SQL Server Encryption Unplugged

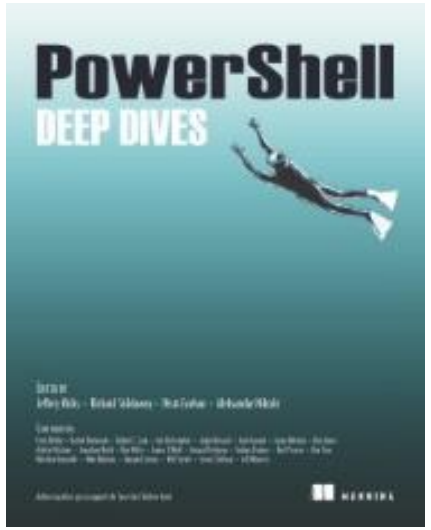
In Depth Look at Encryption

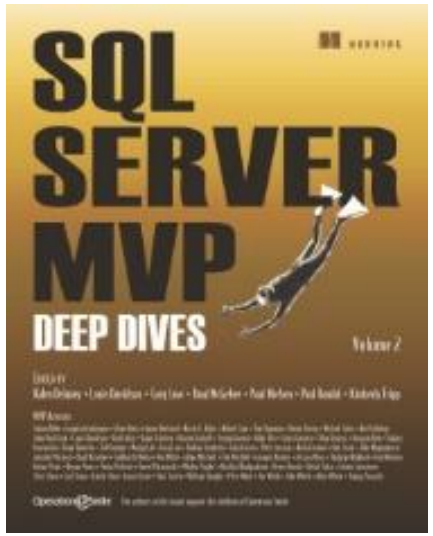
Ben Miller

SQL Architect

Ben Miller & Associates, Inc.

DATA PLATFORM SUMMIT 2020
VIRTUAL | ROUND-THE-CLOCK
Empowering Data & AI Transformation



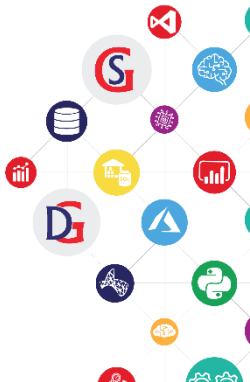


DBADuck
ben miller

Microsoft CERTIFIED Master
SQL Server® 2008

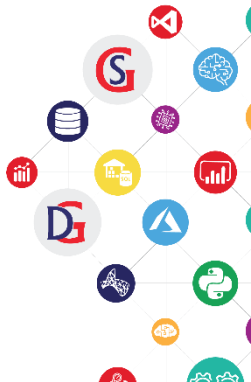
MVP
Microsoft®
Most Valuable
Professional

- Data Platform MVP (since 2009)
- Certified Master SQL Server
- Author
- Group Leadership (SQL, PASS PowerShell)
- Consultant
- PowerShell DBA and Trainer
- @DBAduck everywhere
- Email: ben@benmiller.net
- https://dbaduck.com



