Fortanix®

# Integration Guide

## USING DATA SECURITY MANAGER WITH MSSQL SERVER TDE – STANDALONE SERVER

*VERSION 1.0*

Fortanix, Inc. | 3910 Freedom Circle | Suite 104 | Santa Clara, CA 95052| United States of America | +1 (650) 943-2484 |
info@fortanix.com | www.fortanix.com

## TABLE OF CONTENTS

## 1.0 INTRODUCTION

This document is a step-by-step guide to implement Microsoft SQL Transparent Data Encryption (TDE) using the Fortanix DSM.

**NOTE**: Ensure that you have performed the steps from *Data Security Manager with MSSQL TDE Integration Guide – Before You Begin Guide.*

## 2.0 ENABLING SQL FEATURES

Run the following commands if Extensible Key Management (EKM) is not supported or enabled in the SQL Server Edition:

```
sp_configure 'show advanced', 1
GO
RECONFIGURE
GO
sp_configure 'EKM provider enabled', 1
GO
RECONFIGURE
GO
```

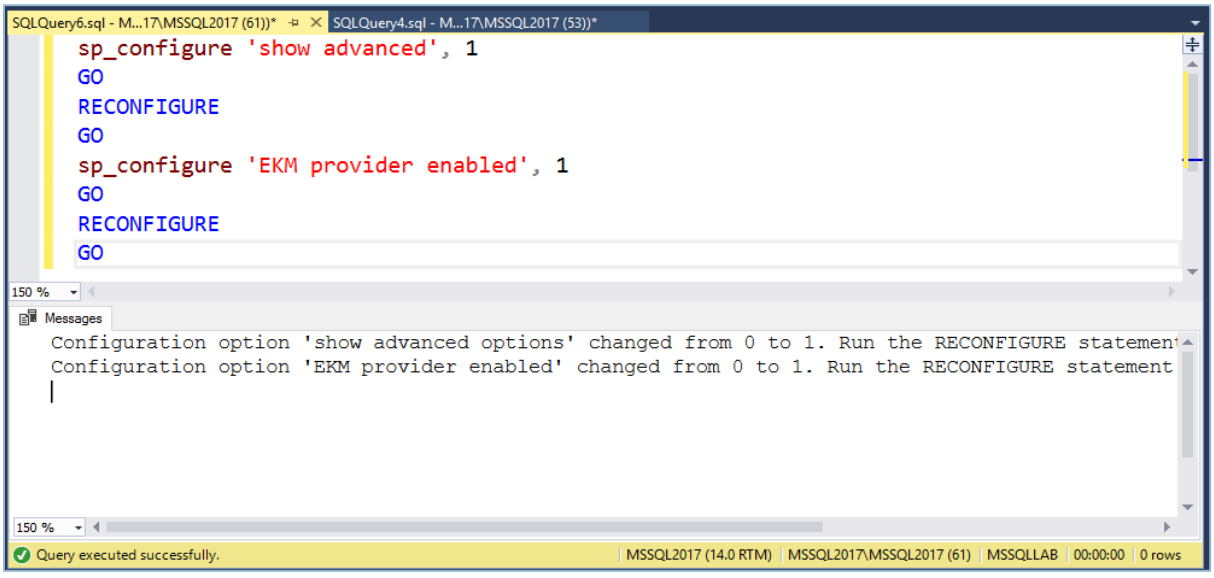


FIGURE 1: RUN COMMANDS FOR ERROR SCENARIO

## 3.0 CREATING CRYPTOGRAPHIC PROVIDER

Run the following commands to use the correct location of the `EKM DLL`:

```
CREATE CRYPTOGRAPHIC PROVIDER EKM_Prov
FROM FILE = 'C:\Program Files\Fortanix\KmsClient\FortanixKmsEkmProvi
der.dll' ;
GO
```

Where,

- `EKM_Prov` refers to the name of the provider defined by the user.

