Microsoft

Azure
SQL Data Warehouse
Security

Industry-leading security and compliance

Enterprise-grade security

Defense-in-Depth

Industry-leading compliance

ISO 27001

SOC 1 Type 2

SOC 2 Type 2

PCI DSS Level 1

Cloud Controls Matrix

ISO 27018

Content Delivery and Security Association

Shared Assessments

FedRAMP JAB P-ATO

HIPAA / HITECH

FIPS 140-2

21 CFR Part 11

FERPA

DISA Level 2

CJIS

IRS 1075

ITAR-ready

Section 508 VPAT

European Union Model Clauses

EU Safe Harbor

United Kingdom G-Cloud

China Multi Layer Protection Scheme

China GB 18030

China CCCPPF

Singapore MTCS Level 3

Australian Signals Directorate

New Zealand GCIO

Japan Financial Services

ENISA IAF

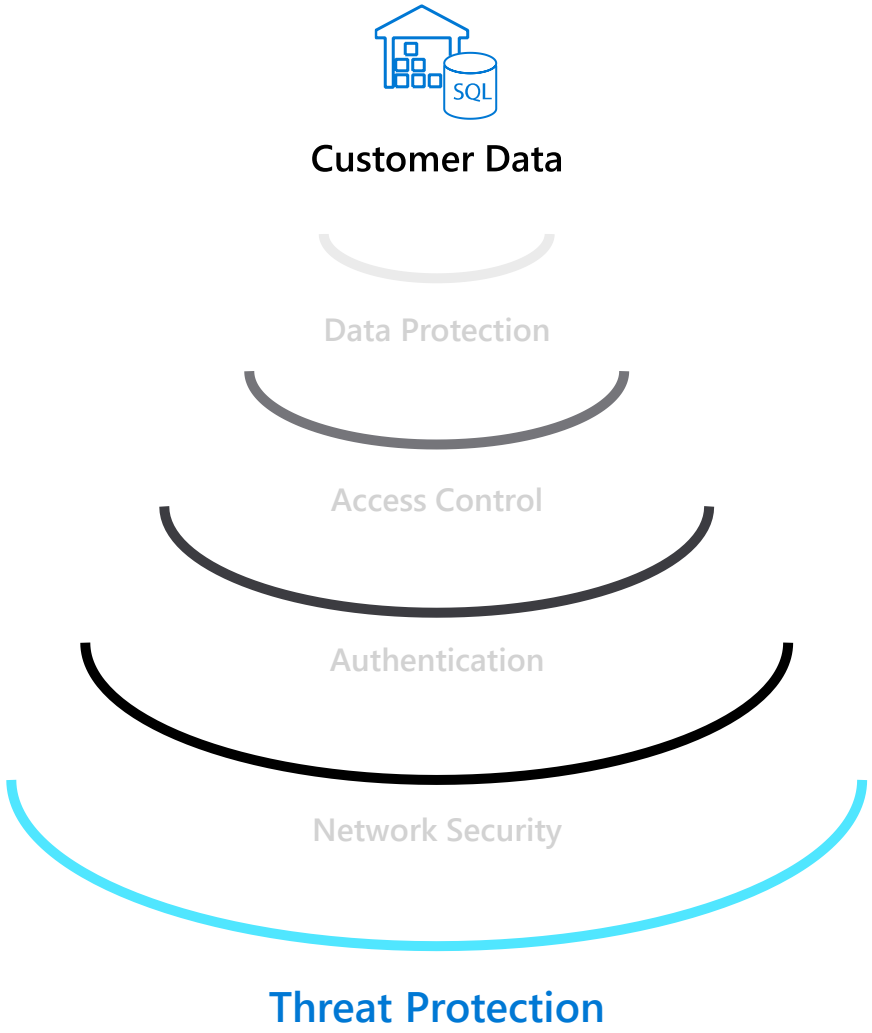
Threat Protection - Business requirements

How do we enumerate and track potential SQL vulnerabilities?

To mitigate any security misconfigurations before they become a serious issue.

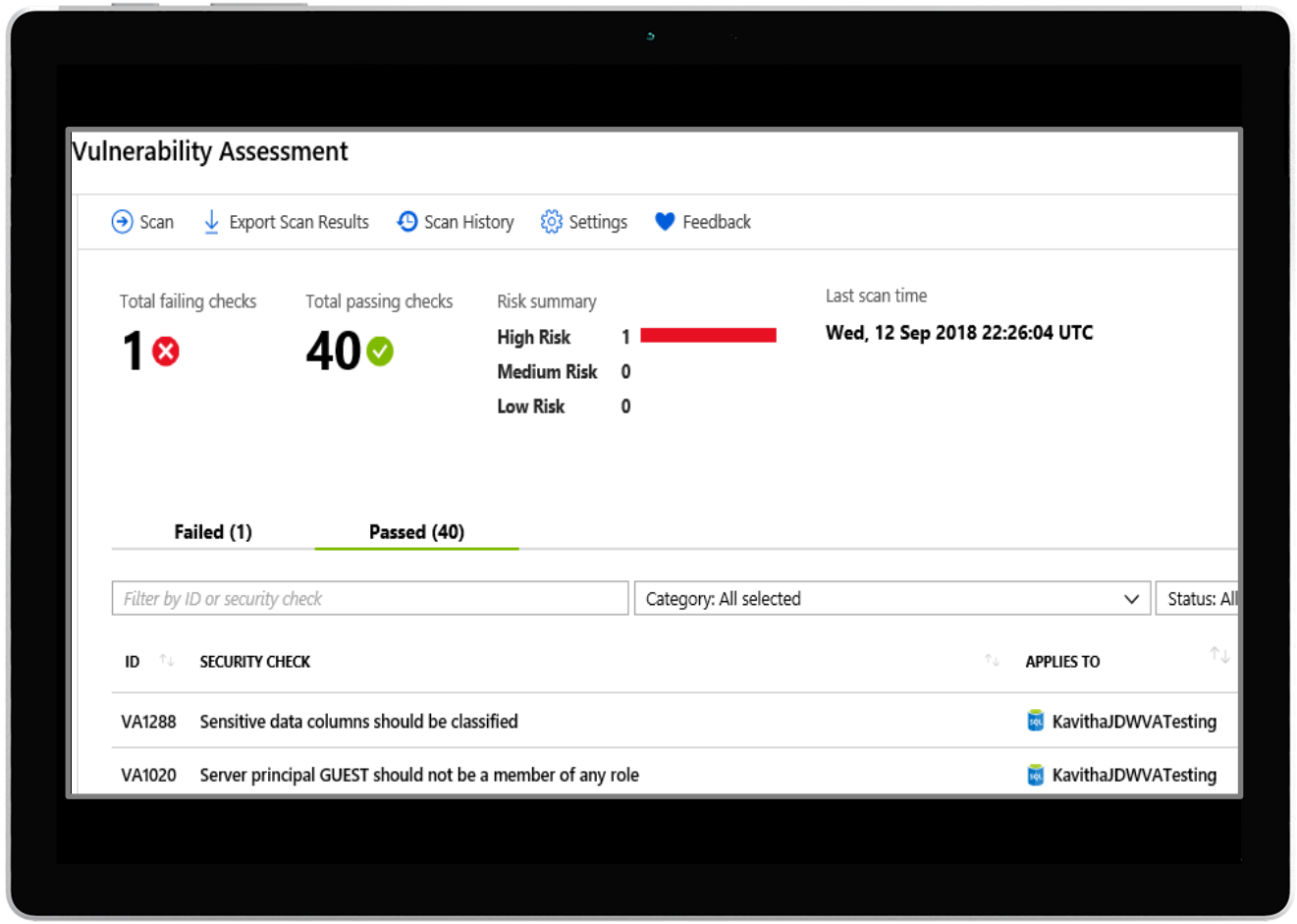
How do we discover and alert on suspicious database activity?

To detect and resolve any data exfiltration or SQL injection attacks.



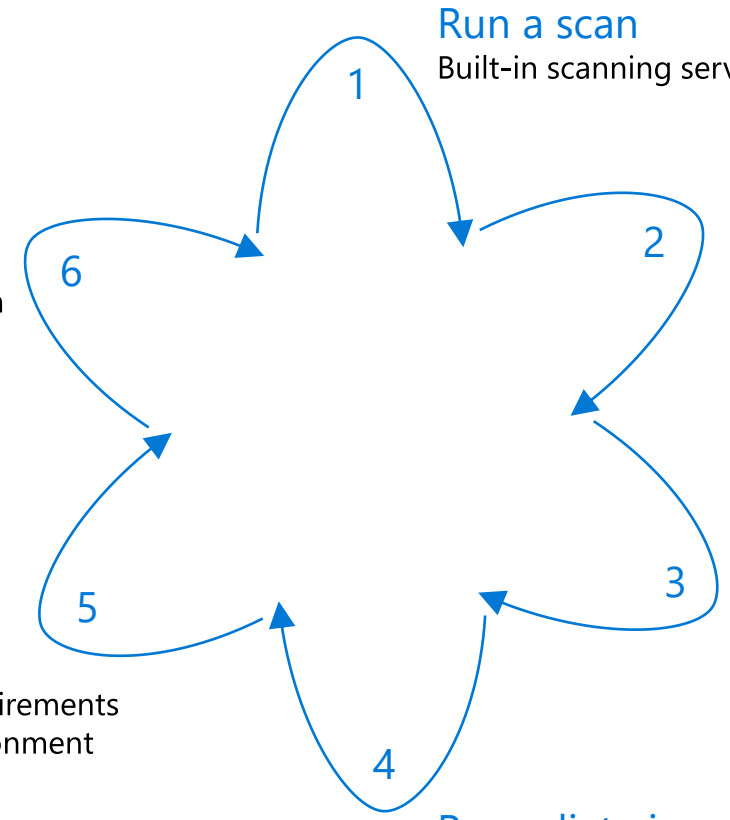
SQL vulnerability assessment

Discover, track, and remediate security misconfigurations



- ✓ Identify security misconfigurations
- ✓ Actionable remediation steps
- ✓ Security baseline tuned to your environment
- ✓ Manual/periodic scans
- ✓ Coherent reports for auditors

Using vulnerability assessment



Run a scan
Built-in scanning service

View a report
All-up assessment of security state

Drill-down to results
View detailed results and understand how they impact database security

Remediate issues
Run scripts that resolve vulnerabilities directly within the report

Set a baseline
Customize scan requirements based on your environment

Detect deviations
Subsequent scans will alert on deviations from your baseline

SQL Auditing

SQL Auditing

- **Server-level vs. database-level auditing**
- **Default auditing policy includes all actions plus**

