Administration Guide

USING DATA SECURITY MANAGER WITH MSSQL SERVER – ALWAYS ENCRYPTED

VERSION 1.0

Fortanix, Inc. | 3910 Freedom Circle | Suit 104 | Santa Clara, CA 95052 | United States of America | +1 (628) 400-2043 |

✉info@fortanix.com | 🌐 www.fortanix.com

TABLE OF CONTENTS

1.0 INTRODUCTION

This document describes the steps to integrate the Fortanix Data Security Manager (DSM) with Microsoft SQL Always Encrypted Server.

For more information, refer to the https://learn.microsoft.com/en-us/sql/relational-databases/security/encryption/always-encrypted-database-engine?view=sql-server-ver15

1.1 PREREQUISITES

Ensure the following:

- The port 443 must be accessible from the SQL target machine to Fortanix DSM.

PROTOCOL	INBOUND/ OUTBOUND	PORT NUMBER	LOAD BALANCER USE (YES/NO)	PURPOSE
TCP	Outbound	443	No	HTTPS – Used for calling REST API. MS-SQL server will access the cluster/SaaS URL on this port. Each individual node will also need this port open.

- The SQL Server must be installed and configured on the target machine.
- Administrators are privileged to access SQL Server Management Studio from the target machine.

2.0 DEFINITIONS

- **Fortanix Data Security Manager** -

  Fortanix DSM is the cloud solution secured with Intel® SGX. With Fortanix DSM, you can securely generate, store, and use cryptographic keys and certificates, as well as secrets, such as passwords, API keys, tokens, or any blob of data.

- **Accounts** -

  A Fortanix DSM account is the top-level container for security objects managed by the Fortanix DSM. An account is generally associated with an organization, rather than an individual. Security objects, groups, and applications belong to exactly one account. Different accounts are fully isolated from each other. *See support for more information.*

- **Users** -

  Users are associated with an email address. A user can be a member of one or more accounts. Depending on permissions, users can:

  - Perform management operations like adding or modifying users or groups
  - Create security objects
  - Change properties of security objects
  - Review logs of Fortanix DSM activity

  ⚠ **Users cannot perform cryptographic operations. Only applications can perform cryptographic operations.**

- **Groups** -

  A group is a collection of security objects created by and accessible by users and applications which belong to the group. The user who creates a group automatically gets assigned the role of the group administrator. You can add more users to the group in the role of administrators or auditors. You can also add applications to the group to enable the applications to create and use security objects in that group. *See support for more information.*

Access policies are set at the group level, so all security objects in a group share the same access policy. Any number of users and/or applications can be assigned to a group. *Some examples of usage of groups are given in the Authorization section*.

Quorum policies can also be set at group level. A Quorum policy mandates that all security sensitive operations in that group would require a quorum approval. Such operations include using a key for cryptographic operations or deleting or updating a group. *See Quorum Policy for more information.*

- **Applications** -

  An application (app) can use Fortanix DSM to generate, store, and use security objects, such as cryptographic keys, certificates, or an arbitrary secret. Applications can authenticate to Fortanix DSM using an API key (a secret token) or a TLS client certificate. An application can interact with Fortanix DSM using the REST APIs or using the PKCS#11, JCE, or CNG providers. *See support for more information.*

- **Fortanix Data Security Manager Security Objects** –

  A security object is any datum stored in Fortanix DSM (for example a key, a certificate, a password, or other security objects). Each security object is assigned to exactly one group. users and applications assigned to the group have permission to see the security object and to perform operations on it. *See support for more information.*

3.0 SQL SERVER ALWAYS ON SETUP

This integration uses the following setup to demonstrate the Always Encrypted with Fortanix DSM:

1. A Windows Server machine, as a Domain Controller.
2. A Windows Server machine, with SQL Server and Fortanix CNG 32-bit client installed.
3. A Windows 10 Professional machine with Fortanix CNG 64-bit client install to the test column decryption.

3.1 SUPPORTED VERSIONS

This SQL Always Encrypted integration is tested on the following versions:

- Microsoft SQL Server 2019
- Microsoft Server Management Studio 19 (v19.1)
- Fortanix DSM 4.19
- Fortanix CNG Client 4.19 (32-bit)

4.0 INTEGRATION STEPS

4.1 CONFIGURING FORTANIX DSM

Perform the following tasks to configure Fortanix DSM:

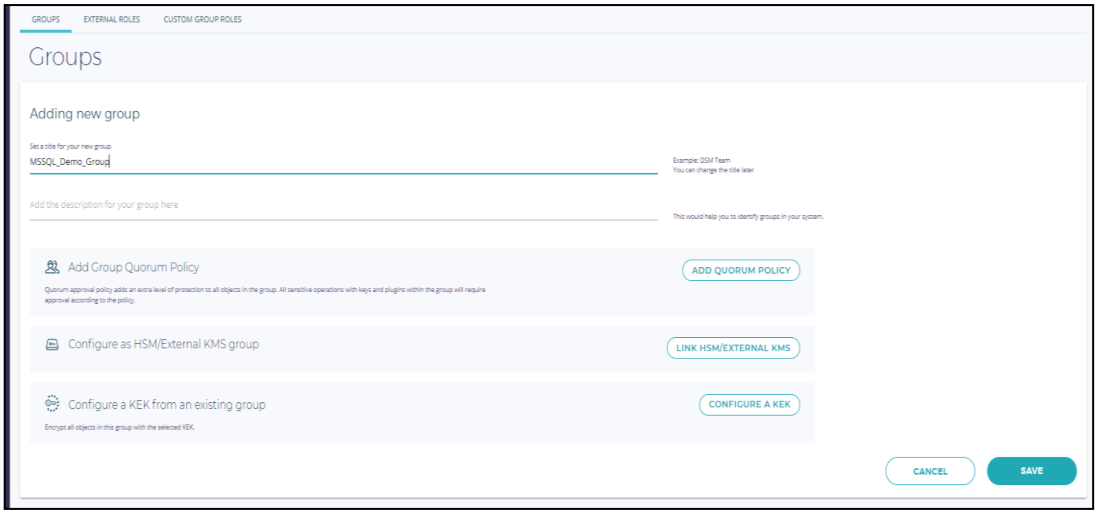
4.1.1 CREATING GROUPS

A Fortanix DSM group is a collection of security objects created by and accessible by users and applications that belong to the group. The user who creates a group automatically gets assigned the role of group administrator. You can add more users to the group in the role of administrators or auditors. You can also add applications to the group to enable the applications to create and use security objects in that group.

To add a group, specify the following:

- The title of the group (required).
- A short description for the group (not mandatory).
- Users in your account as members.

- Applications in your account to add to the group so that they can use the security objects in the group. *Refer to "Section 4.1.2- Creating Apps" to know the steps for creating the app.*
- Add a quorum approval policy (optional). A group administrator may enable a quorum approval policy for a group, which mandates that all security-sensitive operations in that group would require a quorum approval.



Groups

Adding new group

Set a title for your new group
MSSQL_Demo_Group

Example: DSM Team
You can change the title later

Add the description for your group here

This would help you to identify groups in your system.

Add Group Quorum Policy

Quorum approval policy adds an extra level of protection to all objects in the group. All sensitive operations with keys and plugins within the group will require approval according to the policy.

Configure as HSM/External KMS group

Configure a KEK from an existing group

Encrypt all objects in this group with the selected KEK.