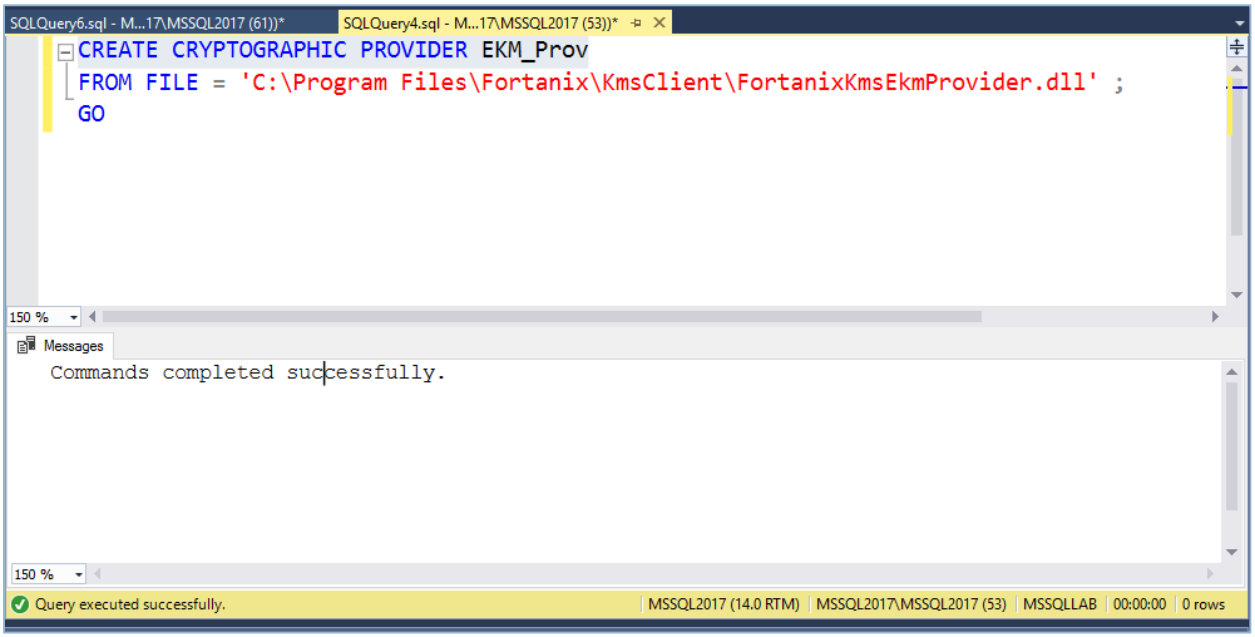


FIGURE 2: CREATE CRYPTOGRAPHIC PROVIDER

## 4.0 CREATING CREDENTIALS (SYSADMIN)

This section describes the steps to create the credentials to generate the Master Encryption Key (MEK) on the Fortanix DSM using the SQL administrator.

The SQL administrator requires permission to connect to Fortanix DSM to generate the key.

1. Perform the following steps to get the API key:
   a. Log in to the Fortanix DSM.
   b. From the UI left panel, click the **Apps** tab.
   c. Click **COPY API KEY** to copy the API key of your application and then paste the DSM API key as the value for the `SECRET` parameter in the next command.