  BATCH_COMPLETED_GROUP

  SUCCESSFUL_DATABASE_AUTHENTICATION_GROUP

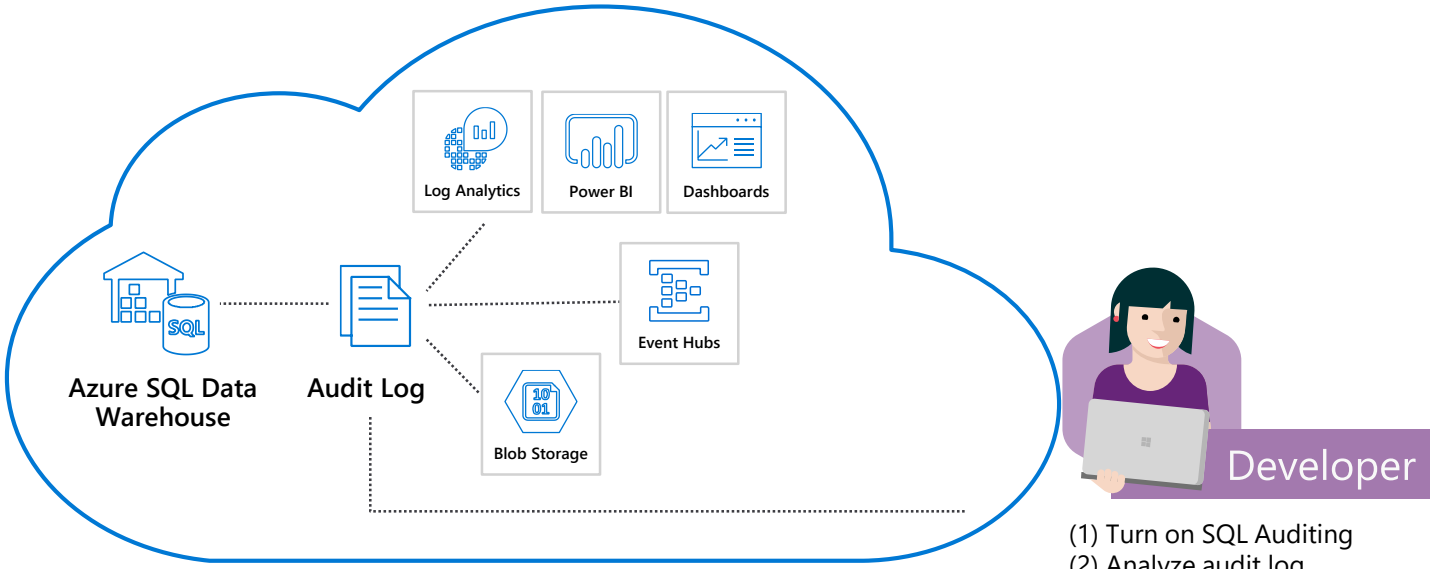
  FAILED_DATABASE_AUTHENTICATION_GROUP
- **Use PowerShell or RestAPI to customize the audited events**

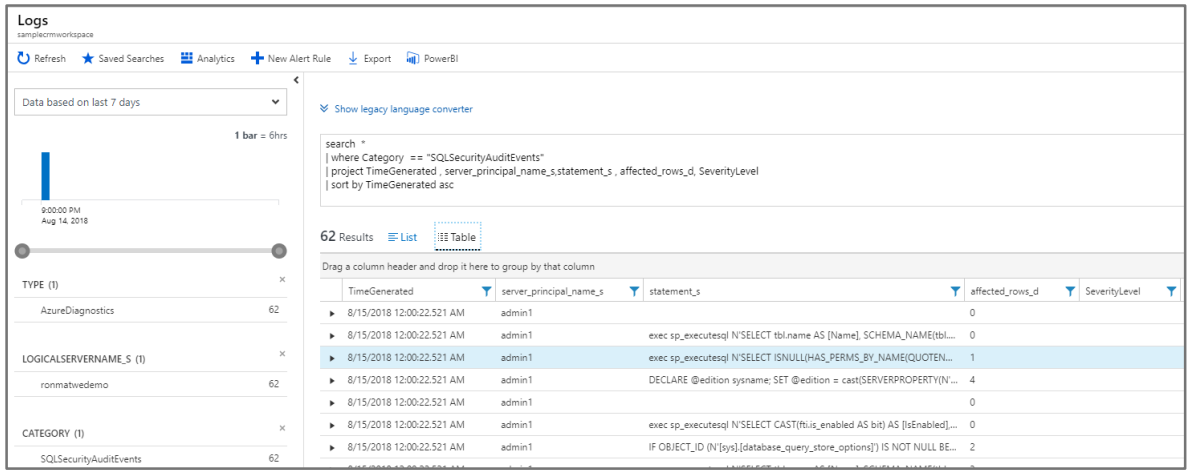
Analyze Audit Logs and reports

- **Azure Monitor Logs**
  - Auditing blade – View audit logs
  - Open in OMS
  - Log Analytics blade
- **Event Hub**
- **Azure Storage Account**
  - Azure Storage Explorer
  - Auditing blade – View audit logs
  - System function sys.fn_get_audit_file
  - SSMS
  - PowerBI
  - PowerShell (query extended events files)

SQL auditing in Azure Log Analytics and Event Hubs

Gain insight into database audit log

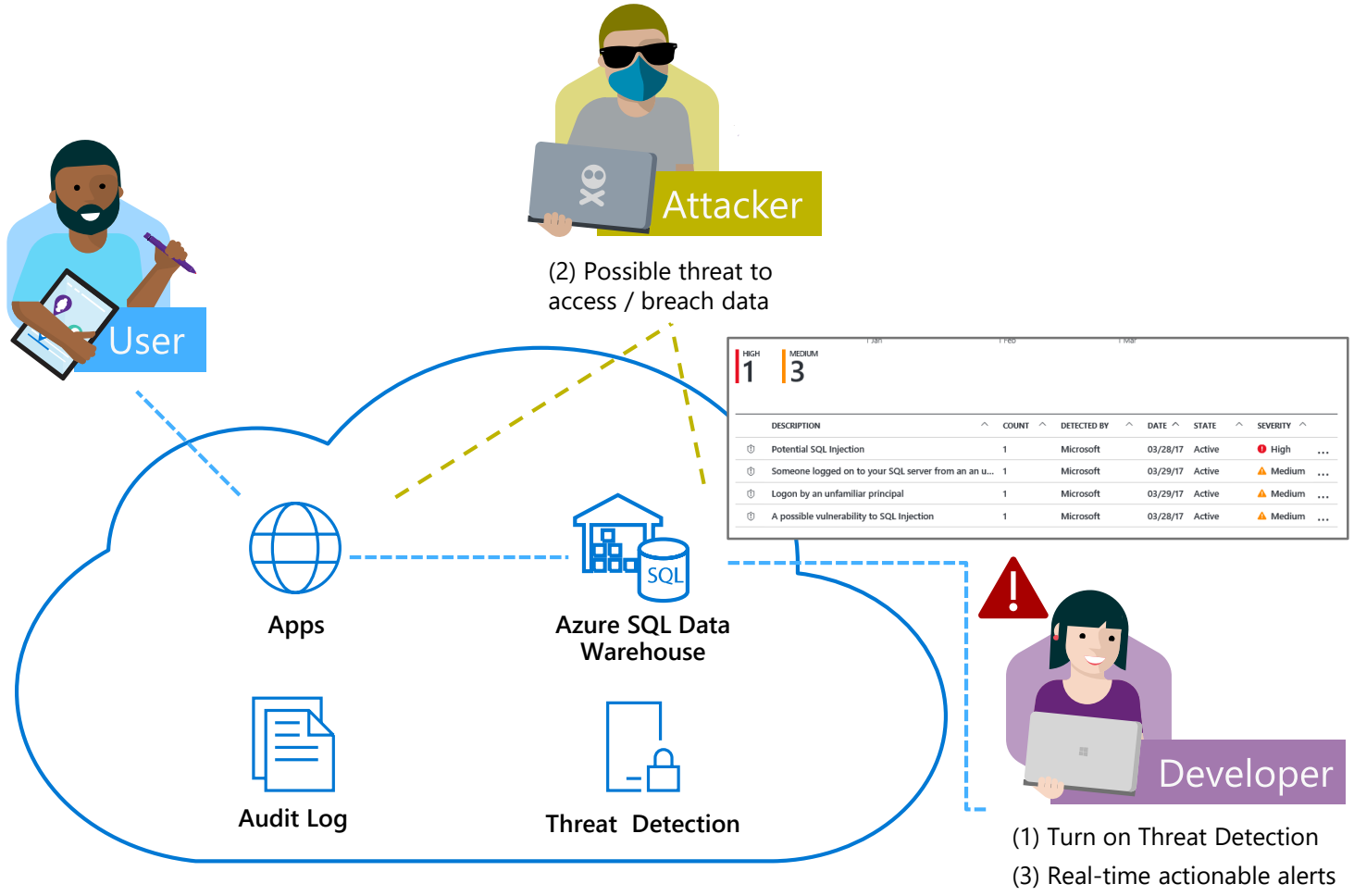




- ✓ **Configurable via audit policy**
- ✓ **SQL audit logs can reside in**
  - Azure Storage account
  - Azure Log Analytics
  - Azure Event Hubs
- ✓ **Rich set of tools for**
  - Investigating security alerts
  - Tracking access to sensitive data

SQL threat detection

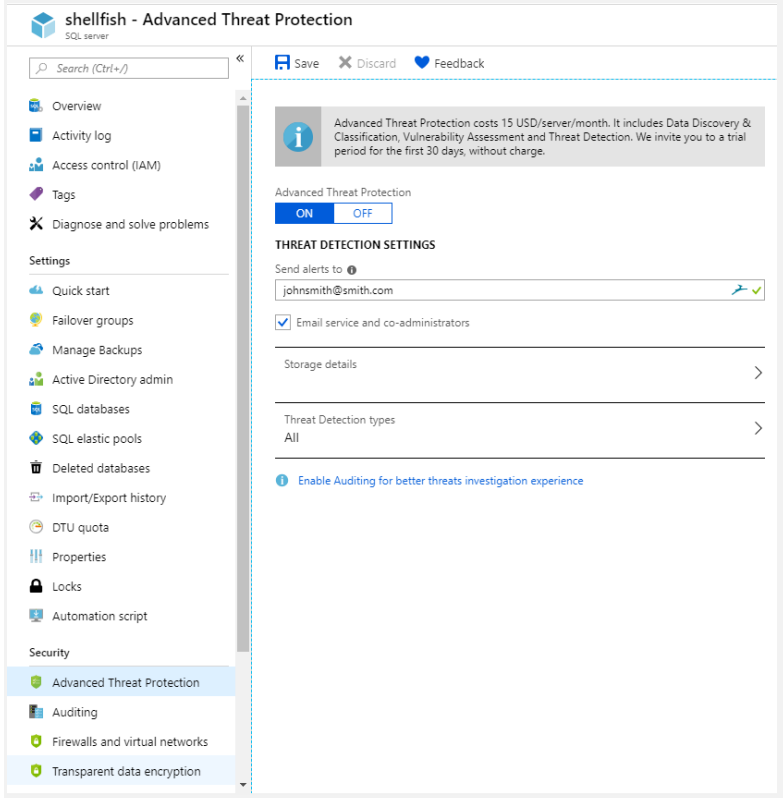
Detect and investigate anomalous database activity



