Fortanix®

Integration Guide

USING DATA SECURITY MANAGER WITH MSSQL SERVER TDE – BACKUP AND RESTORE

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 describes the step-by-step procedure to backup and restore the Microsoft SQL server Transparent Data Encryption (TDE) enabled database, which is protected by Fortanix Data Security Manager (DSM).

To perform the restoration, the target database MSSQL server must point to the same asymmetric key, which was previously created on the source database MSSQL server.

When transparent data encryption is enabled in the database, the database backup files are encrypted as well. The following error appears on the screen when the user tries to restore a TDE enabled database backup to a different server:

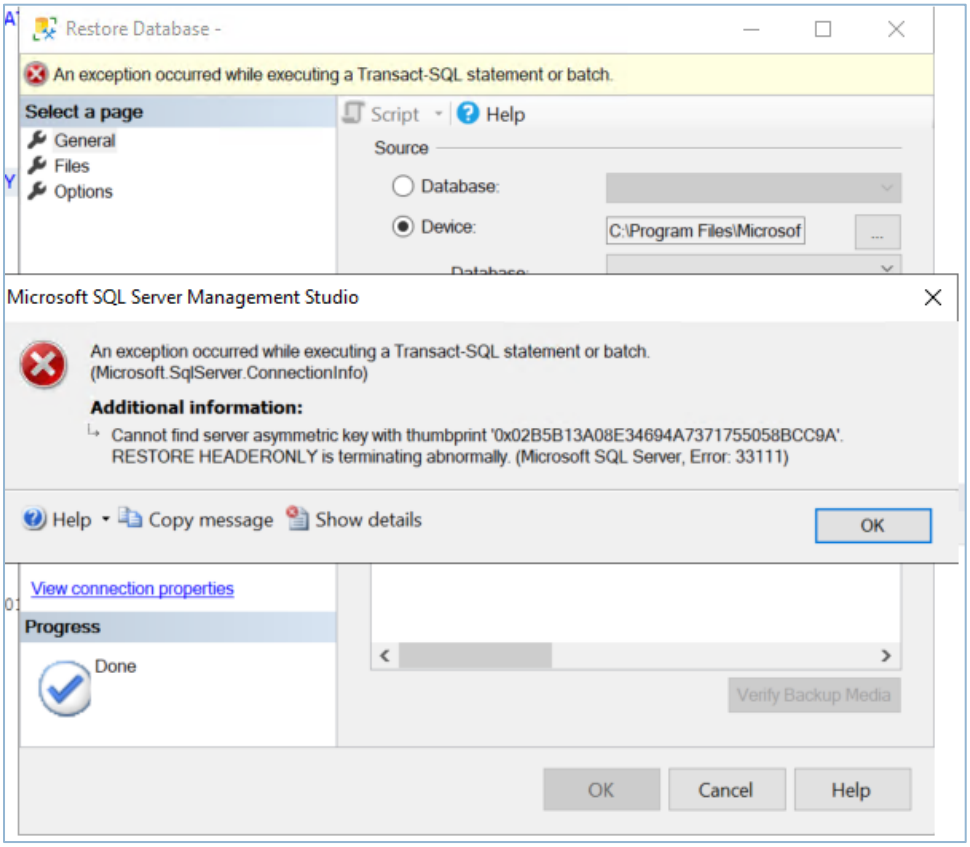

FIGURE 1: ERROR DIALOG BOX

2.0 BACKING UP THE DATA FROM SOURCE

This section lists the steps for taking a backup of your database from the source server. This backup contains the data in encrypted format, which also contains the Data Encryption Key (DEK) protected by the Fortanix master key.

In the given example, we will use the database name as **`employee`** and we are backing it up from the Object Explorer or T-SQL command.

1. Right-click the desired database (**`Company`**).
2. Select **Tasks** and click the **Back Up** option from the context menu.