Agenda

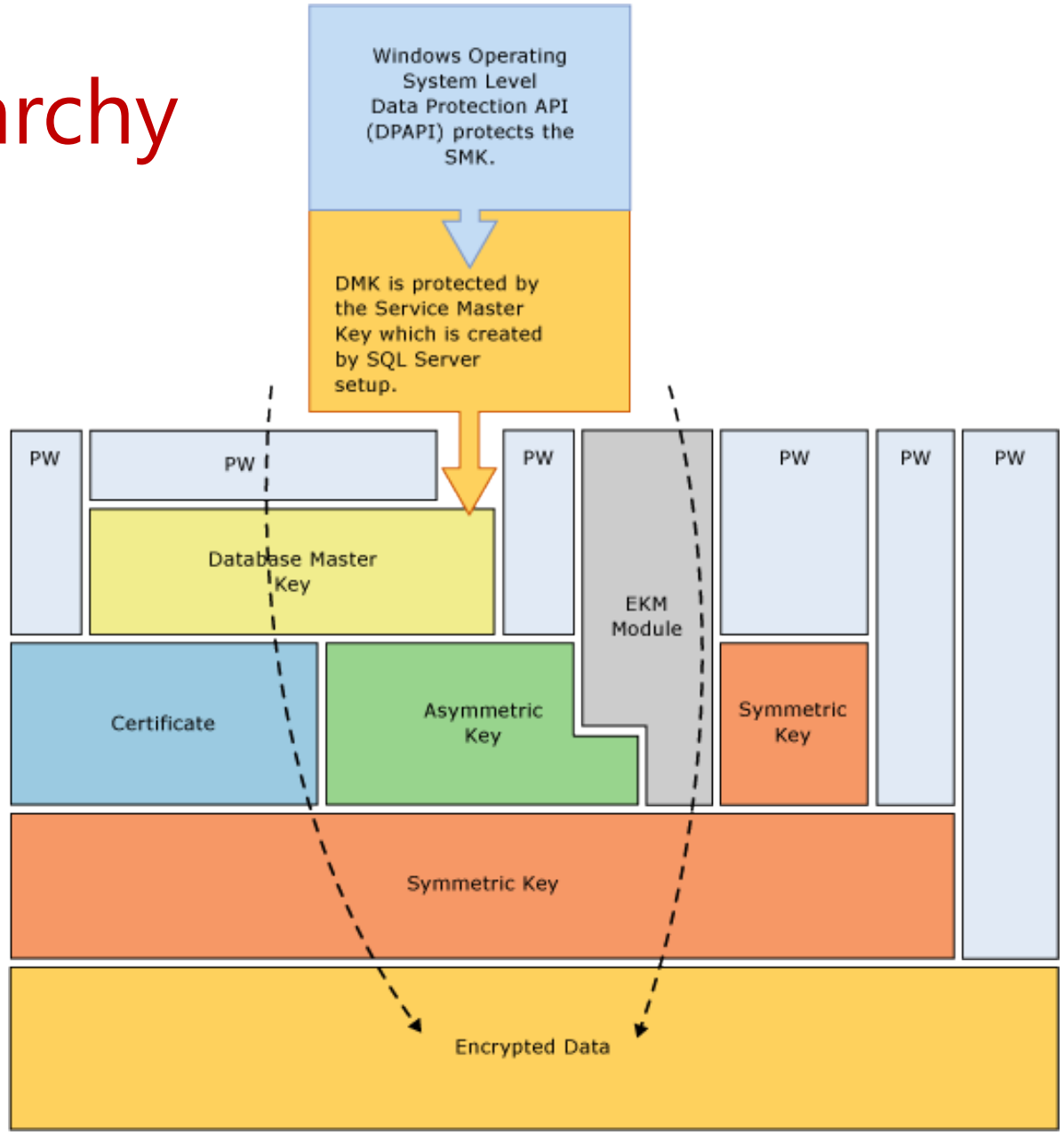
- Encryption Hierarchy
- Transparent Data Encryption (TDE)
- Cell Encryption
- Always Encrypted
- Backup Encryption