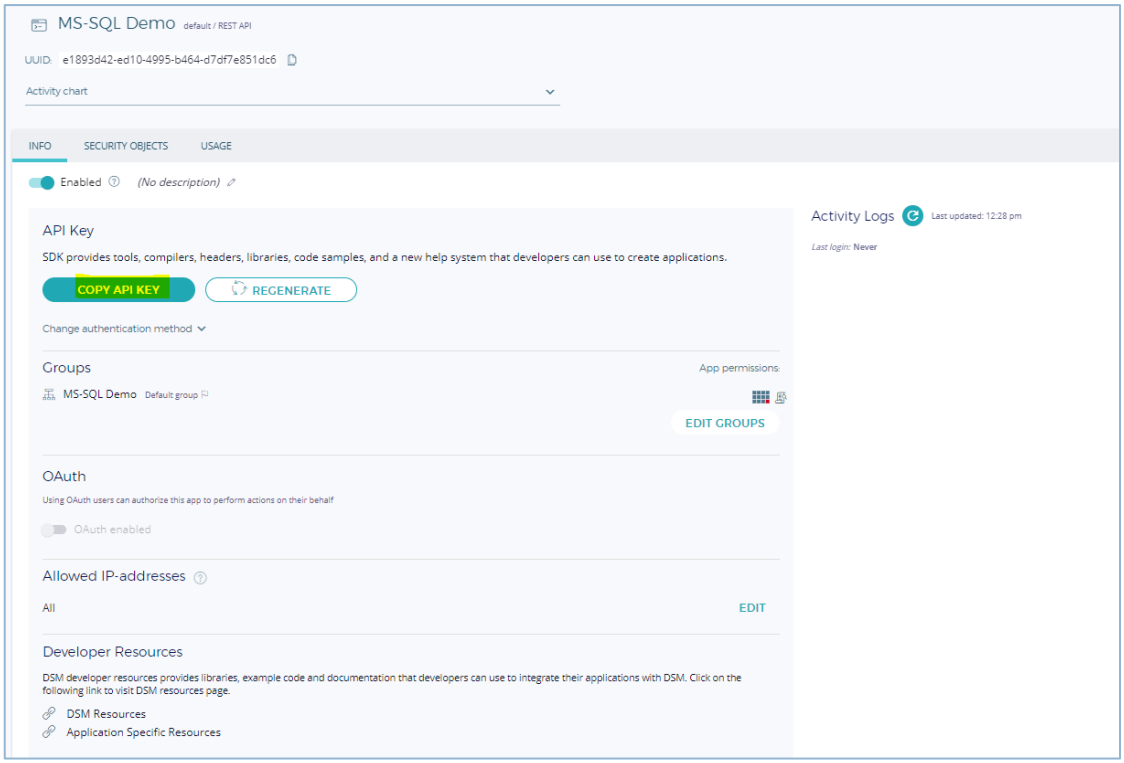


**FIGURE 3: COPY API KEY**

2. Run the following commands to create a credential using the copied API key in your SQL Server Studio that will be used by the system administrators:

```
CREATE CREDENTIAL sa_ekm_tde_cred
WITH IDENTITY = 'Identity1',
SECRET = '<DSM API KEY>'
```

```
FOR CRYPTOGRAPHIC PROVIDER EKM_Prov ;
GO
```

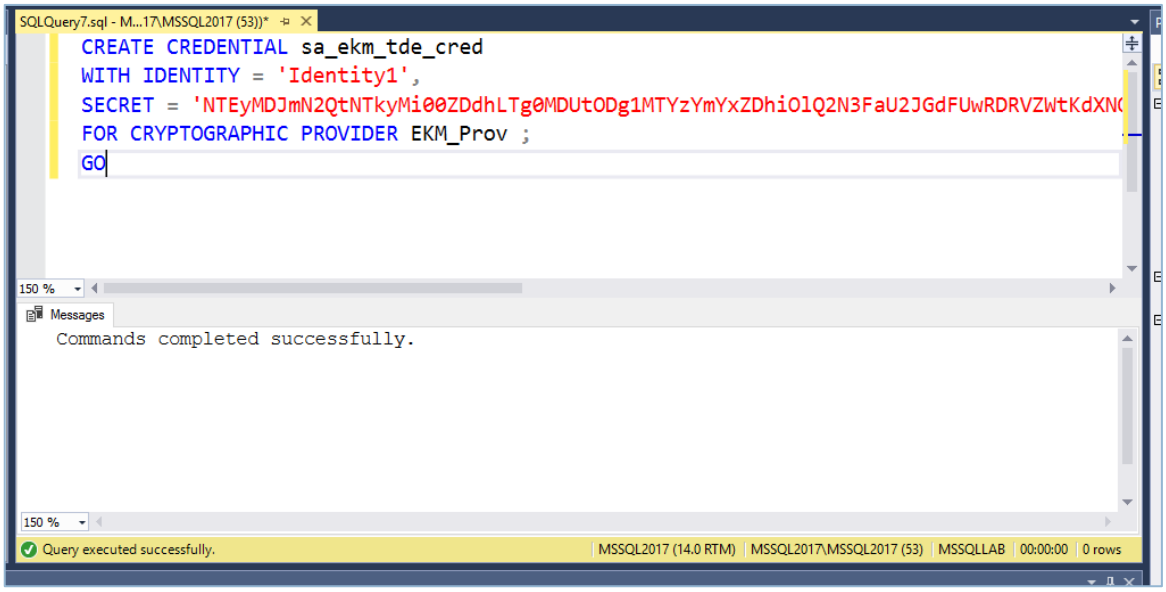


**FIGURE 4: CREATE CREDENTIAL**

3. Add the credential to a high privileged user such as your own domain login in the format `[DOMAIN\login]`:

```
ALTER LOGIN "<Domain>\Administrator"
ADD CREDENTIAL "sa_ekm_tde_cred";
GO
```

Run the following commands in case there is no domain, and the machine is part of a workgroup or standalone:

```
ALTER LOGIN "LOCALHOST\Administrator"
ADD CREDENTIAL "sa_ekm_tde_cred";
GO
```