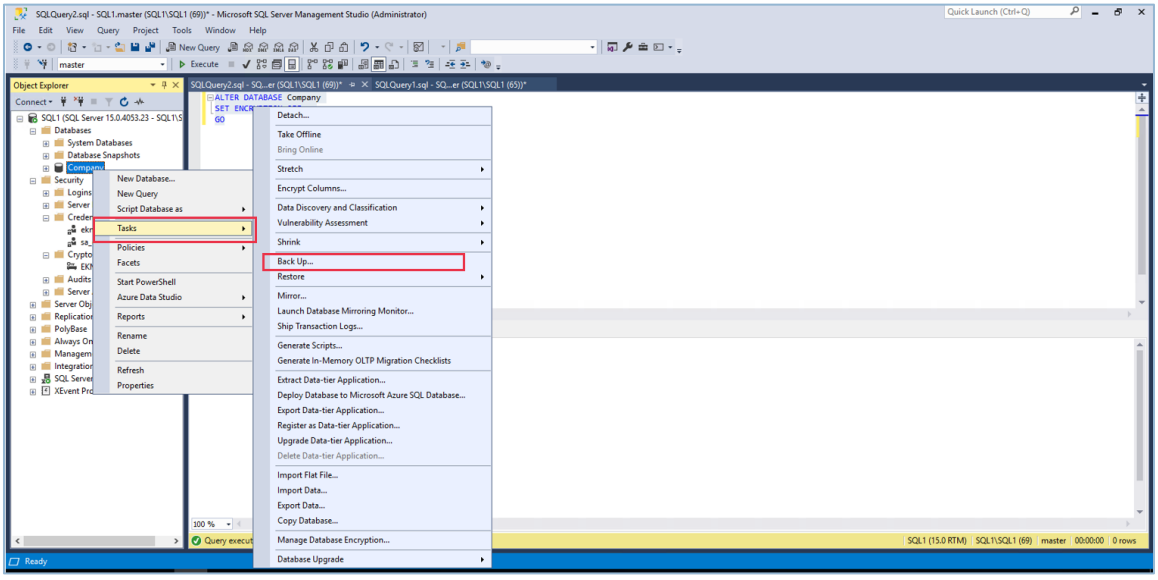

FIGURE 2: TAKE BACKUP OF SOURCE SERVER

a) Select the backup path.

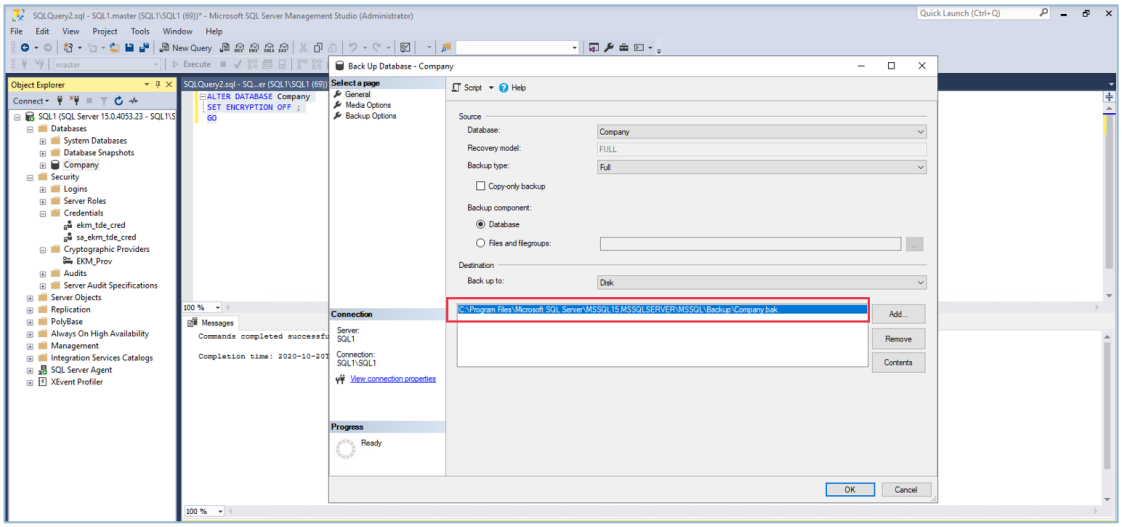

FIGURE 3: SELECT THE BACKUP PATH

b) Backup completed successfully.