FIGURE 1: ADDING NEW GROUP

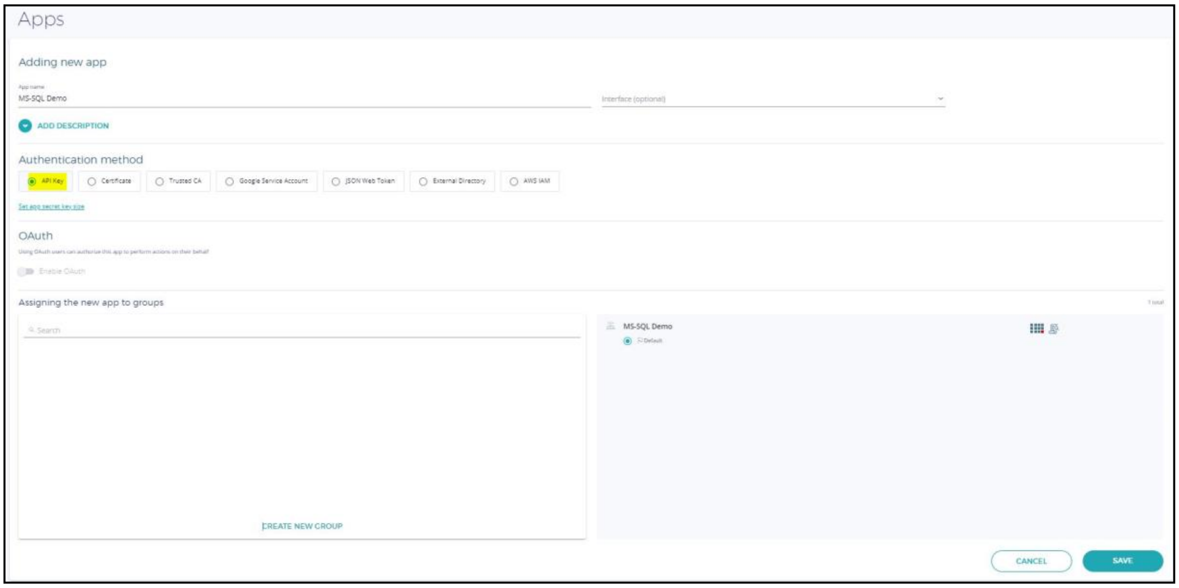
4.1.2 CREATING APPS

An application can use Fortanix DSM to generate, store, and use security objects, such as cryptographic keys, certificates, or an arbitrary secret. Examples of applications include web servers, PKI servers, key vaults, and so on. An application can interact with Fortanix DSM using the REST APIs or the PKCS#11, JCE, or CNG providers.

To add an application, specify the following:

- Name of the application (required).
- Type of the application. Select the value as **interface**.
- A short description of the application.
- Select the authentication method as **API key**.
- Assign the app to the MSSQL group as created in the *"Section 4.1.3- Creating Group"*.

After the application has been added, you can use the API key to authenticate the CNG client to Fortanix DSM and start making calls to do cryptographic operations.



Apps

Adding new app

App name: MS-SQL Demo

Interface (optional)

Authentication method: ● API Key ○ Certificate ○ Trusted CA ○ Google Service Account ○ JSON Web Token ○ External Directory ○ AWS IAM

OAuth

Using OAuth users can authorize this app to perform actions on their behalf

Enable OAuth

Assigning the new app to groups

MS-SQL Demo ● Default

FIGURE 2: ADDING NEW APP

4.2 FORTANIX CNG CLIENT

The Fortanix CNG Provider must be installed on every target machine. *Refer to https://support.fortanix.com/hc/en-us/articles/360018084132-CNG-EKM (32-bit) to download the CNG Provider.*

`FortanixKmsClient.msi` installs the Fortanix CNG Provider, as well as an EKM provider and the PKCS#11 library. Next, to configure the CNG client Fortanix CNG Provider communicates with Fortanix DSM for crypto operations.

4.2.1 INSTALLATION

Perform the following steps to complete the installation on your machine:

1. On the **Fortanix KMS Client Setup** dialog box, click the **Next** button.
2. Select the checkbox for **I accept the terms in the License Agreement** and click the **Next** Button.
3. Enter the location for installing the **Fortanix KMS Client** as **C:\Program Files\Fortanix\KMS Client**.
4. Click the **Install** button to install the Fortanix KMS client.
5. After the installation is done, click the **Finish** button.

4.2.2 CONFIGURING CNG CLIENT

The Fortanix KMS Server URL and proxy information are configured in the Windows registry for the current user.

1. Run the following command to navigate to `FortanixKmsClientConfig.exe` file:

```
cd C:\Program Files (x86)\Fortanix\KmsClient\
```

2. The user key store uses the current user configuration.

   For example, run the following command to configure the Fortanix KMS Server URL for the current user:

```
 FortanixKmsClientConfig.exe user --api-endpoint {KMS_URL}
```

Where,

- `KMS_URL` refers to the Fortanix DSM URL. On-premises customers use KMS URL and SaaS. The customers can use the following URLs based on the region.
  - Europe: https://eu.smartkey.io/
  - APAC: https://apac.smartkey.io/

- United States of America: https://amer.smartkey.io/

  For example,

```
FortanixKmsClientConfig.exe user --api-endpoint https://<fortanix_dsm_url>
```

3. To configure proxy information, add `--proxy http://proxy.com` or `--proxy none` to unconfigure proxy.
4. Run the following command to configure the API key as created in *Section 4.1.2: Creating Apps* for the user keystore:

```
FortanixKmsClientConfig.exe user --api-key <api_key>
```

4.3 SQL ALWAYS ENCRYPTED

4.3.1 CREATE SAMPLE DATABASE

For testing the integration, a sample database is created. However, you can use the existing database table to encrypt the required column.

1. Open the SQL Server Management Studio and connect to the database.
2. Run the following commands to create database `employee`:

```
CREATE DATABASE employee
```

3. 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
```

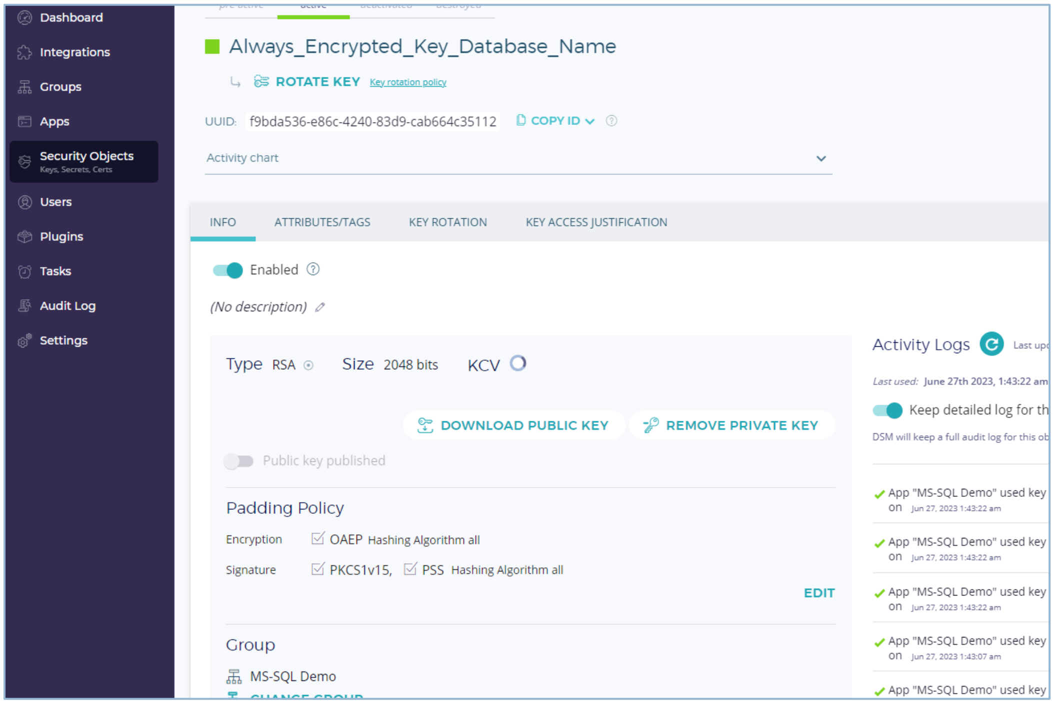
4. Run the following commands to insert data into the table:

```
insert into employee values ('Adam','Parker','1','5000')
insert into employee values ('John','Doe','2','4500')
insert into employee values ('Peter','Williams','2','4500')
GO
```

4.3.2 CREATE COLUMN MASTER KEY

The column master keys are key-protecting keys that are used to encrypt the column encryption keys. The column master keys will be stored on the Fortanix DSM. The database only contains metadata about the column master keys such as type of key store and location. The column master key metadata is stored in the `sys.column_master_keys` (Transact-SQL) catalog view.

1. Log in to the Fortanix DSM user interface and create an RSA key with the following permissions, such as **Always_Encrypted_Key_Database_Name**.
    - Encrypt
    - Decrypt
    - Sign
    - Verify



Always_Encrypted_Key_Database_Name

UUID: f9bda536-e86c-4240-83d9-cab664c35112

Enabled

(No description)

Type RSA Size 2048 bits KCV

Public key published

Padding Policy

Encryption ☑ OAEP Hashing Algorithm all

Signature ☑ PKCS1v15, ☑ PSS Hashing Algorithm all

Group

MS-SQL Demo

Activity Logs Last upd

Last used: June 27th 2023, 1:43:22 am

Keep detailed log for th

DSM will keep a full audit log for this ob

✓ App "MS-SQL Demo" used key on Jun 27, 2023 1:43:22 am

✓ App "MS-SQL Demo" used key on Jun 27, 2023 1:43:22 am

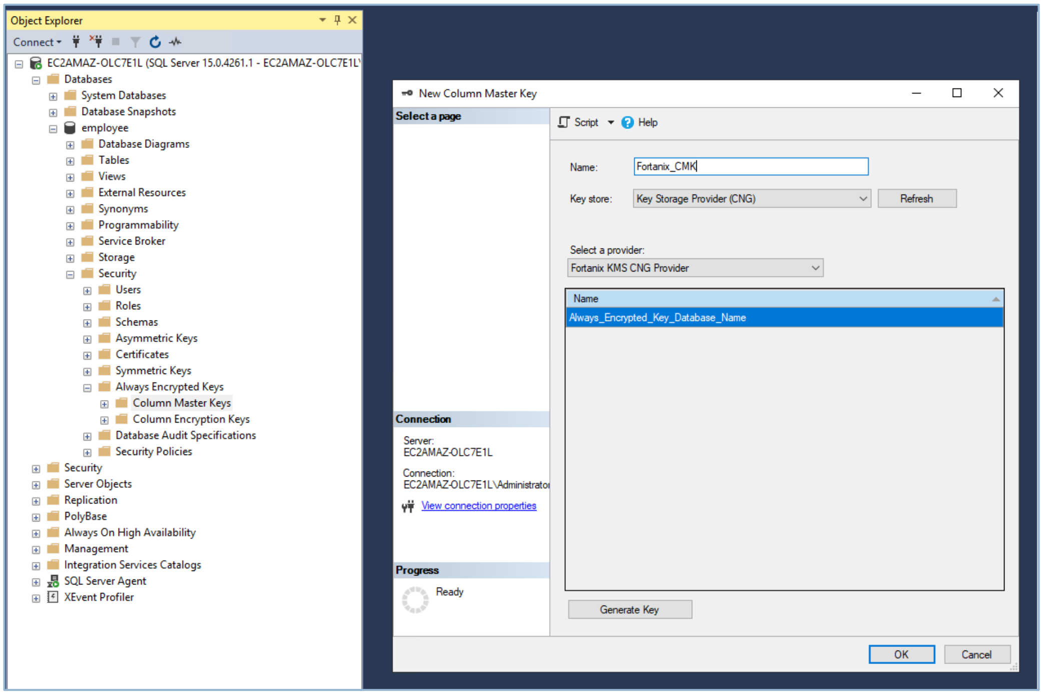
✓ App "MS-SQL Demo" used key on Jun 27, 2023 1:43:22 am

✓ App "MS-SQL Demo" used key on Jun 27, 2023 1:43:07 am

✓ App "MS-SQL Demo" used key

FIGURE 3: CREATE KEY

2. After the key is created, log in to the SQL Server Management Studio.
3. Navigate to the **Databases** → **employee** → **Security** → **Always Encrypted Keys** → **Column Master Keys.** Right click on the folder Column Master Keys and select the **New Column Encryption Key** option to create the Column Master Keys.
4. Enter the following details:
   - **Name:** Enter the required name of the key.
   - **Key Store:** Select the **Key Storage Provider (CNG)** option from the drop down menu.
   - **Select a provider:** Select the **Fortanix KMS CNG Provider** option from the drop down menu.



Name: Fortanix_CMK

Key store: Key Storage Provider (CNG)

Select a provider: Fortanix KMS CNG Provider

Name
Always_Encrypted_Key_Database_Name

Server: EC2AMAZ-OLC7E1L

Connection: EC2AMAZ-OLC7E1L\Administrator

FIGURE 4: COLUMN MASTER KEYS

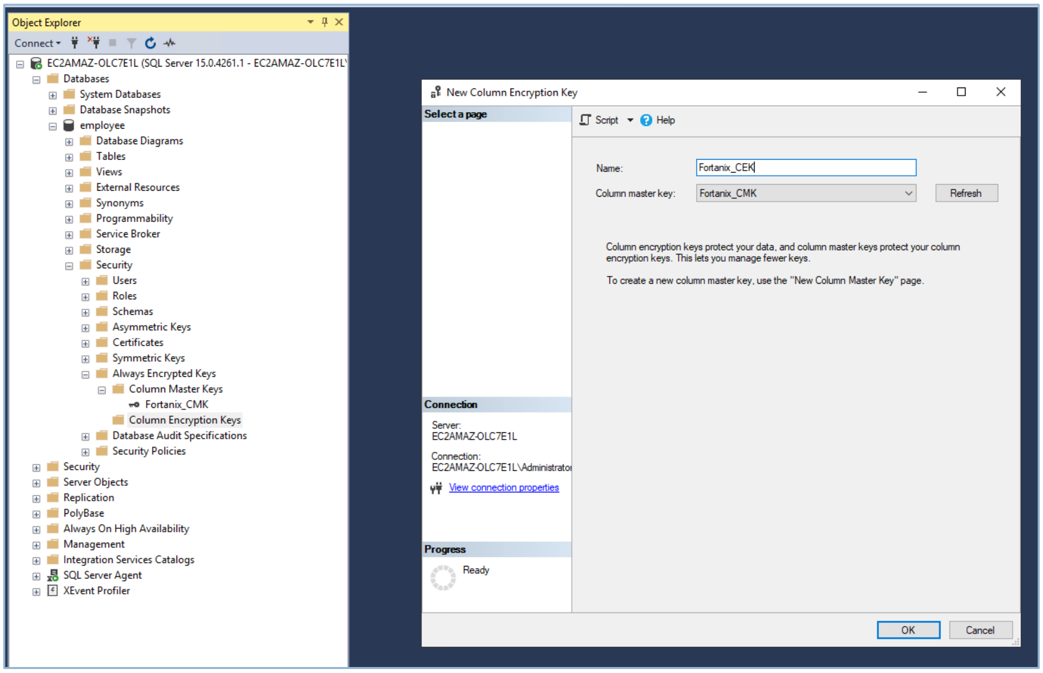
5. Click the **OK** button.

4.3.3 CREATE COLUMN ENCRYPTION KEY

The column encryption keys are content-encryption keys used to encrypt the data in the database columns. You can encrypt one or more columns with the same column encryption key or use multiple column encryption keys depending on your application requirements. The column encryption keys are themselves encrypted, and only the encrypted values of the column encryption keys are stored in the database (as part of the column encryption key metadata). The column encryption key metadata is stored in the `sys.column_encryption_keys` (Transact-SQL) and `sys.column_encryption_key_values` (Transact-SQL) catalog views. The column encryption keys used with the AES-256 algorithm are 256-bit long.

1. Navigate to the **Databases** → **employee** → **Security** → **Always Encrypted Keys** → **Column Encryption Keys** to create the Column Encryption Keys.
2. Enter the following details:
   - **Name:** Enter the name of the column encryption key.

- **Column master key:** Select the same column master key as created in *Section 4.2.2 – Create Column Master Key*. For example, select **Fortanix_CMK** key from the drop down menu.



Name: Fortanix_CEK

Column master key: Fortanix_CMK

Column encryption keys protect your data, and column master keys protect your column encryption keys. This lets you manage fewer keys.

To create a new column master key, use the "New Column Master Key" page.

Server:
EC2AMAZ-OLC7E1L

Connection:
EC2AMAZ-OLC7E1L\Administrator

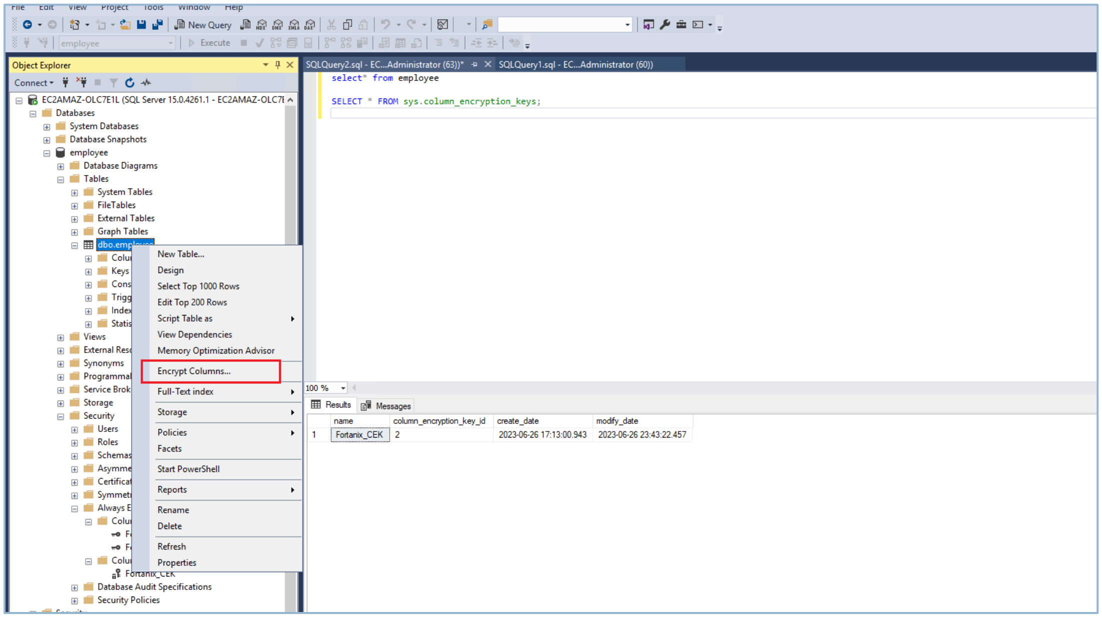
FIGURE 5: COLUMN MASTER ENCRYPTION KEY

3. Click the **OK** button.

4.3.4 ENCRYPT COLUMNS USING ALWAYS ENCRYPTED KEY

Perform the following steps:

1. Navigate to the **Databases** → **employee** → **Tables**. Right click the required table and select the **Encrypt Columns** option to encrypt the columns.

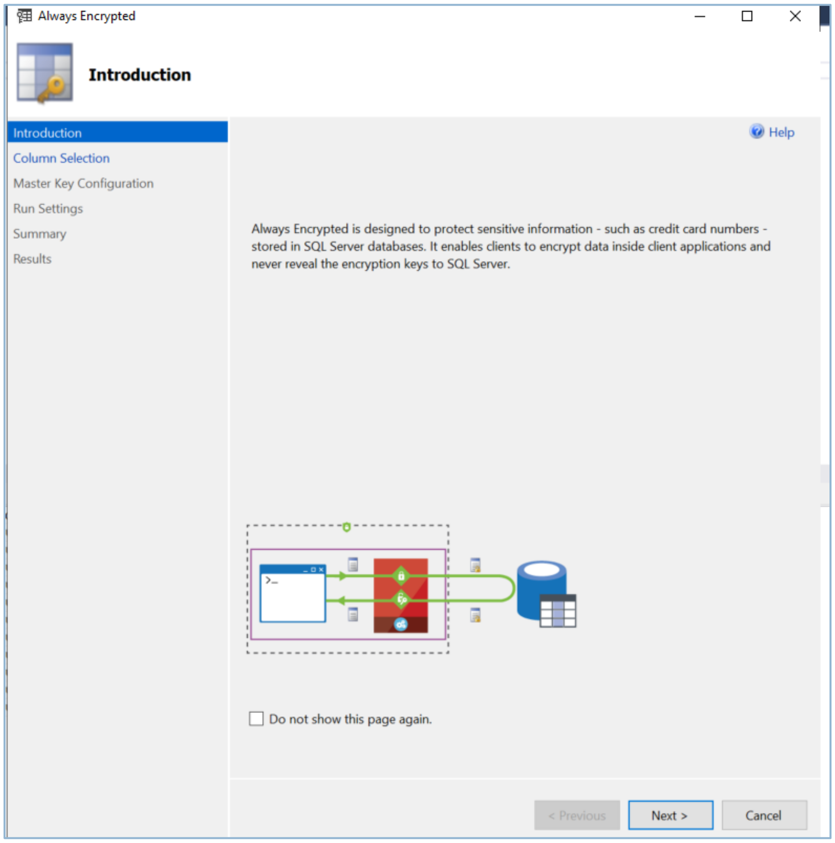