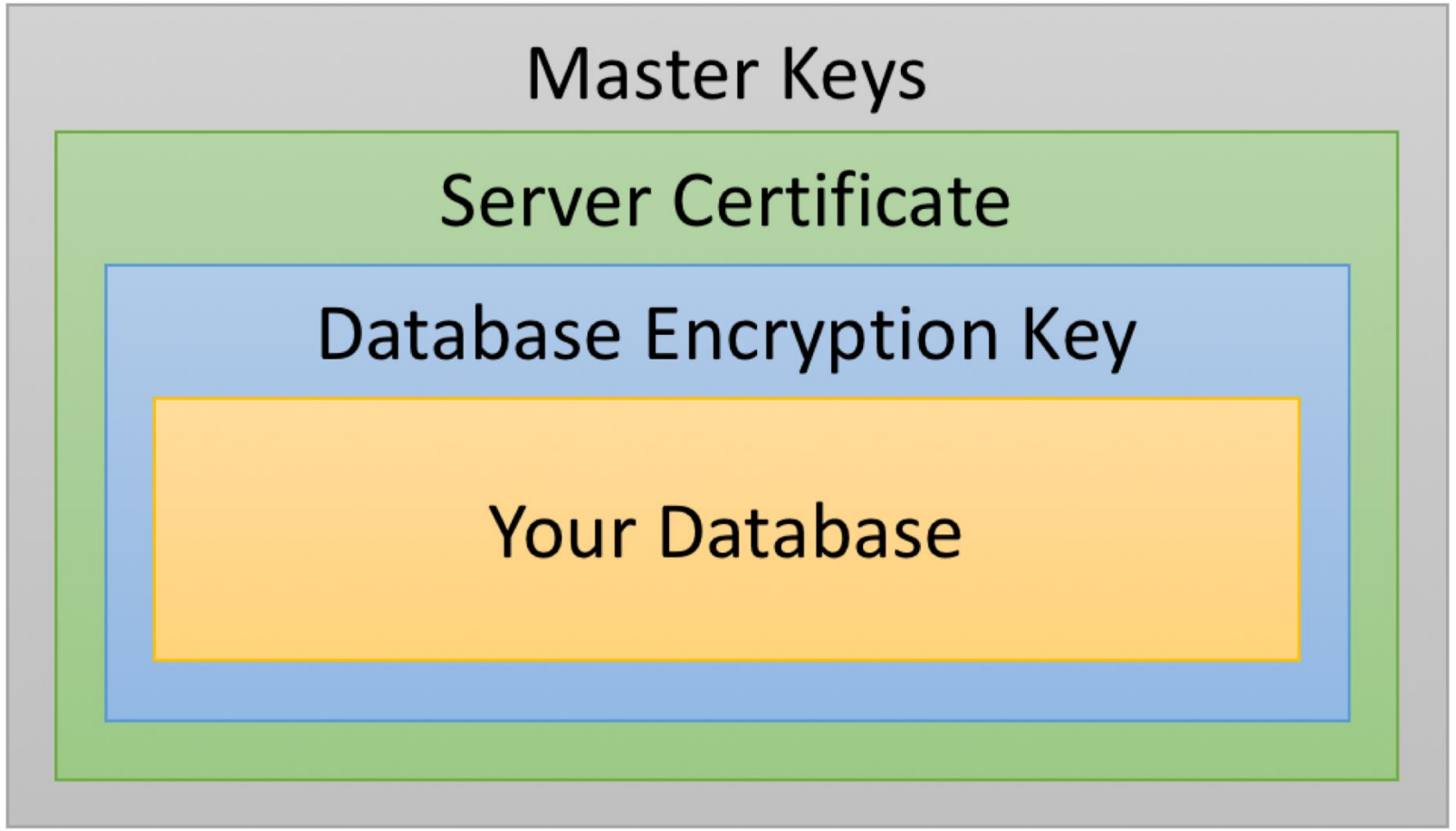
ENCRYPTION HIERARCHY

Master Keys, Certificates, Symmetric Keys

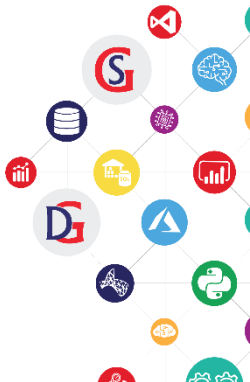
Encryption Hierarchy



Another Look at the Hierarchy



Encryption Hierarchy Simplified

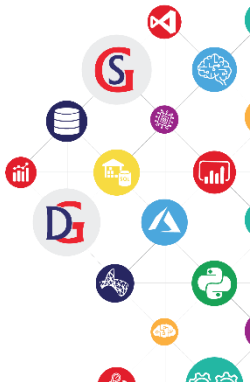


TRANSPARENT DATA ENCRYPTION (TDE)

Master Key, Server Certificate, Database Encryption Key

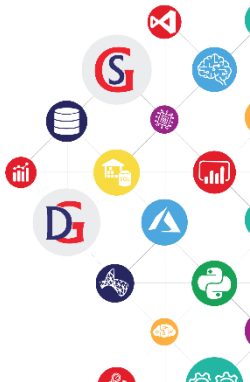
Transparent Data Encryption (TDE)

- Data Encrypted at Rest Only – not in transit
- Requirements
  - Master Key in master database
  - Certificate in master database (Server Certificate)
  - Database Encryption Key in user database
- Satisfies the Encryption Hierarchy
- Protect Your Keys!!!!



