Data Platform Security in Azure

Robin Lester

Microsoft

GDPR with data - common themes

- Key tooling to consider
  - Discovery - Azure Data Catalog
  - SQL encryption technologies
- Right to be forgotten – do I have to delete emails, backups etc?
  - Rights can not overwrite the rights of others
  - As long as processes are in place backup strategy can still stand

Azure Key Vault

Organizations need to safeguard certificates deployed into their VMs.

1

Developers need to safeguard config secrets of their Azure cloud services.

e.g. Storage account key
SQL connection string

2

Organizations need to control encryption keys used by their OWN apps.

3

Organizations need to control encryption keys used by SaaS services.

4

What is Azure Key Vault?

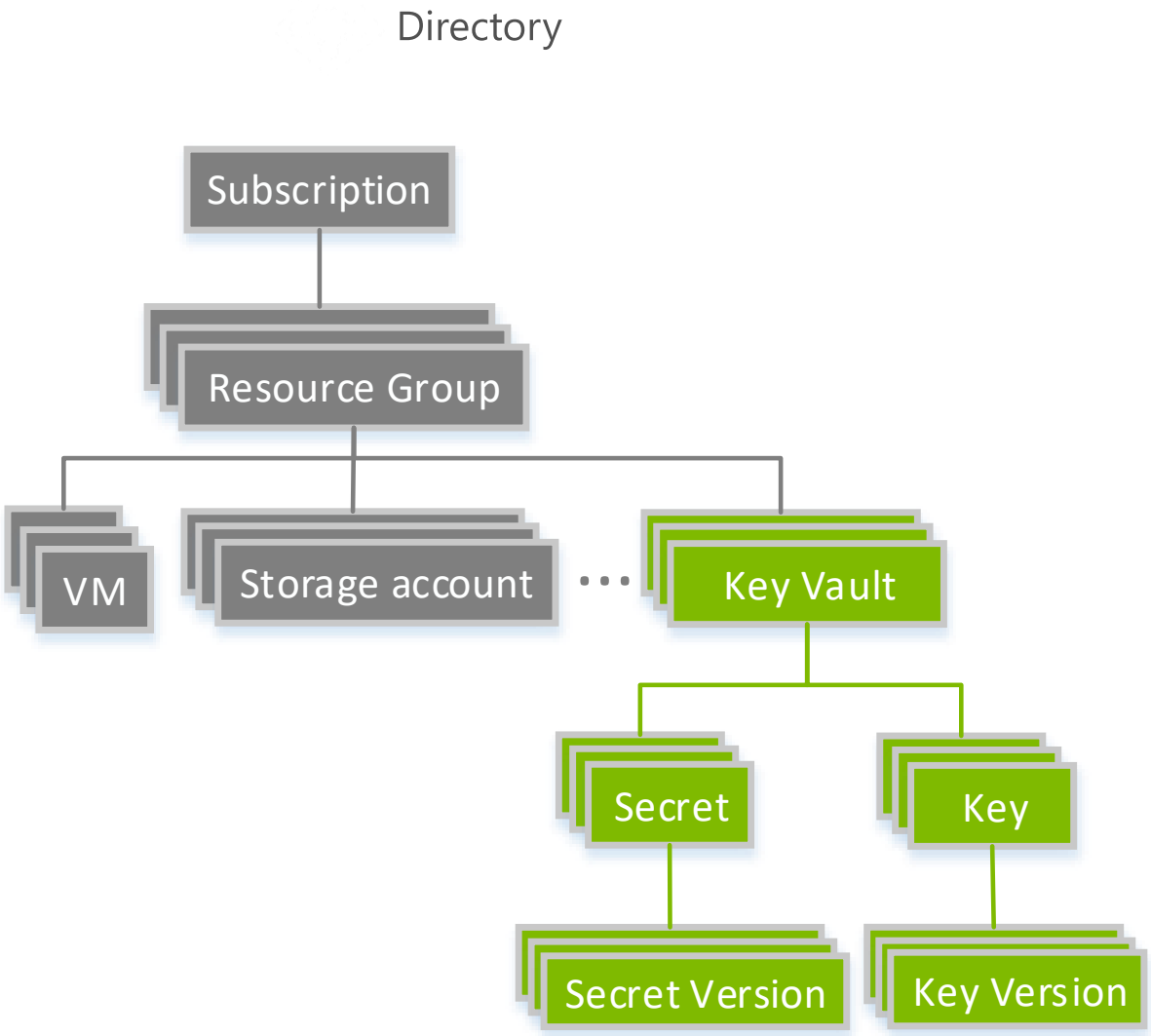
- An Azure resource provider that lets you
  - Store & manage SECRETS (esp app config), and release them at runtime to authorized apps & users.
  - Store & manage KEYS, and perform cryptographic operations on behalf of authorized apps & users.

- Backed by Hardware Security Modules
  - All secrets and keys are protected at rest with key chain terminating in HSMs.
  - Keys marked as ‘HSM-protected’ are protected even at runtime with HSMs.

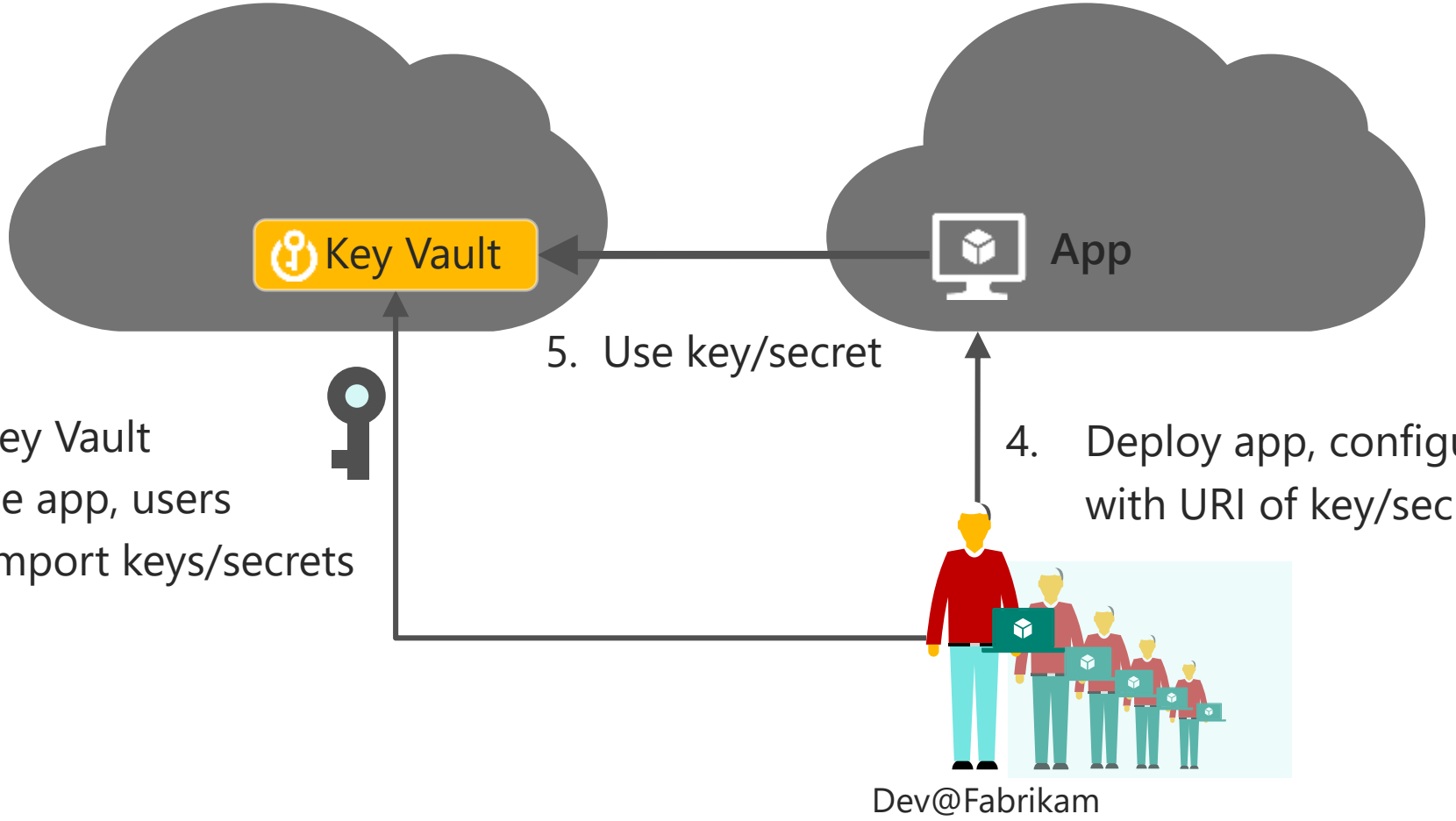
Terminology

- Key Vault
  - Container for related keys and secrets that are managed together.
  - Unit of access control, unit of billing.
  - An Azure resource, like a storage account.
- Secret
  - What: Any sequence of bytes under 25KB. E.g. SQL connection string, Storage account key.
  - How used: Authorized users/apps write and read back the secret value.
- Key
  - What: A cryptographic key. RSA 2048.
  - How used: A key cannot be read back. Caller must ask the service to decrypt / sign with the key.

Key Vault within Azure object model

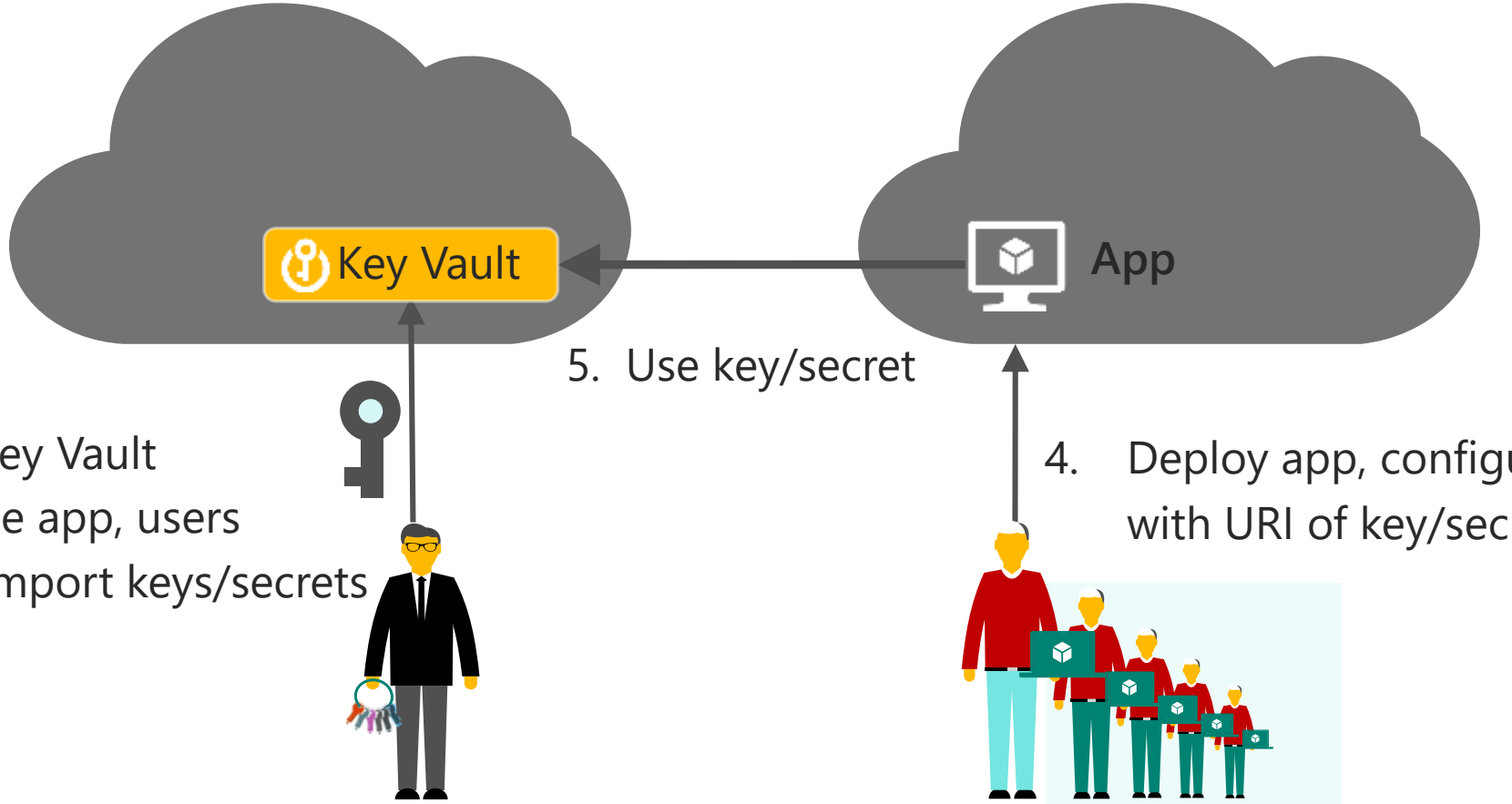


Phase 1: Developer builds/tests application



1. Create Key Vault
2. Authorize app, users
3. Create/import keys/secrets

Phase 2: App moves into pilot / pre-prod



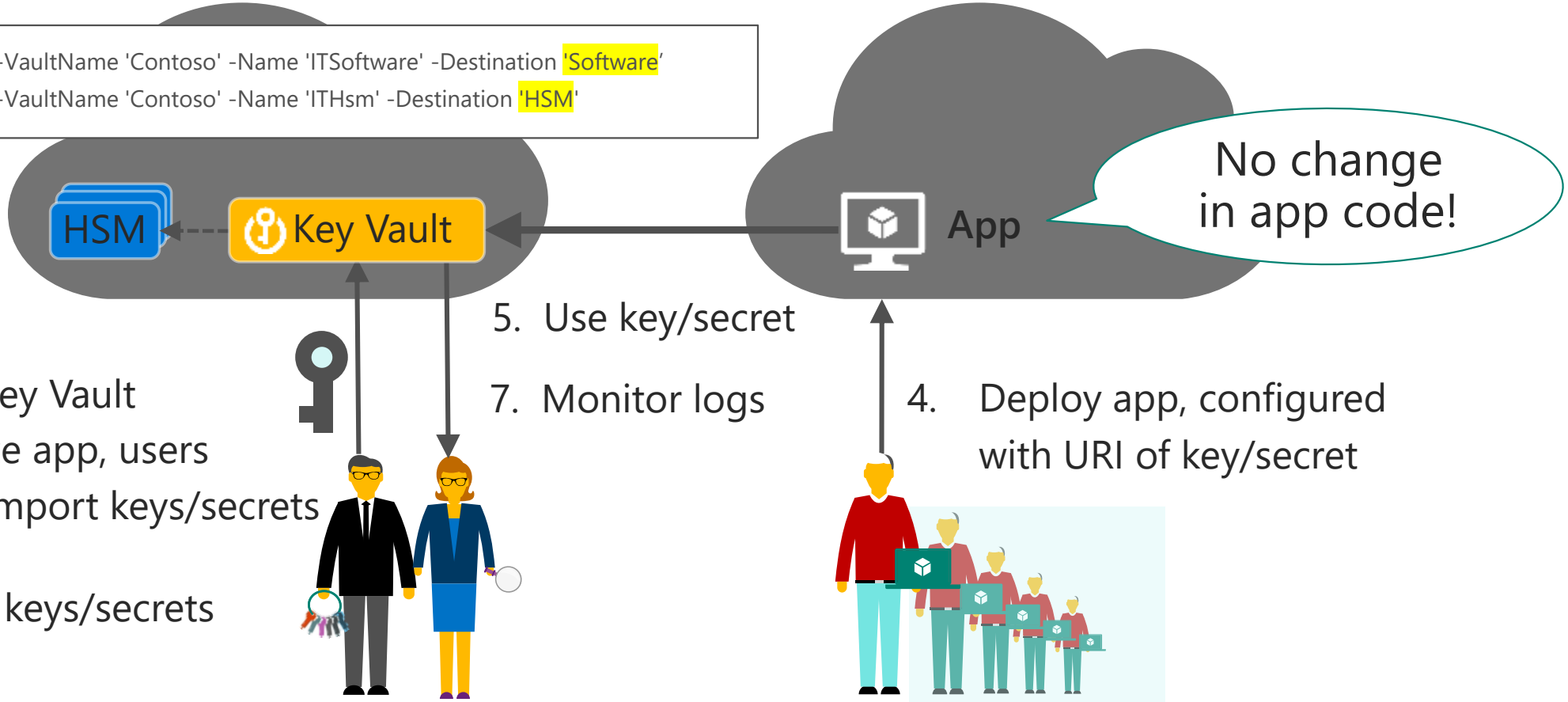
1. Create Key Vault
2. Authorize app, users
3. Create/import keys/secrets

Privileged@Fabrikam

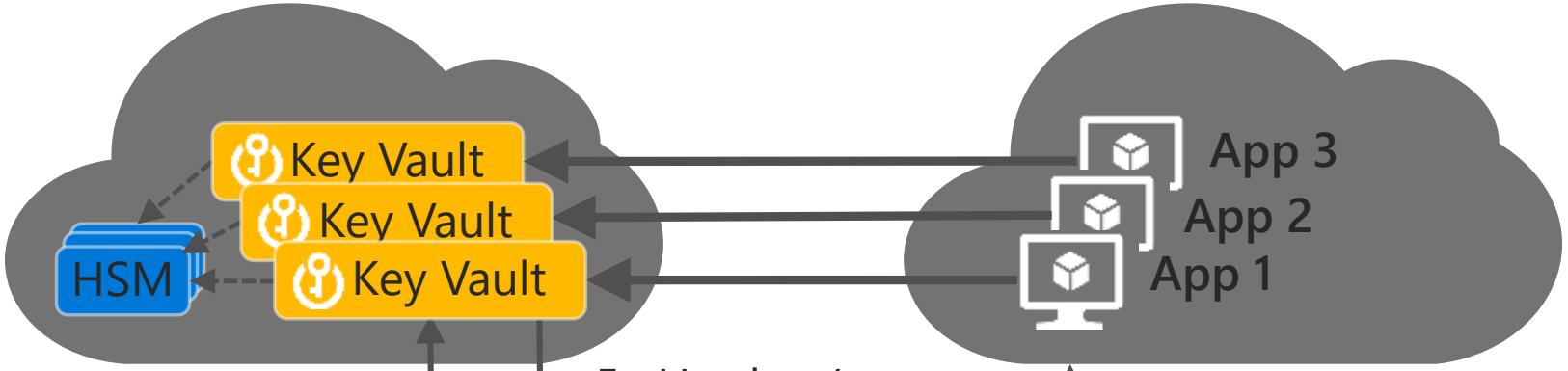
Dev@Fabrikam

Phase 3: App moves into production

```
Add-AzureKeyVaultKey -VaultName 'Contoso' -Name 'ITSoftware' -Destination 'Software'
Add-AzureKeyVaultKey -VaultName 'Contoso' -Name 'ITHsm' -Destination 'HSM'
```