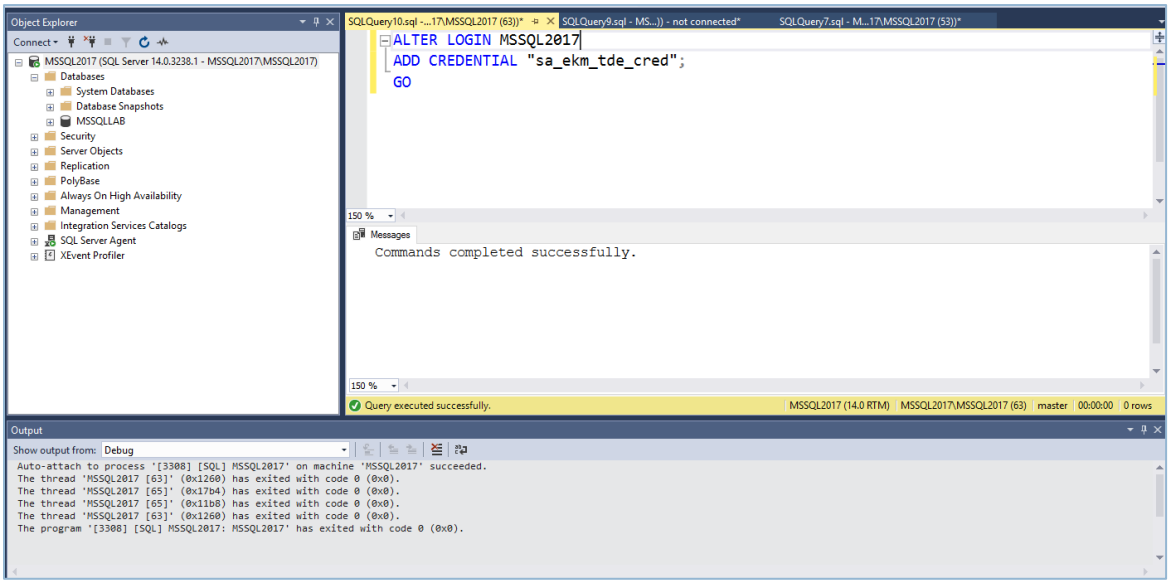


**FIGURE 5: COMMAND FOR NO DOMAIN**

If you are not an administrator and hence unable to alter the login, open the Object Explorer and map the credentials as shown in the following image:

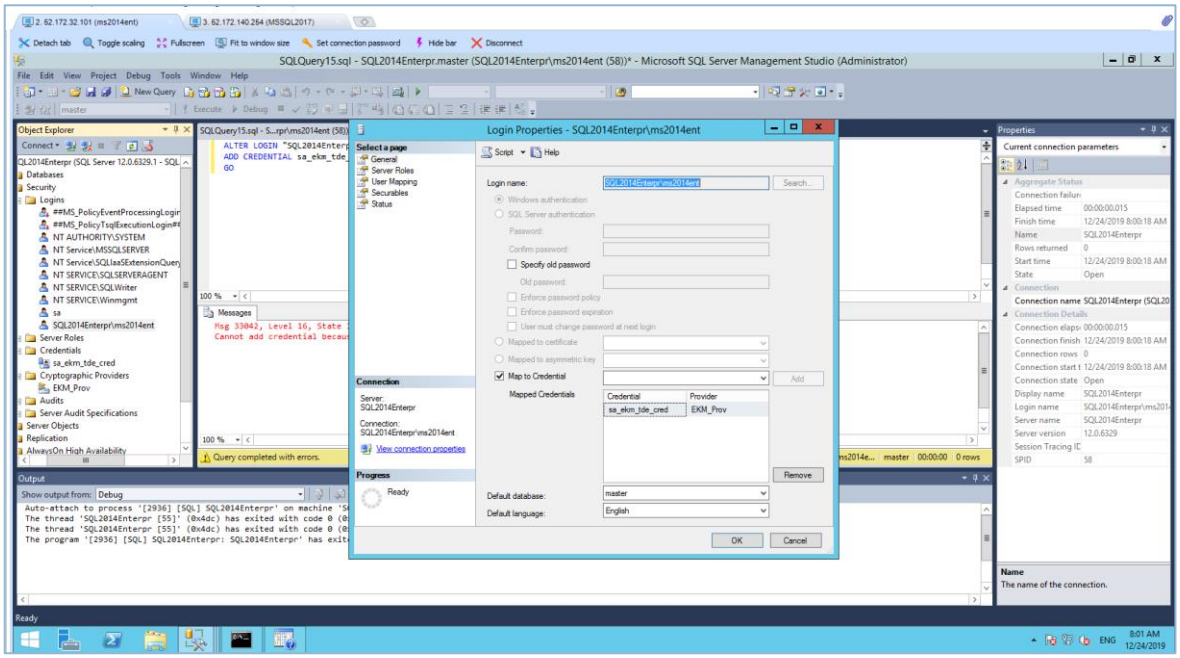


**FIGURE 6: MAP CREDENTIALS**

## 5.0 CREATING ASYMMETRIC KEYS (MEK)

The MSSQL admin has the credentials associated with creating the Master Encryption Key (MEK) on Fortanix DSM. This section describes the steps to create the asymmetric keys.

Run the following commands to create an asymmetric key stored inside the EKM provider:

```
USE master;
GO
CREATE ASYMMETRIC KEY ekm_login_key
FROM PROVIDER [EKM_Prov]
WITH ALGORITHM = RSA_2048,
PROVIDER_KEY_NAME = 'SQL_Server_Key';
GO
```

Where,

- `ekm_login_key` refers to the master key alias on the MSSQL database.
- `EKM_Prov` refers to the Fortanix EKM Provider.
- `SQL_Server_Key` refers to the key created on the Fortanix DSM.