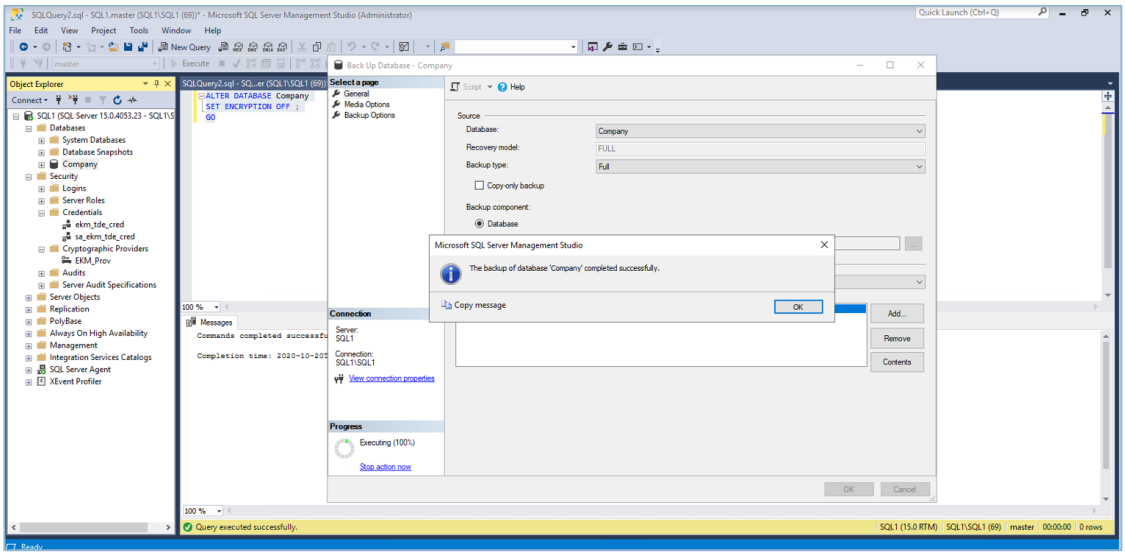

FIGURE 4: BACKUP COMPLETED

3. Move the backup database to the target server.
4. Log in to the secondary target server.

3.0 CONFIGURING TDE ON TARGET SERVER

3.1 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
```

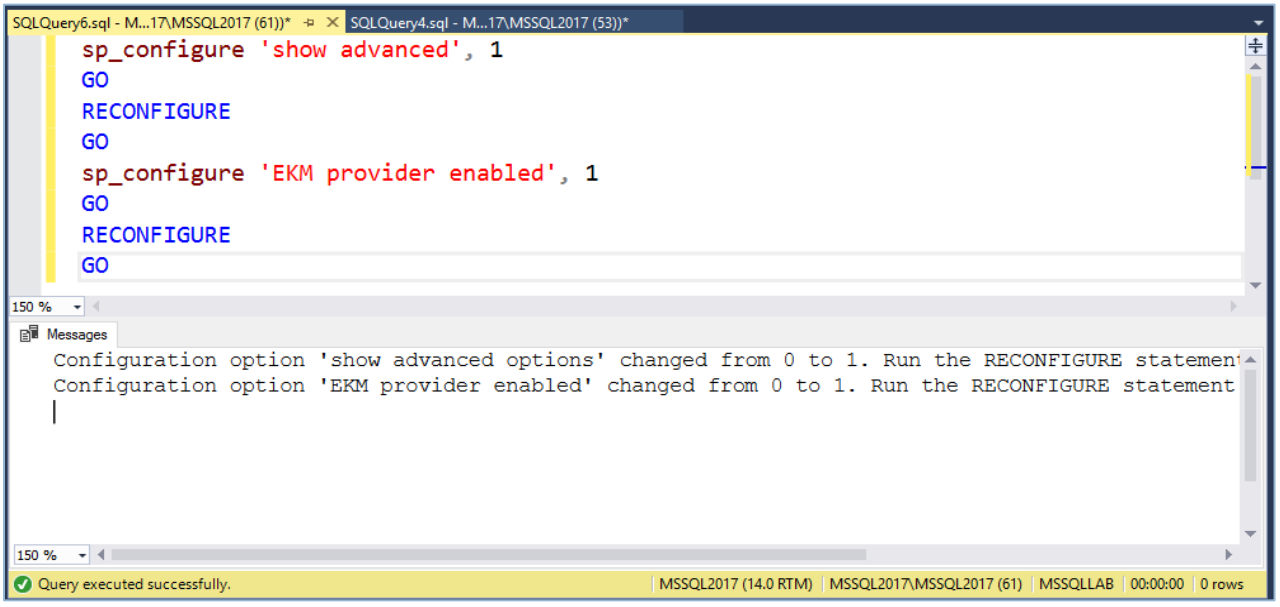

FIGURE 5: RUN COMMANDS FOR ERROR SCENARIO

3.2 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.

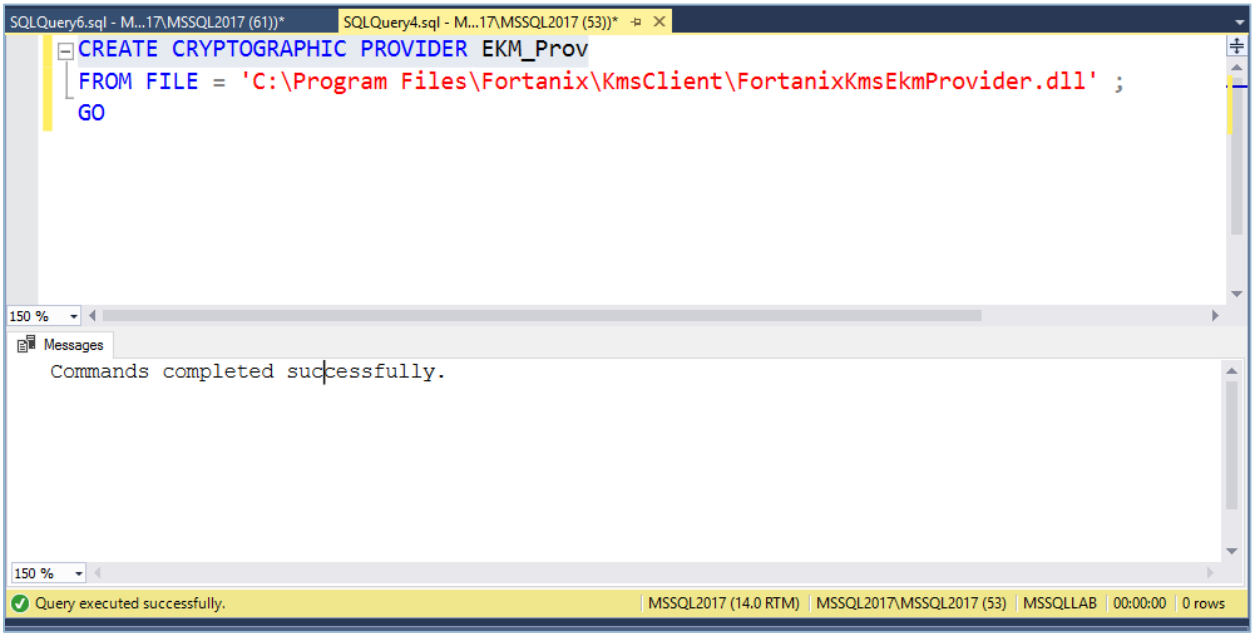

FIGURE 6: CREATE CRYPTOGRAPHIC PROVIDER

3.3 CREATING CREDENTIALS

This section describes the steps to create the credentials to generate the master key on the Fortanix DSM using the SQL admin.

The SQL admin 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.

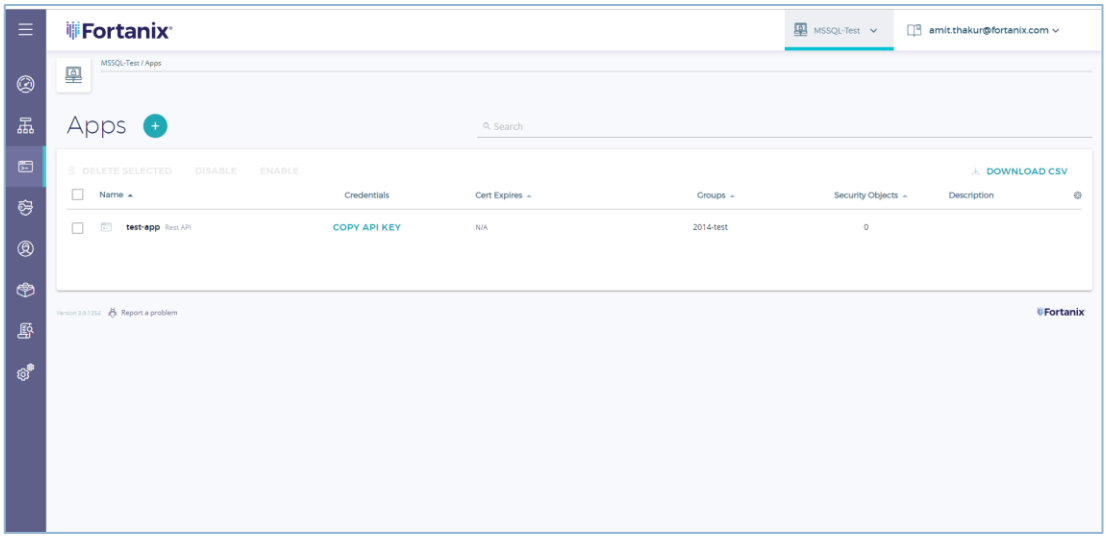

FIGURE 7: COPY API KEY

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
```