```
select* from employee

SELECT * FROM sys.column_encryption_keys;
```

	name	column_encryption_key_id	create_date	modify_date
1	Fortanix_CEK	2	2023-06-26 17:13:00.943	2023-06-26 23:43:22.457

FIGURE 6: ENCRYPT COLUMN

2. On the **Introduction** screen, click the **Next** button.



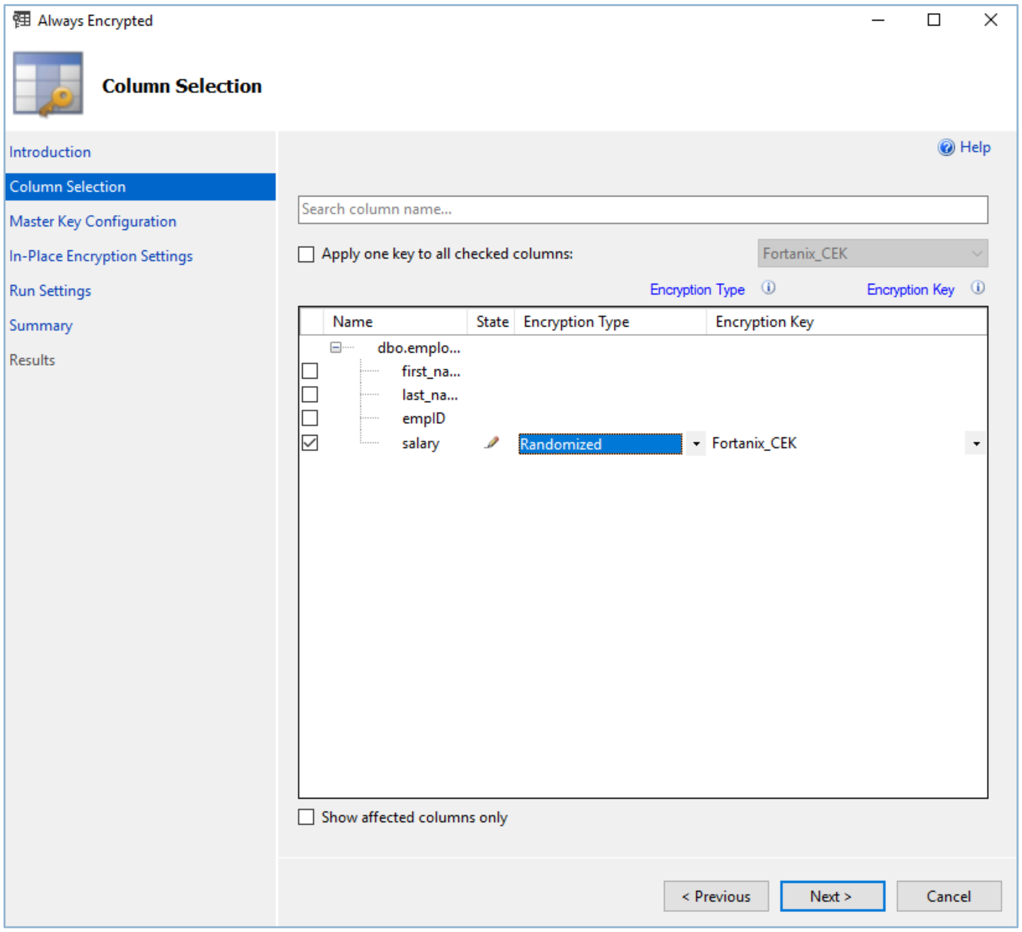
Always Encrypted is designed to protect sensitive information - such as credit card numbers - stored in SQL Server databases. It enables clients to encrypt data inside client applications and never reveal the encryption keys to SQL Server.

☐ Do not show this page again.

FIGURE 7: INTRODUCTION

3. On the **Column Selection** screen, select the following:
   a. **Encryption Type:** Chose the required option from the drop down menu:
      - **Deterministic encryption** always generates the same encrypted value for a given plaintext value.
      - **Randomized encryption** uses a method that encrypts data in a less predictable manner.

   b. **Encryption Key:** Chose the same key name as created in *Section 4.2.3: Create Column Encryption Key*.