**NOTE:** It is recommended to add versions to the Fortanix DSM keys for an easier key rotation process.

For example:

```
USE master;
GO
CREATE ASYMMETRIC KEY ekm_login_key_v1
FROM PROVIDER [EKM_Prov]
WITH ALGORITHM = RSA_2048,
PROVIDER_KEY_NAME = 'SQL_Server_Key_v1';
GO
```

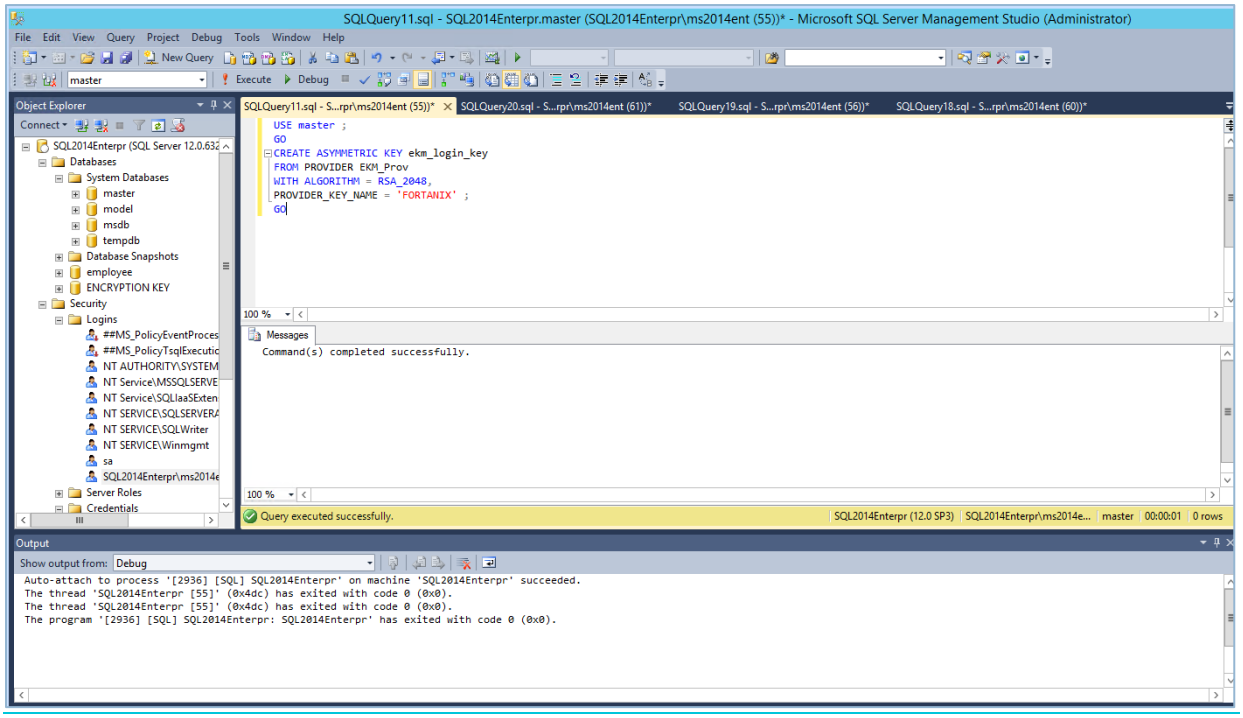


**FIGURE 7: CREATE ASYMMETRIC KEY**

## 6.0 CREATING CREDENTIALS (DB ENGINE)

Run the following commands to create a credential that will be used by the database engine:

```
USE master;
CREATE CREDENTIAL ekm_tde_cred
WITH IDENTITY = 'Identity2',
SECRET = '<DSM API KEY>'
FOR CRYPTOGRAPHIC PROVIDER EKM_Prov;
```

Where,

- `ekm_tde_cred` refers to the name of the credentials.
- `Identity2` refers to the identity name. The value can be any name.
- `EKM_Prov` refers to the Fortanix EKM Provider.
- `SECRET` refers to the Fortanix DSM API Key. *Refer to the "Section 4- Creating Credentials (SysAdmin)" to get the DSM API Key.*