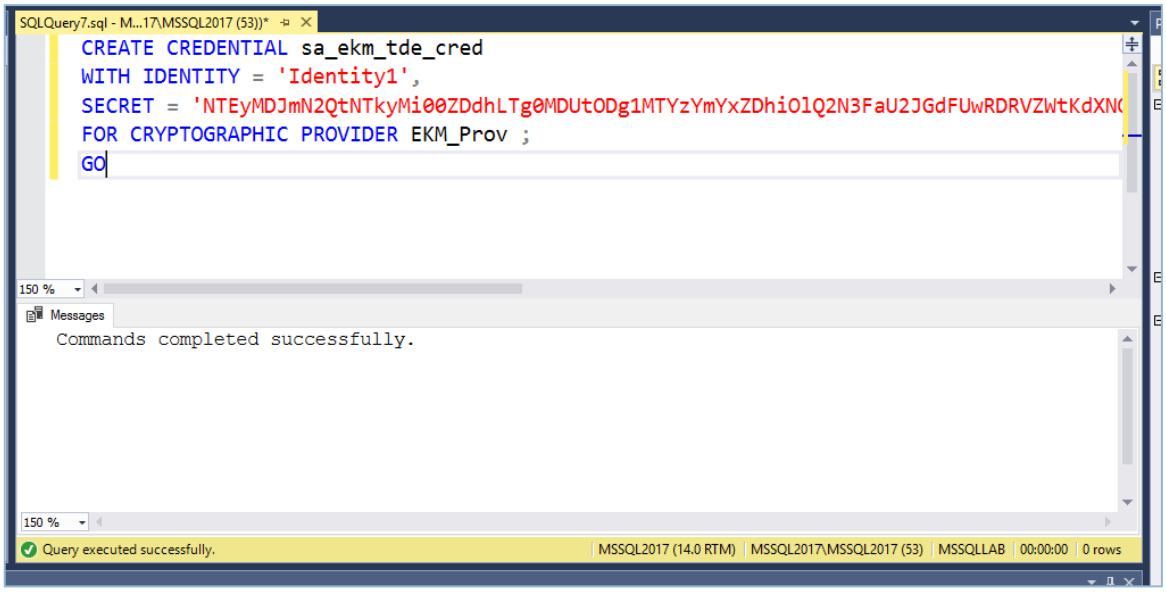

FIGURE 8: CREATE CREDENTIAL

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

```
ALTER LOGIN EC2AMAZ-1RDPAEU\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
```