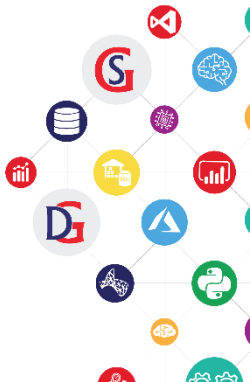
TDE Advantages

- No schema changes needed
- Page Level Encryption
- Overhead estimated at 1-2% (Your mileage may vary)
- Secure backups by default
- Invisible to the User
- “At Rest” Encryption



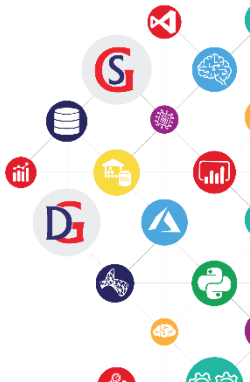
TDE Disadvantages

- Backup Compression no longer effective
  - Update: SQL 2016 and above you get compression now
- Enterprise Edition required (now in Standard in 2019)
- Not as versatile as Cell (Column) Level encryption
- TempDB is encrypted when one database is encrypted
  - TempDB is not decrypted until Service Restart



TDE ala Azure Key Vault

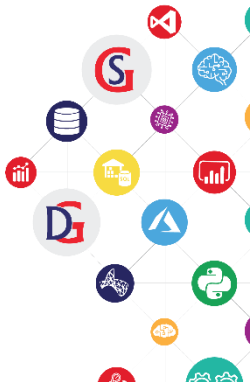
- Enterprise Key Management Solution
  - Where keys are not stored in the database but externally
  - Needs an install to be the EKM provider
  - Requires set up of Azure
  - Requires Internet Access
- Needs access to the Key Vault at least every 8 hours
  - Bricks your database if connectivity lost for more
- Alternative when access is lost
  - Change the encryption key to local key



Azure Key Vault Setup

- Azure AD Principal in the form of an App Registration
  - Name it and allow access to “this organizational directory only”
- Create a Secret (New Client Secret)
  - Name it and set to Never Expire
  - Copy the Value of the Client Secret (right then, you will not see it again)
- Copy and save the Application (client) ID for use later
- Create the Key Vault (enable soft delete)

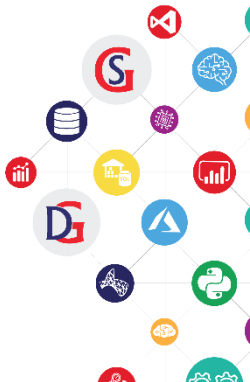
- Set up Transparent Data Encryption (TDE) Extensible Key Management with Azure Key Vault - SQL Server | Microsoft Docs



Azure Key Vault Setup (part 2)

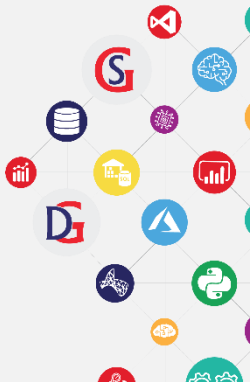
- Add an Access Policy to the Vault
  - Include Get, List, Unwrap Key, Wrap Key permissions
  - Select Principal you just created (App Registration)
- Create a Key
  - Generate, Name it, RSA, 2048, No Dates, Enabled
- Install the SQL Server Connector
  - 1.0.5.0 (version 15.0.2000.367) – File date September 11, 2020 (microsoft.com)
- Configure SQL Server to use the EKM Provider