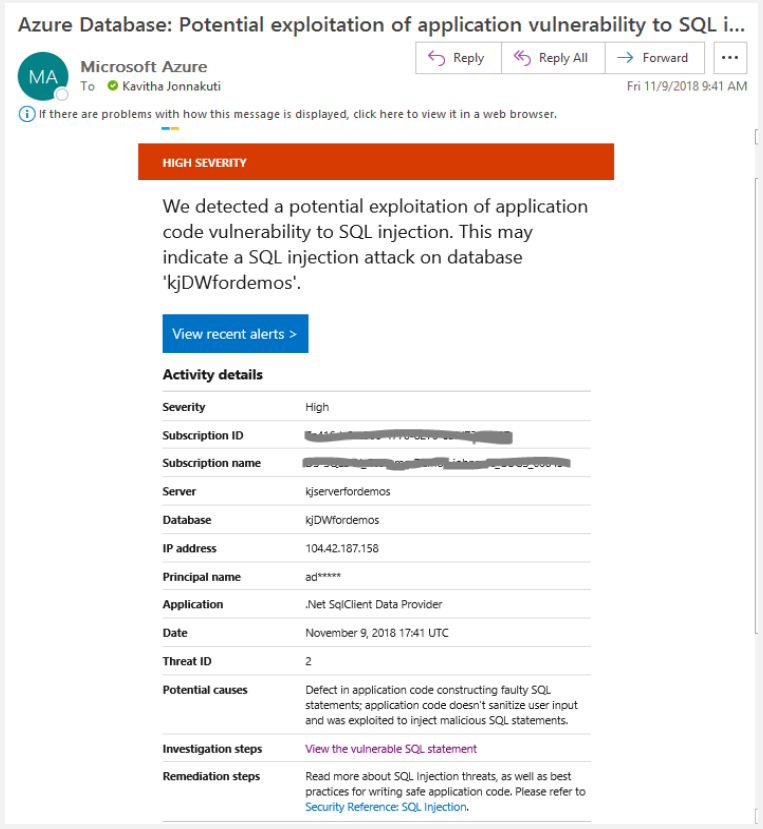
- ✓ Detects potential SQL injection attacks
- ✓ Detects unusual access & data exfiltration activities
- ✓ Actionable alerts to investigate & remediate
- ✓ View alerts for your entire Azure tenant using Azure Security Center

How threat detection works

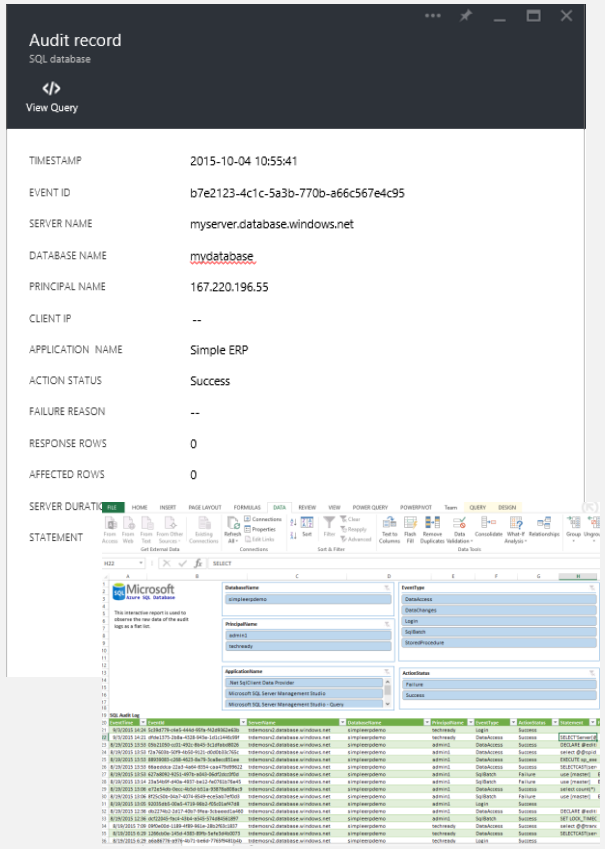
Set up



Alert

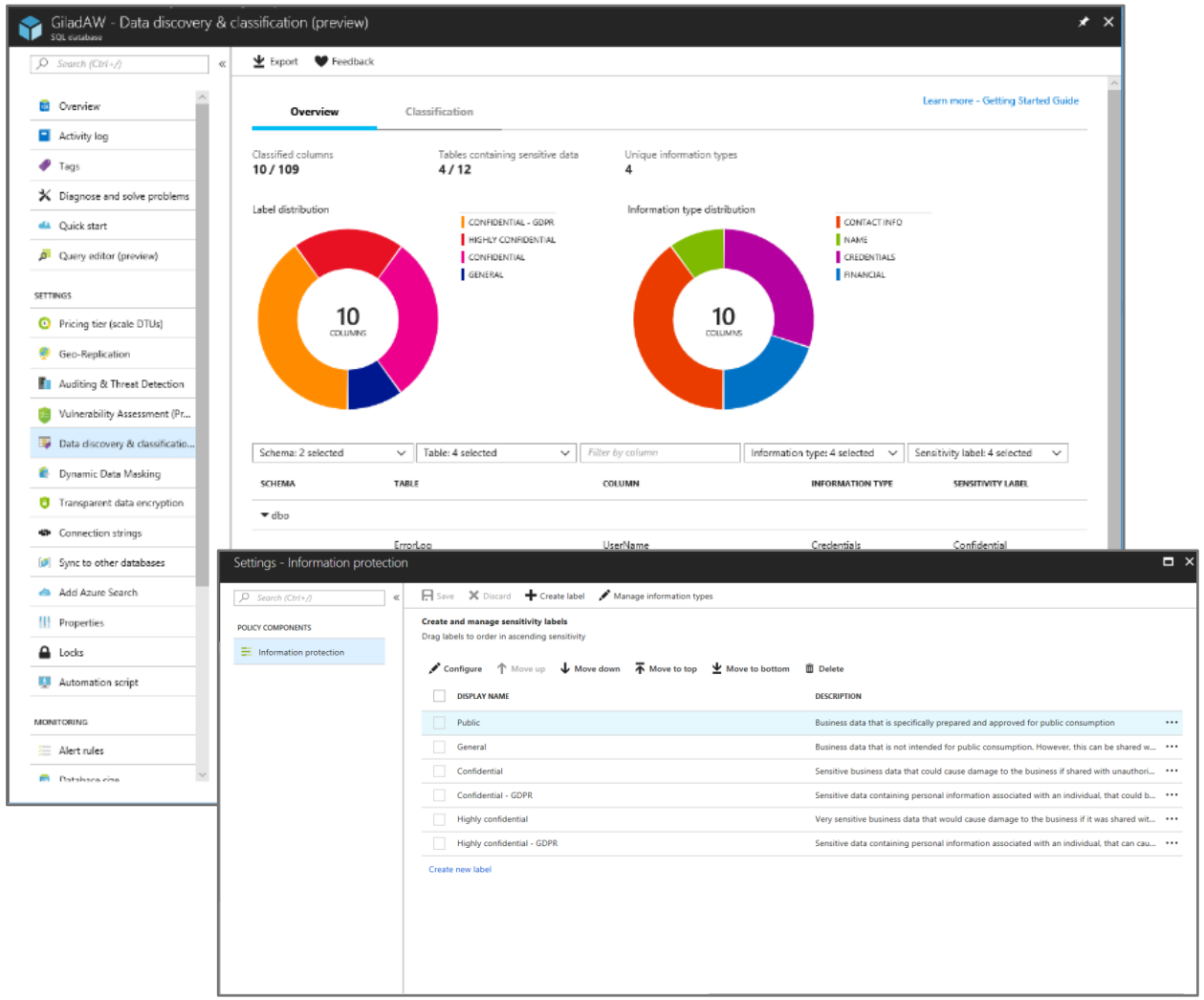


Explore



SQL Data Discovery & Classification

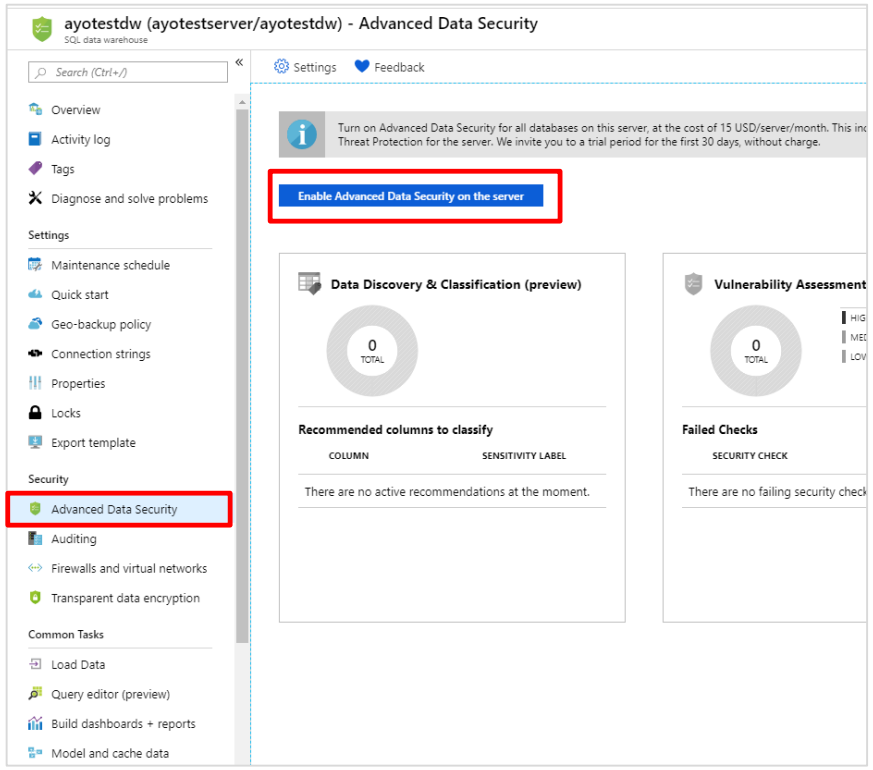
Discover, classify, protect and track access to sensitive data



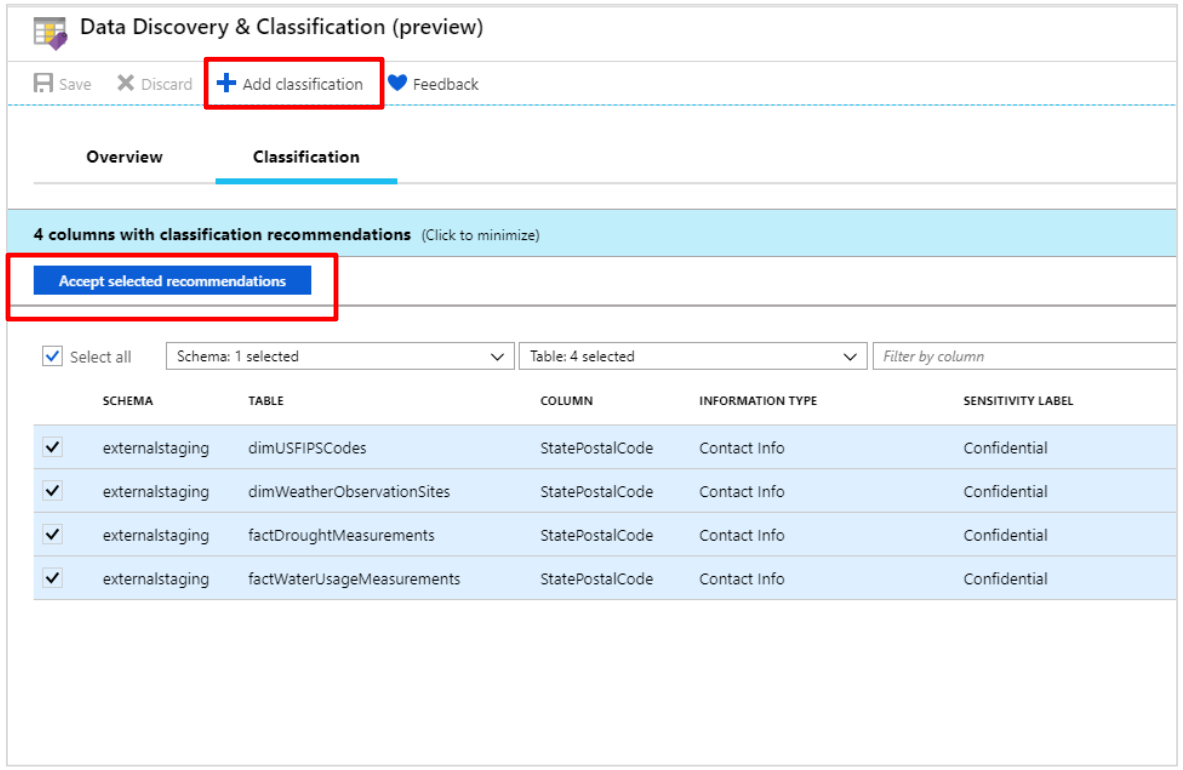
- ✓ Automatic discovery of columns with sensitive data
- ✓ Add persistent sensitive data labels
- ✓ Audit and detect access to the sensitive data
- ✓ Manage labels for your entire Azure tenant using Azure Security Center