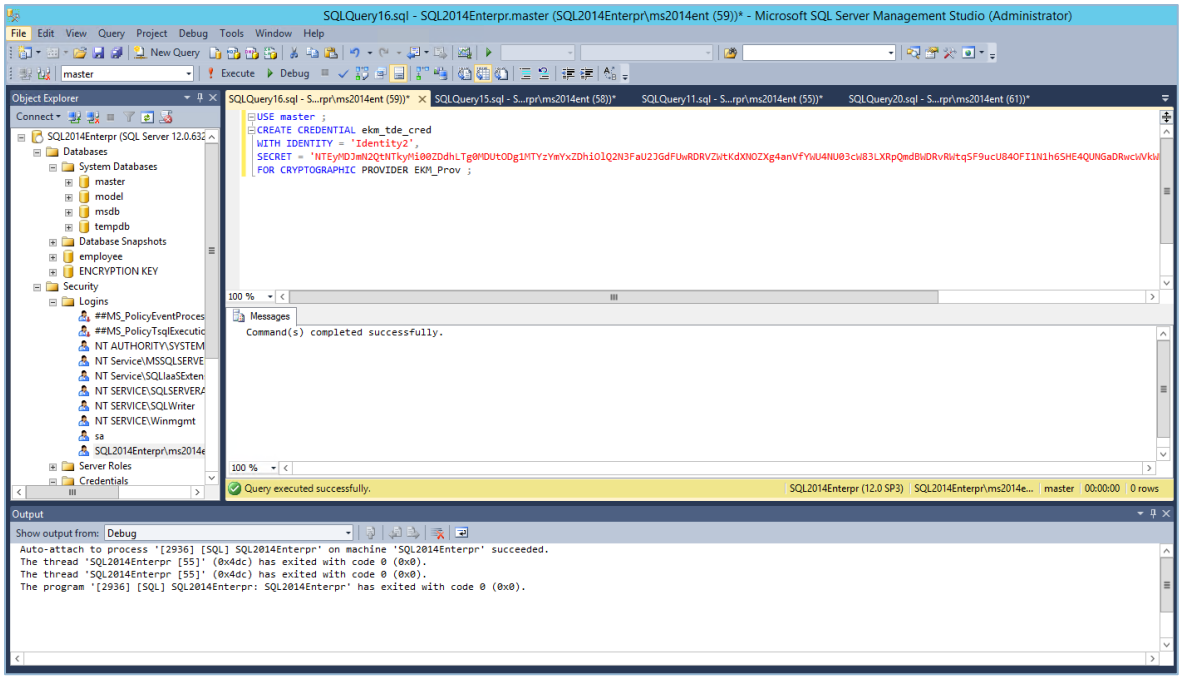


**FIGURE 8: CREATE CREDENTIAL FOR DATABASE ENGINE**

## 7.0 CREATING LOGIN (DB ENGINE)

Run the following commands to create a login from an asymmetric key and map credentials to the login:

```
CREATE LOGIN EKM_Login
FROM ASYMMETRIC KEY ekm_login_key ;
GO
ALTER LOGIN EKM_Login
ADD CREDENTIAL ekm_tde_cred ;
GO
```

Where,

- `ekm_login_key` refers to the master key alias on MSSQL database. This key is already created in *"Section 5- Creating Asymmetric keys"*.

- `EKM_Login` refers to the login name.
- `ekm_tde_cred` refers to the key created on the Fortanix DSM. This credential is already created in "*Section 6- Creating Credentials*".