Phase 4: Scale, deploy more apps in minutes



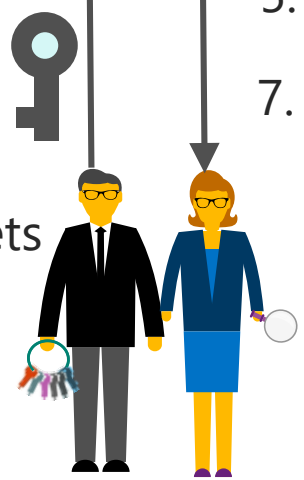
5. Use key/secret

7. Monitor logs

1. Create Key Vault
2. Authorize app, users
3. Create/import keys/secrets

6. Manage keys/secrets

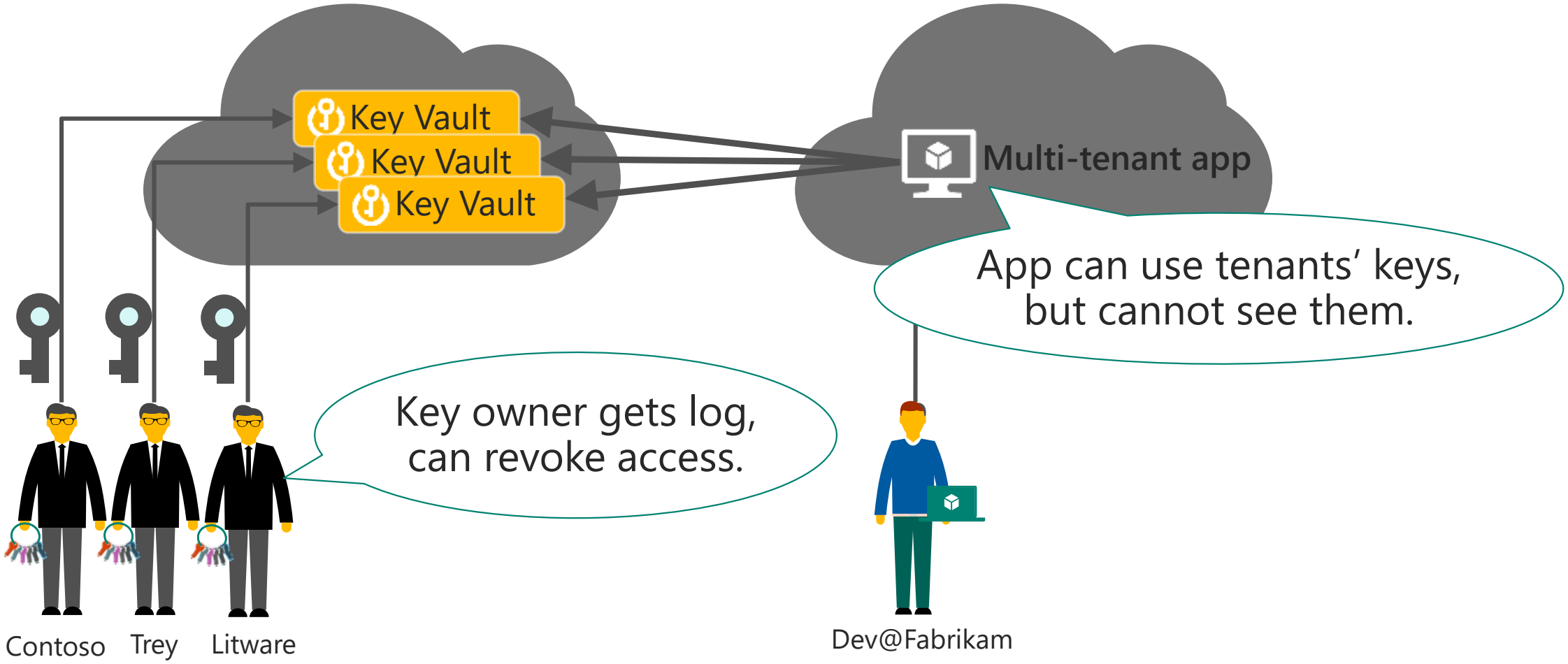
4. Deploy app, configured with URI of key/secret



CISO@Fabrikam

Dev@Fabrikam

Multi-tenant app offers customer-managed keys



KeyVault Application Setup

- Create a local certificate
- Create an application in Azure Active Directory
  - Using the certificate
- Create a KeyVault
- Allow the application permission to the KeyVault

```
$certificateName = "$applicationName" + "cert"
$myCertThumbprint = (New-SelfSignedCertificate -Type Custom -Subject
```

```
Set-AzureRmKeyVaultAccessPolicy
-VaultName $vaultName `
-ObjectId $servicePrincipal.Id `
-PermissionsToKeys all `
-PermissionsToSecrets all `
-PermissionsToCertificate all
```

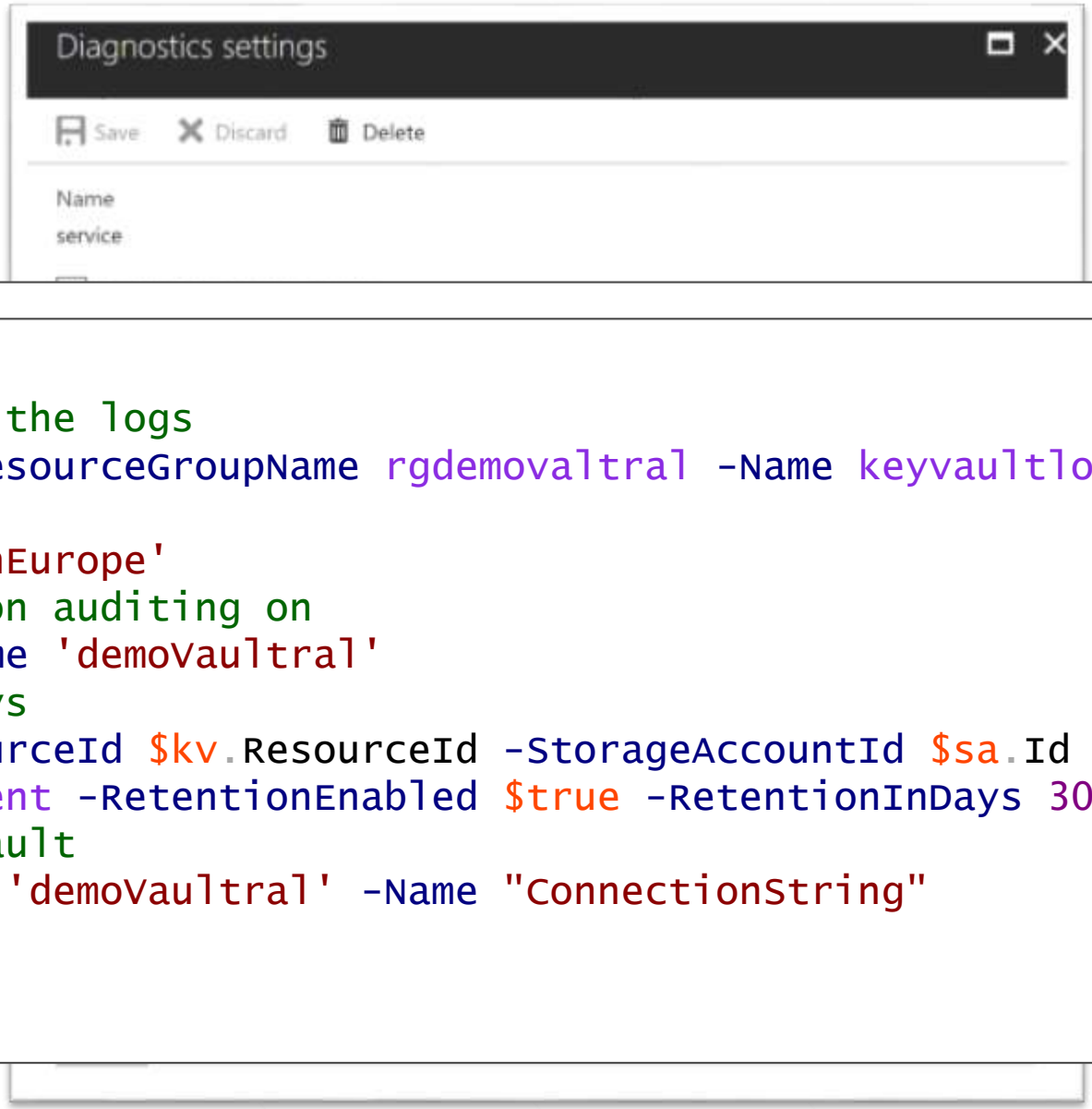
```
$now = [System.DateTime]::Now
$oneYearFromNow = $now.AddYears(1)
```

Connecting an app to the Keyvault

- Demo

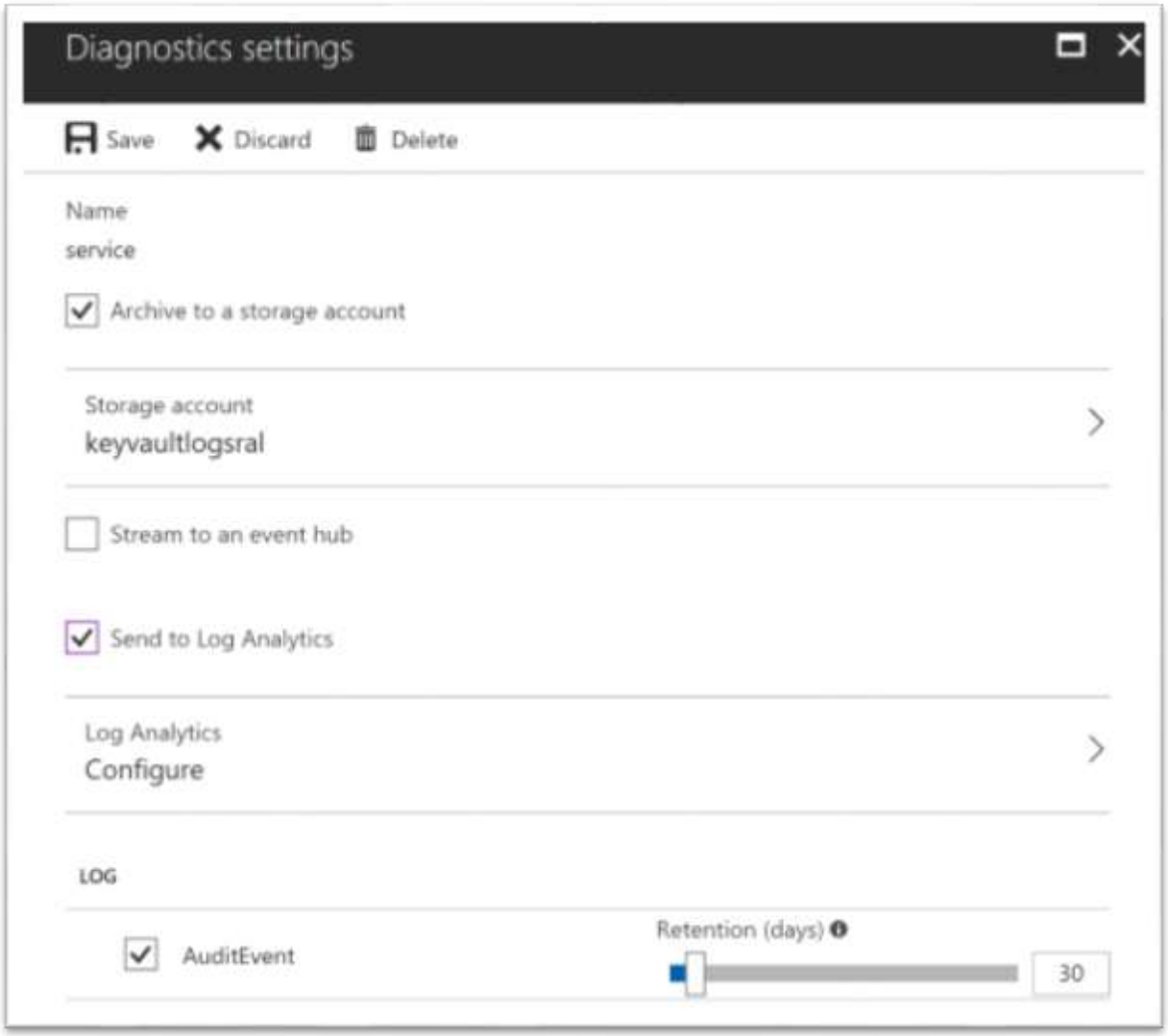
Admin

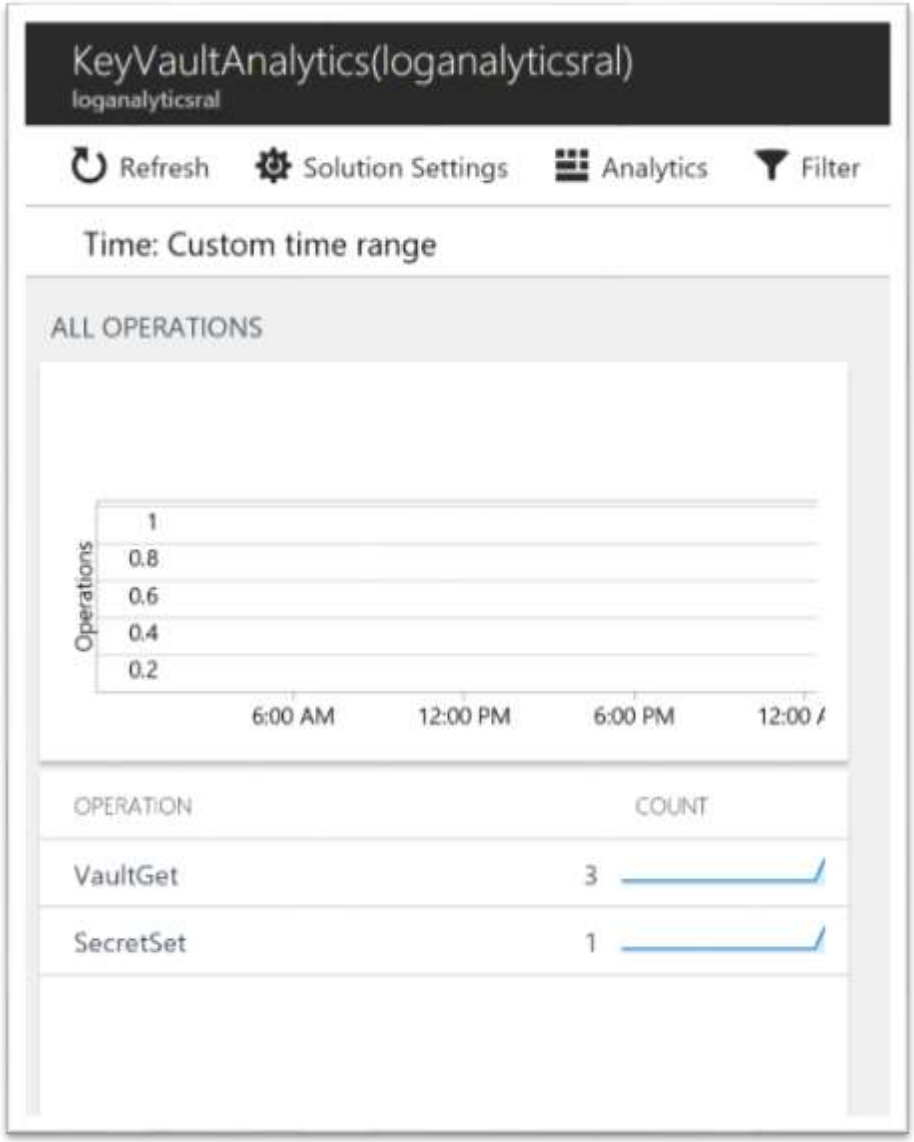
- Demo




```
#Set up logging
#Create an new storage account for the logs
$sa = New-AzureRmStorageAccount -ResourceGroupName rgdemovaltral -Name keyvaultlogsral
`
-Type Standard_LRS -Location 'NorthEurope'
#Get the keyvault we want to turn on auditing on
$kv = Get-AzureRmKeyVault -VaultName 'demoVaultral'
#Turn on logging - retention 30 days
Set-AzureRmDiagnosticSetting -ResourceId $kv.ResourceId -StorageAccountId $sa.Id `
-Enabled $true -Categories AuditEvent -RetentionEnabled $true -RetentionInDays 30
#Test the log by querying the keyvault
Get-AzureKeyVaultSecret -VaultName 'demoVaultral' -Name "ConnectionString"
```

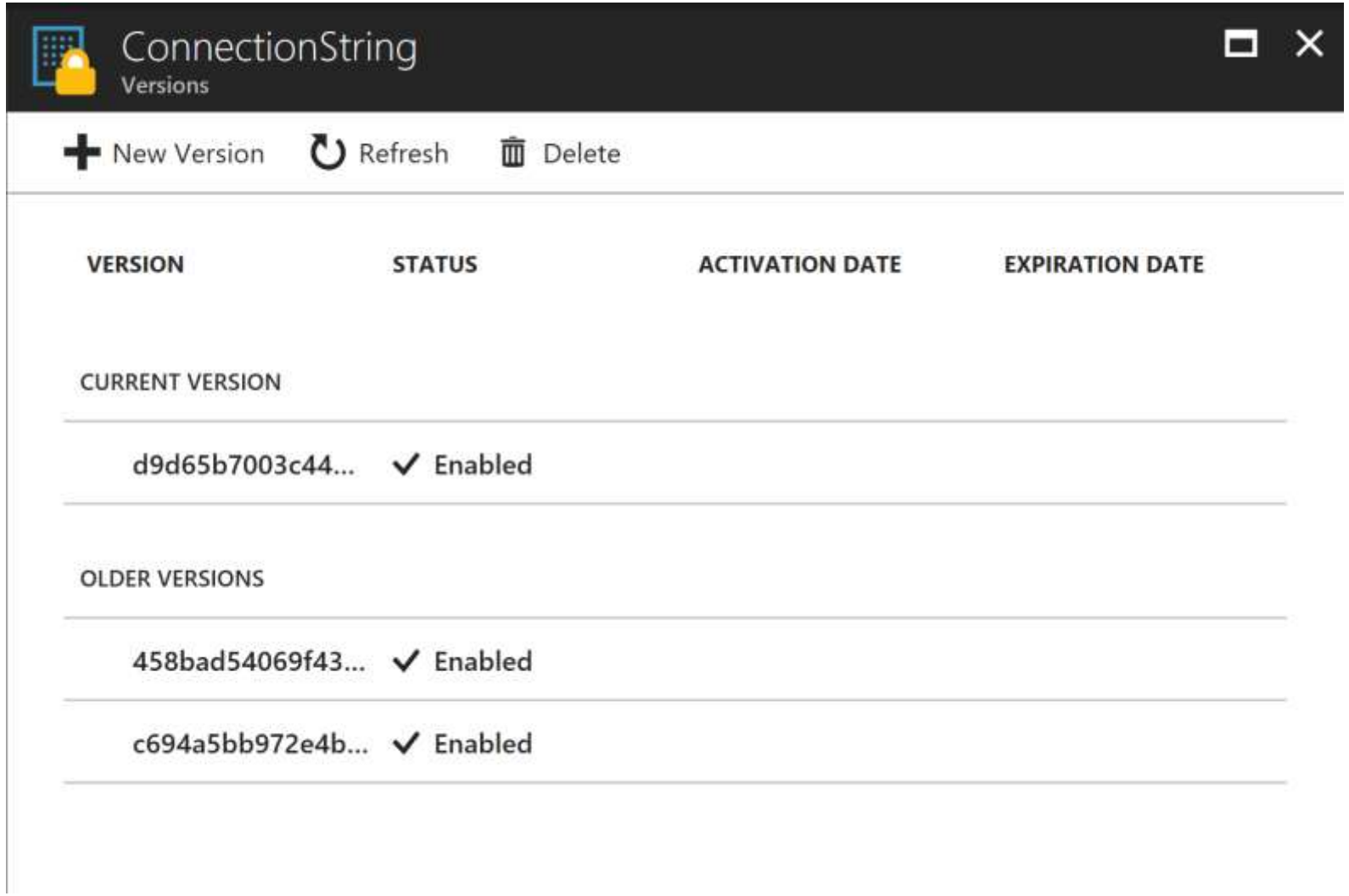
Log Analytics



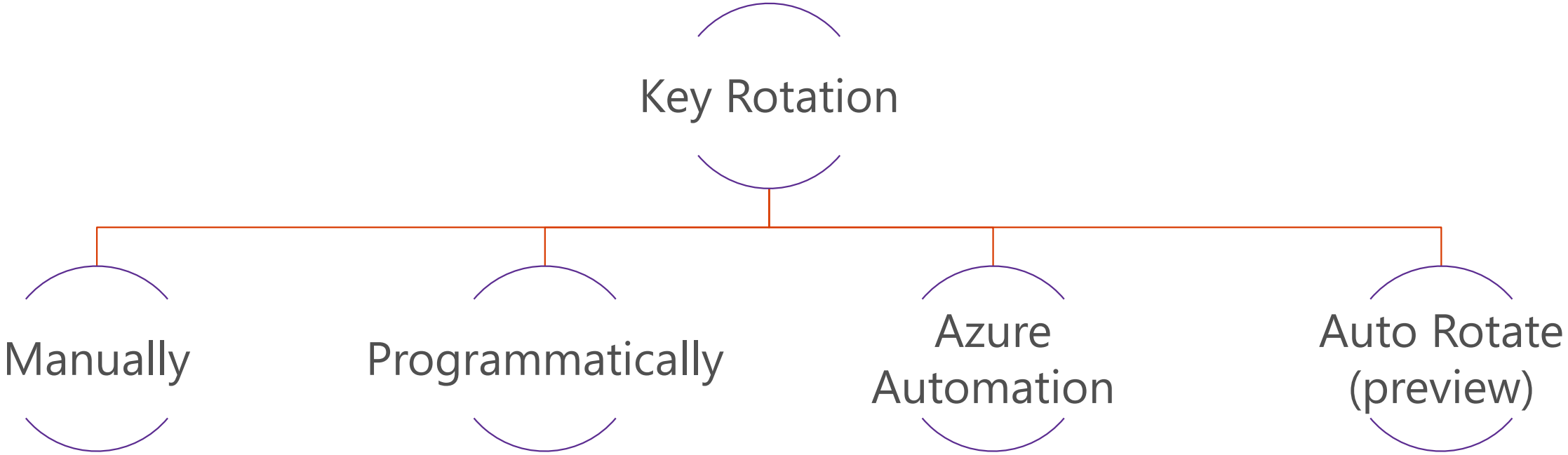


Key Vault Versioning

- Demo
- Encrypt
  - Should always be the latest version
- Decrypt
  - Should be the version that was used to encrypt the version of the data