SQL Data Discovery & Classification - setup

Step 1: Enable Advanced Data Security on the logical SQL Server

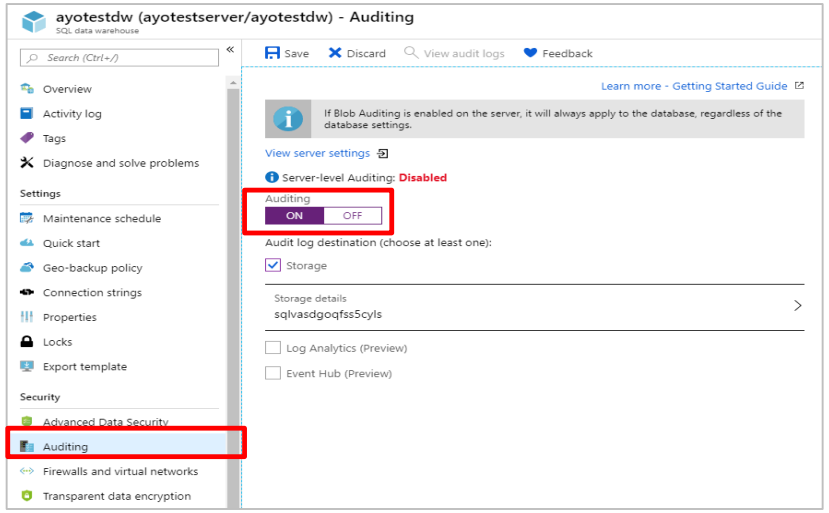


Step 2: Use recommendations and/or manual classification to classify all the sensitive columns in your tables

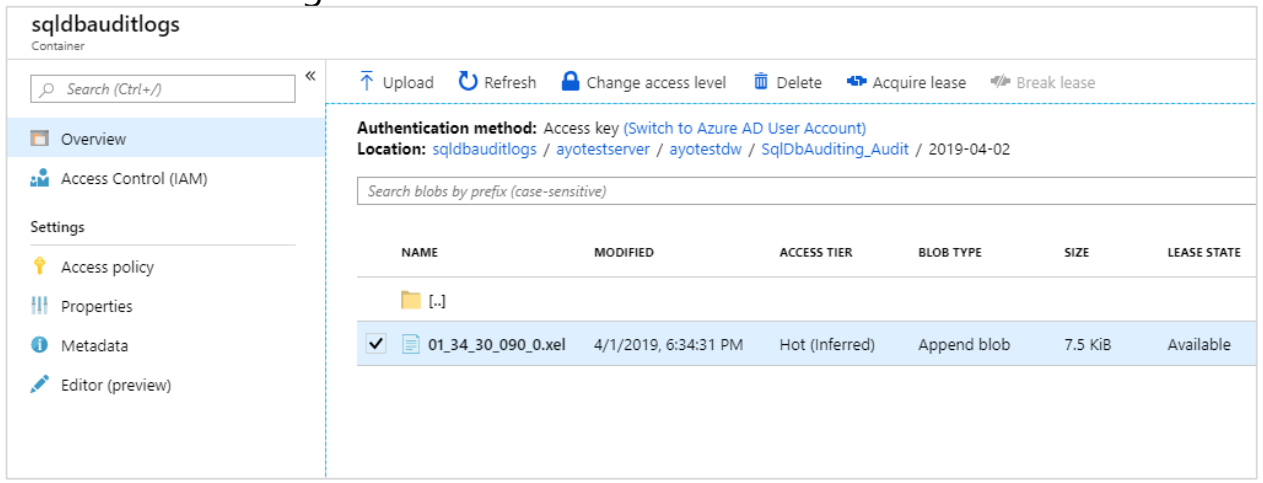


SQL Data Discovery & Classification – audit sensitive data access

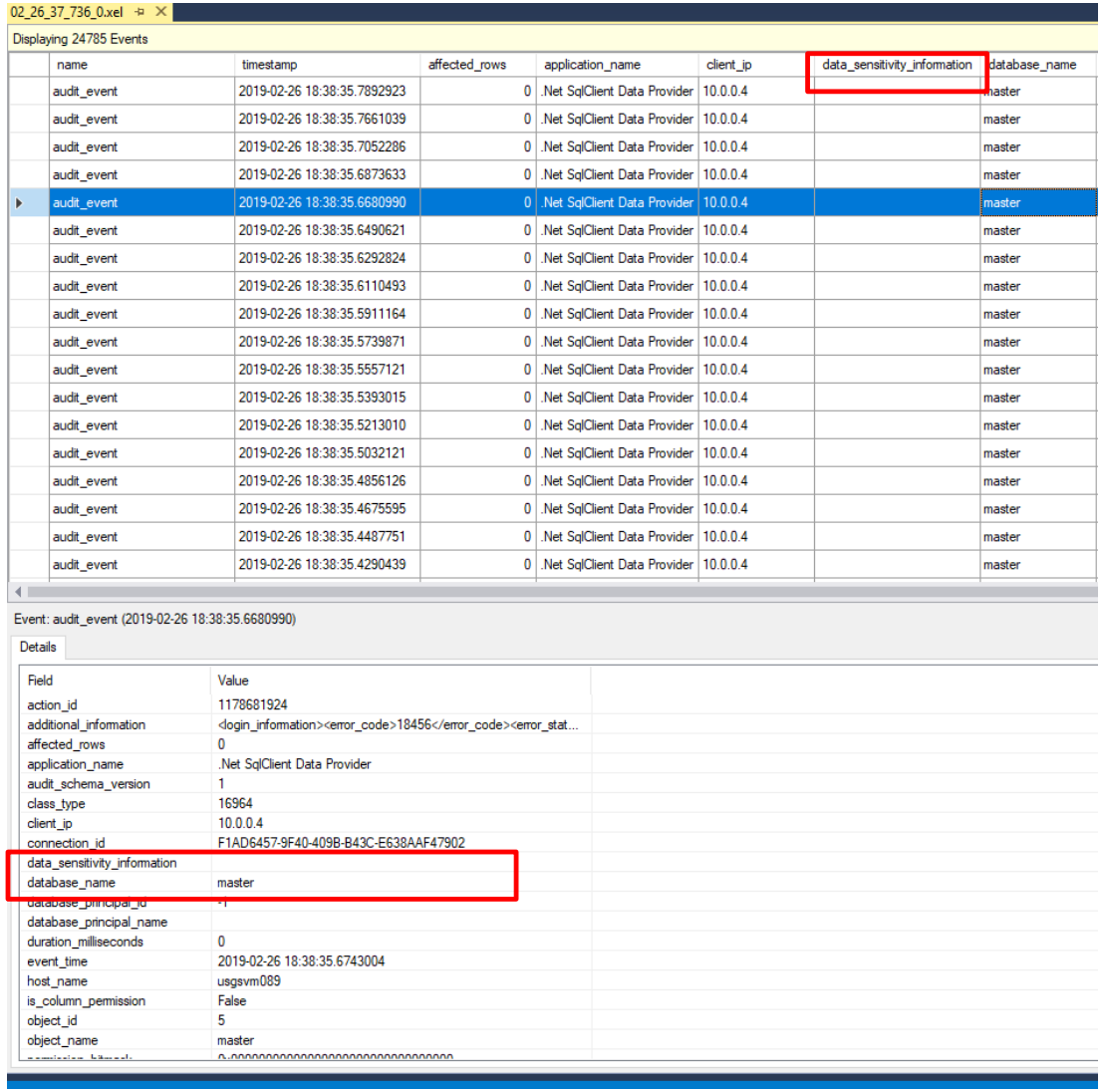
Step 1: Configure auditing for your target Data warehouse. This can be configured for just a single data warehouse or all databases on a server.



Step 2: Navigate to audit logs in storage account and download 'xel' log files to local machine.



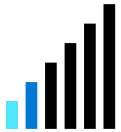
Step 3: Open logs using extended events viewer in SSMS. Configure viewer to include 'data_sensitivity_information' column



Network Security - Business requirements

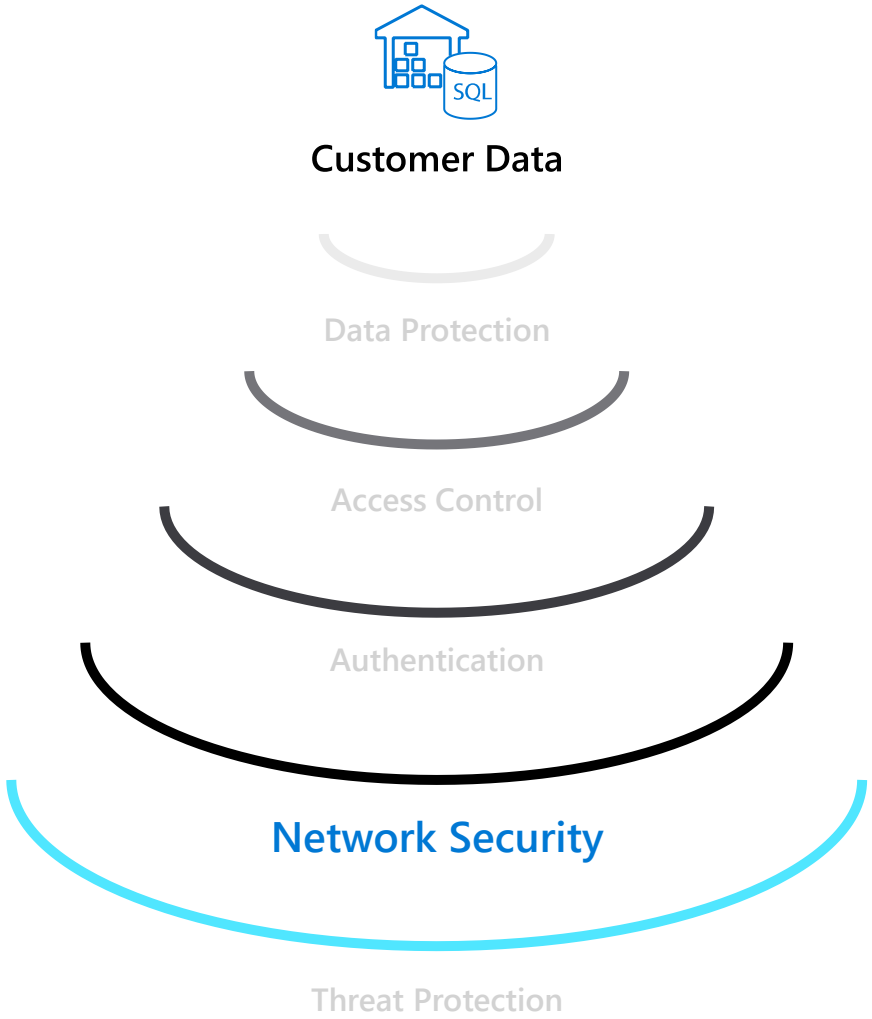
How do we implement network isolation?