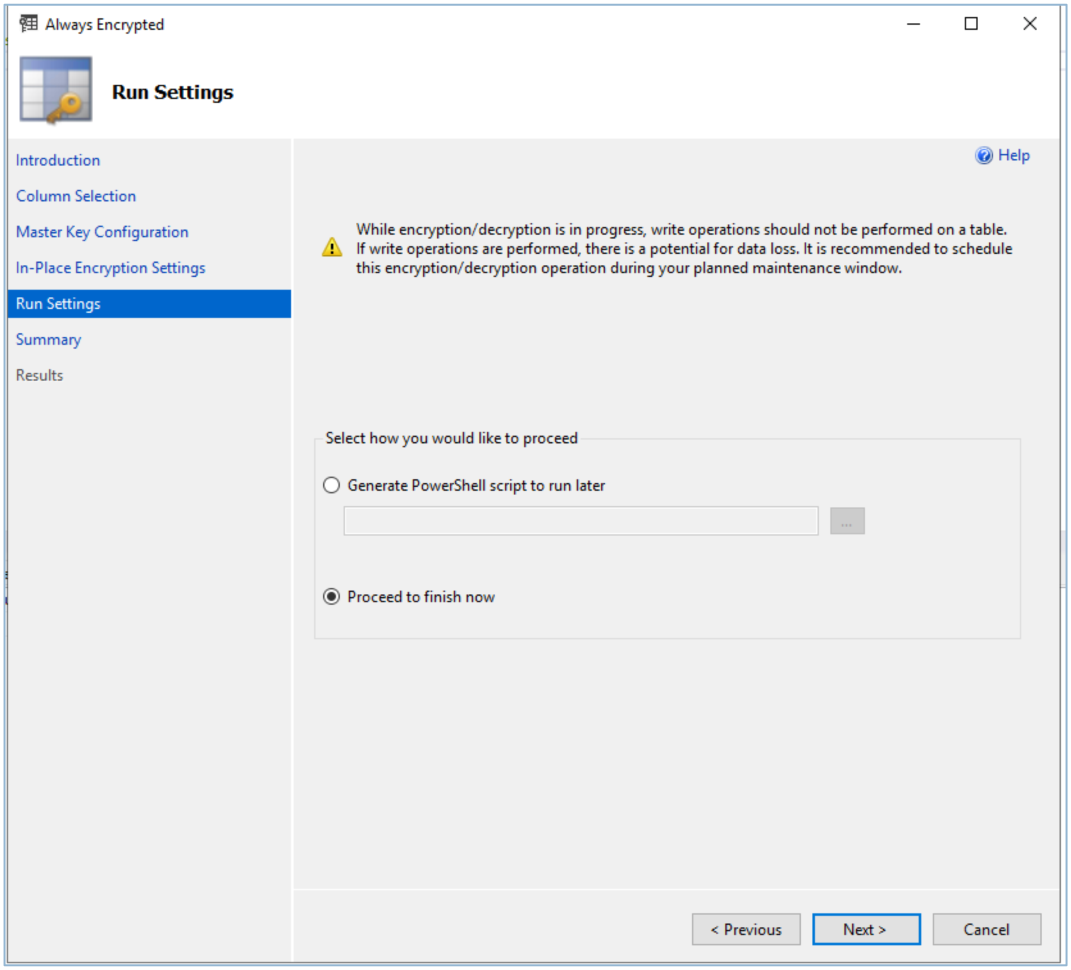
□ Apply one key to all checked columns: Fortanix_CEK

	Name	State	Encryption Type	Encryption Key
	⊟ dbo.emplo...			
□	first_na...			
□	last_na...			
□	empID			
☑	salary		Randomized	Fortanix_CEK

□ Show affected columns only

FIGURE 8: COLUMN SELECTION

4. Click the **Next** button.
5. On the **Run Settings** screen, select the **Proceed to finish now** radio button and click the **Next** button.



While encryption/decryption is in progress, write operations should not be performed on a table. If write operations are performed, there is a potential for data loss. It is recommended to schedule this encryption/decryption operation during your planned maintenance window.

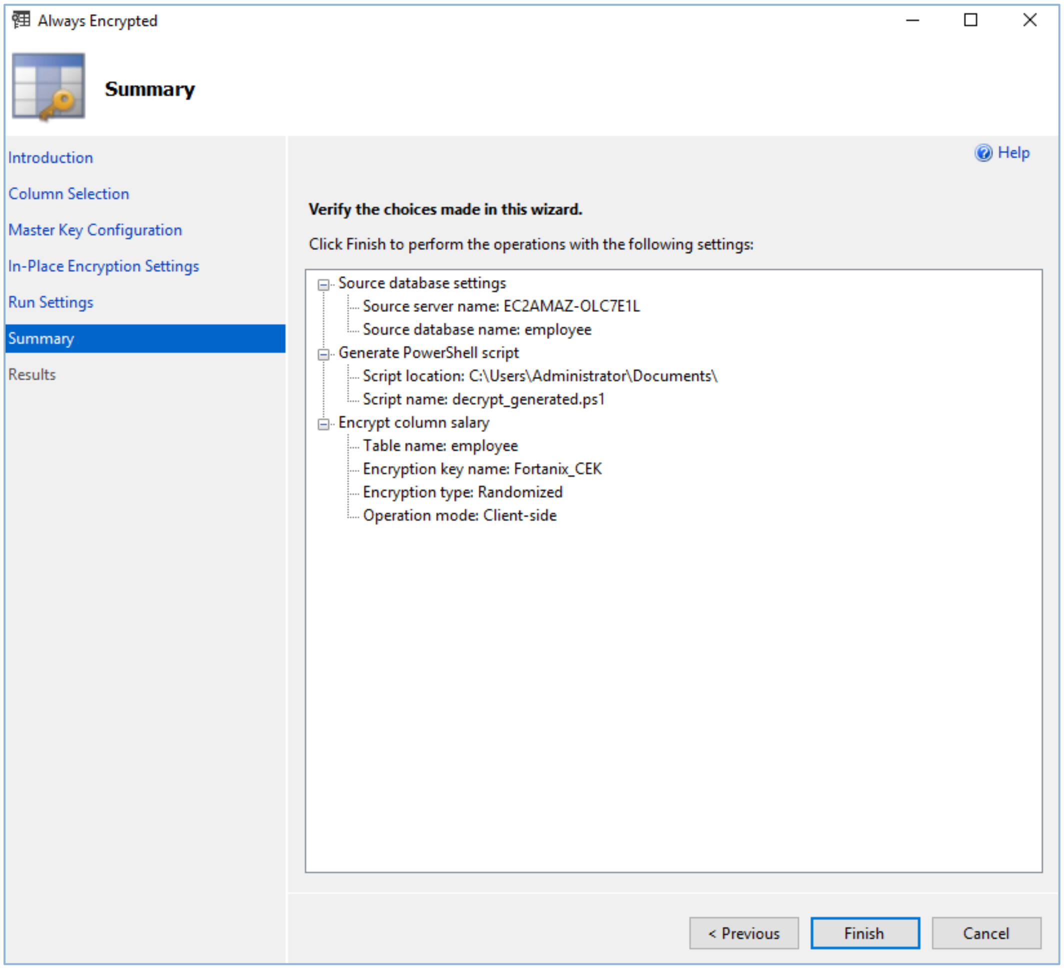
Select how you would like to proceed

○ Generate PowerShell script to run later

● Proceed to finish now

FIGURE 9: RUN SETTINGS

6. On the **Summary** screen, wait until the results are processing.



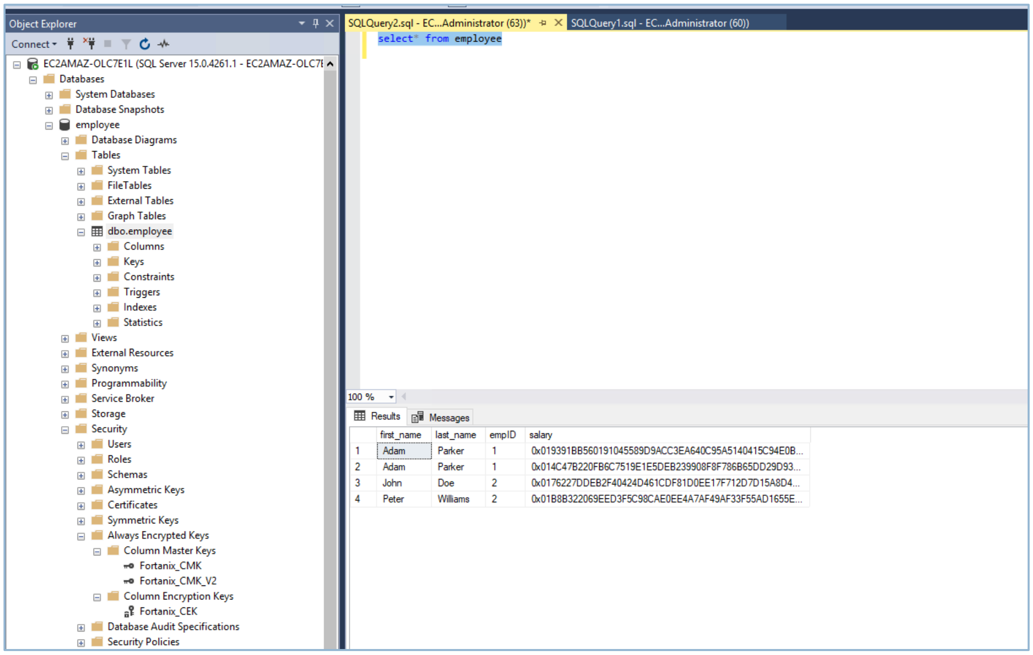
Verify the choices made in this wizard.

Click Finish to perform the operations with the following settings:

```
Source database settings
    Source server name: EC2AMAZ-OLC7E1L
    Source database name: employee
Generate PowerShell script
    Script location: C:\Users\Administrator\Documents\
    Script name: decrypt_generated.ps1
Encrypt column salary
    Table name: employee
    Encryption key name: Fortanix_CEK
    Encryption type: Randomized
    Operation mode: Client-side
```

FIGURE 10: SUMMARY

7. Click the **Finish** Button to view the results.



```
select * from employee
```

	first_name	last_name	empID	salary
1	Adam	Parker	1	0x019391BB560191045589D9ACC3EA640C95A5140415C94E0B...
2	Adam	Parker	1	0x014C47B220FB6C7519E1E5DEB239908F8F786B65DD29D93...
3	John	Doe	2	0x0176227DDEB2F40424D461CDF81D0EE17F712D7D15A8D4...
4	Peter	Williams	2	0x01B8B322069EED3F5C98CAE0EE4A7AF49AF33F55AD1655E...

FIGURE 11: RESULTS

4.3.5 VERIFY ALWAYS ENCRYPTED COLUMNS

Perform the following steps on the testing server or the application server to view encrypted columns in plain text format:

NOTE: Ensure that the Fortanix CNG 64-bit client must be installed and configured with Fortanix endpoint and API key.

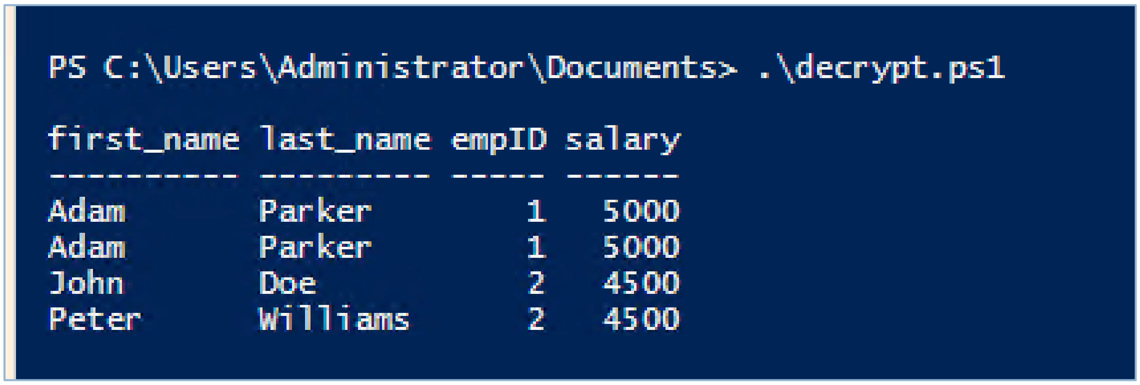
1. Log into the Windows 10 professional machine.
2. Install Fortanix Client 64-bit CNG Client. For more information, refer to *Section 4.2: Fortanix CNG Client*.
3. Run the following command to install the SQL Server PowerShell module:

```
install-Module -Name SqlServer -AllowClobber
```

4. Use the following sample script to decrypt the data:

```
# Import the SqlServer module.
Import-Module "SqlServer"
# Connect to your database.
$serverName = "<server_name>"
```

```
 $databaseName = "<database_name>"

 # Change the authentication method in the connection string, if
 needed.
 $connStr = "Data Source=$serverName;Initial
 Catalog=employee;Integrated Security=True;Column Encryption Setting
 = Enabled"

 #Tesing using SQL login
 #$pwd = read-host -AsSecureString -Prompt "Password"
 #$connStr = "Data Source=$serverName; User Id=applogin;Initial
 Catalog=$databaseName; Password
 =Fortanix123!;TrustServerCertificate=true;Column Encryption Setting
 = Enabled"

 # Invoke the query to view the encrypted data
Invoke-Sqlcmd -ConnectionString $connStr -Query "SELECT * FROM
dbo.employee" | Format-Table -AutoSize
```



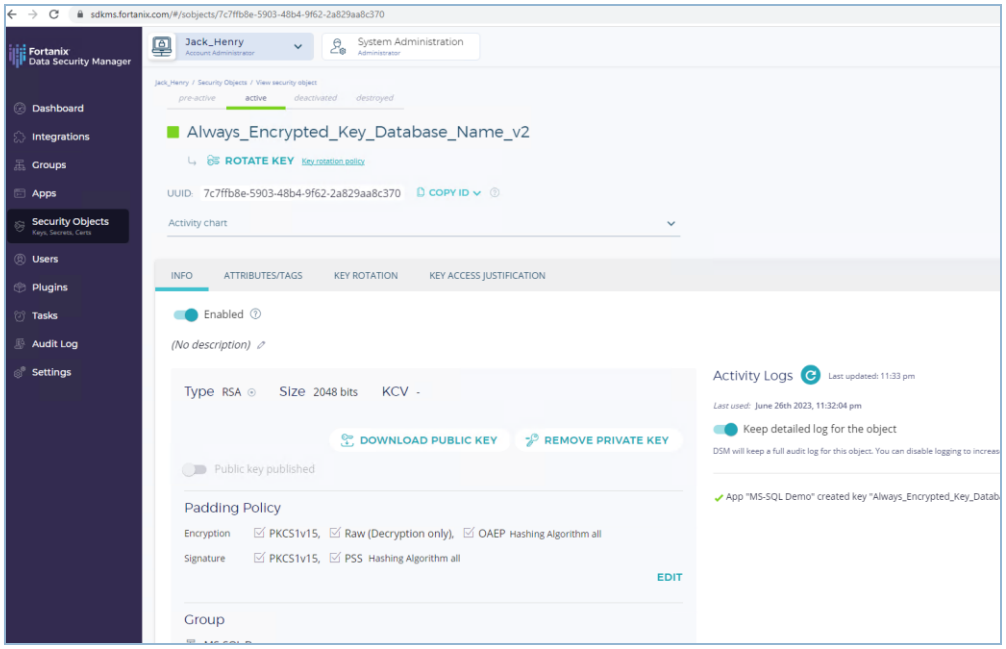
```
PS C:\Users\Administrator\Documents> .\decrypt.ps1

