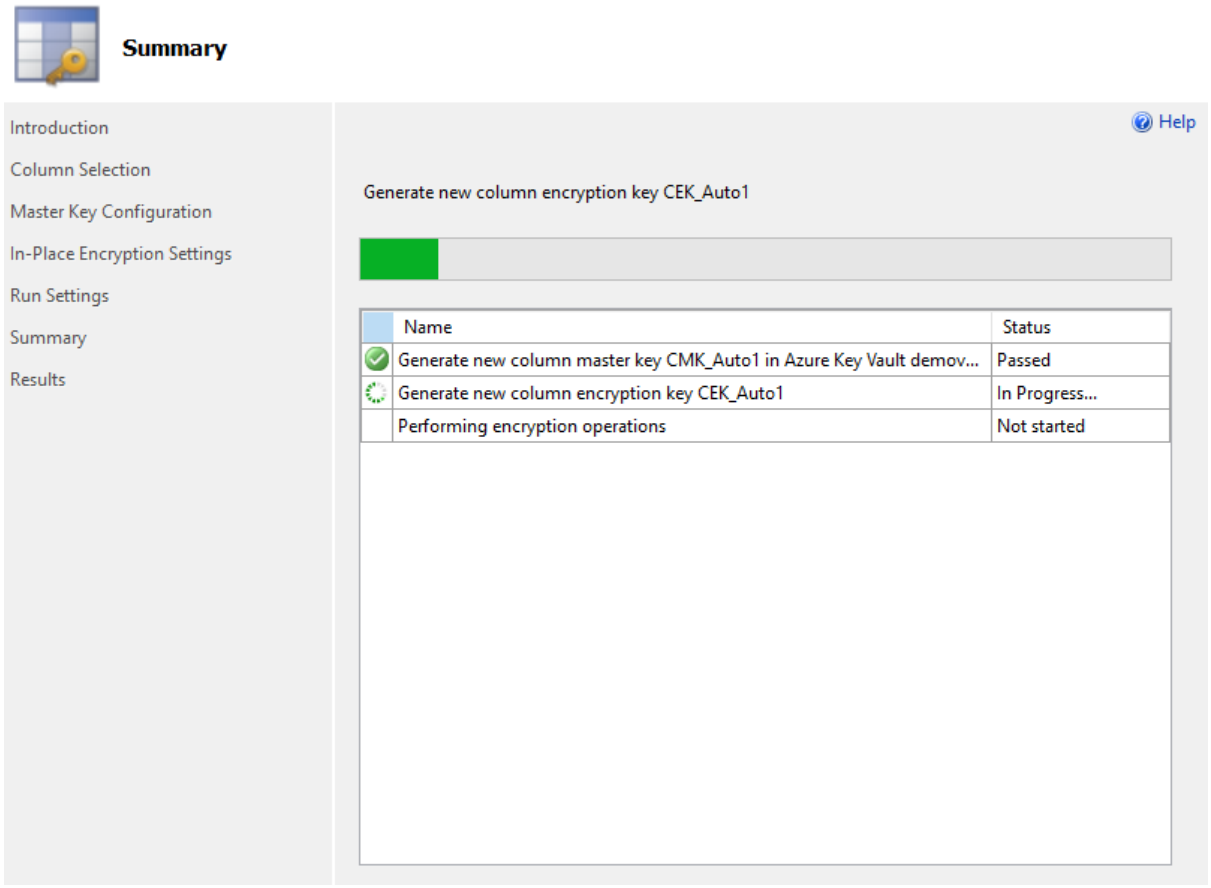
11. On the next option choose Azure key Vault and click sign in.