Key Vault


Key Rotation
Manually
Programmatically
Azure Automation
Auto Rotate (preview)

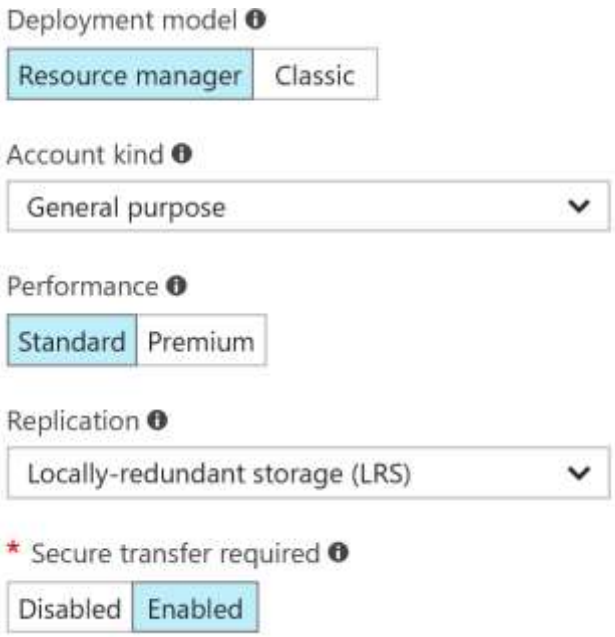
Storage

Blob Storage

- Account keys 512-bit strings
- Primary and secondary keys for key rotation
- Manual Key Vault rotation
  - App is using key 1
  - Regenerate key 2
  - Swap app to use key 2
  - Regenerate key 1
- Or use key vault for key rotation

Blob Storage

- Shared Access Signatures (SAS)
  - Give the client just the permissions they need for a limited amount of time
  - Can revoke without affecting anyone else
  - Requests can be restricted to an IP address and can be restricted to https
- Secure Transfer Required
  - Enforced https connections



Blob Storage Encryption Options

Azure Disk Encryption

- VM disks encrypted
- Windows – Bitlocker
- Linux - DM-Crypt
  - RHEL 7.2
  - CentOS 7.2n
  - Ubuntu 16.04.
- Integration to key vault

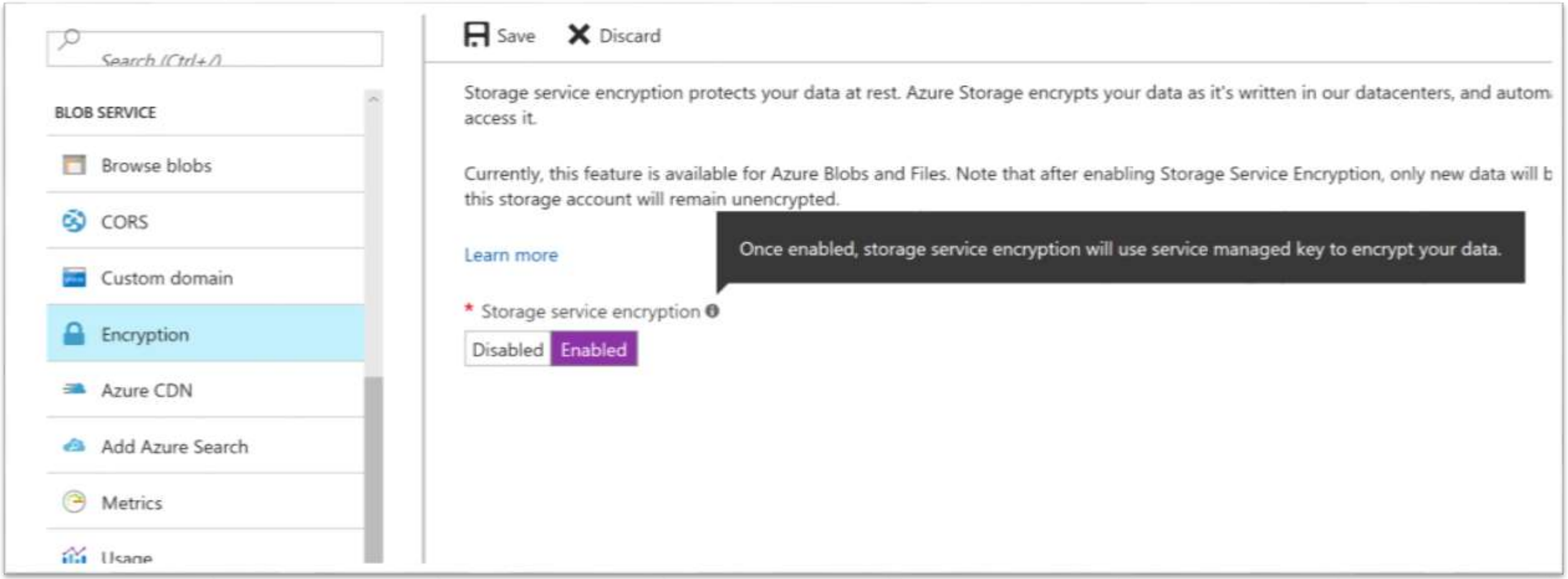
Client-side Encryption

- Encrypt files before they land in blob storage
  - Encrypted on the wire and at rest
- Can use certificates stored in Key Vault

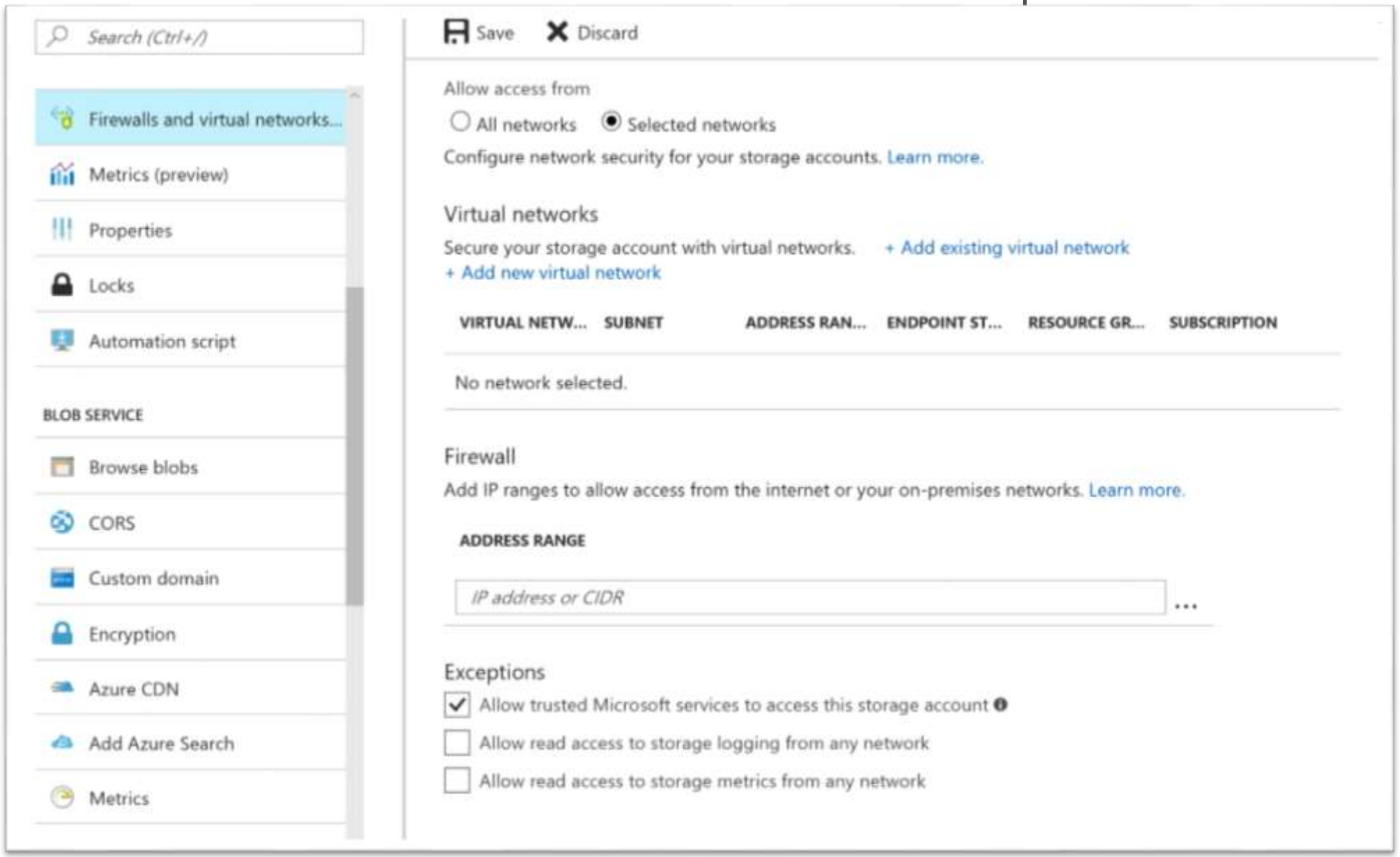
Storage Service Encryption

- Data is automatically encrypted
  - Keys managed by Microsoft (currently)

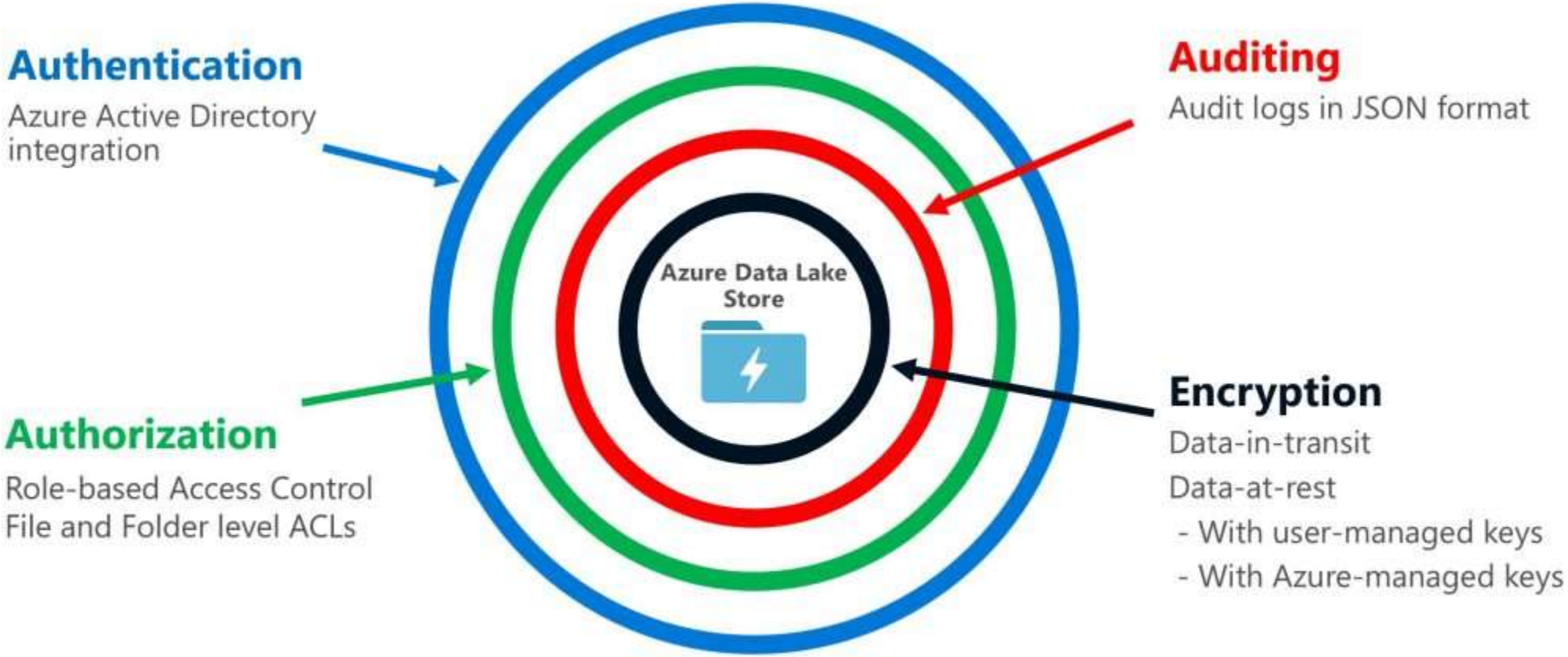
Azure Storage Service Encryption for Data at Rest