Data at different levels of security needs to be accessed from different locations.

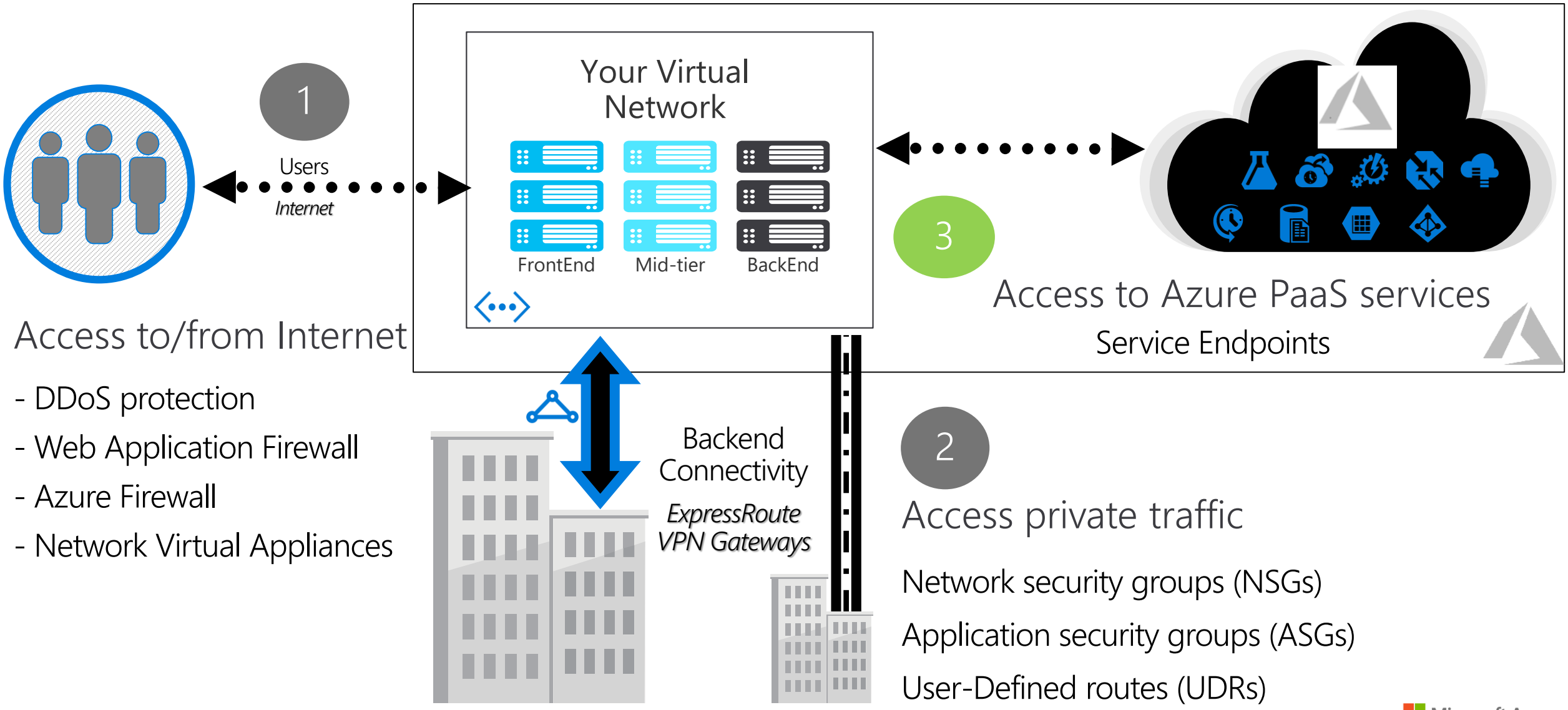


How do we achieve separation?

Disallowing access to entities outside the company's network security boundary.



Azure Networking - Application Access Patterns



Access to/from Internet

- DDoS protection
- Web Application Firewall
- Azure Firewall
- Network Virtual Appliances

Access private traffic

Network security groups (NSGs)

Application security groups (ASGs)

User-Defined routes (UDRs)

Microsoft Azure

Securing with firewalls

Overview

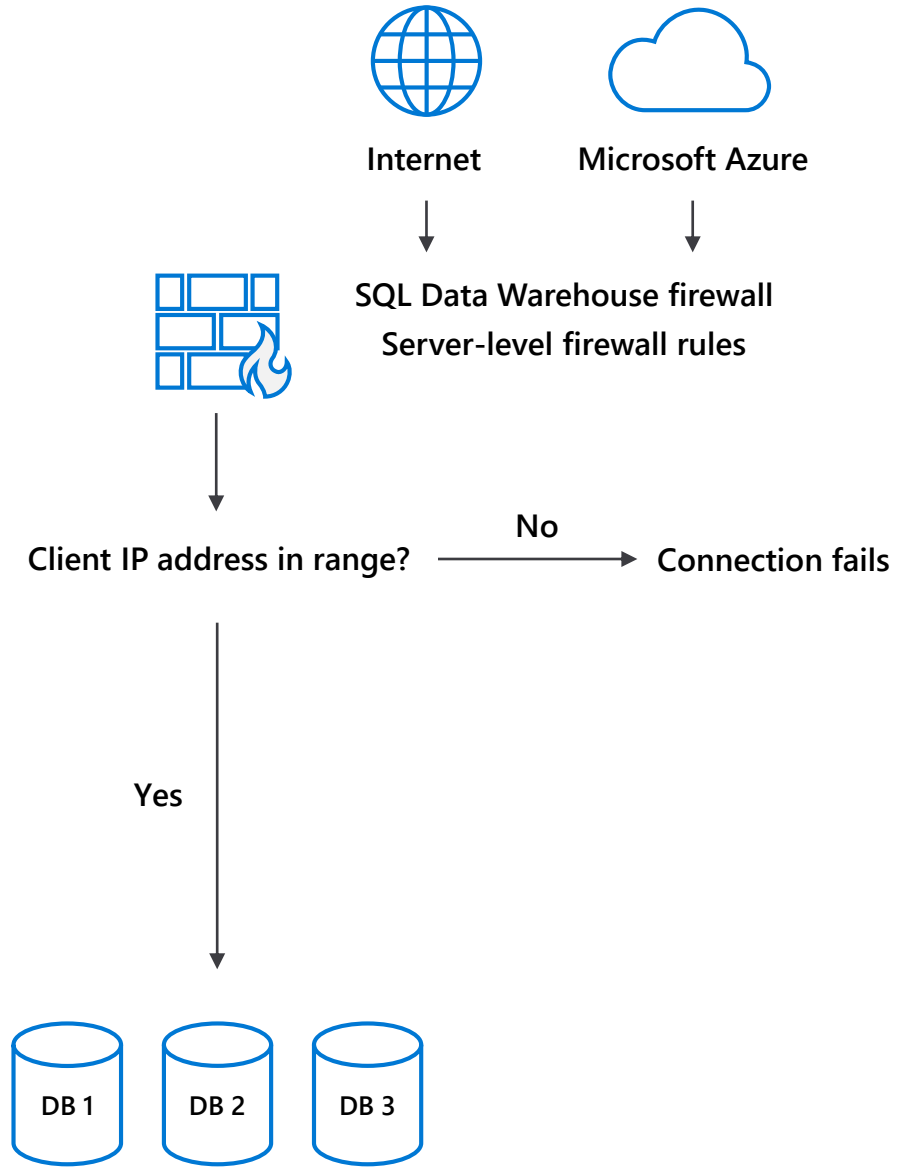
By default, all access to your Azure SQL Data Warehouse server is blocked by the firewall.

Firewall also manages virtual network rules that are based on virtual network service endpoints.

Rules

Allow specific or range of whitelisted IP addresses.

Allow Azure applications to connect.

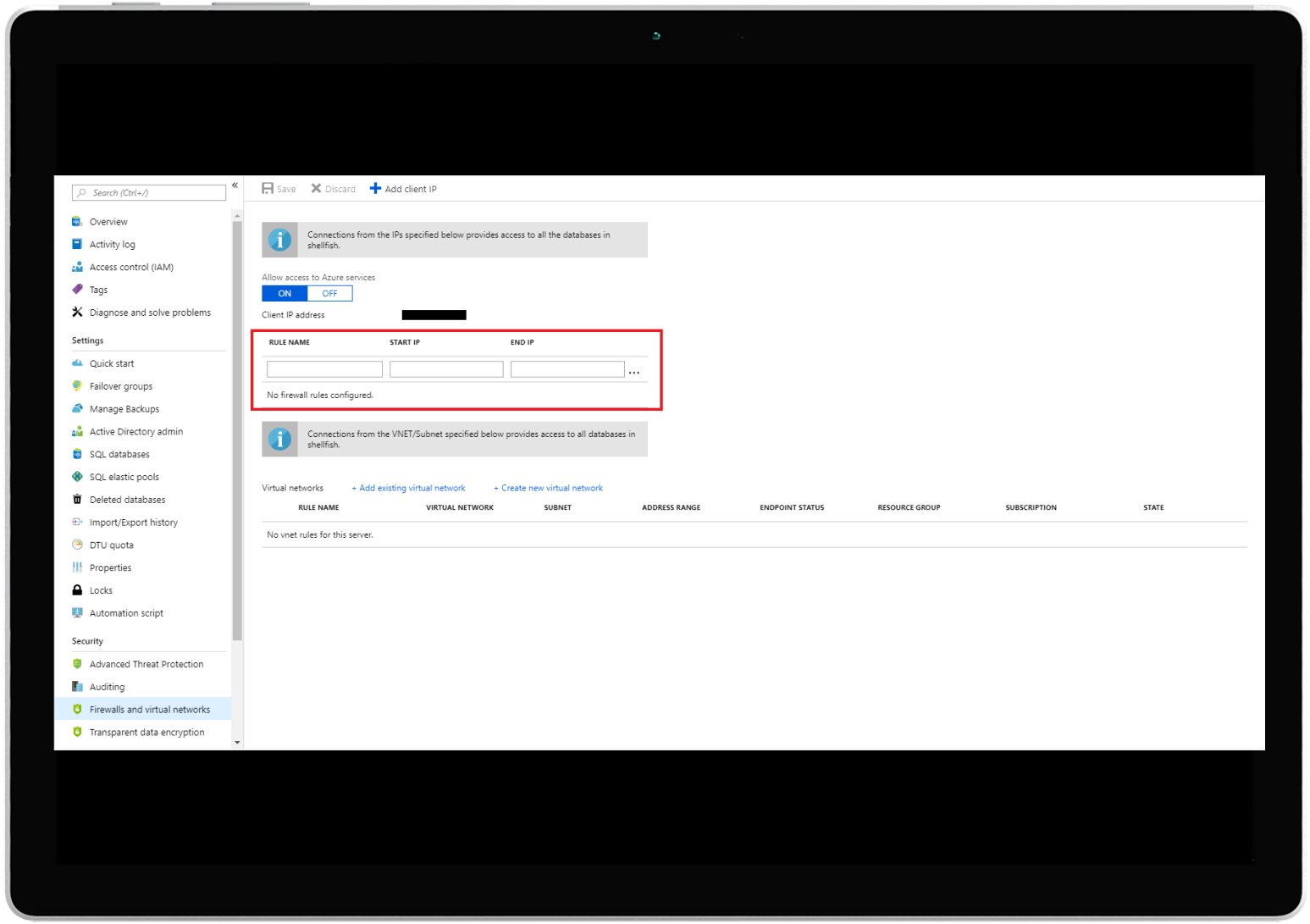


Firewall configuration on the portal

By default, Azure blocks all external connections to port 1433

Configure with the following steps:

SQL Data Warehouse Resource:
Server name > Firewalls and virtual networks



Firewall configuration using PowerShell/T-SQL

Windows PowerShell Azure cmdlets

New-AzureRmSqlServerFirewallRule

Get-AzureRmSqlServerFirewallRule

Set-AzureRmSqlServerFirewallRule

Transact SQL

sp_set_firewall_rule

sp_delete_firewall_rule

```
# PS Allow external IP access to SQL DW
PS C:\> New-AzureRmSqlServerFirewallRule
          -ResourceGroupName "myResourceGroup" `
          -ServerName $servername `
          -FirewallRuleName "AllowSome"
          -StartIpAddress "0.0.0.0"
          -EndIpAddress "0.0.0.0“

-- T-SQL Allow external IP access to SQL DW
EXECUTE sp_set_firewall_rule
          @name = N'ContosoFirewallRule’,
          @start_ip_address = '192.168.1.1’,
          @end_ip_address = '192.168.1.10'
```

Virtual network service endpoints

Overview

Extend VNET identity to the service.