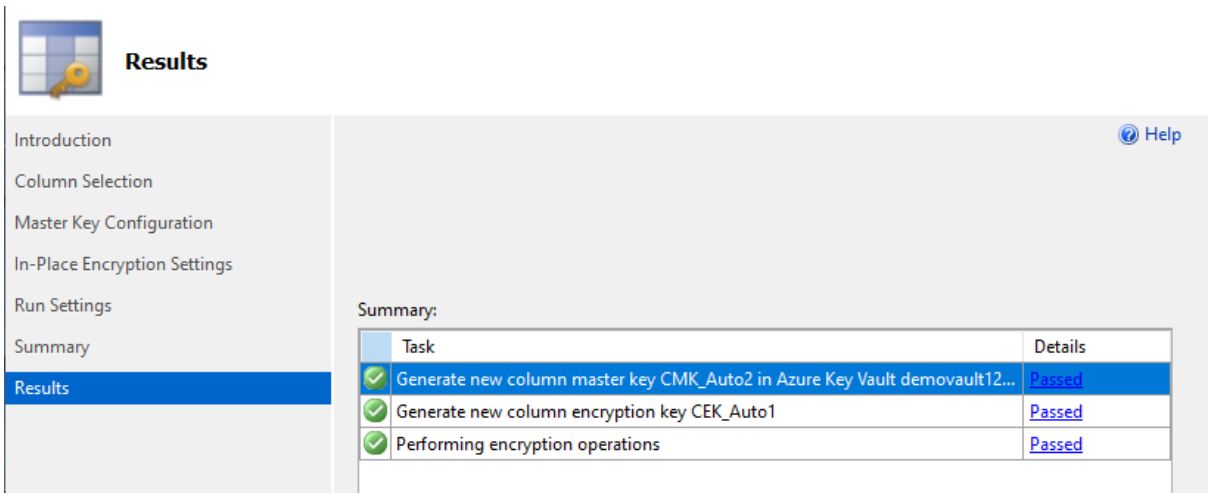
12. After signing in you will see your subscription name and the key vault name.



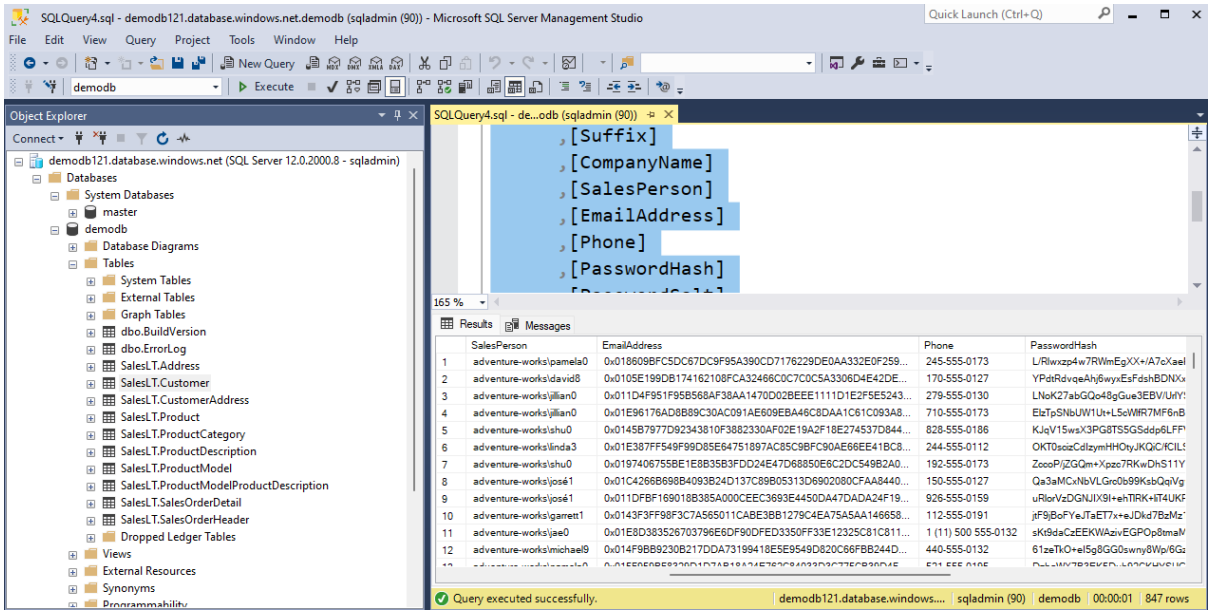
13. Then finish up the things and you will see that a summary page has been opened and here you can see that the encryption is happening. It might take 10-15 minutes to complete.



14. Once the results are completed you will see every tasked has been given the status of passed.



15. Now if we fire up the query once again we can see that our email has been encrypted.



16. If you go to the key vault and open up the keys you will see a key.