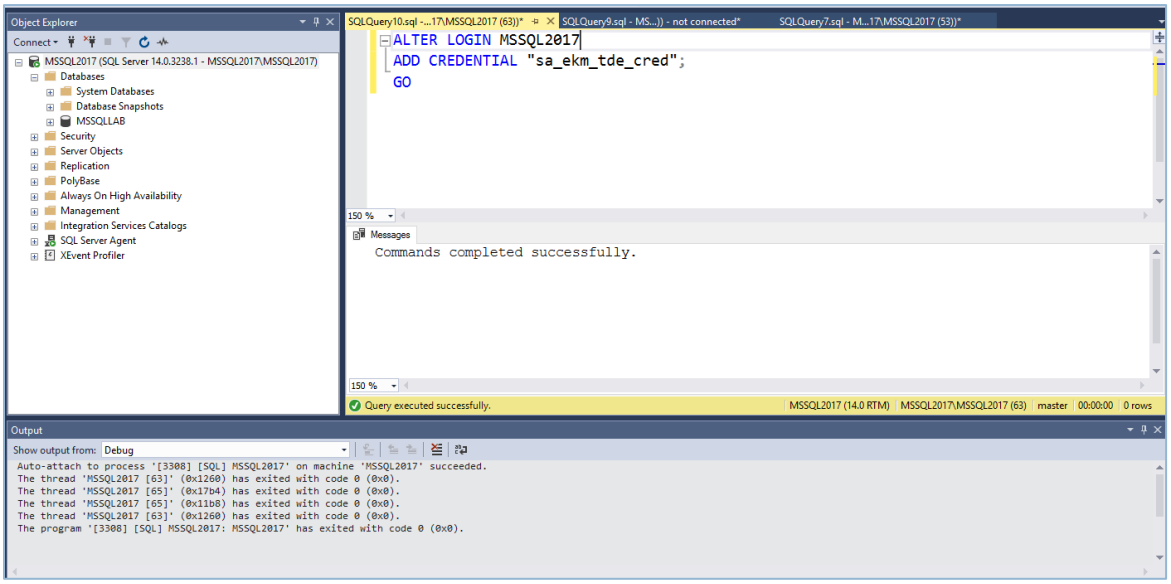

FIGURE 9: COMMAND FOR NO DOMAIN

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:

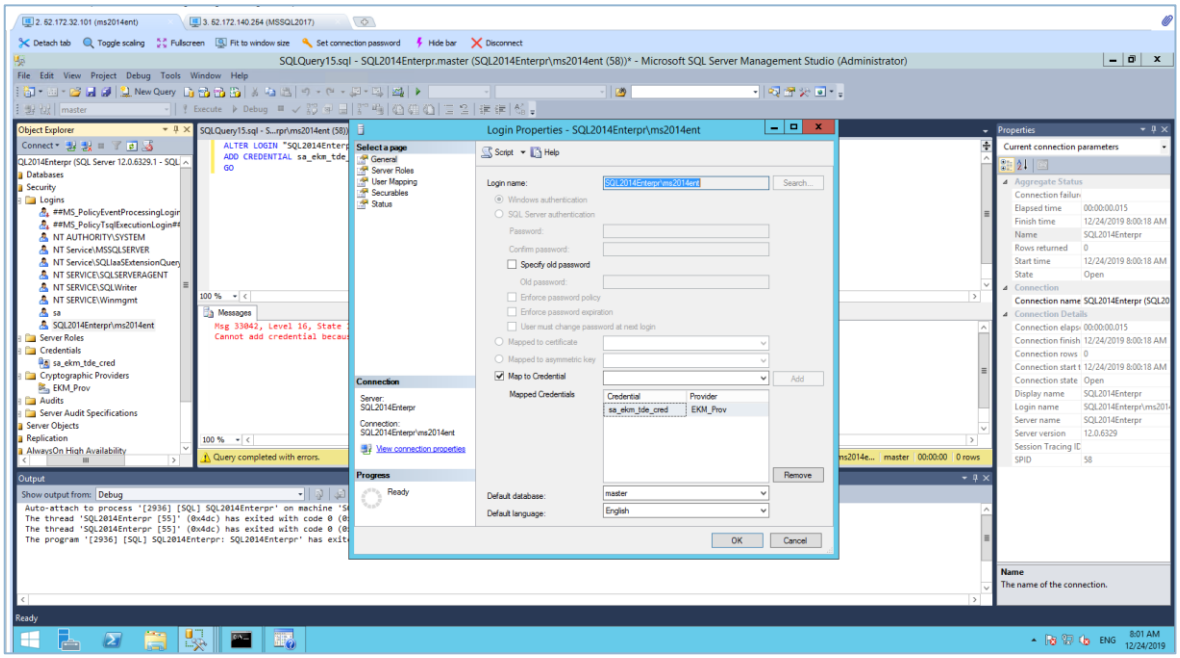

FIGURE 10: MAP CREDENTIALS

3.4 CREATING ASYMMETRIC KEY

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 key from the existing key in the Fortanix DSM.