first_name last_name empID salary
---------- --------- ----- ------
Adam       Parker        1   5000
Adam       Parker        1   5000
John       Doe           2   4500
Peter      Williams      2   4500
```

4.3.6 ROTATE ALWAYS ENCRYPTED KEY

Rotating the Always Encrypted Keys is the process of replacing an existing key with a new one. You may need to rotate a key if it has been compromised, or to comply with your organization's policies or compliance regulations that mandate that the cryptographic keys must be rotated regularly.

- **Rotate column encryption key**: This involves decrypting the existing data with current key and re-encrypting it using the new column encryption key.
- **Rotate column master key**: This involves decryption the column encryption key and protecting it with new column master key. *For more information, refer to https://learn.microsoft.com/en-us/sql/relational-databases/security/encryption/rotate-always-encrypted-keys-using-ssms?view=sql-server-ver16.* Perform the following steps:

  1. **Creating New Column Master Key:** Create new security object in Fortanix DSM as described in *Section 4.2.2: Create Column Master Key*.



Always_Encrypted_Key_Database_Name_v2

UUID: 7c7ffb8e-5903-48b4-9f62-2a829aa8c370

Enabled

(No description)

Type RSA Size 2048 bits KCV -

Public key published

Padding Policy

Encryption ☑ PKCS1v15, ☑ Raw (Decryption only), ☑ OAEP Hashing Algorithm all

Signature ☑ PKCS1v15, ☑ PSS Hashing Algorithm all

Activity Logs Last updated: 11:33 pm

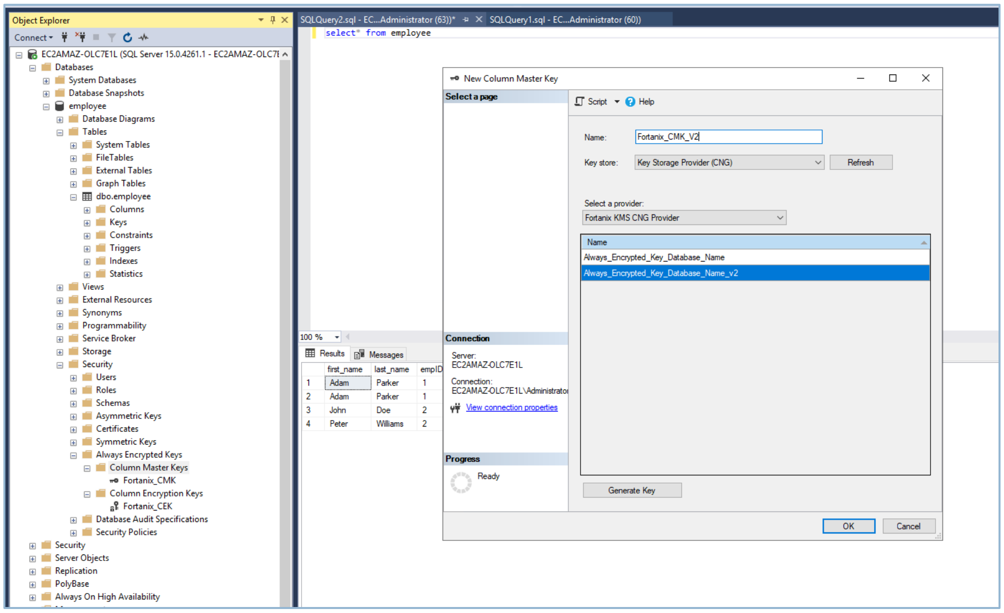
Last used: June 26th 2023, 11:32:04 pm

Keep detailed log for the object

DSM will keep a full audit log for this object. You can disable logging to increas

✓ App "MS-SQL Demo" created key "Always_Encrypted_Key_Datab

FIGURE 12: KEY CREATED ON FORTANIX DSM



```
select* from employee
```

Name: Fortanix_CMK_V2

Key store: Key Storage Provider (CNG)

Select a provider: Fortanix KMS CNG Provider

Name
Always_Encrypted_Key_Database_Name
Always_Encrypted_Key_Database_Name_v2

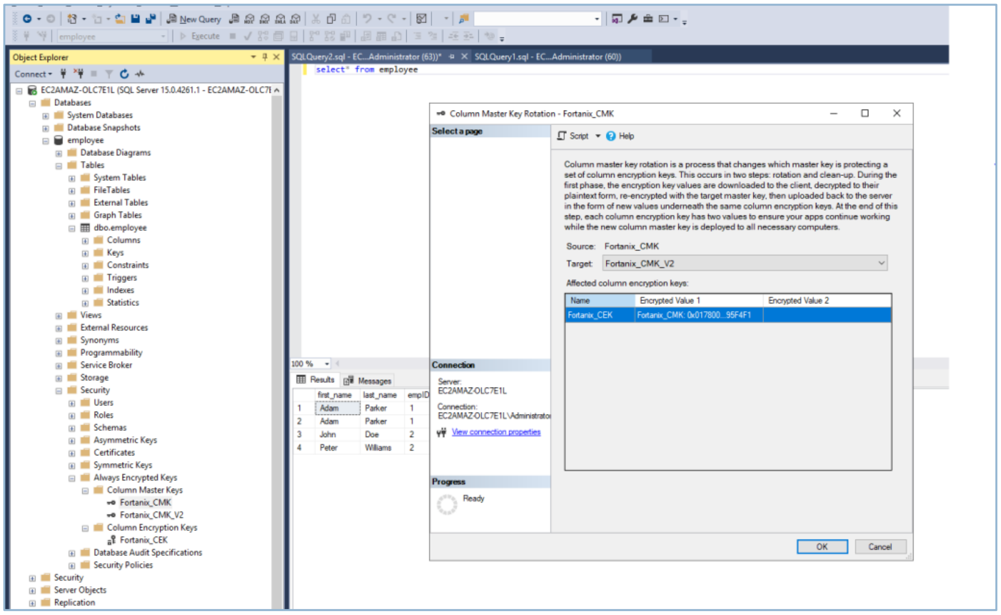
Server: EC2AMAZ-OLC7E1L

Connection: EC2AMAZ-OLC7E1L\Administrator

	first_name	last_name	empID
1	Adam	Parker	1
2	Adam	Parker	1
3	John	Doe	2
4	Peter	Williams	2

FIGURE 13: KEY CREATED ON SQL SERVER

2. **Rotating the Key:** After the key is rotated, the affected column encryption key will have two encrypted values: one value encrypted with the existing column master key, and a new value encrypted with the new column master key.
   a. Navigate to the **Security** → **Always Encrypted Keys** → **Column Master Keys** folder and locate the column master key that you want to rotate.
   b. Right-click on the column master key and select the **Rotate** option.
   c. In the **Column Master Key Rotation** dialog box, select the name of your **new column master key** that you created in *Step 1: Creating New Column Master Key in the Target field*.
   d. Review the list of the column encryption keys, protected by the existing column master keys. These keys will be affected by the rotation.
   e. Click the **OK** button.



```
select* from employee
```

	first_name	last_name	empID
1	Adam	Parker	1
2	Adam	Parker	1
3	John	Doe	2
4	Peter	Williams	2

Column Master Key Rotation - Fortanix_CMK

Column master key rotation is a process that changes which master key is protecting a set of column encryption keys. This occurs in two steps: rotation and clean-up. During the first phase, the encryption key values are downloaded to the client, decrypted to their plaintext form, re-encrypted with the target master key, then uploaded back to the server in the form of new values underneath the same column encryption keys. At the end of this step, each column encryption key has two values to ensure your apps continue working while the new column master key is deployed to all necessary computers.

Source: Fortanix_CMK

Target: Fortanix_CMK_V2

Affected column encryption keys:

Name	Encrypted Value 1	Encrypted Value 2
Fortanix_CEK	Fortanix_CMK: 0x017800...95F4F1	

Server:
EC2AMAZ-OLC7E1L

Connection:
EC2AMAZ-OLC7E1L\Administrator

Ready

FIGURE 14: ROTATE THE KEY

3. **Configure Application with New Column Master Key:** Ensure that all your client applications query database columns that are protected with the rotated Fortanix column master key can access the new column master key.

   The column master key is stored in Fortanix DSM, the application must be implemented so that it can authenticate to Fortanix DSM and has permission to access the new column master key.

4. **Cleaning Up:** After you have configured all your applications to use the new column master key, remove the values of column encryption keys that are encrypted with the old column master key from the database. Removing old values will ensure that you are ready for the next rotation.

NOTE: Each column encryption key is protected with a column master key to be rotated, must have exactly one encrypted value.

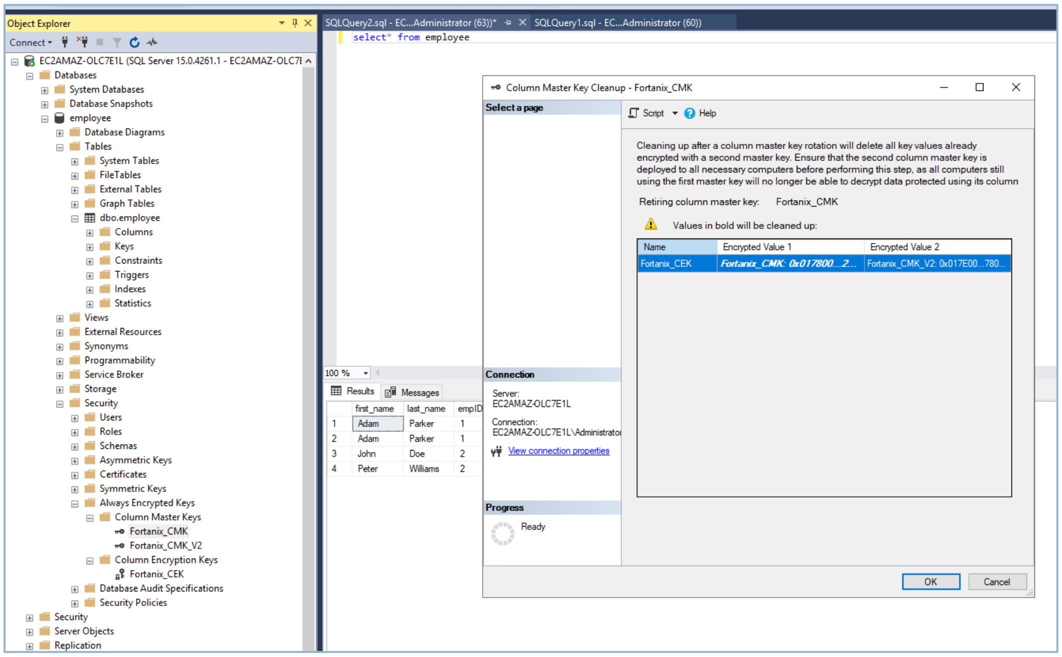
WARNING: If you remove the value of a column encryption key before its corresponding column master key has been made available to an application, the application will no longer be able to decrypt the database column.

a. Navigate to the **Security** → **Always Encrypted Keys** folder and locate the existing column master key that you want to replace.

b. Right-click on your existing column master key and select the **Cleanup** option.

c. Review the list of column encryption key values to be removed.

d. Click the **OK** button.