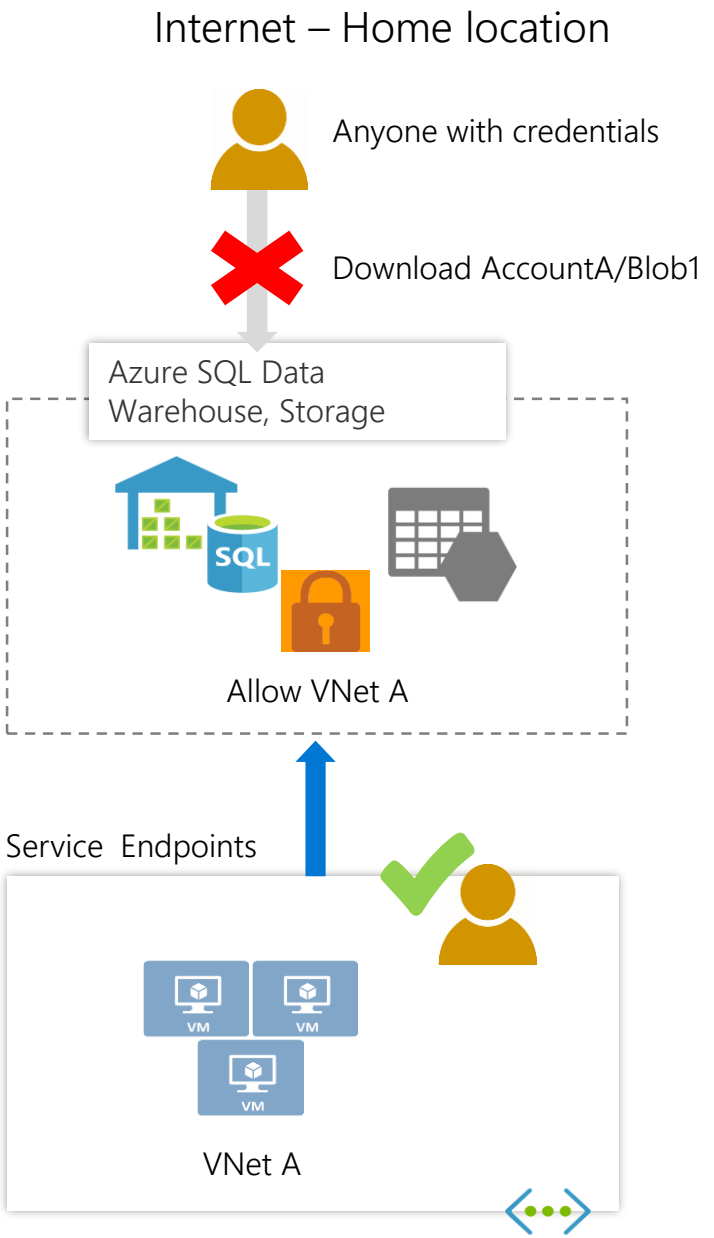
Secure critical Azure resources to only your VNET.

Traffic remains on the Azure backbone.

Virtual network Rules

Firewall security feature that allows communications from only specified subnets in virtual networks.

Finer granular security control than "**Allow access to Azure Services.**"



VNET configuration on Azure portal

Configure with the following steps:

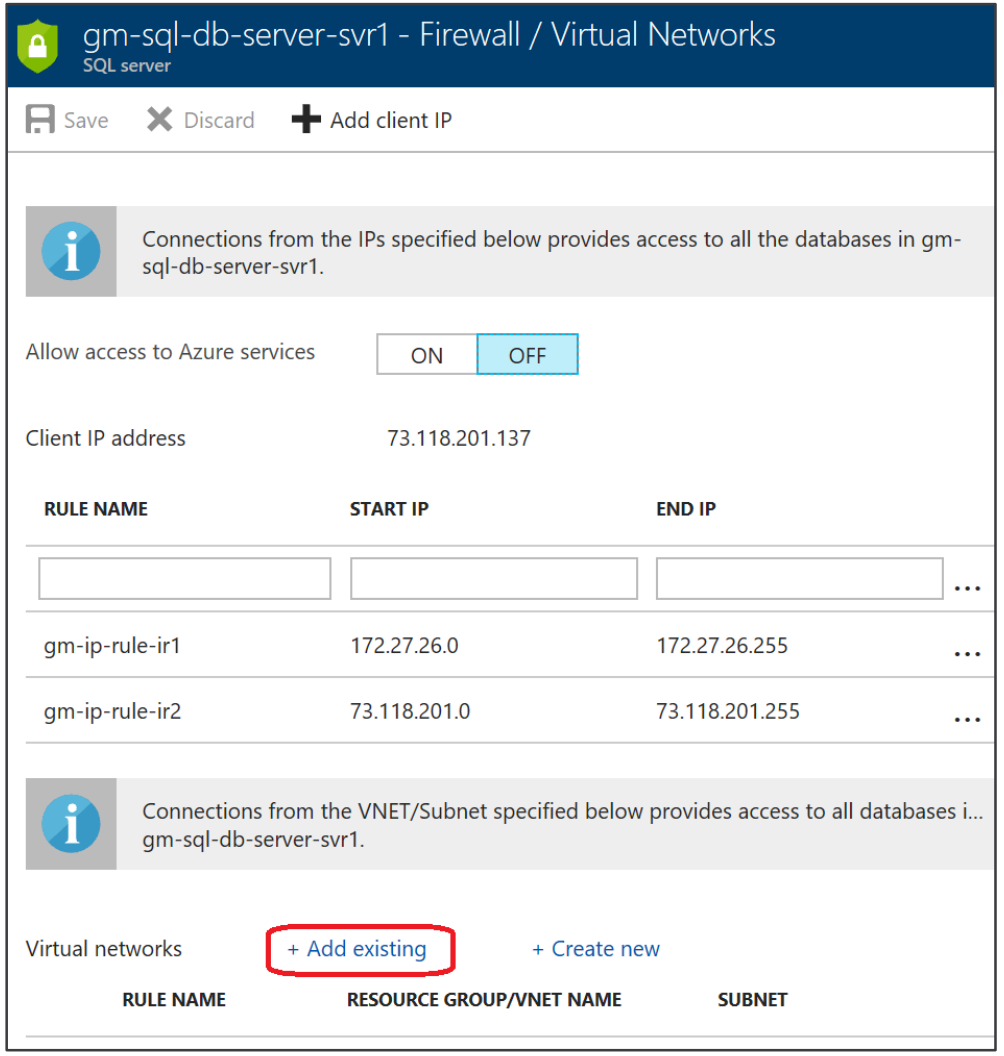
SQL Data Warehouse Resource:
Server name > Firewalls and virtual networks

REST API and PowerShell alternatives available

Note:

By default, VMs on your subnets cannot communicate with your SQL Data Warehouse.

There must first be a virtual network service endpoint for the rule to reference.