- Set up Transparent Data Encryption (TDE) Extensible Key Management with Azure Key Vault - SQL Server | Microsoft Docs



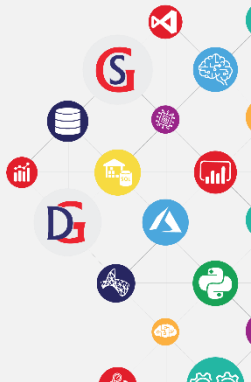
Configure SQL Server sp_configure

```
-- Enable advanced options.
USE master;
GO

EXEC sp_configure 'show advanced options', 1;
GO
RECONFIGURE;
GO

-- Enable EKM provider
EXEC sp_configure 'EKM provider enabled', 1;
GO
RECONFIGURE;
```



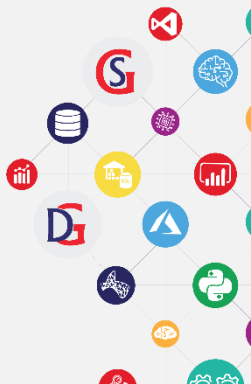
Create EKM Provider

```
-- Path Cannot be more than 256 characters
CREATE CRYPTOGRAPHIC PROVIDER AzureKeyVault_EKM
FROM FILE = 'C:\Program Files\SQL Server Connector for Microsoft Azure Key Vault\Microsoft.AzureKeyVaultService.EKM.dll';
GO
```



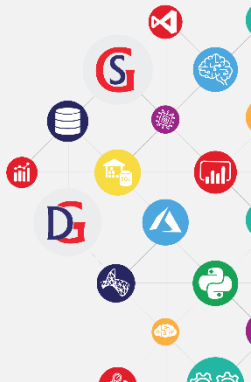
Create SQL CREDENTIAL

```
USE master;
CREATE CREDENTIAL sysadmin_ekm_cred
WITH IDENTITY = 'ContosoEKMKeyVault',     -- for public Azure
-- WITH IDENTITY='ContosoEKMKeyVault.vault.usgovcloudapi.net',--for Azure Government
-- WITH IDENTITY='ContosoEKMKeyVault.vault.azure.cn', -- for Azure China 21Vianet
-- WITH IDENTITY='ContosoEKMKeyVault.vault.microsoftazure.de', -- for Azure Germany
SECRET = '<----Application (Client) ID --><--Azure AD app (Client) ID secret-->'
FOR CRYPTOGRAPHIC PROVIDER AzureKeyVault_EKM;

-- Add the credential to the SQL Server administrator's domain login
ALTER LOGIN [<domain>\<login>]
ADD CREDENTIAL sysadmin_ekm_cred;
```



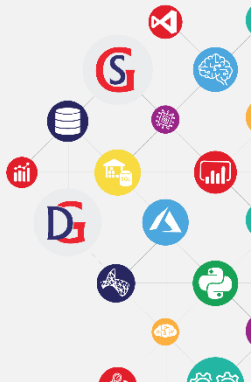
Create Asymmetric Key and Login

```
-- Create Asymmetric Key from EKM (Key Vault)
-- This key will be created from the one
-- you created in Key vault
CREATE ASYMMETRIC KEY EKMSampleASYKey
FROM PROVIDER [AzureKeyVault_EKM]
WITH PROVIDER_KEY_NAME = 'KeyCreatedInKeyVault',
CREATION_DISPOSITION = OPEN_EXISTING;

--Create a Login that will associate the asymmetric key to this login
CREATE LOGIN TDE_Login
FROM ASYMMETRIC KEY EKMSampleASYKey;
```



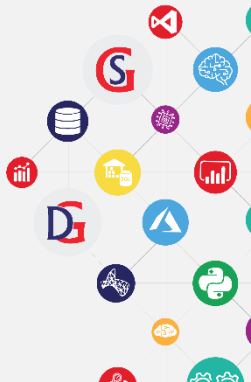
More Plumbing for the Credential

```
--Now drop the credential mapping from the original association
ALTER LOGIN [<domain>\<login>]
DROP CREDENTIAL sysadmin_ekm_cred;

--Now add the credential mapping to the new login for TDE
ALTER LOGIN TDE_Login
ADD CREDENTIAL sysadmin_ekm_cred;
```