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

```
USE master
CREATE ASYMMETRIC KEY ekm_login_key FROM PROVIDER EKM_Prov
WITH PROVIDER_KEY_NAME='SQL_Server_Key',
CREATION_DISPOSITION = OPEN_EXISTING;
GO
```

NOTE:

- `ekm_login_key` is the key name on SQL server created on the source server.
- `SQL_Server_Key` is the key name on Fortanix DSM created on the source server.

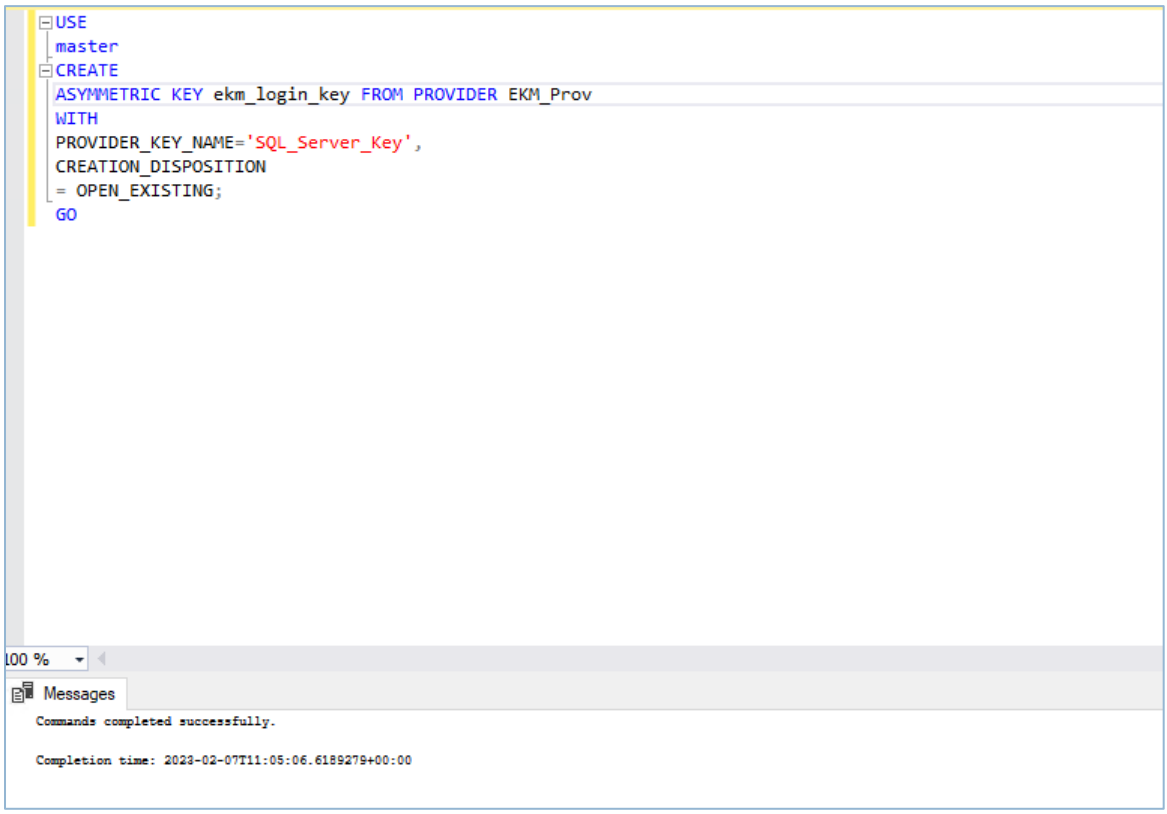

FIGURE 11: CREATING ASYMMETRIC KEY

3.5 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.