Private Link

Overview

Secure and scalable way to access Azure resources

No need for gateways, NAT, or Public IP addresses

Brings Azure services inside customer's private VNet

Supports

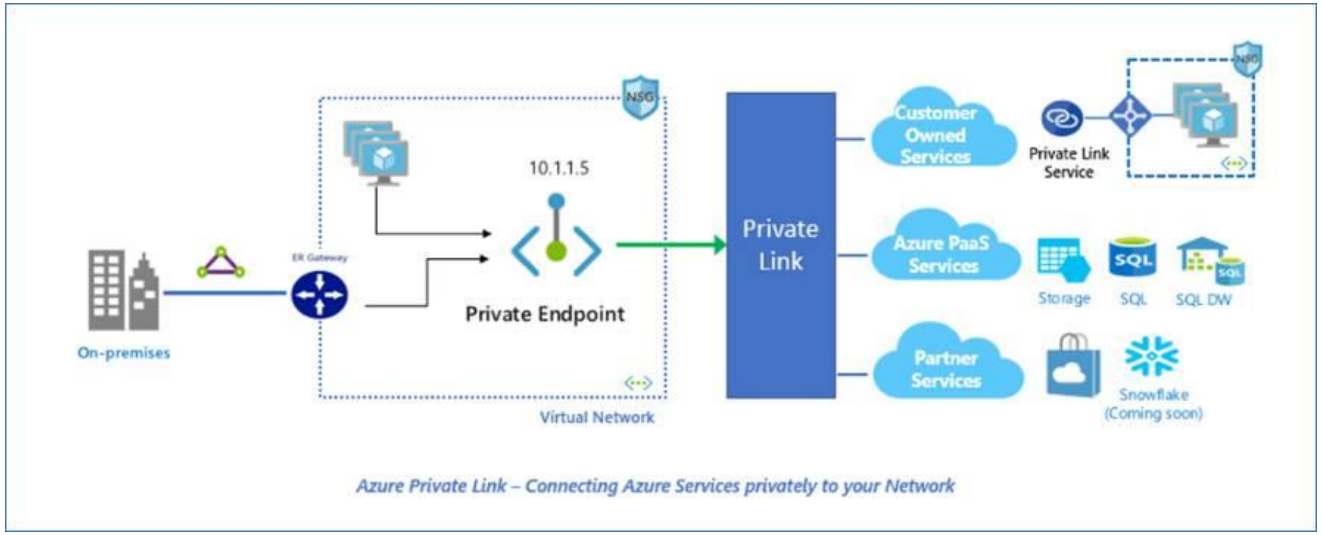
Azure SQL DW

Azure SQL DB

Azure Data Lake Storage Gen2

Azure Storage

Customer-owned services



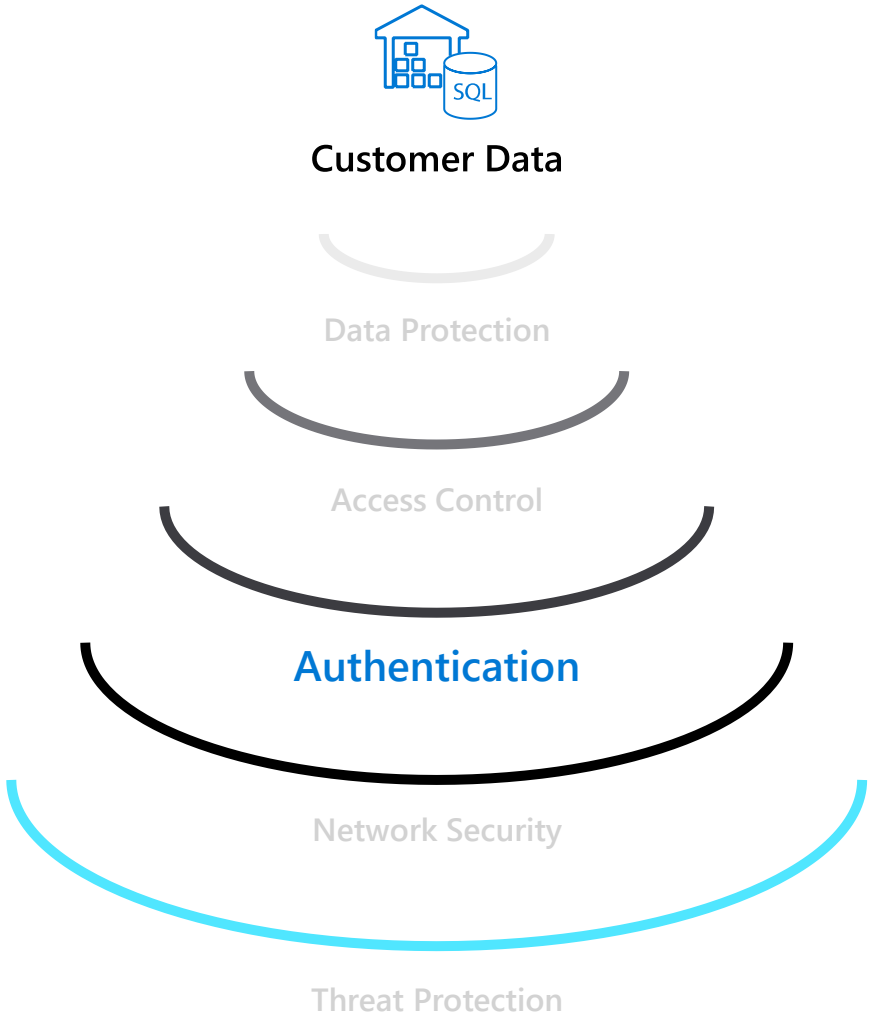
Azure Private Link – Connecting Azure Services privately to your Network

Authentication - Business requirements

How do I configure Azure Active Directory with Azure SQL Data Warehouse?

I want additional control in the form of multi-factor authentication

How do I allow non-Microsoft accounts to be able to authenticate?



Azure Active Directory authentication

Overview

Manage user identities in one location.

Enable access to Azure SQL Data Warehouse and other Microsoft services with Azure Active Directory user identities and groups.

Benefits

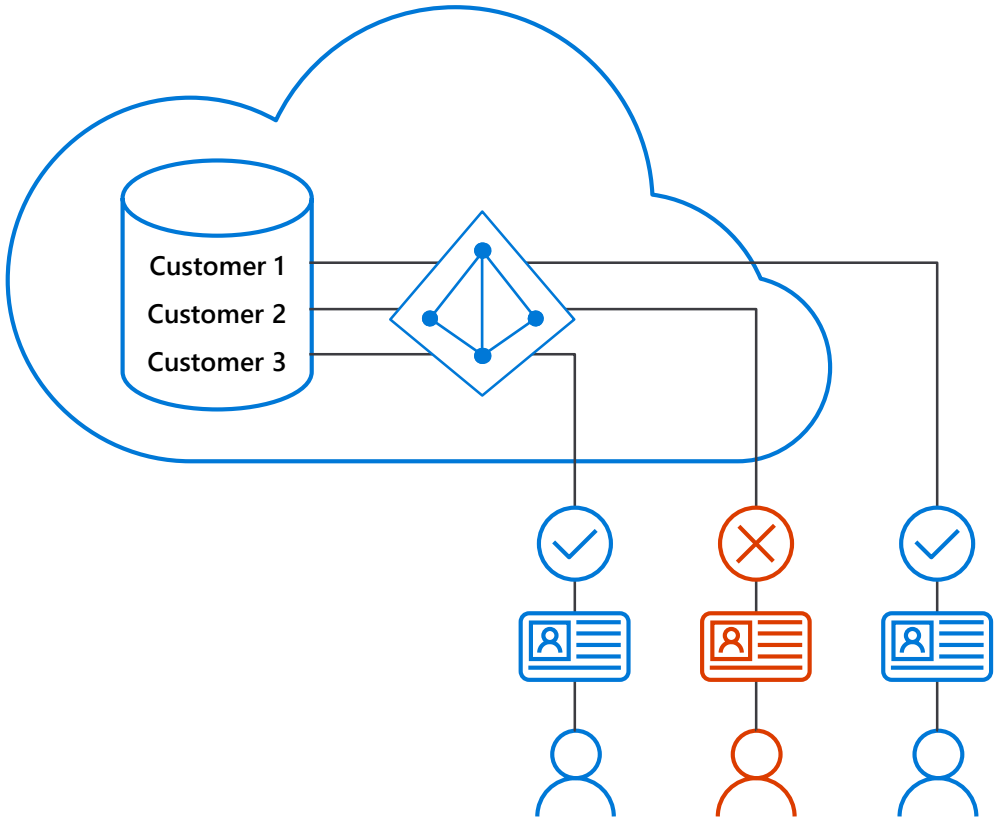
Alternative to SQL Server authentication

Limits proliferation of user identities across databases

Allows password rotation in a single place

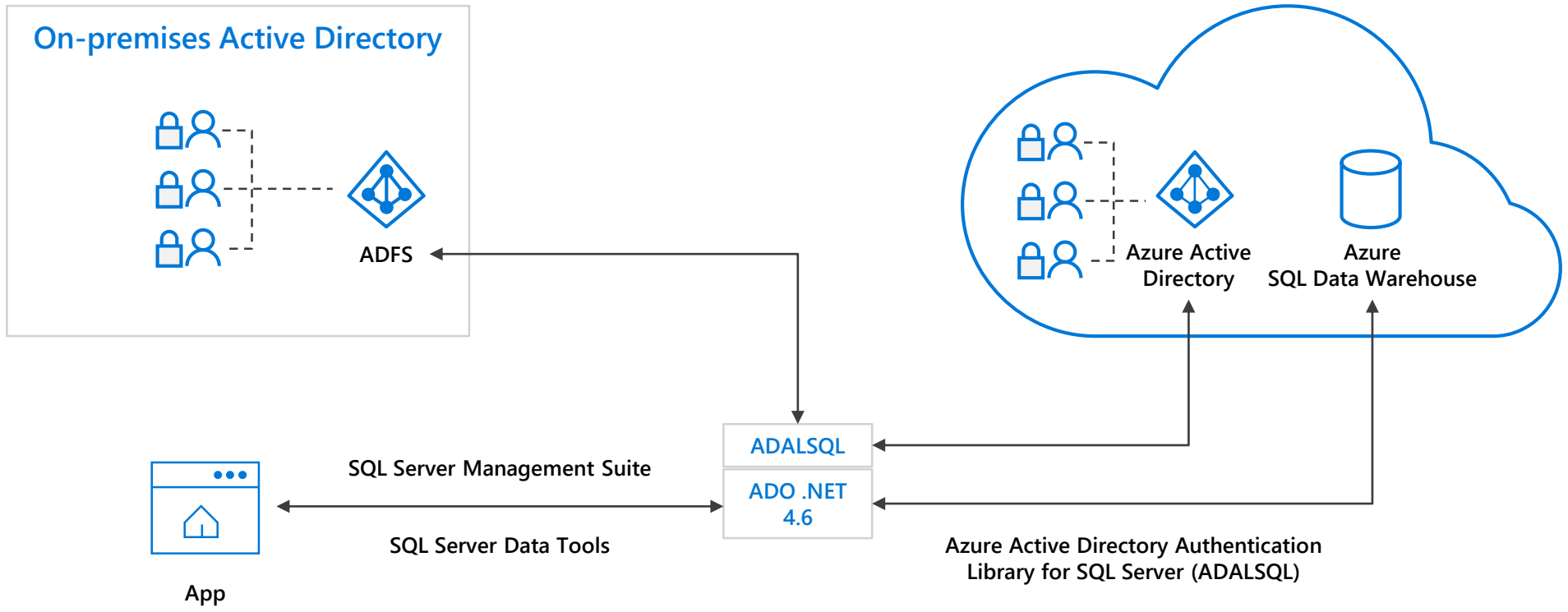
Enables management of database permissions by using external Azure Active Directory groups

Eliminates the need to store passwords



Azure Active Directory trust architecture

Azure Active Directory and Azure SQL Data Warehouse



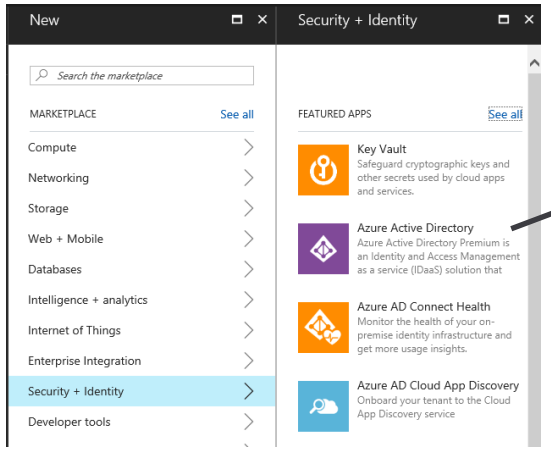
Support for MFA (SSMS)

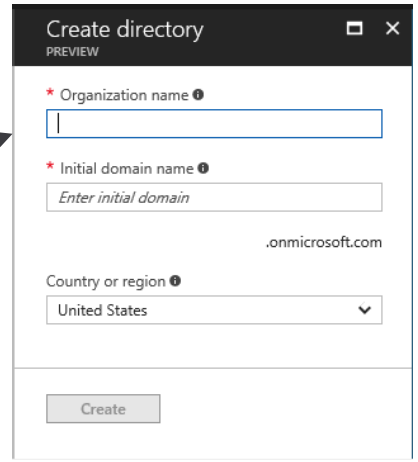
Setting up AAD

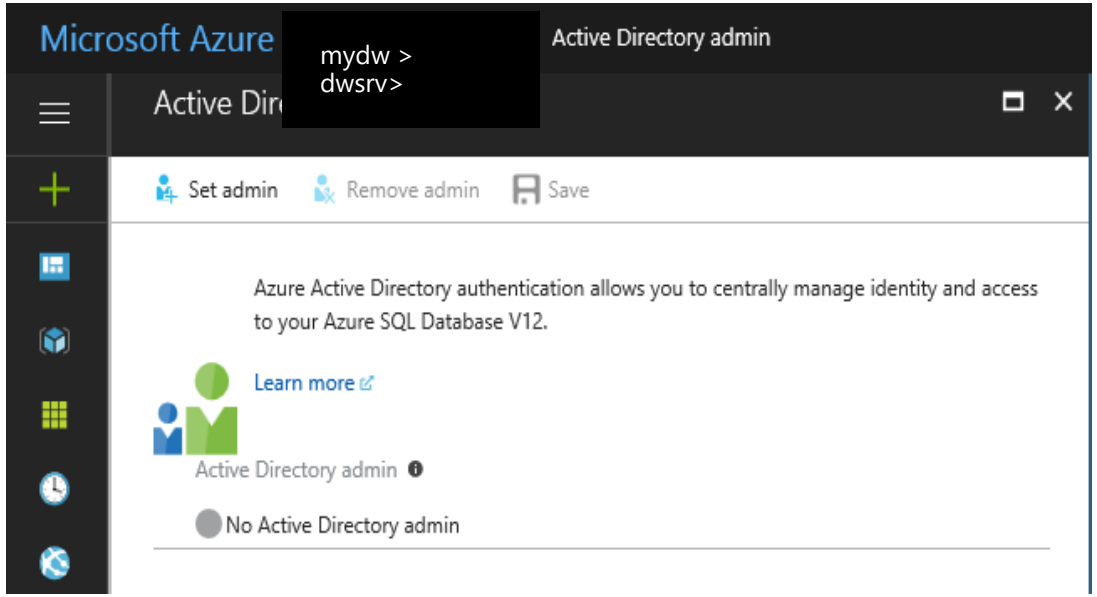
Associate on-premises AD with the Azure Subscription (as service admin via the portal)

OR

Create and populate Azure AD.