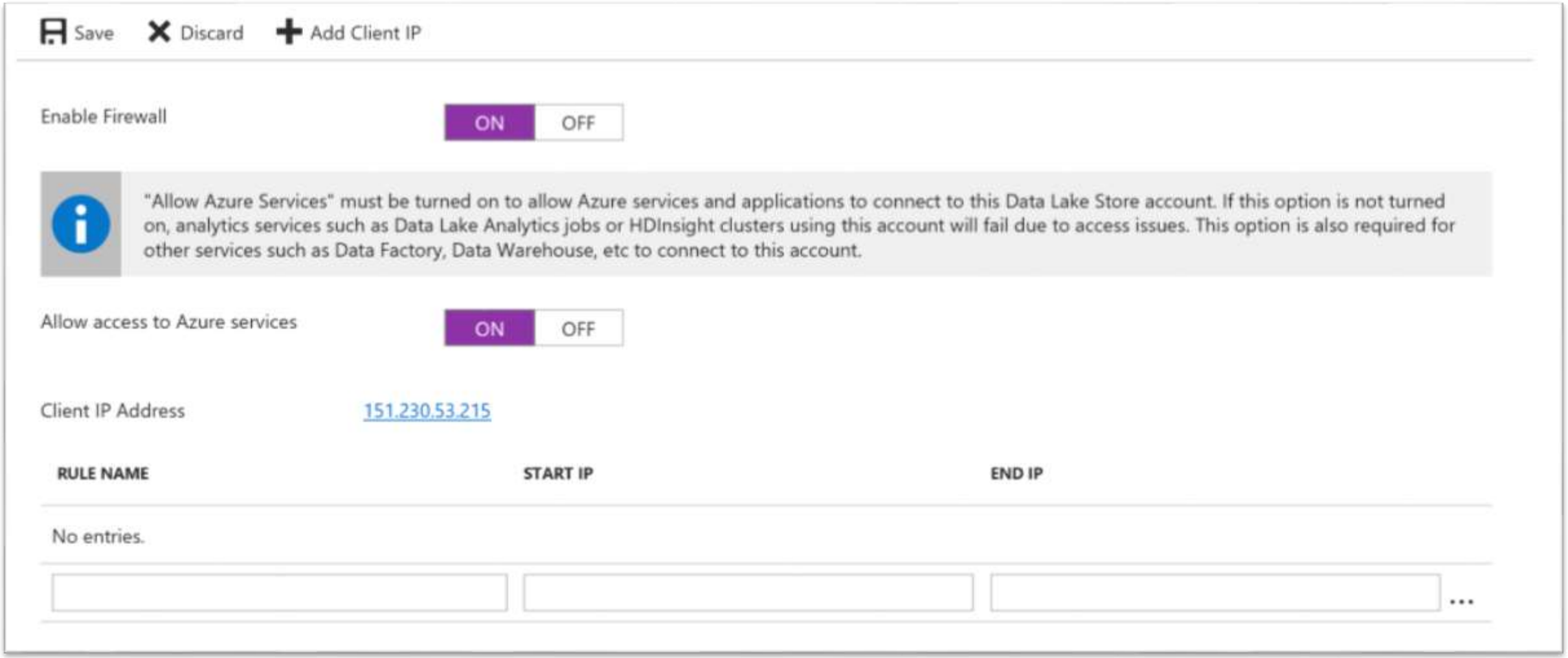
Blob Virtual Network Service Endpoints



Data lake Store - Always encrypted, role-based security, ACLs and Auditing

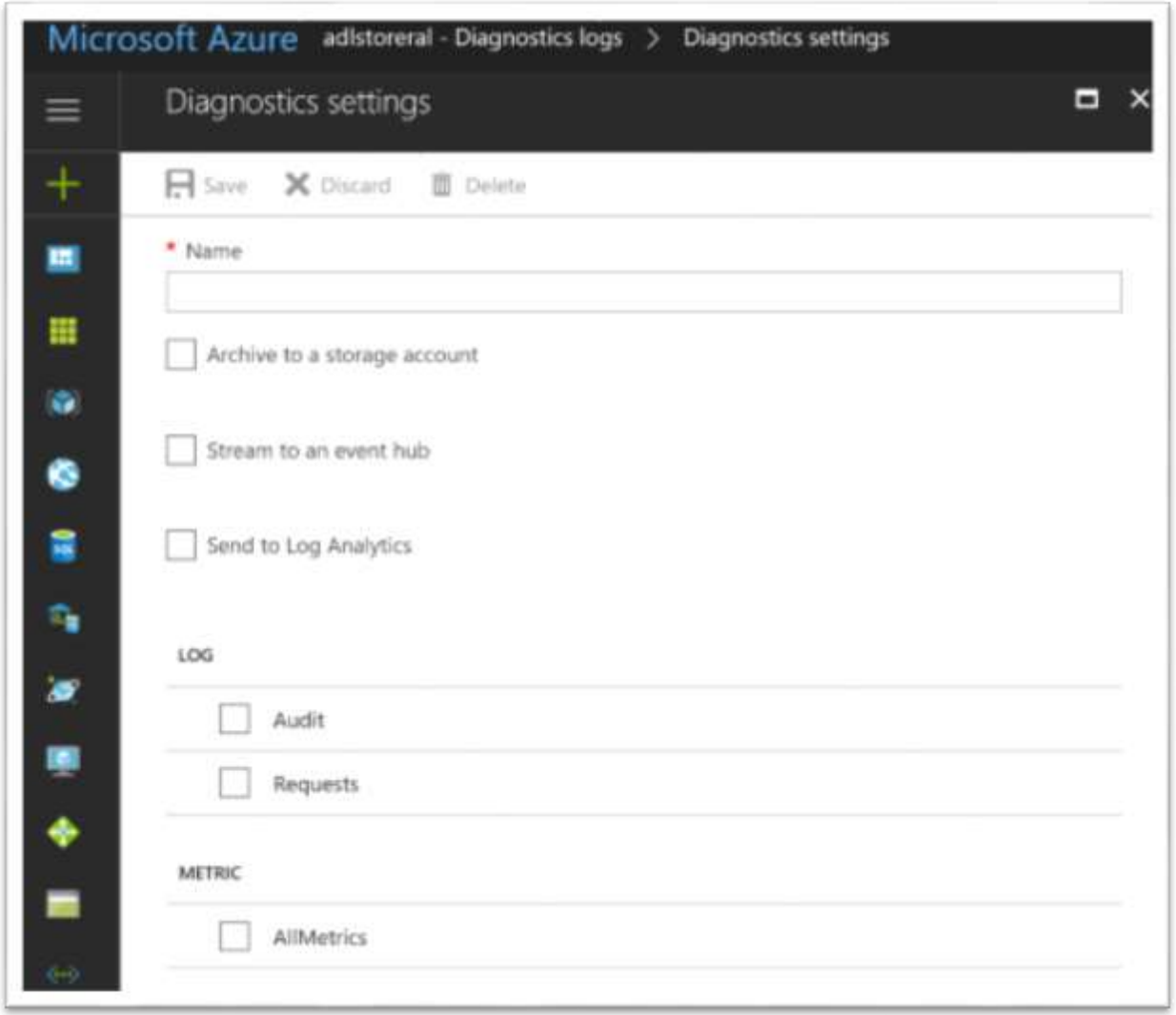


Data Lake Store Firewall



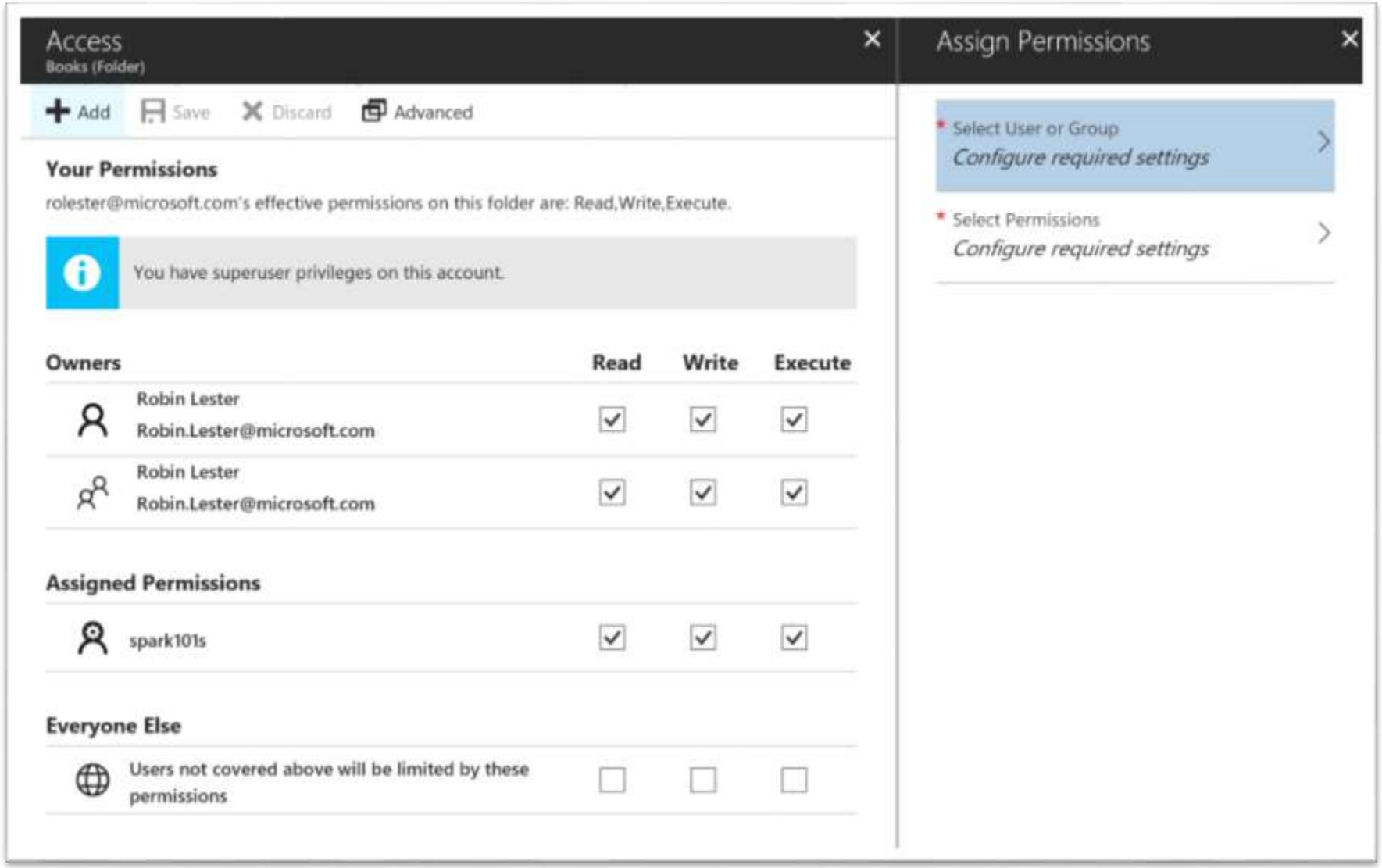
Data lake Store

- Enabling Auditing



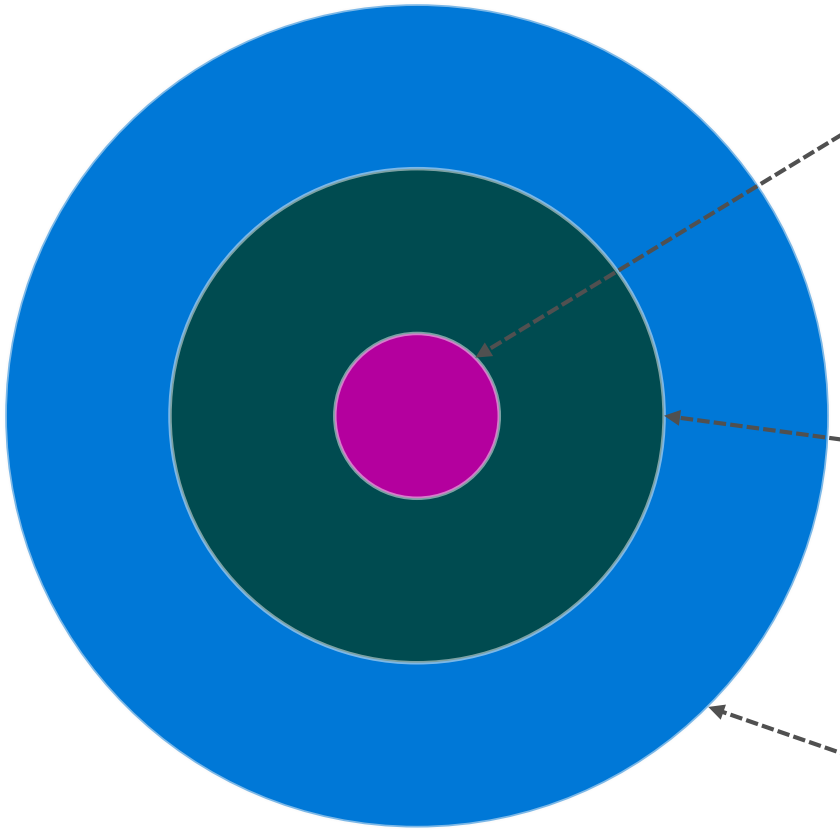
Data lake Store

- Azure AD Integrations



Azure SQL Database

Layered Approach to Security



Securing your database

- Firewall
- User Permissions
- Connection string encryption

Securing your data

- Column Level Encryption

Securing changes to your data

- Application and user activity auditing

SQL Security Landscape

Application Access

Dynamic Data Masking
Limit exposure of sensitive data by masking it to non-privileged users

Row-level Security
Keep multi-tenant databases secure from unauthorized access by other users who share the same tables

Proactive Monitoring

Auditing
Track database activity and log directly to your Azure storage account

SQL Threat Detection
Receive alerts on anomalous database activity in the form of common database threats

Database Access

SQL Authentication
SQL built-in identity management and server authentication (aka SQL logins)

SQL Permissions
SQL agent roles that enable management of permissions, e.g. for data selection and modification

SQL Firewall
Prevention of access to the SQL server unless the IP address of the SQL client is specified

Azure Active Direct. Authentication
Azure central identity management, which combats proliferation of identities

Data Encryption

Encryption in flight
Encrypt data that is transmitted across the network, protecting against snooping & man-in-the-middle attacks

Encryption-at-rest (TDE)
Server-side encryption of the database content on physical storage, protecting against offline media attacks

Always Encrypted
Client-side encryption of sensitive data using keys that are never revealed to the database system

Cloud-only

Transparent Data Encryption

Protect sensitive data stored in a SQL database from unauthorized access

Encrypted at rest, in flight, and while in use

SQL Server does not have the keys (nor does it need the keys)

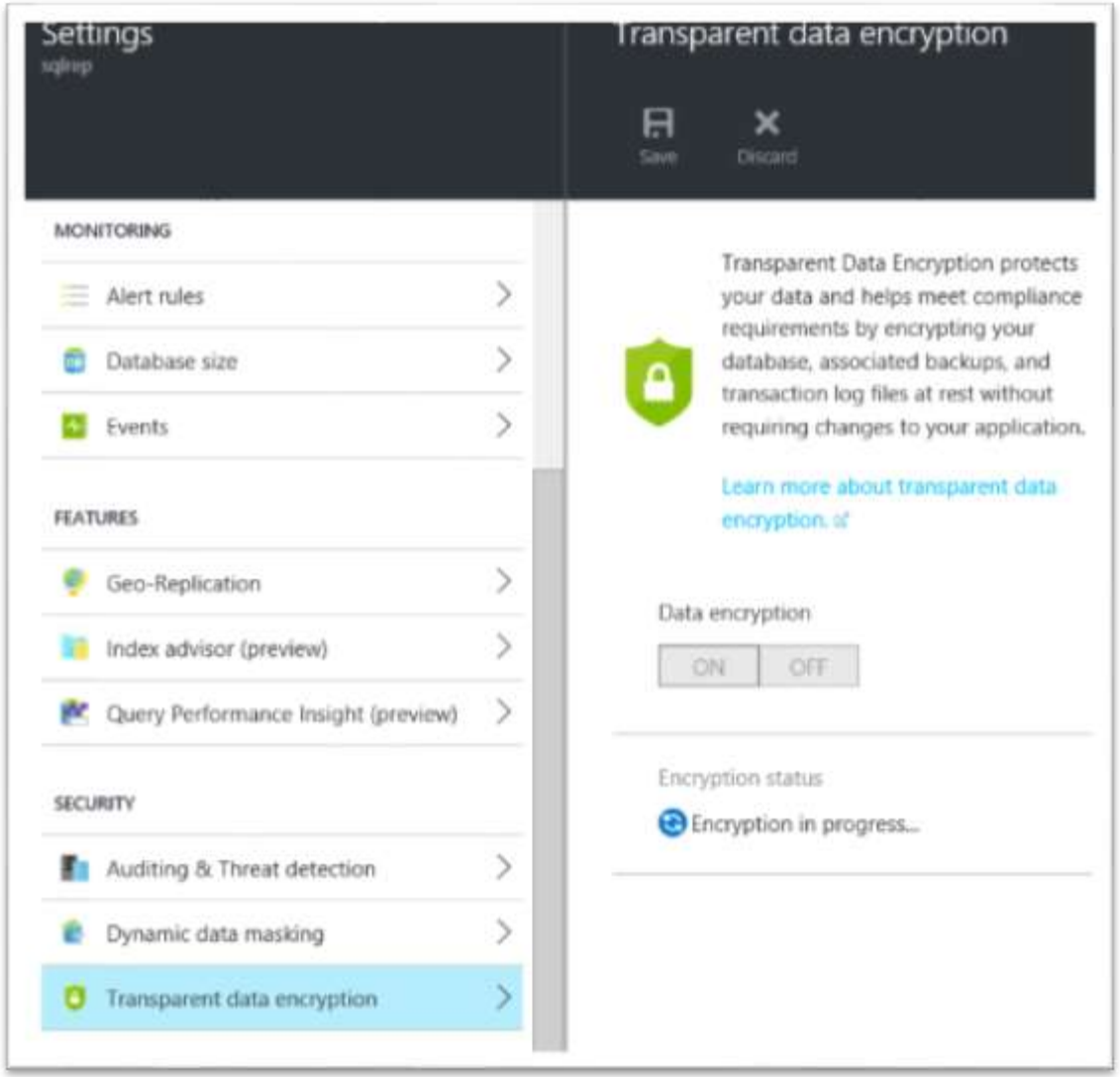
Keep application changes to a minimum

Encryption/decryption of data done transparently in TCE-enabled client driver

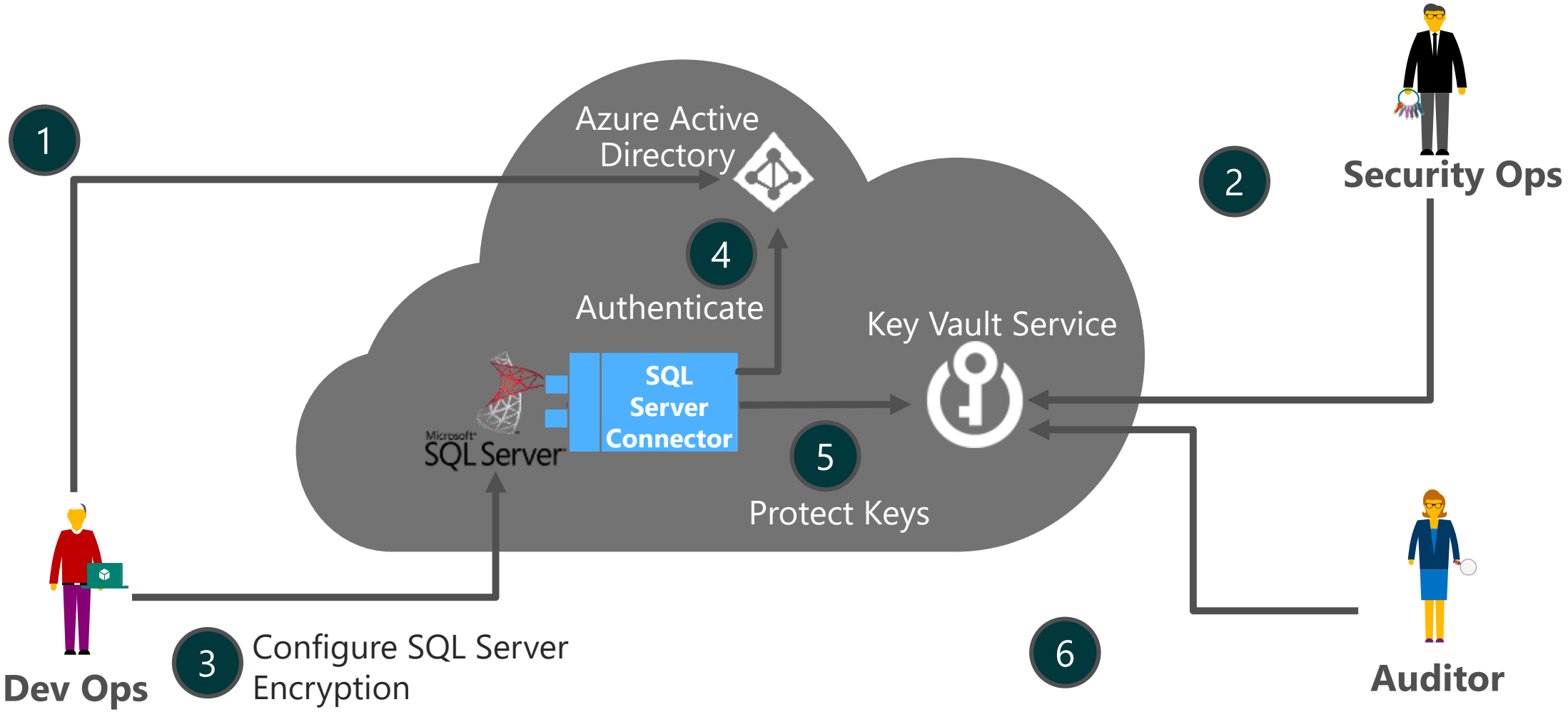
Support for equality operations (include joins) on encrypted data