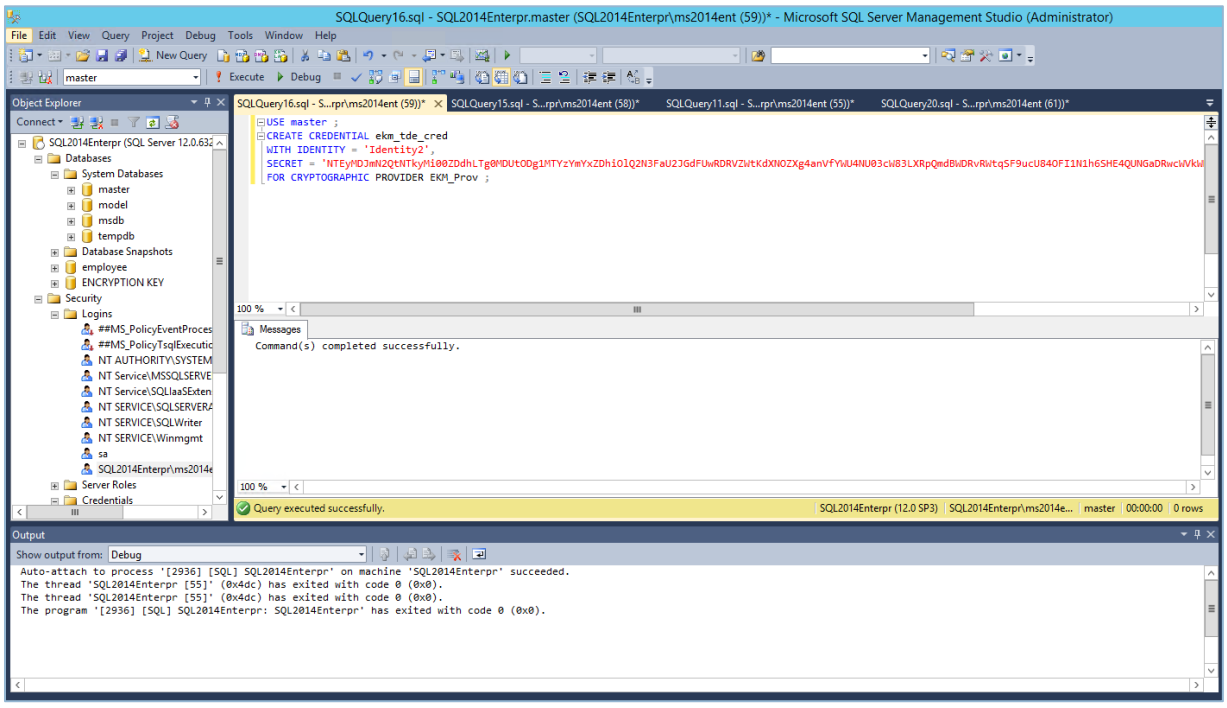

FIGURE 12: CREATE CREDENTIAL FOR DATABASE ENGINE

3.6 CREATING LOGIN (DB ENGINE)

Run the following commands to add a login used by TDE and add the new credential 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 the MSSQL database. This key is already created in "*Section 3.4- 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 3.3- Creating Credentials*".