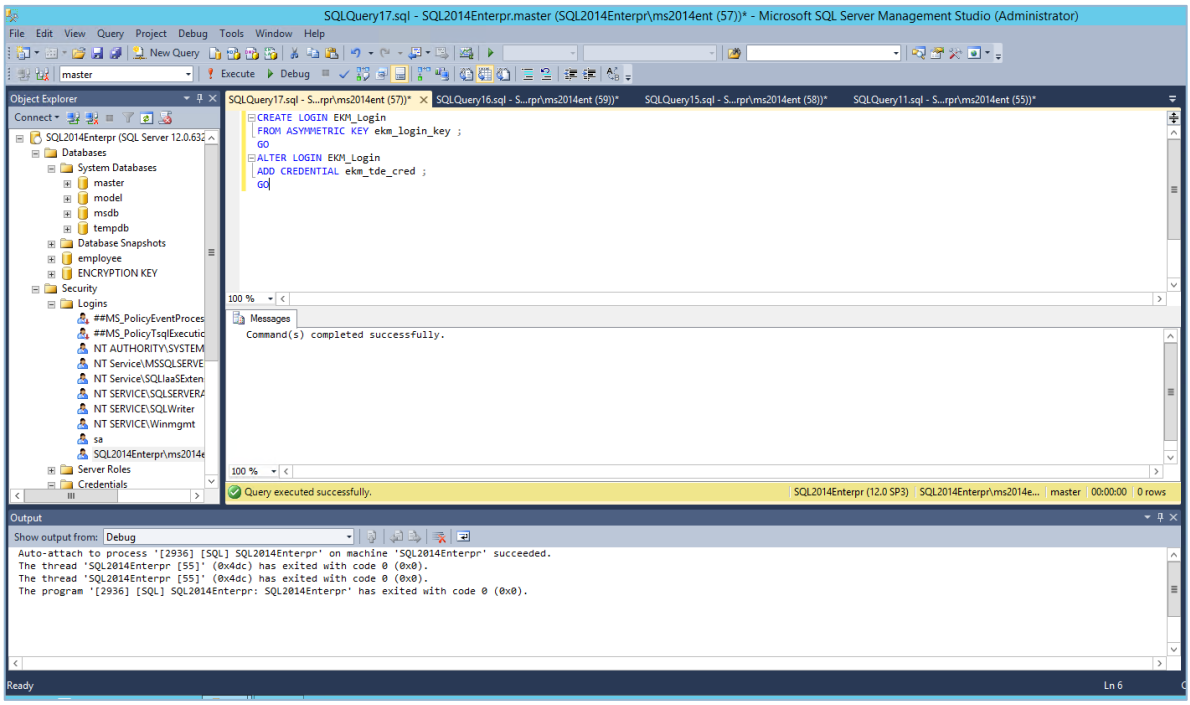


**FIGURE 9: ADD NEW CREDENTIAL TO LOGIN**

## 8.0 CREATING SAMPLE DATABASE

This section describes the steps for creating sample database to enable TDE.

1. Run the following commands to create database `employee`:

```
CREATE DATABASE employee
```

2. Run the following commands to create table `employee`:

```
USE employee
CREATE TABLE employee (first_name VARCHAR(128),last_name VARCHAR(128
),empID DECIMAL,salary DECIMAL(6));
GO
```

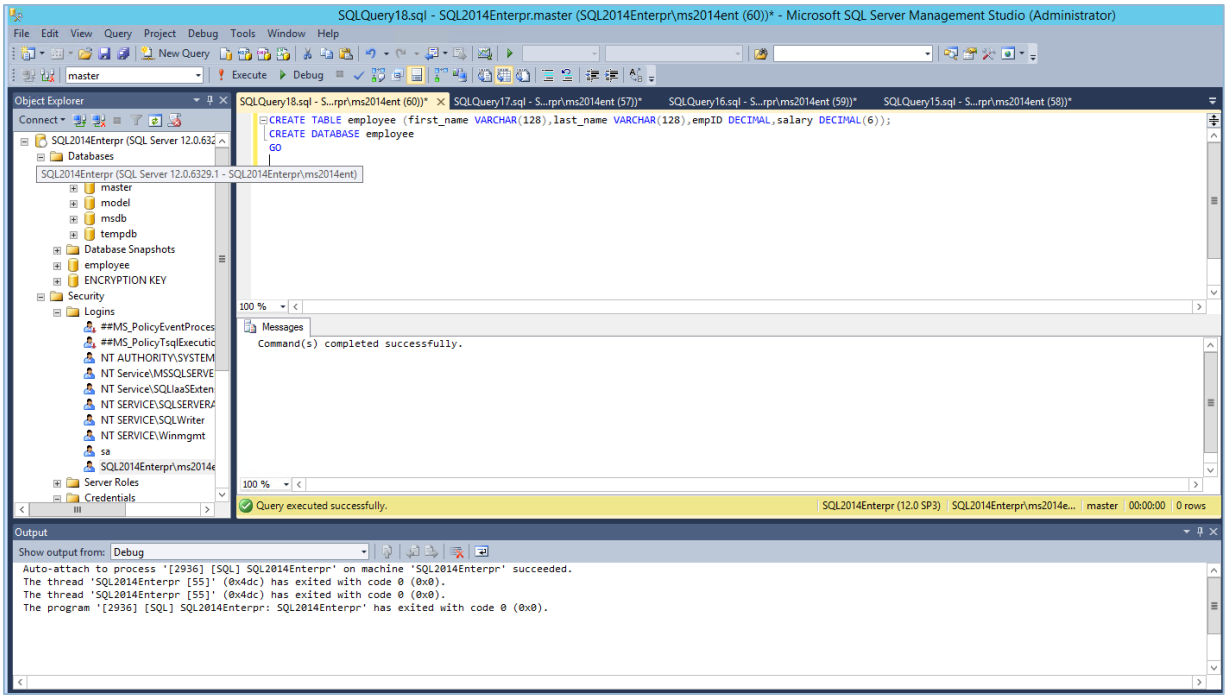


**FIGURE 10: CREATE TABLE**

## 9.0 CREATING DATA ENCRYPTION KEY (DEK)

Run the following commands to create the Data Encryption Key (DEK) that will be used for TDE:

```
USE employee
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM  = AES_256
ENCRYPTION BY SERVER ASYMMETRIC KEY ekm_login_key ;
GO
```

Where,

- `employee` refers to the database name.
- `ekm_login_key` refers to the master key alias on the MSSQL database.