Azure manages encryption keys

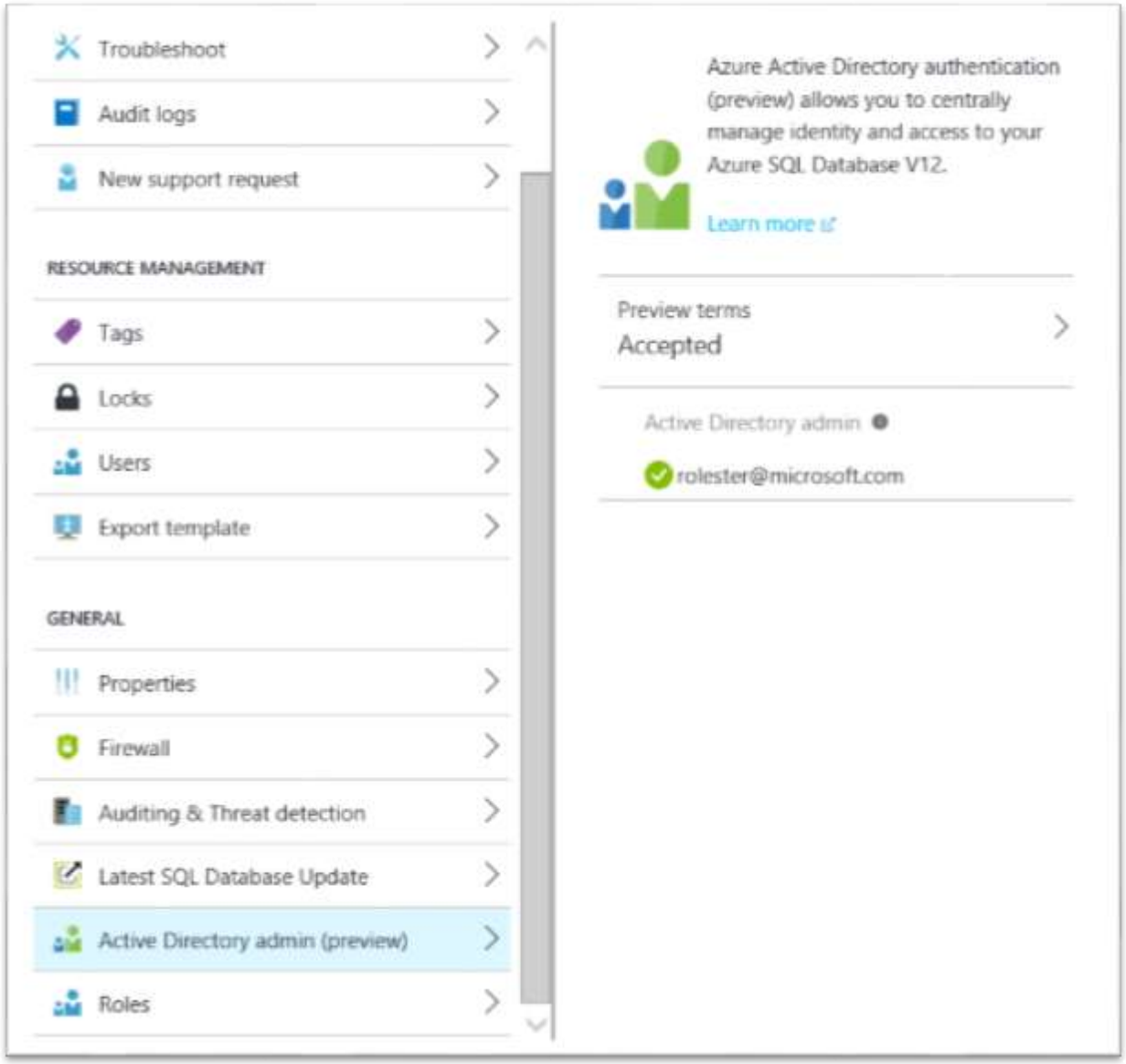
TDE

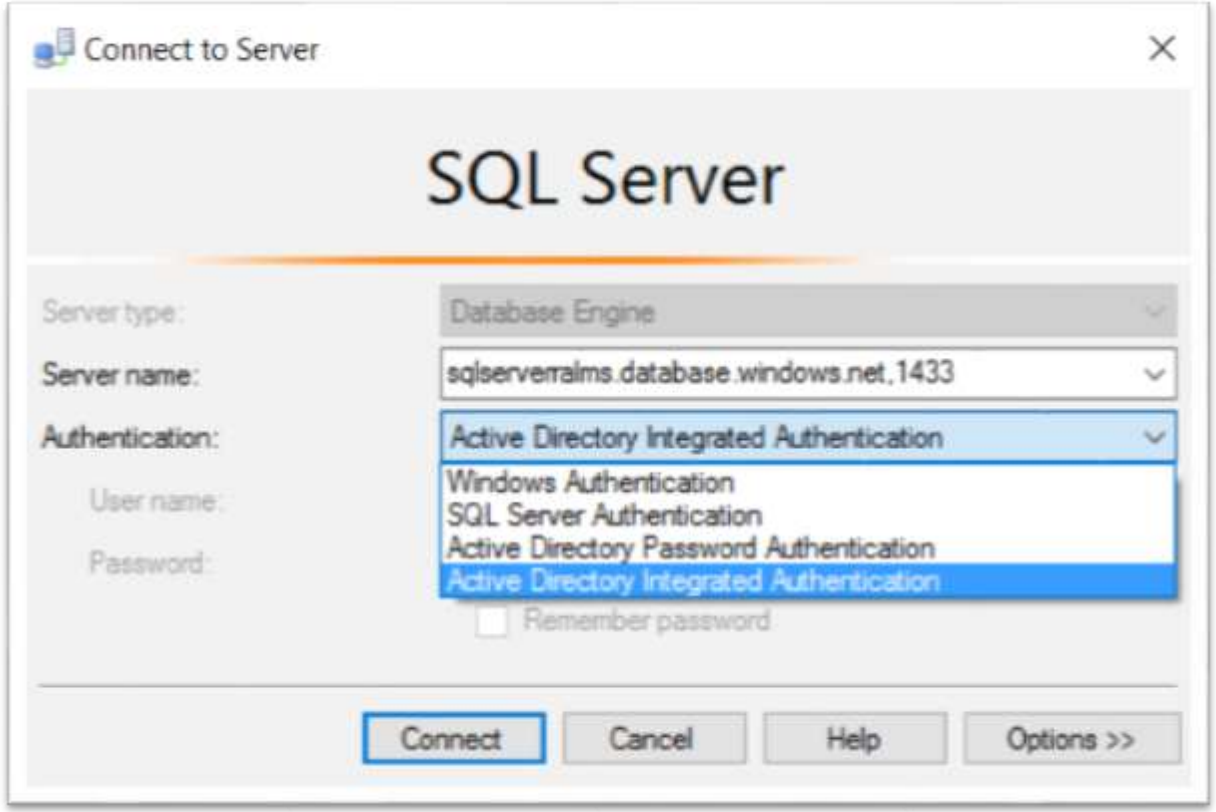

Settings
Transparent data encryption
Save
Discard
MONITORING
Alert rules
Database size
Events
FEATURES
Geo-Replication
Index advisor (preview)
Query Performance Insight (preview)
SECURITY
Auditing & Threat detection
Dynamic data masking
Transparent data encryption
Transparent Data Encryption protects your data and helps meet compliance requirements by encrypting your database, associated backups, and transaction log files at rest without requiring changes to your application.
Learn more about transparent data encryption.
Data encryption
ON
OFF
Encryption status
Encryption in progress...