```
select* from employee
```

	first_name	last_name	empID
1	Adam	Parker	1
2	Adam	Parker	1
3	John	Doe	2
4	Peter	Williams	2

Server:
EC2AMAZ-OLC7E1L

Connection:
EC2AMAZ-OLC7E1L\Administrator

View connection properties

Cleaning up after a column master key rotation will delete all key values already encrypted with a second master key. Ensure that the second column master key is deployed to all necessary computers before performing this step, as all computers still using the first master key will no longer be able to decrypt data protected using its column

Retiring column master key: Fortanix_CMK

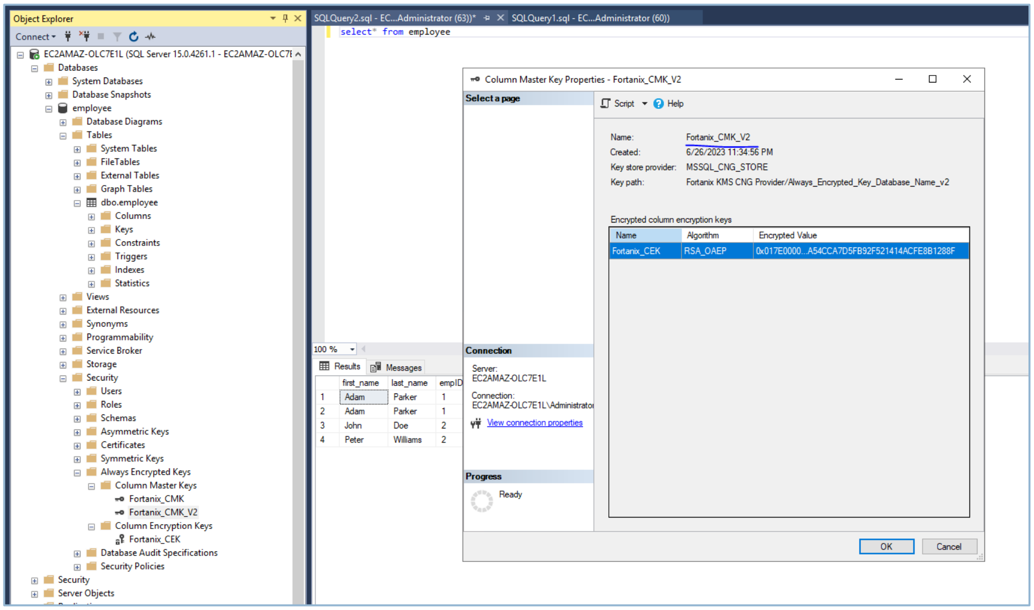
Values in bold will be cleaned up:

Name	Encrypted Value 1	Encrypted Value 2
Fortanix_CEK	***Fortanix_CMK: 0x017800...2...***	Fortanix_CMK_V2: 0x017E00...780...

FIGURE 15: CLEAN UP

5. **After Rotation:** To verify if that rotation is done successfully, refer to the following figures:

a. **Column Master Key:**



```
select* from employee
```

	first_name	last_name	empID
1	Adam	Parker	1
2	Adam	Parker	1
3	John	Doe	2
4	Peter	Williams	2

Column Master Key Properties - Fortanix_CMK_V2

Name: Fortanix_CMK_V2
Created: 6/26/2023 11:34:56 PM
Key store provider: MSSQL_CNG_STORE
Key path: Fortanix KMS CNG Provider/Always_Encrypted_Key_Database_Name_v2

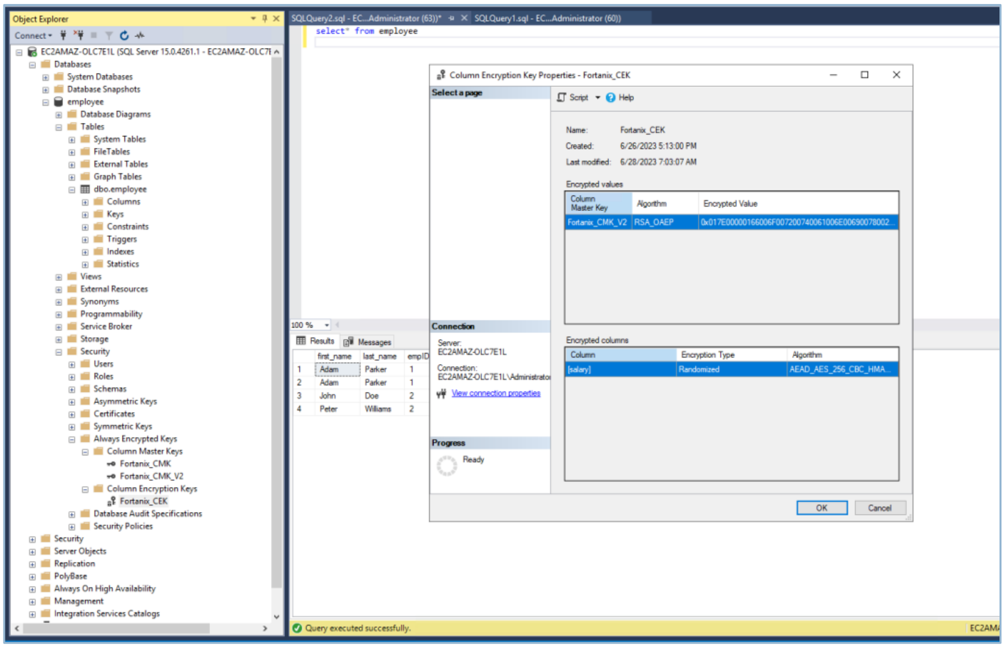
Encrypted column encryption keys

Name	Algorithm	Encrypted Value
Fortanix_CEK	RSA_OAEP	0x017E0000...A54CCA7D5FB92F521414ACFE8B1288F

Server:
EC2AMAZ-OLC7E1L
Connection:
EC2AMAZ-OLC7E1L\Administrator

FIGURE 16: COLUMN MASTER KEY

b. **Column Encryption Key:**



```
select* from employee
```

	first_name	last_name	empID
1	Adam	Parker	1
2	Adam	Parker	1
3	John	Doe	2
4	Peter	Williams	2

Column Encryption Key Properties - Fortanix_CEK

Name: Fortanix_CEK
Created: 6/26/2023 5:13:00 PM
Last modified: 6/28/2023 7:03:07 AM

Encrypted values

Column Master Key	Algorithm	Encrypted Value
Fortanix_CMK_V2	RSA_OAEP	0x017E00000166006F007200740061006E00690078002...

Encrypted columns

Column	Encryption Type	Algorithm
[salary]	Randomized	AEAD_AES_256_CBC_HMA...

Server:
EC2AMAZ-OLC7E1L
Connection:
EC2AMAZ-OLC7E1L\Administrator

Query executed successfully.

FIGURE 17: COLUMNS ENCRYPTION KEY

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/16916974311700-Fortanix-Data-Security-Manager-with-Microsoft-SQL-Server-Always-Encrypted

5.2 DOCUMENT UPDATES

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

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

NOTICE: This document was produced by Fortanix, Inc. (Fortanix) and contains information which is proprietary and confidential to Fortanix. The document contains information that may be protected by patents, copyrights, and/or other IP laws. If you are not the intended recipient of this material, please destroy this document and inform info@fortanix.com immediately.