Create Database and Encrypt

```
--Create a test database that will be encrypted with the Azure key vault key
CREATE DATABASE TestTDE

--Create an ENCRYPTION KEY using the ASYMMETRIC KEY (EKMSampleASYKey)
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_256
ENCRYPTION BY SERVER ASYMMETRIC KEY EKMSampleASYKey;

--Enable TDE by setting ENCRYPTION ON
ALTER DATABASE TestTDE
SET ENCRYPTION ON;
```



Demo

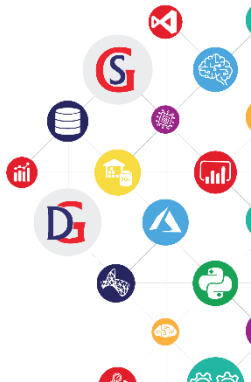
All Things TDE

CELL ENCRYPTION

Symmetric Key, Certificate

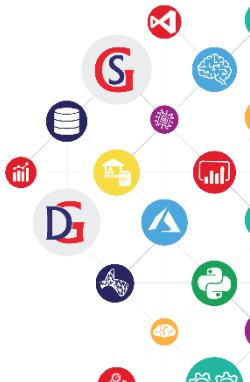
Cell Encryption

- Keys you can use
  - Certificate
  - Asymmetric Key
  - Symmetric Key
- Still want to use the Encryption Hierarchy so the keys can be opened without a password.
- Cell becomes varbinary(x) and cannot be indexed or searched
- Data is encrypted in Buffer Pool



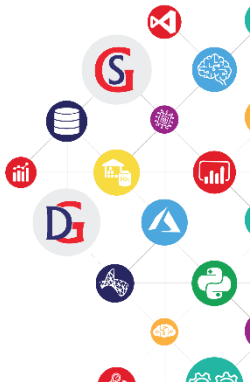
Symmetric/Asymmetric Keys

- Symmetric Keys
  - DO NOT USE RC4 and RC4_128 algorithms
  - ALGORITHM of TRIPLE_DES_3KEY use a 192-bit key.
  - ALGORITHM of TRIPLE_DES use a 128-bit key.
- Asymmetric Keys
  - Ownership
    - Can be Windows Logins, SQL Server Logins, and Application Roles
    - Cannot be Groups and Roles
- Key Pair and Encrypted by Master Key, Password or EKM Provider
- Can be RSA_512, RSA_1024, RSA_2048



Notes on Encrypting Data

- Notes:
  - For best performance, encrypt data using symmetric keys instead of certificates or asymmetric keys
  - Database master keys are protected by the Service Master Key
  - An Extensible Key Management (EKM) module holds symmetric or asymmetric keys outside of SQL Server
  - The Service Master Key and all Database Master Keys are symmetric keys

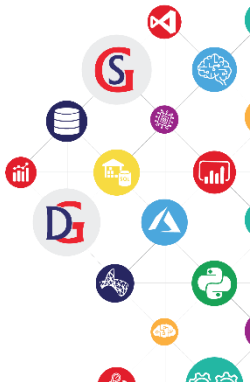


ALWAYS ENCRYPTED

Column Master Key, Column Encryption Key, Certificate, DataTypes

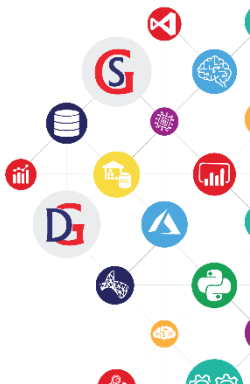
Always Encrypted Primer

- Role Separation vs. Not
  - Beware where you create the certificate used
- Local or Centralized Key Stores (Azure Key Vault or Local Cert Store)
- Certificates
- Expiration Date or Certificate Authority Chain is not validated
- X509 standard cert
- Must contain the Private Key (recommend > 2048 bits)
- Must be created for Key Exchange
- Must be on the Client that will be used to connect to SQL