Enable TDE with customer-managed keys

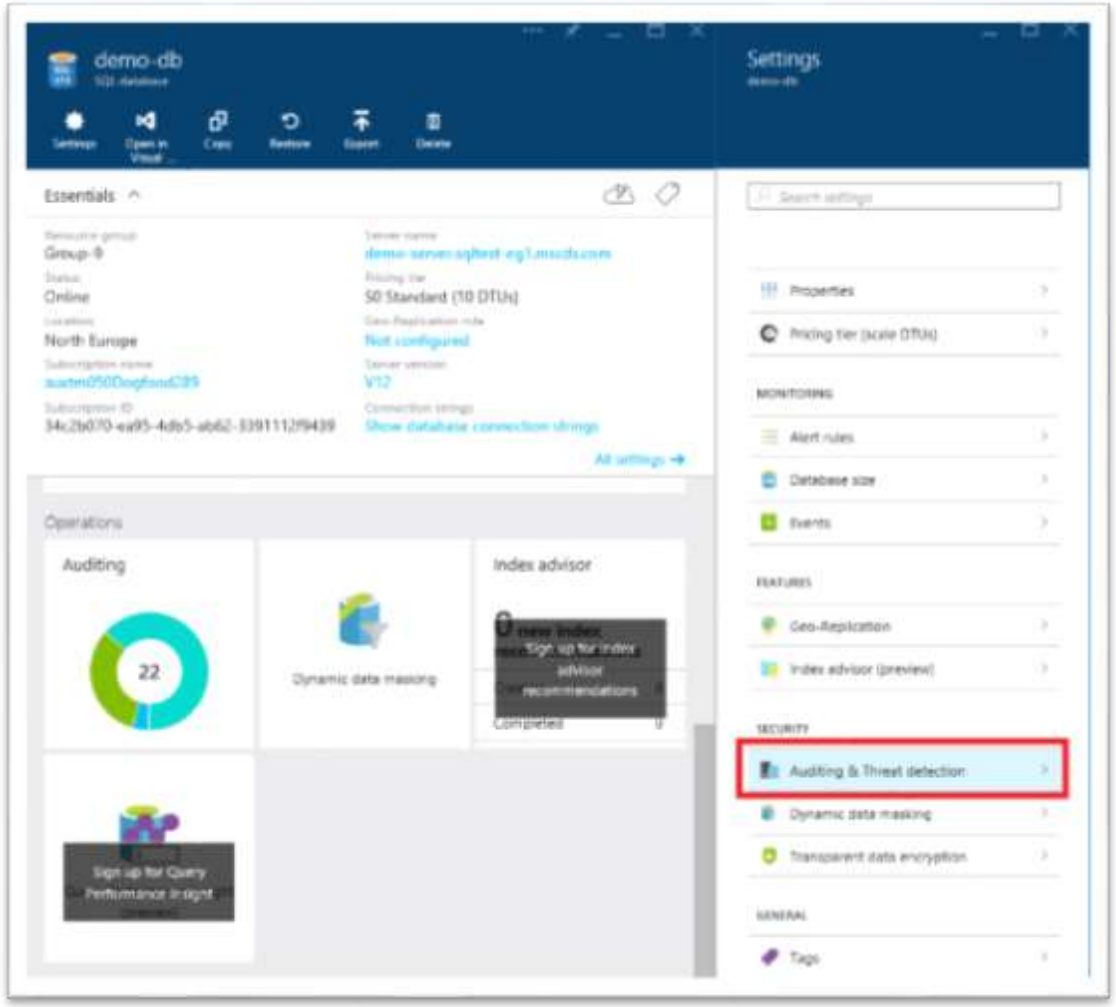


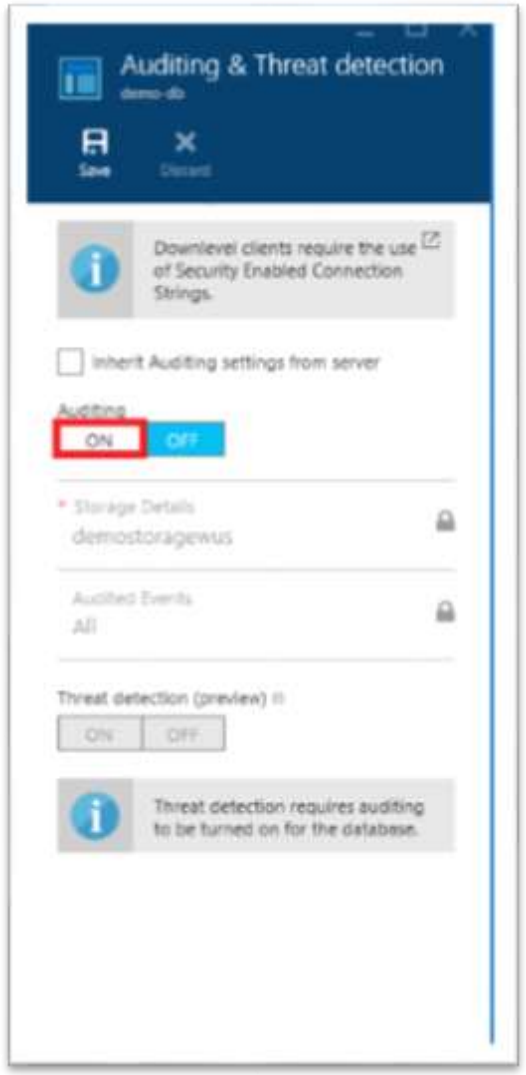
Azure Active Directory Integration

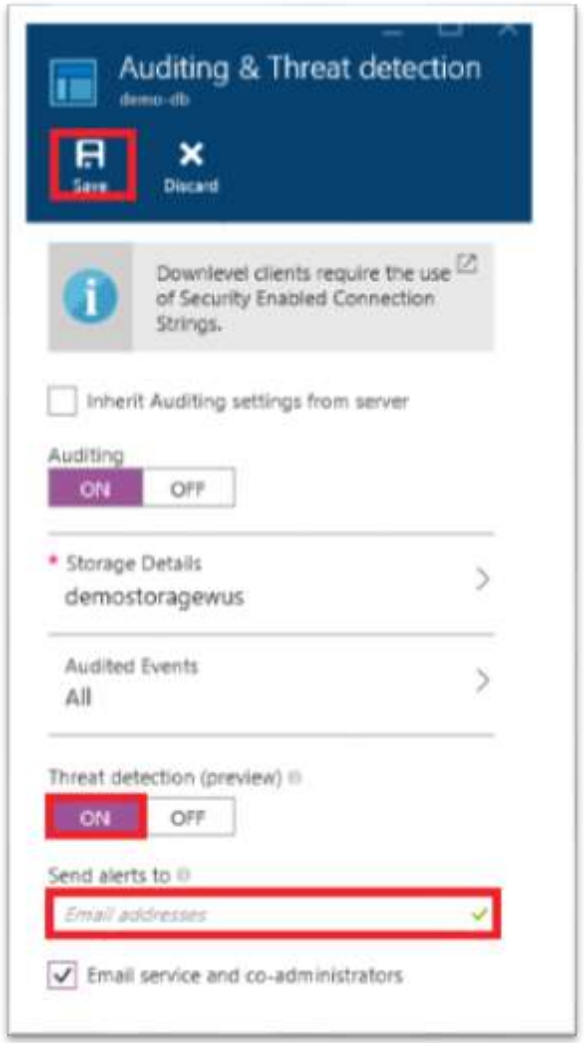




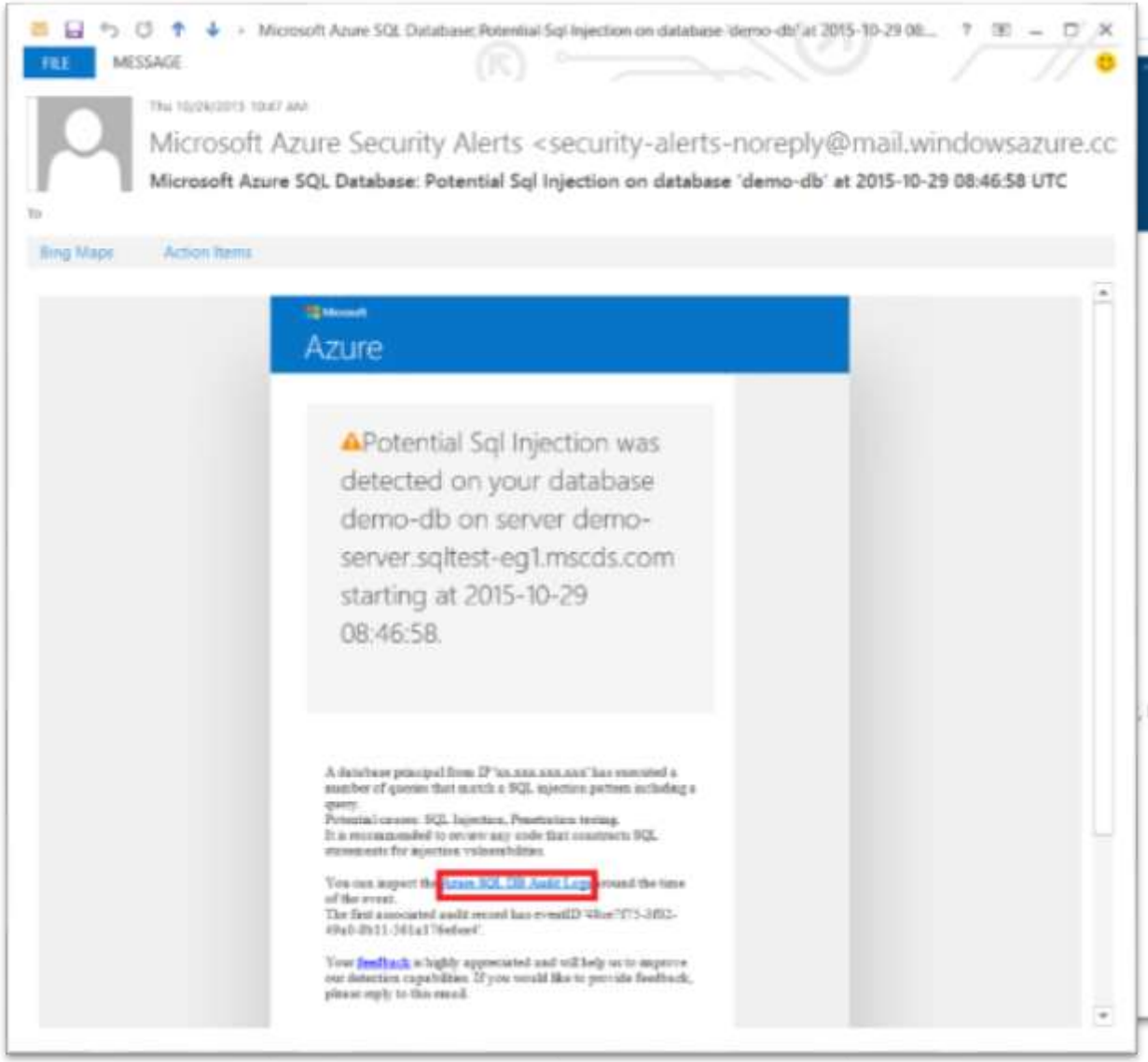
Threat Detection

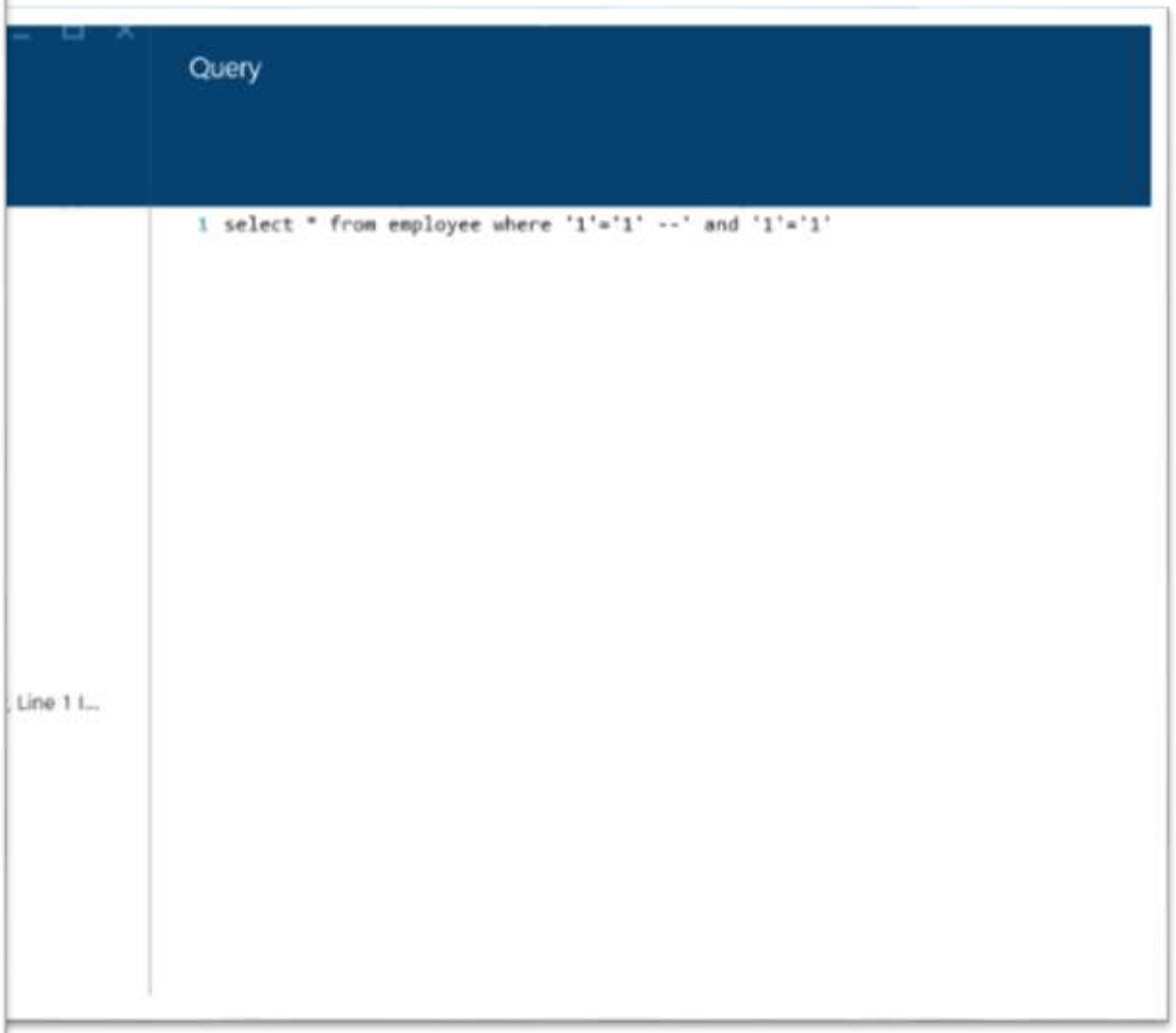






Threat Detection





Auditing

Gain insight into database events and streamline compliance-related tasks

Configurable to track and log database activity

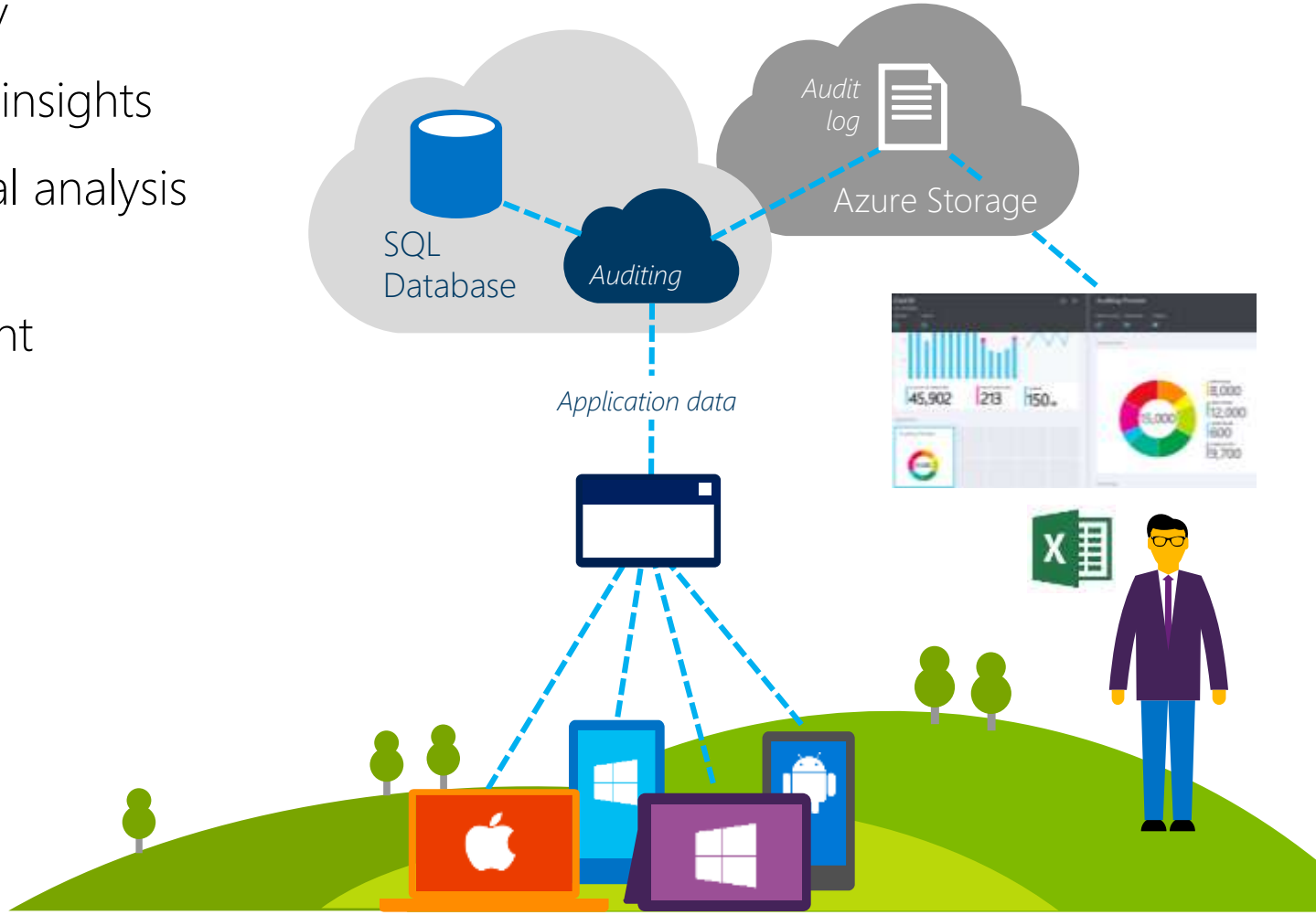
Dashboard views in the portal for at-a-glance insights

Pre-defined Power View reports for deep visual analysis on Audit log data

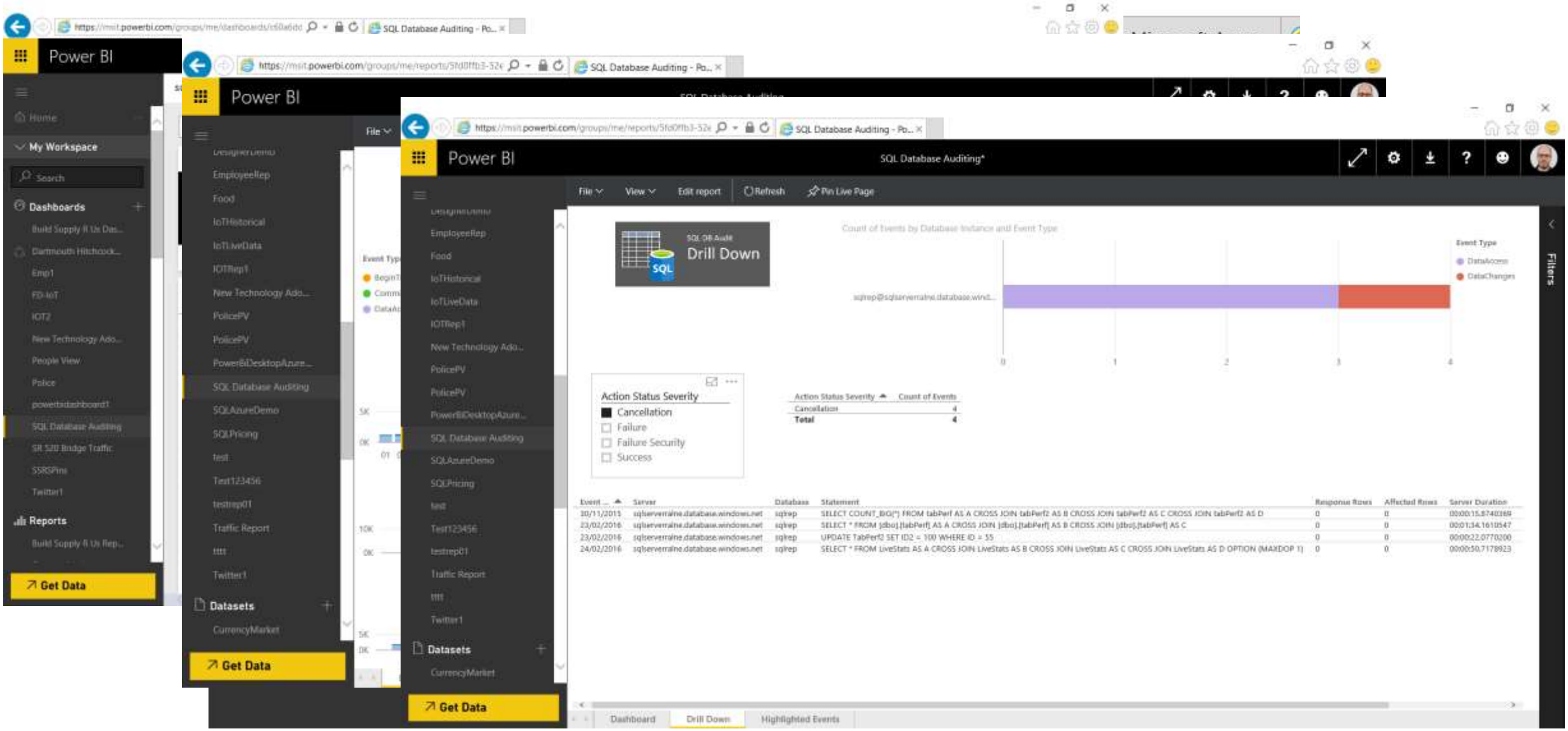
Audit logs reside in your Azure Storage account

Available in Basic, Standard, and Premium

Access via the new Azure preview portal

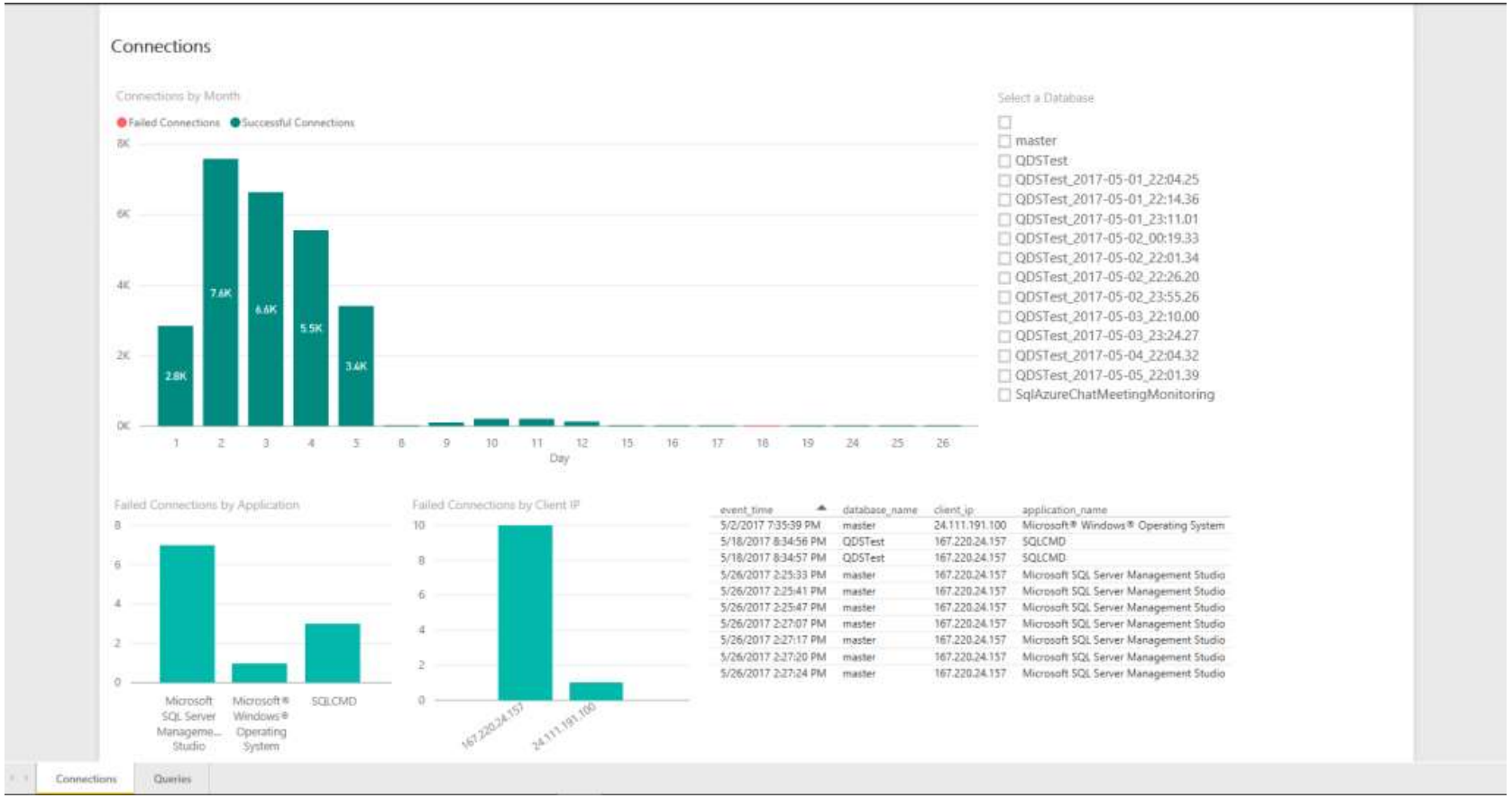


Viewing Audit Logs (only table storage – Deprecated this month)



Audit Logs on blob storage

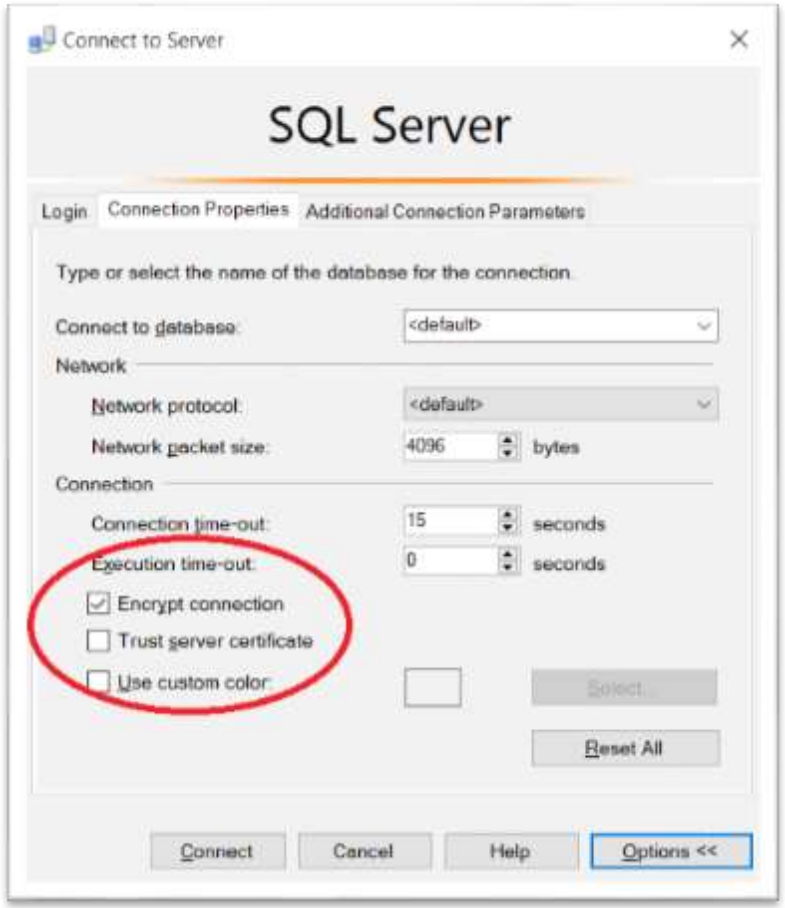
https://blogs.msdn.microsoft.com/azuresqldbsupport/2017/05/26/sql-azure-blob-auditing-basic-power-bi-dashboard/



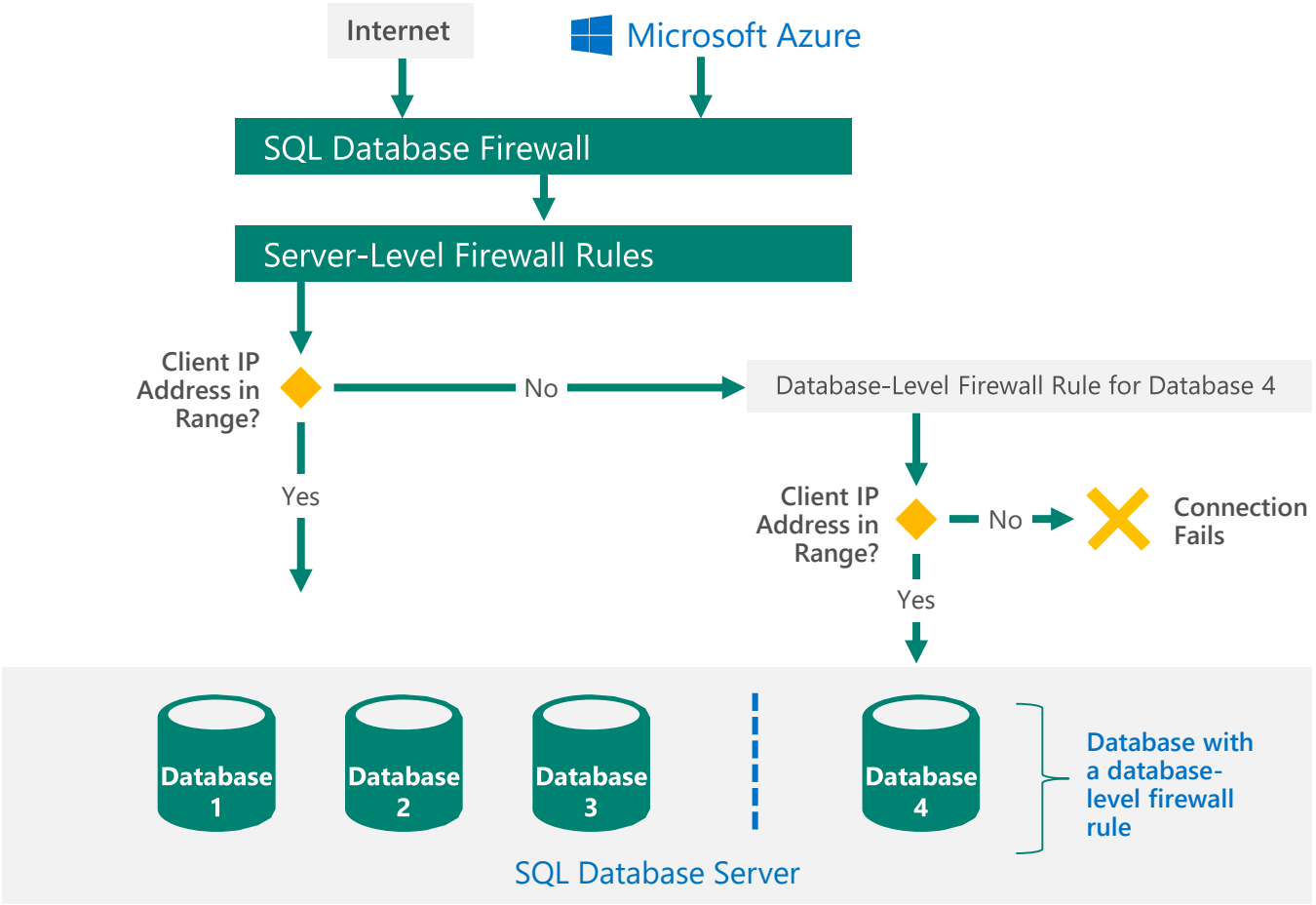
Demo

SQL Network Connections

- Azure DB Firewall
- Site to Site VPN
- Express route
- Connection is always encrypted
- Valid certificate protects against man-in-the-middle attacks (only if connection set to not trust server certificate)



