

Migrate TDE Enabled DB to MI

Last updated by | Vitor Tomaz | May 17, 2022 at 10:17 AM PDT

Contents

- [Migration Scenario](#)
 - [Issue](#)
 - [Investigation/Analysis](#)
- [BYOK TDE](#)
- [Backup/Restore TDE Encrypted Database](#)
- [Classification](#)

Migration Scenario

Issue

When customer has on-premise database which has enabled TDE, they would like to migrate the database using backup/restore or DMS without disabling TDE.

Investigation/Analysis

They need to follow the public documents: [Migrate certificate of TDE protected database to Azure SQL Database Managed Instance](#) 

A few things to be careful:

- PowerShell commands in official documents only works on Azure PowerShell 5.1 version. If the PowerShell version is higher than 5.1, it will not work since "-Encoding Byte" is not supported. Please replace with below:

```
$fileContentBytes = Get-Content 'C:/full_path/TDE_Cert.pfx' -AsByteStream
$base64EncodedCert = [System.Convert]::ToBase64String($fileContentBytes)
$securePrivateBlob = $base64EncodedCert | ConvertTo-SecureString -AsPlainText -Force
$password = "<password>"
$securePassword = $password | ConvertTo-SecureString -AsPlainText -Force
Add-AzSqlManagedInstanceTransparentDataEncryptionCertificate -ResourceGroupName "<resourceGroupName>" `
    -ManagedInstanceName "<managedInstanceName>" -PrivateBlob $securePrivateBlob -Password $securePas
```


- A migrated certificate is used for restore of the TDE-protected database only. Soon after restore is done, the migrated certificate gets replaced by a different protector, either a service-managed certificate or an asymmetric key from the key vault, depending on the type of the TDE you set on the instance.
 - You can use TSQL to check the current encryption thumbprint:

```
select b.name, b.is_encrypted, a.key_algorithm, a.encryptor_thumbprint, a.key_length, a.encryptor_type fr
```

- o Powershell script to check the thumbprint of the Key Vault Key configured on Managed Instance

```
Get-AzSqlInstanceKeyVaultKey -ResourceGroupName <RG Name> -InstanceName <Managed Instance Name> -KeyId '<
```

BYOK TDE

SQL Managed Instance support customer managed asymmetric key(also known as BYOK) in Azure Key Vault. Certificate is not supported. [Manage Transparent Data Encryption in a Managed Instance using your own key from Azure Key Vault](#) 

Backup/Restore TDE Encrypted Database

If TDE is enabled with **service managed key**, you cannot backup a TDE enabled database in MI. The backup in MI has to be performed over a URL that points to the Azure blob storage. Having TDE enabled disables this capability since the certificate cannot be backed up along with encrypted DB.

Following error message shows up in this situation:

```
System.Data.SqlClient.SqlError: The backup operation for a database with service-managed transparent data encr
```

If TDE is enabled with **customer managed key**, you can backup a TDE enabled database in MI to Azure blob storage.

Classification

Root cause tree:

How good have you found this content?



-