Always Encrypted Requirements

- Certificate Created on the Client
- Column Master Key
- Column Encryption Key
- String Columns
  - Must be Latin1_General_BIN2 collation if DETERMINISTIC encryption
- Column Encryption Types
  - DETERMINISTIC – Can be used in Indexes and searched
  - RANDOMIZED – Cannot be used in Indexes and not valid for search
- ALGORITHM
  - One supported right now: AEAD_AES_256_CBC_HMAC_SHA_256
- Temporal Tables cannot have Always Encrypted Columns



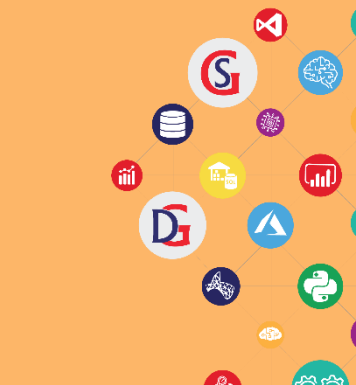
Always Encrypted More…

- Triggers are partially supported
  - Cannot Select Encrypted Columns from inserted/deleted tables
- In Memory OLTP does not support Encrypted Columns
- Foreign Keys must match Encryption Type
- .NET Framework 4.6 to allow the connection string parameter
  - Column Encryption Setting=Enabled
- More…….

- Very good reference with examples of Error Messages (Aaron Bertrand)
  - https://blogs.sentryone.com/aaronbertrand/t-sql-tuesday-69-always-encrypted-limitations/

Demo

Always Encrypted



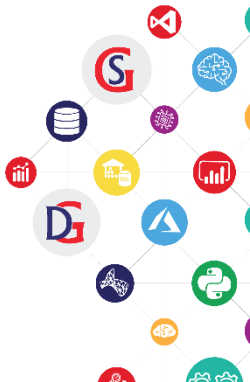
BACKUP ENCRYPTION

Asymmetric Key, Certificate

Backup Encryption

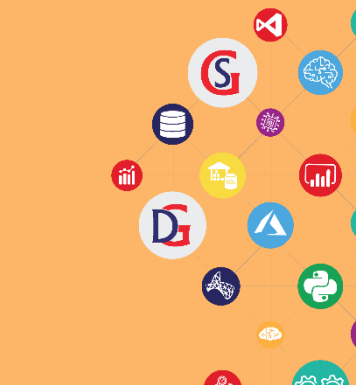
- SQL 2014+
- Certificate or Asymmetric Key
  - Asymmetric Key must reside in EKM
- Encryption Algorithms (AES 128, 192, 256 and TripleDES)
- New Media Set – cannot add to an existing

- Permissions:
  - db_backupoperator membership
  - VIEW DEFINITION on the certificate in master db
  - Or Obviously, Sysadmin membership



Demo

Backup Encryption



Thank You For Attending!

ben@benmiller.net

@DBAduck

DATA PLATFORM SUMMIT 2020

VIRTUAL | ROUND-THE-CLOCK

Empowering Data & AI Transformation

Special Thanks To

Microsoft

for supporting

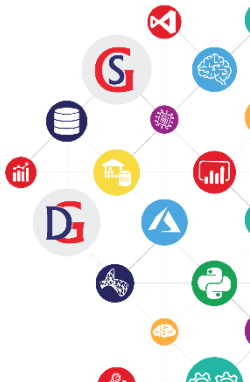
DataPlatformGeeks & SQLServerGeeks

Community Initiatives

SQLServerGeeks

Data Platform Summit

DataPlatformGeeks



Thank You

Three Ways to Win Prizes

Post your selfie with hash tag **#DPS2020**

Give Session & Conference Feedback

Visit our Sponsors & Exhibitors

Follow us on Twitter **@TheDataGeeks @DataAISummit**

DATA PLATFORM SUMMIT 2020
VIRTUAL | ROUND-THE-CLOCK
Empowering Data & AI Transformation