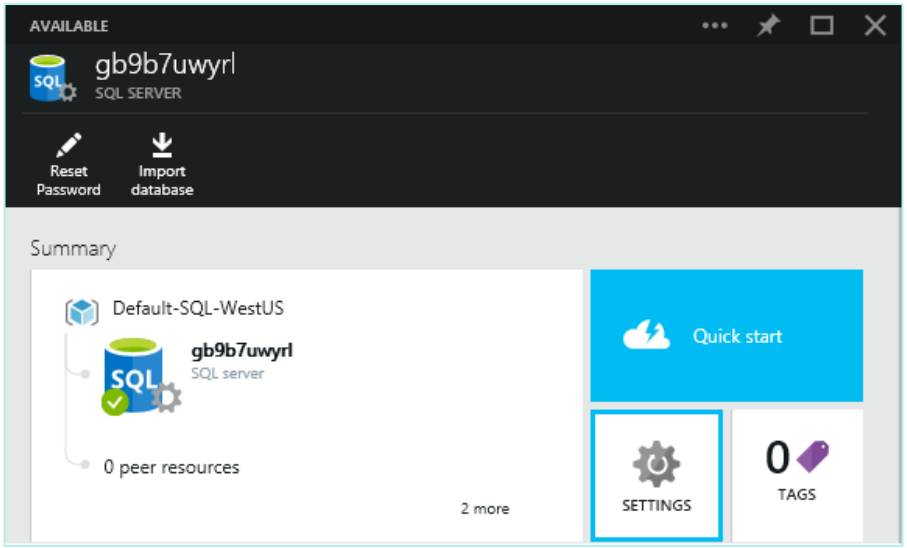
SQL Database Firewalls

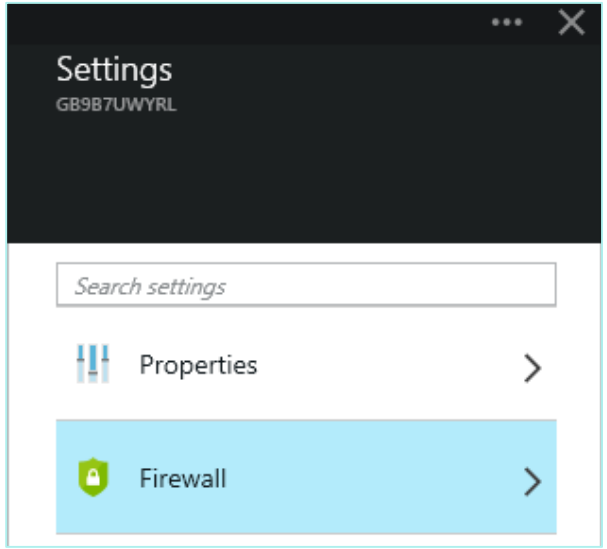


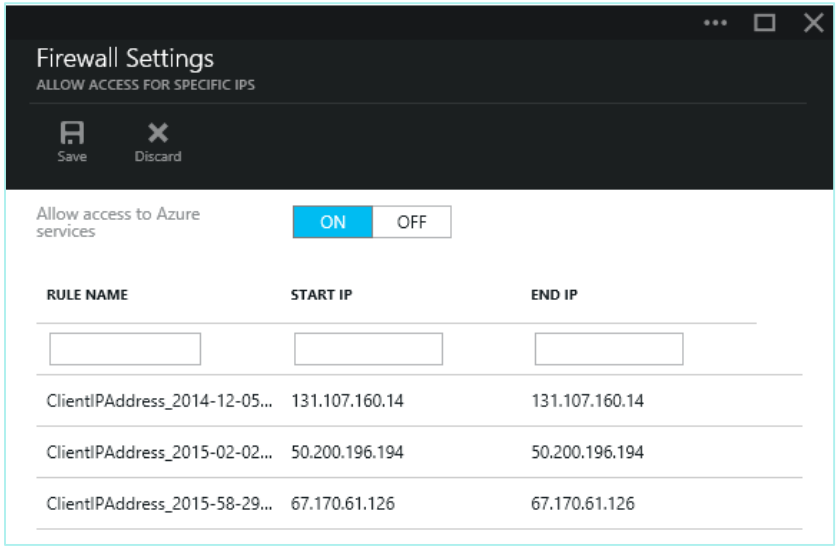
Windows Azure Platform

Demo

Firewall configuration using portals







- By default, Azure blocks all external connections to port 1433.

Azure SQL Database – Connecting to other services

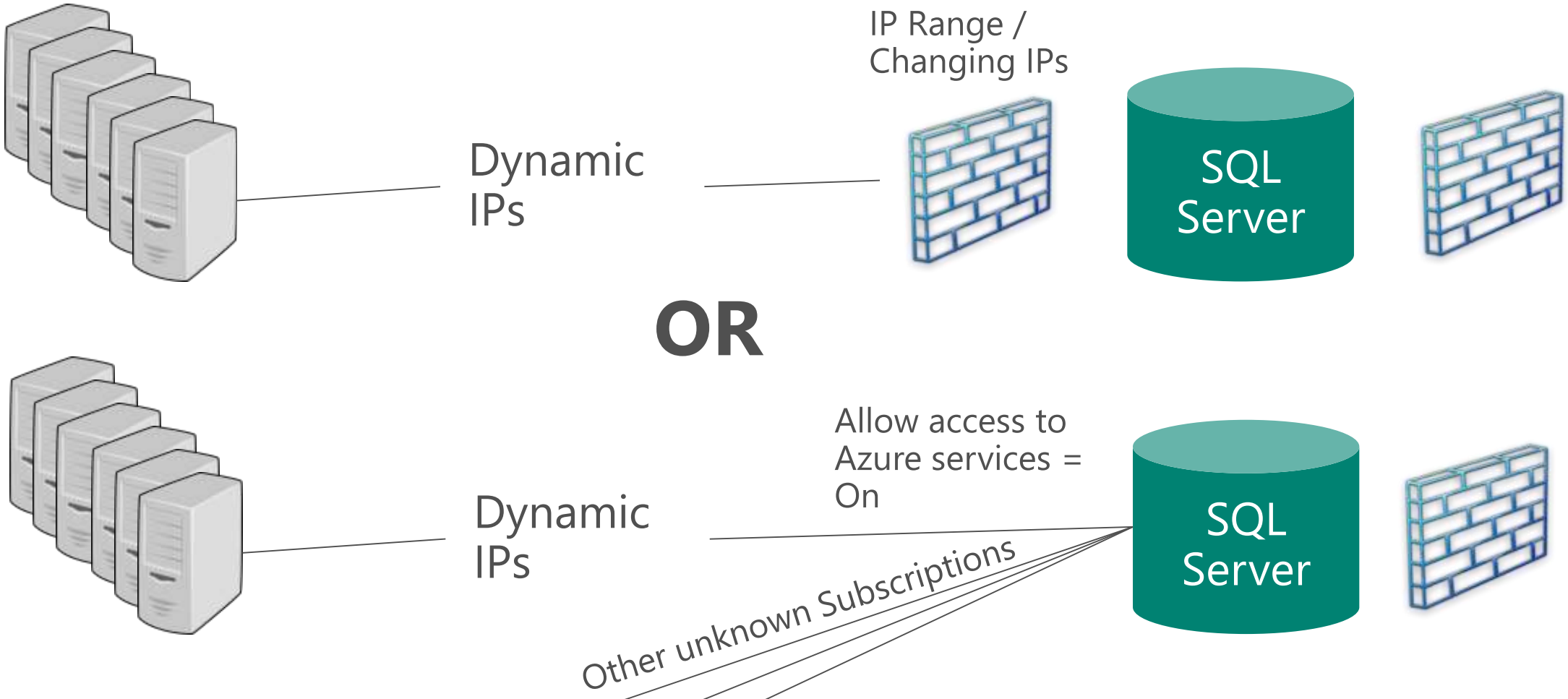
Allow access to Azure services

On = **All** Azure IPs
and **All** Subnets

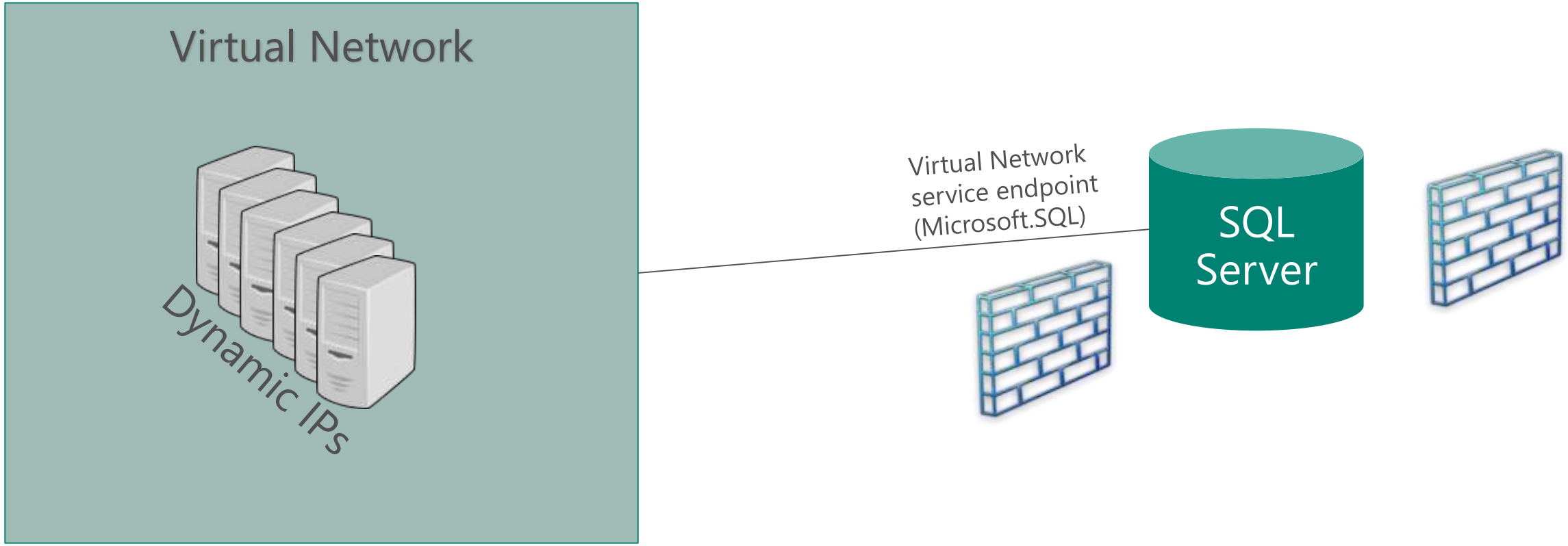
Off = **No** Azure IPs
and **No** Subnets

SQL Database Networking

- VMs in Azure generally have dynamic IPs (Static are expensive at scale and management requirement)



SQL Database Virtual Network service endpoints



SQL Database Virtual Network service endpoints

- Considerations
  - Server level only
  - Azure Resource Manager only
  - Site to site VPN and Express route need to have their IPs explicitly stated in the firewall

Row-level security

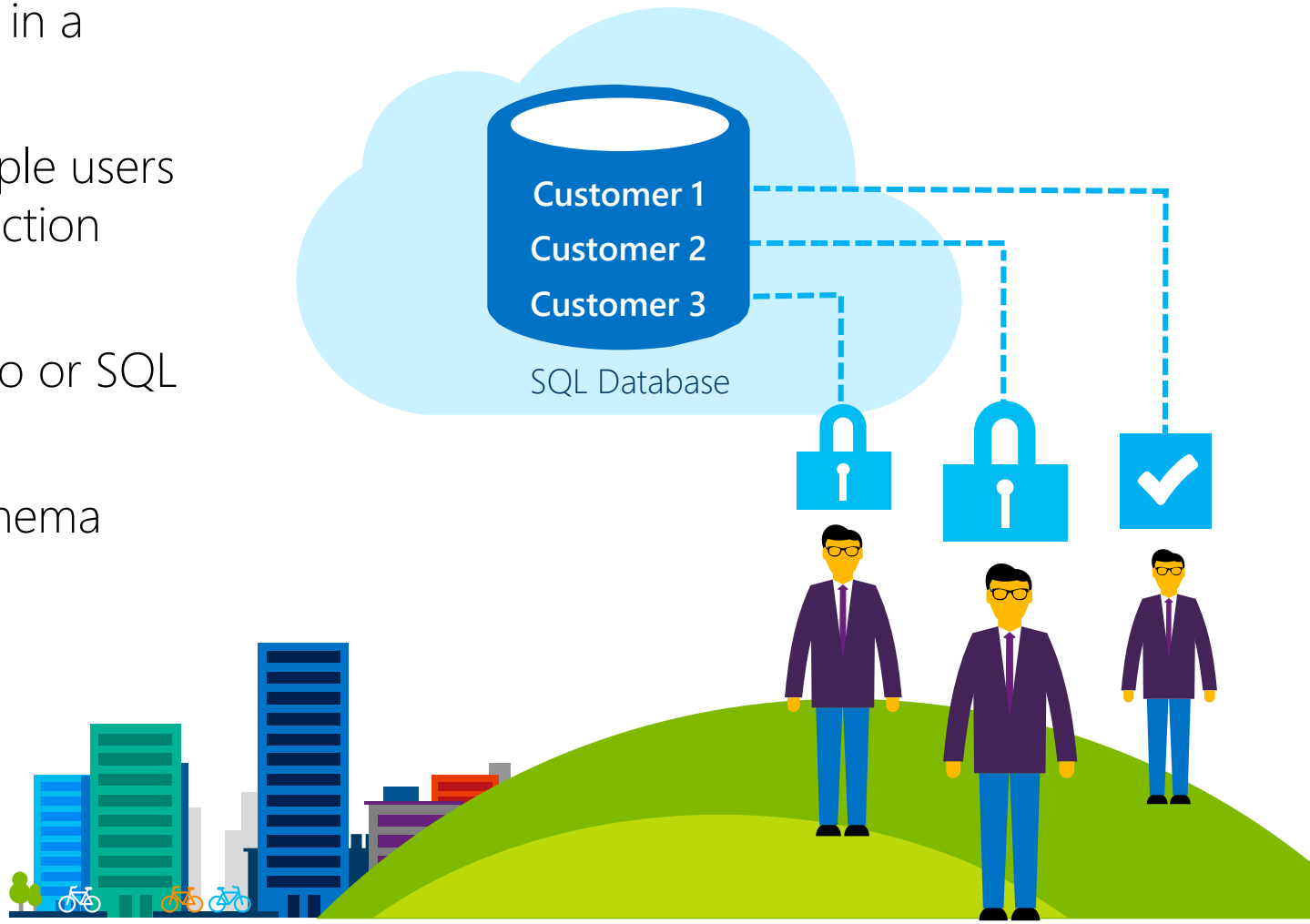
Protect data privacy by ensuring the right access across rows

Fine-grained access control over specific rows in a database table

Help prevent unauthorized access when multiple users share the same tables, or to implement connection filtering in multitenant applications

Administer via SQL Server Management Studio or SQL Server Data Tools

Enforcement logic inside the database and schema bound to the table.



Row Level Security

- Demo

Common RLS Use Cases…

- Traditional RLS workloads
  - Custom business logic to determine which rows each user can SELECT, INSERT, UPDATE, DELETE based on their role, department, security level, etc.
  - Target sectors: Finance, insurance, healthcare, oil/gas, Federal, etc.
- Multi-tenant databases
  - Ensuring tenants can only access their own rows of data in a shared database, with enforcement logic in the database rather than in the app tier
  - E.g. multi-tenant shards with elastic database tools on Azure SQL Database
- Reporting, analytics, data warehousing
  - Different users access same database through various reporting tools, and work with different subsets of data based on their identity/role

Dynamic Data Masking

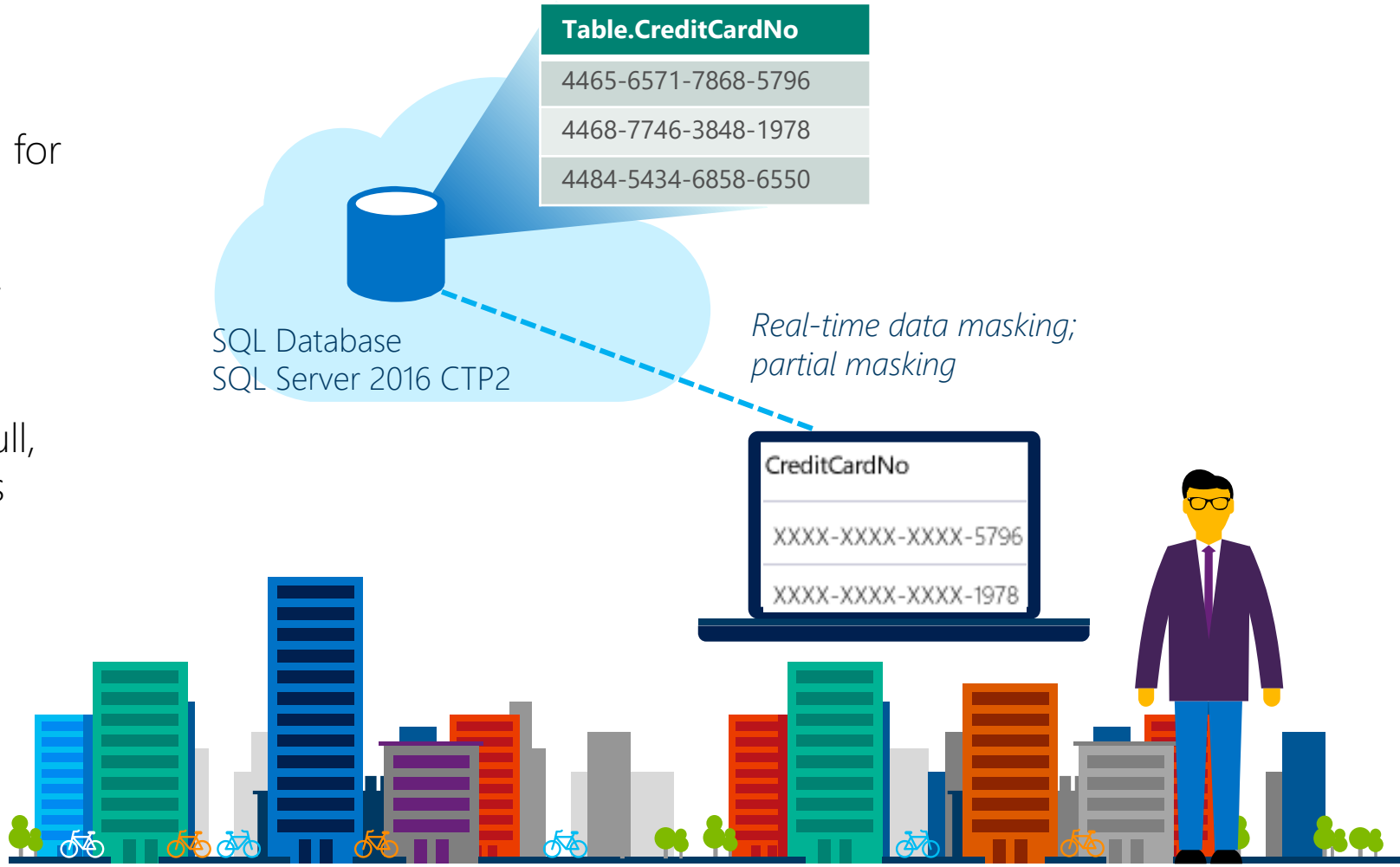
Prevent the abuse of sensitive data by hiding it from users