## 10.0 ENABLING TDE ON DATABASE

Run the following commands to alter the database to enable Transparent Data Encryption (TDE):

```
ALTER DATABASE employee
SET ENCRYPTION ON ;
GO
```

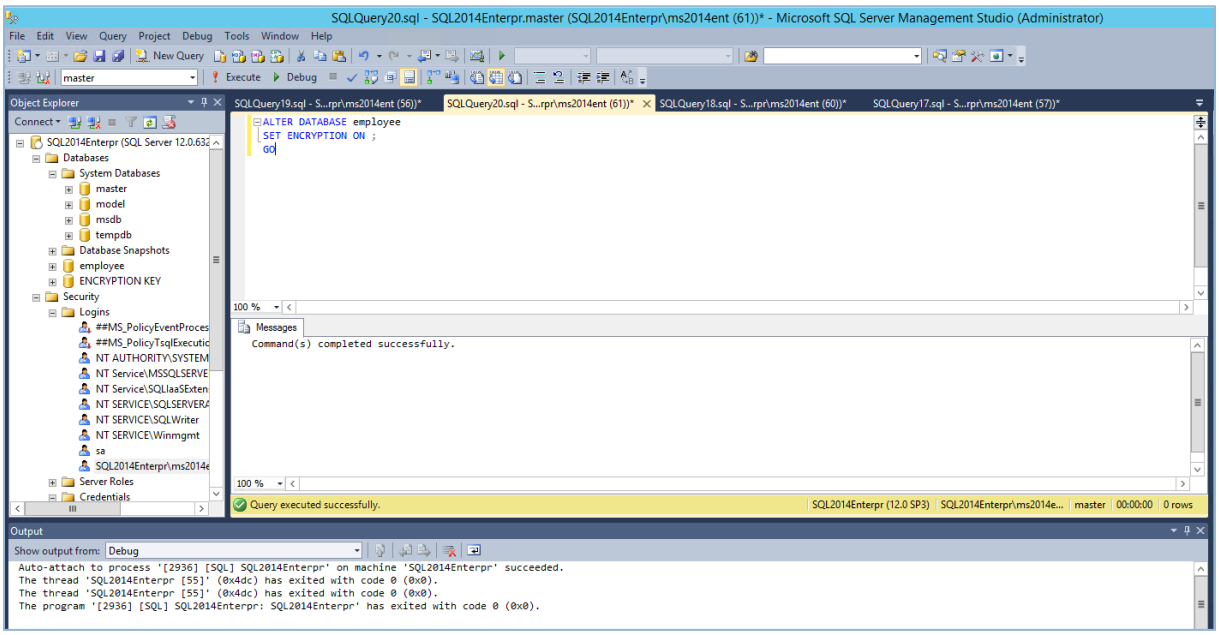


**FIGURE 11: ENABLE TDE**

## 11.0 MONITORING TDE PROGRESS

SQL Server keeps track of the encryption progress, and we can pull that information by querying `sys.dm_database_encryption_keys`. Particularly '`Percent_Complete`' and '`encryption_state`' are the two columns that are required to understand the progress of TDE. '`encryption_state`' column returns an integer value (0-6) which indicates the encryption status of the database and '`percent_complete`' column tells us the percent completed of the DB encryption state change.

| Encryption_state (int) | Description |
|---|---|
| **0** | No database encryption key present, no encryption |
| **1** | Unencrypted |
| **2** | Encryption in progress |
| **3** | Encrypted |
| **4** | Key change in progress |
| **5** | Decryption in progress |
| **6** | Protection changes in progress (The certificate or asymmetric key that is encrypting the database encryption key is being changed). |

The following T-SQL statement can be used to monitor TDE progress or status:

```
SELECT DB_NAME(database_id) AS DatabaseName, encryption_state,
encryption_state_desc =
CASE encryption_state
         WHEN '0'  THEN  'No database encryption key present, no enc
ryption'
         WHEN '1'  THEN  'Unencrypted'
         WHEN '2'  THEN  'Encryption in progress'
         WHEN '3'  THEN  'Encrypted'
         WHEN '4'  THEN  'Key change in progress'
         WHEN '5'  THEN  'Decryption in progress'
         WHEN '6'  THEN  'Protection change in progress (The certifi
cate or asymmetric key that is encrypting the database encryption ke
y is being changed.)'
         ELSE 'No Status'
         END,
percent_complete,encryptor_thumbprint, encryptor_type  FROM sys.dm_d
atabase_encryption_keys
```

The output of this query comes handy to manage TDE.

# 12.0 DOCUMENT INFORMATION

## 12.1 DOCUMENT LOCATION

The latest published version of this document is located at the URL:

https://support.fortanix.com/hc/en-us/articles/12716572654228-Data-Security-Manager-with-Microsoft-SQL-Server-TDE-Guide-Standalone-Server

## 12.2 DOCUMENT UPDATES

This document will typically be updated on a periodic review and update cycle.

For any urgent document updates, please send an email to: support@fortanix.com