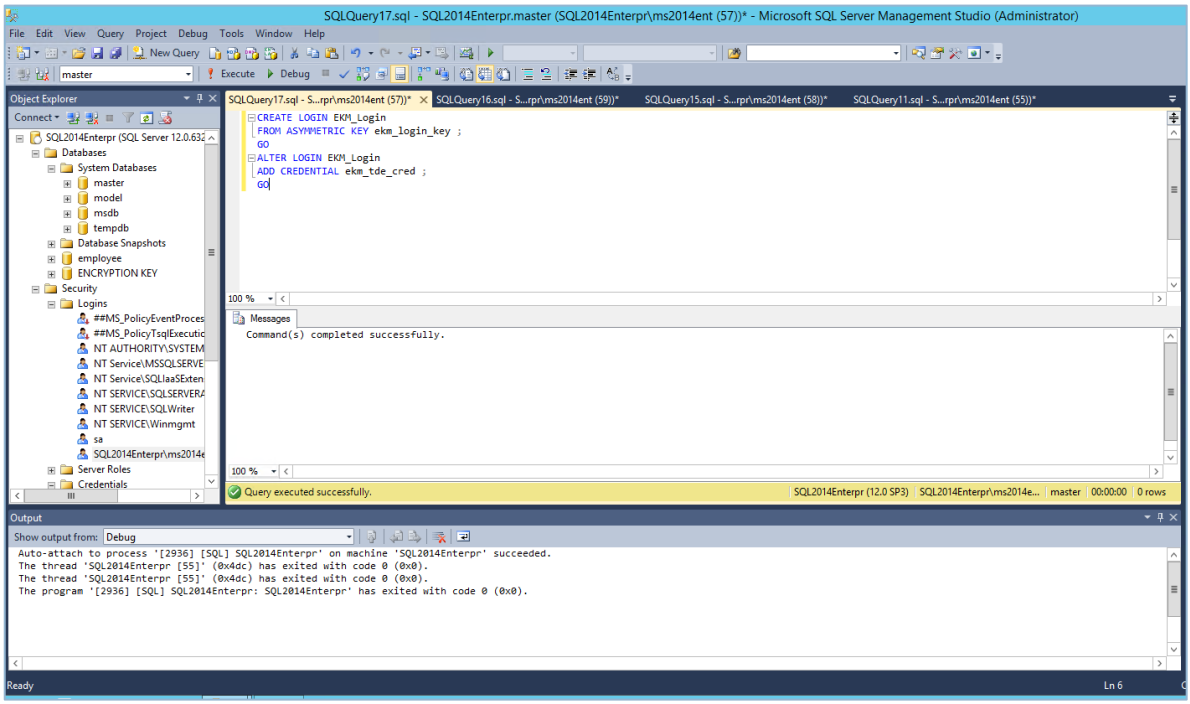

FIGURE 13: ADD NEW CREDENTIAL TO LOGIN

4.0 RESTORING THE ENCRYPTED DATABASE

This section describes the steps for restoring the encrypted backup on the target server. When the backup is encrypted with TDE at the time of restoration, the database tries to unlock the DEK using MEK. The SQL server starts the restoration process only if the respective master key is available on the database.

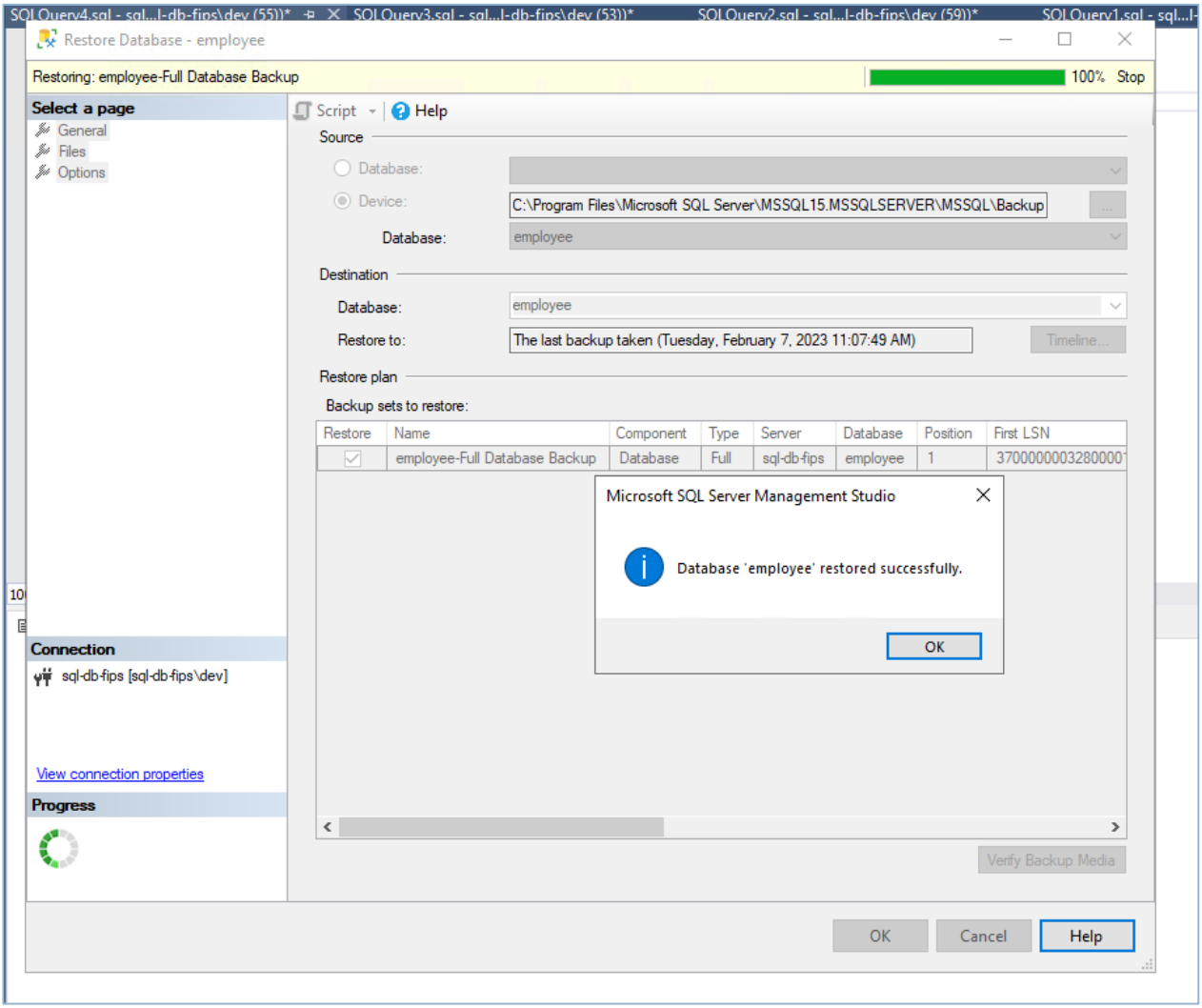

FIGURE 14: RESTORING DATABASE

5.0 DOCUMENT INFORMATION

5.1 DOCUMENT LOCATION

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

https://support.fortanix.com/hc/en-us/articles/12782302548500-Data-Security-Manager-with-Microsoft-SQL-Server-TDE-Guide-Backup-Restore

5.2 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