Configuration made easy in the new Azure portal

Policy-driven at the table and column level, for a defined set of users

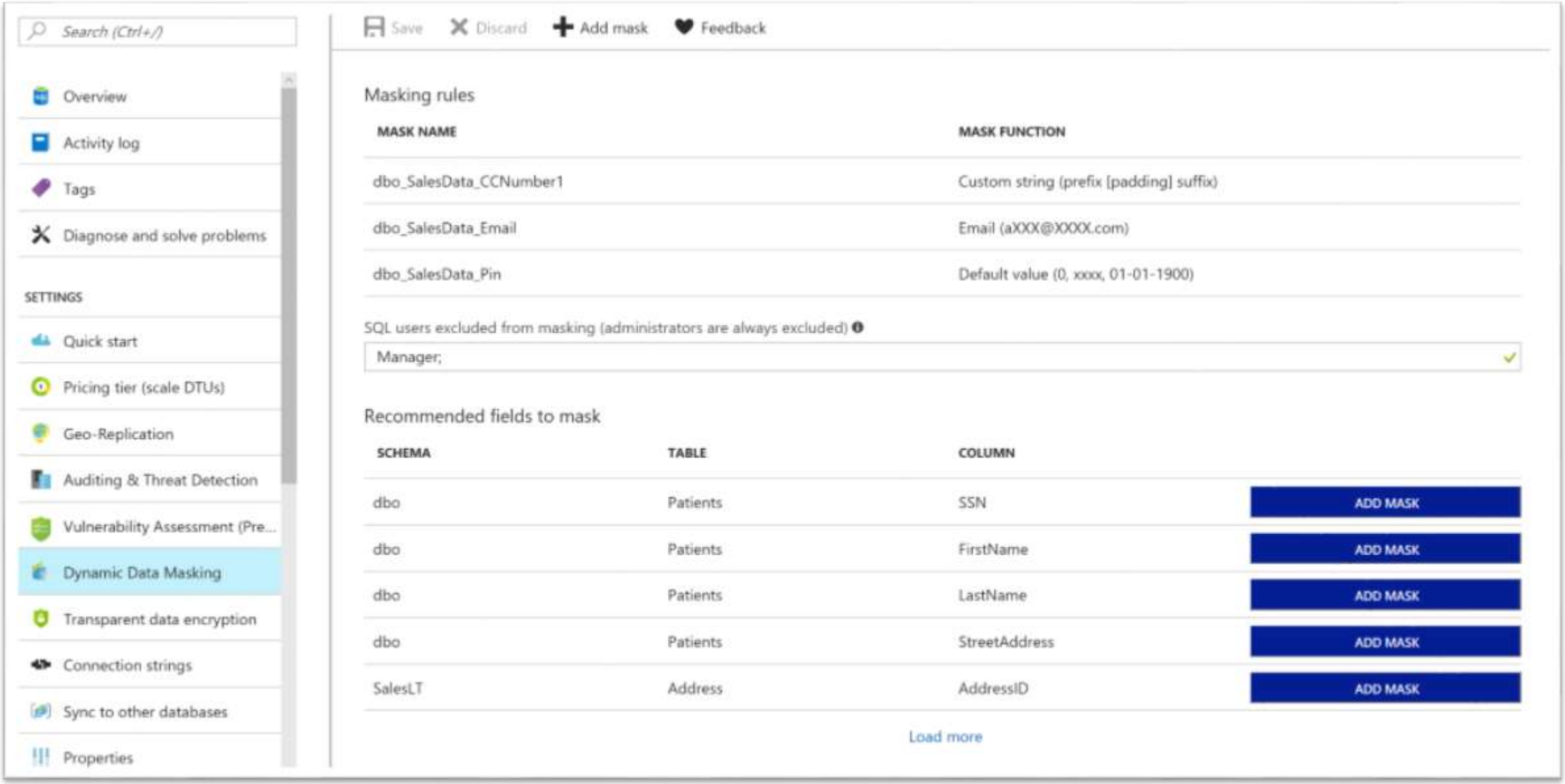
Data masking applied in real-time to query results based on policy

Multiple masking functions available (e.g. full, partial) for various sensitive data categories (e.g. Credit Card Numbers, SSN, etc.)



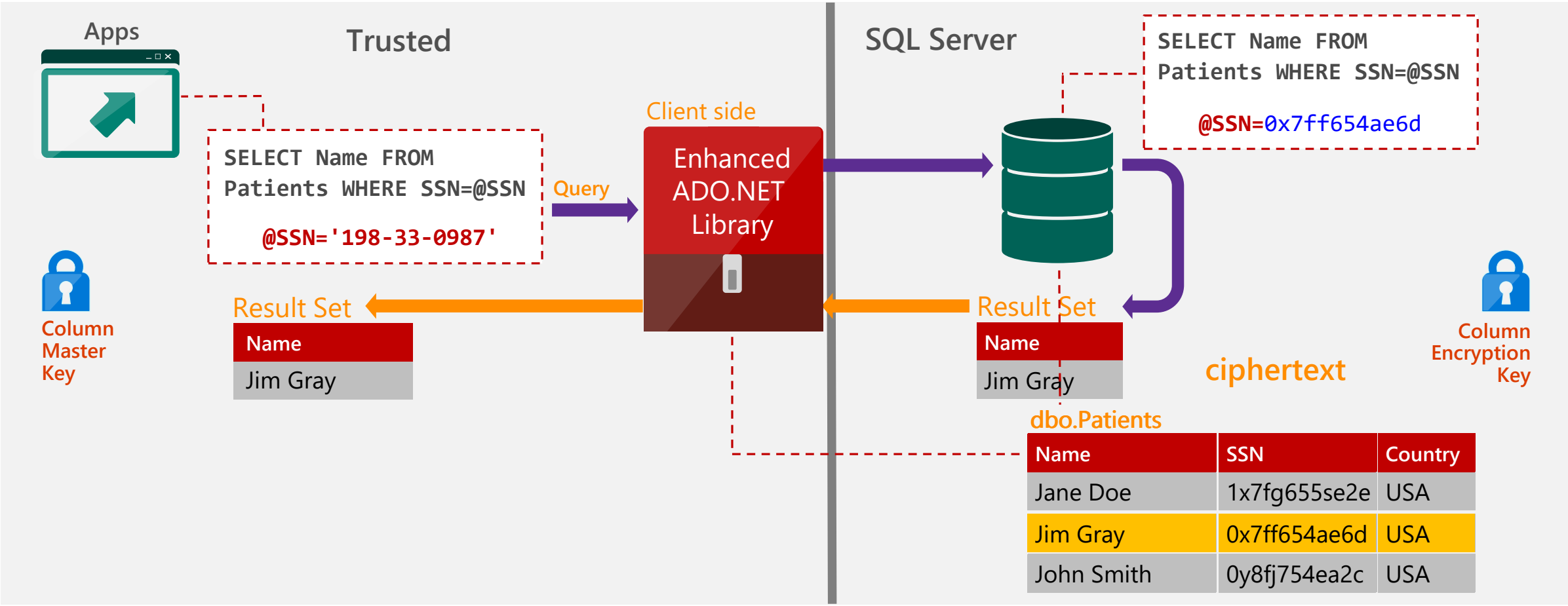
Dynamic Data Masking

- Demo


Search (Ctrl+/)
Overview
Activity log
Tags
Diagnose and solve problems
SETTINGS
Quick start
Pricing tier (scale DTUs)
Geo-Replication
Auditing & Threat Detection
Vulnerability Assessment (Pre...
Dynamic Data Masking
Transparent data encryption
Connection strings
Sync to other databases
Properties
Save
Discard
Add mask
Feedback
Masking rules
MASK NAME
MASK FUNCTION
dbo_SalesData_CCNumber1
Custom string (prefix [padding] suffix)
dbo_SalesData_Email
Email (aXXX@XXXX.com)
dbo_SalesData_Pin
Default value (0, xxxx, 01-01-1900)
SQL users excluded from masking (administrators are always excluded)
Manager;
Recommended fields to mask
SCHEMA
TABLE
COLUMN
dbo
Patients
SSN
ADD MASK
dbo
Patients
FirstName
ADD MASK
dbo
Patients
LastName
ADD MASK
dbo
Patients
StreetAddress
ADD MASK
SalesLT
Address
AddressID
ADD MASK
Load more

Always Encrypted

Help protect data at rest and in motion, on-premises & cloud



Types of Encryption for Always Encrypted

- Randomized encryption
  - Encrypt('123-45-6789') = 0x17cfd50a
  - Repeat: Encrypt('123-45-6789') = 0x9b1fcf32
  - Allows for transparent retrieval of encrypted data but NO operations
  - More secure
- Deterministic encryption
  - Encrypt('123-45-6789') = 0x85a55d3f
  - Repeat: Encrypt('123-45-6789') = 0x85a55d3f
  - Allows for transparent retrieval of encrypted data AND equality comparison
    - E.g. in WHERE clauses and joins, distinct, group by

Two types of encryption available

Randomized encryption uses a method that encrypts data in a less predictable manner

Deterministic encryption uses a method which always generates the same encrypted value for any given plain text value (for equality excretions)

Security

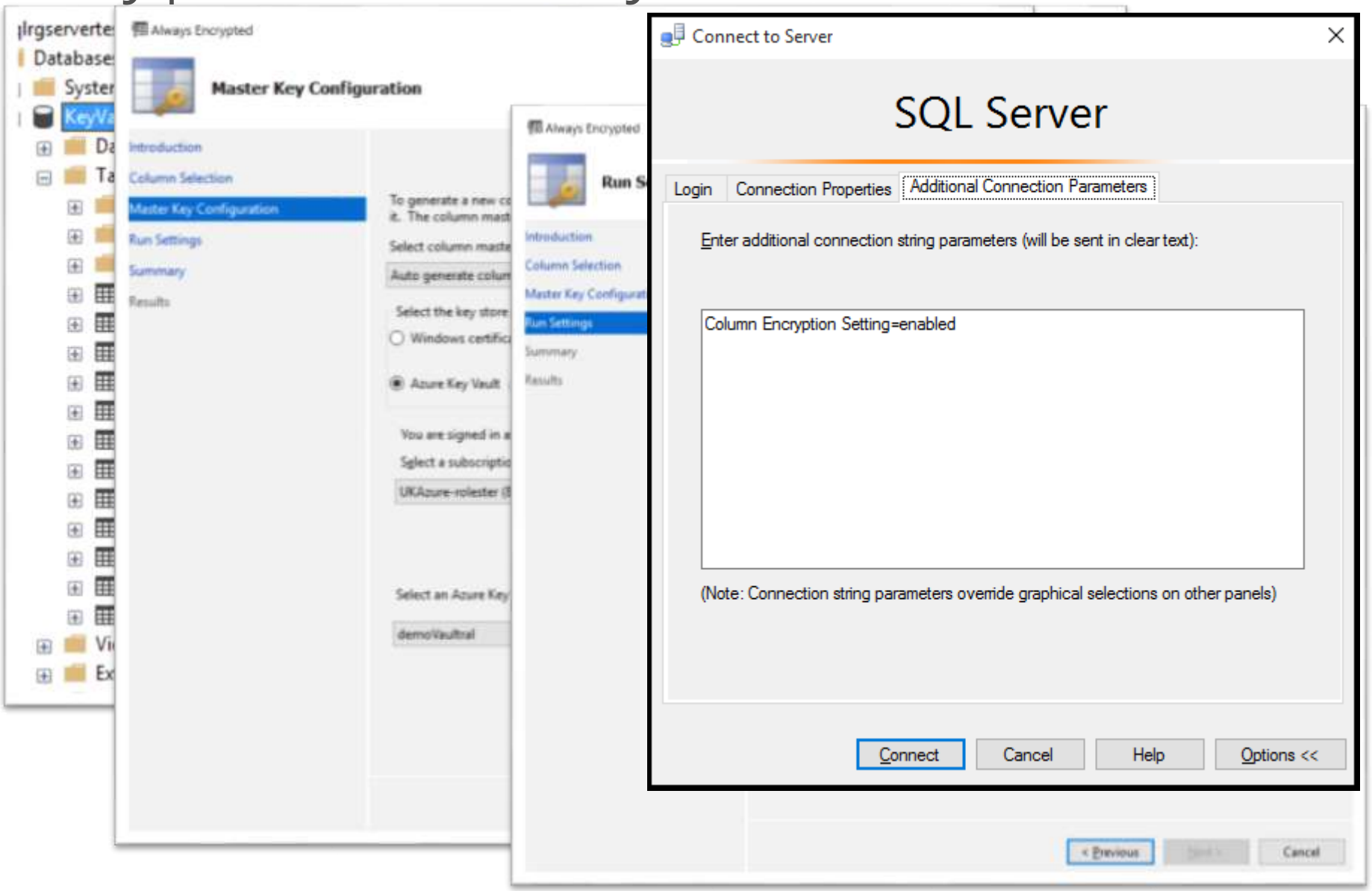
- Deterministic and Randomized just as secure
- Except
  - With deterministic if you know there are a fixed number of records of a certain type in a table you can identify those records by the number of occurrences.
  - EG: if it were know that there were three people that were admins

Name	Security Level
Jane	0x85a55d3f
Jim	0x17cfd50a
Joe	0x9b1fcf32
Julia	0x85a55d3f
Jill	0x85a55d3f
Jerry	0x7a2bda11

  - EG: How many different types of a category are there

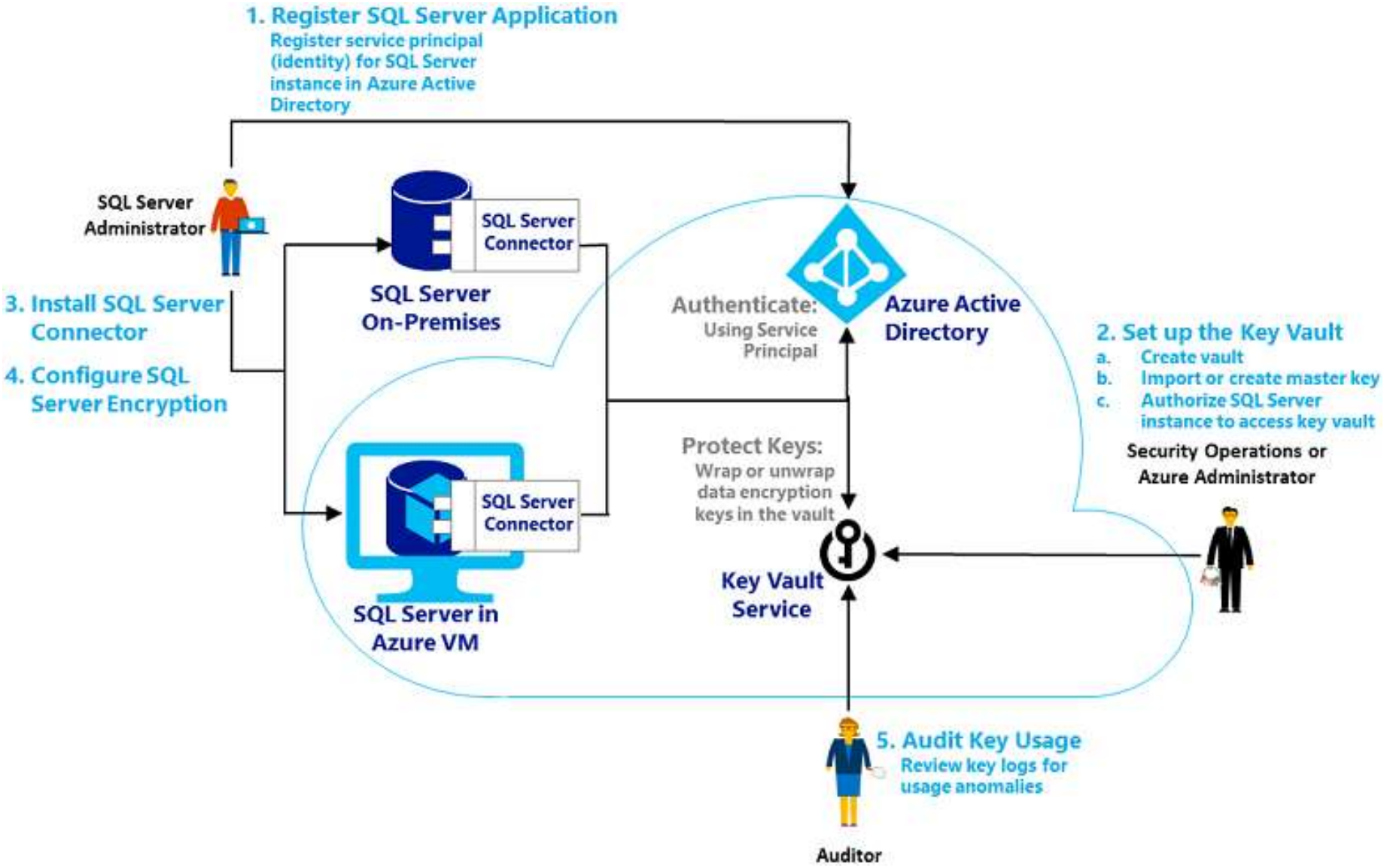
Always Encrypted with Key Vault

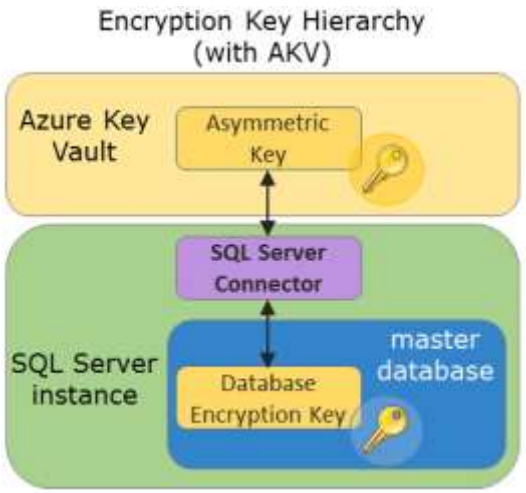
- Demo



Overview of Encryption Technologies

Feature / Capability	Always Encrypted	Transparent Data Encryption	Cell-level Encryption	App-Controlled Encryption
Level of protection	End-to-end	At-rest	At-rest	End-to-end
Can server see sensitive data?	No	Yes	Yes	No
T-SQL operations on encrypted data	Equality comparison	All (after decryption)	All (after decryption)	Possible with the appropriate encryption algo
App development cost to use feature	Low	Very low	High	Very High
Encryption granularity	Column	Database	Cell	Cell


1. Register SQL Server Application
Register service principal (identity) for SQL Server instance in Azure Active Directory
SQL Server Administrator
SQL Server Connector
SQL Server On-Premises
3. Install SQL Server Connector
4. Configure SQL Server Encryption
Authenticate:
Using Service Principal
Azure Active Directory
2. Set up the Key Vault
a. Create vault
b. Import or create master key
c. Authorize SQL Server instance to access key vault
Security Operations or Azure Administrator
Protect Keys:
Wrap or unwrap data encryption keys in the vault
SQL Server Connector
SQL Server in Azure VM
Key Vault Service
5. Audit Key Usage
Review key logs for usage anomalies
Auditor


Encryption Key Hierarchy (with AKV)
Azure Key Vault
Asymmetric Key
SQL Server Connector
SQL Server instance
master database
Database Encryption Key

Microsoft