Configure an AAD admin for your SQL DW server

Create database users that leverage the `FROM EXTERNAL PROVIDER` syntax

```
CREATE USER [billg@microsoft.com] FROM EXTERNAL PROVIDER;
```

SQL authentication

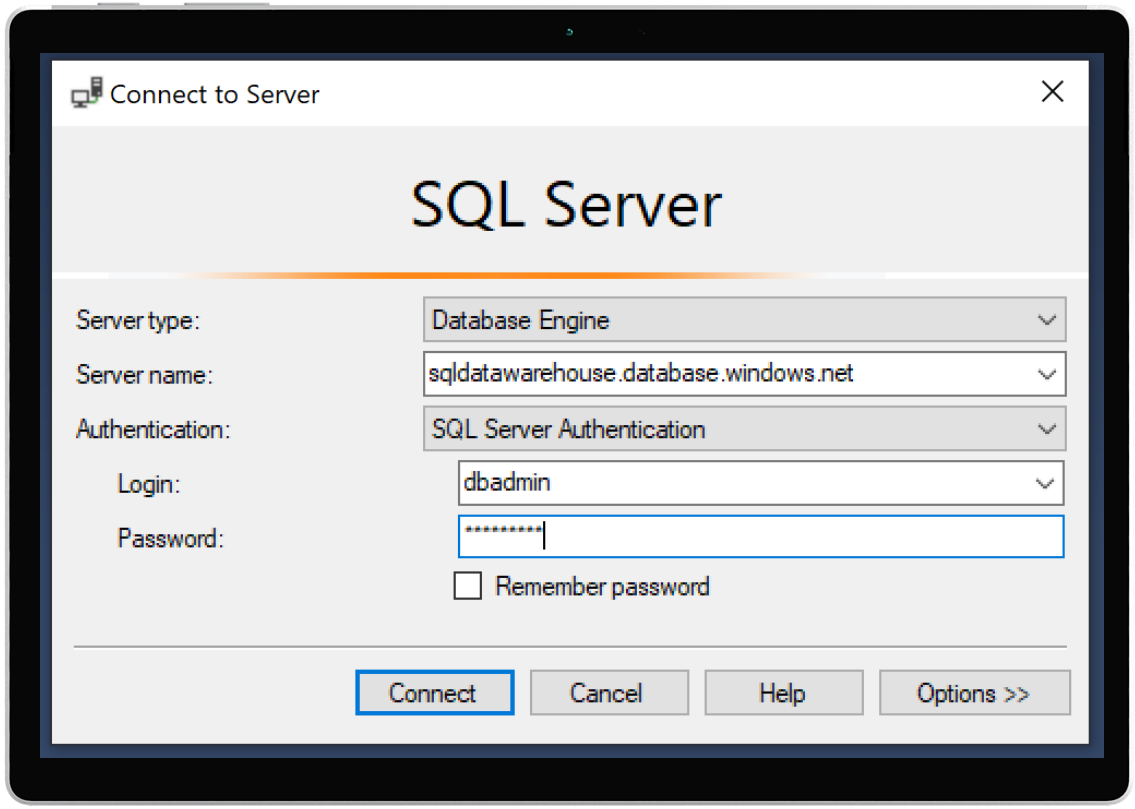
Overview

This authentication method uses a username and password.

When you created the logical server for your data warehouse, you specified a "server admin" login with a username and password.

Using these credentials, you can authenticate to any database on that server as the database owner.

Furthermore, you can create user logins and roles with familiar SQL Syntax.



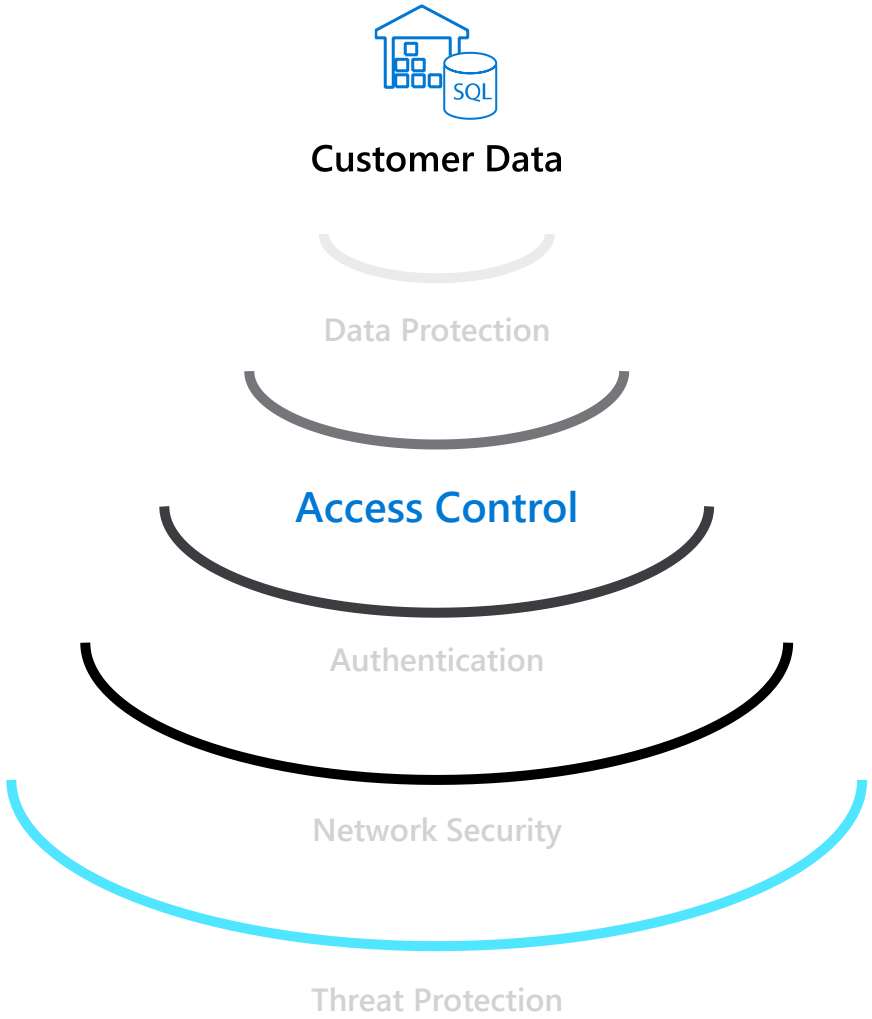

```
-- Connect to master database and create a login
CREATE LOGIN ApplicationLogin WITH PASSWORD = 'Str0ng_password';
CREATE USER ApplicationUser FOR LOGIN ApplicationLogin;

-- Connect to SQL DW database and create a database user
CREATE USER DatabaseUser FOR LOGIN ApplicationLogin;
```

Access Control - Business requirements

How do I restrict access to sensitive data to specific database users?

How do I ensure users only have access to relevant data?

For example, in a hospital only medical staff should be allowed to see patient data that is relevant to them—and not every patient's data.



Object-level security (tables, views, and more)

Overview

GRANT controls permissions on designated tables, views, stored procedures, and functions.

Prevent unauthorized queries against certain tables.

Simplifies design and implementation of security at the database level as opposed to application level.

```
-- Grant SELECT permission to user RosaQdM on table Person.Address in the AdventureWorks2012 database
GRANT SELECT ON OBJECT::Person.Address TO RosaQdM;
GO
-- Grant REFERENCES permission on column BusinessEntityID in view HumanResources.vEmployee to user Wanida
GRANT REFERENCES(BusinessEntityID) ON OBJECT::HumanResources.vEmployee to Wanida with GRANT OPTION;
GO
-- Grant EXECUTE permission on stored procedure HumanResources.uspUpdateEmployeeHireInfo to an application role called Recruiting11
USE AdventureWorks2012;
GRANT EXECUTE ON OBJECT::HumanResources.uspUpdateEmployeeHireInfo TO RECRUITING 11;
GO
```

Row-level security (RLS)

Overview

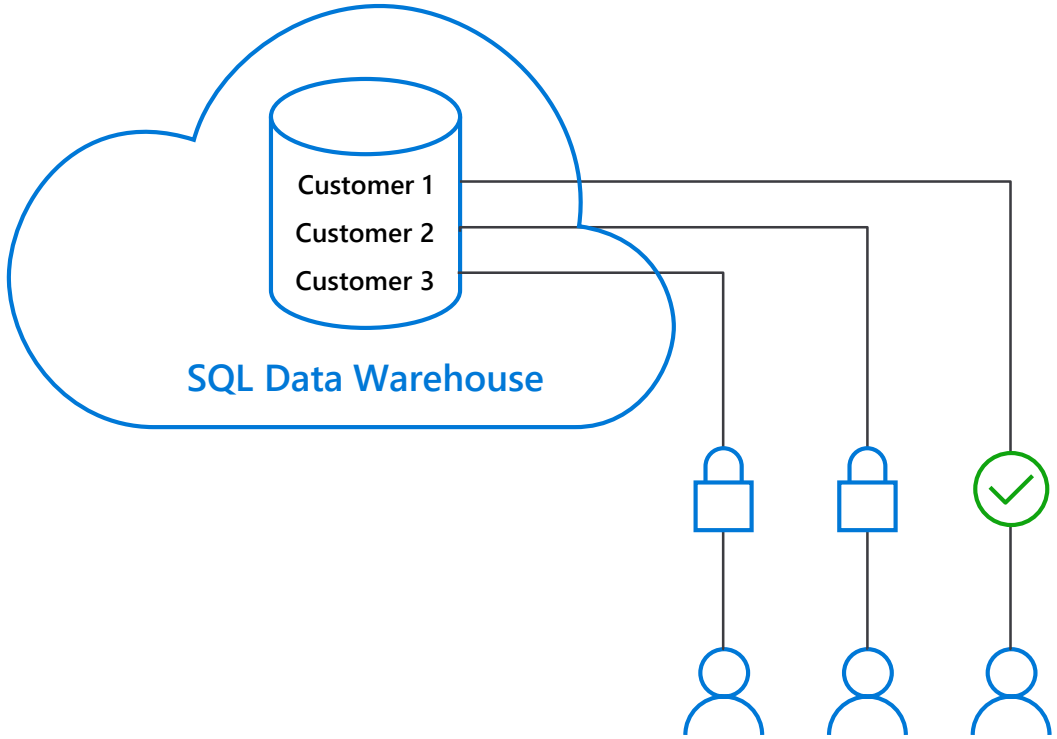
Fine grained access control of specific rows in a database table.

Help prevent unauthorized access when multiple users share the same tables.

Eliminates need to implement connection filtering in multi-tenant applications.

Administer via SQL Server Management Studio or SQL Server Data Tools.

Easily locate enforcement logic inside the database and schema bound to the table.



Row-level security

Creating policies

Filter predicates silently filter the rows available to read operations (SELECT, UPDATE, and DELETE).

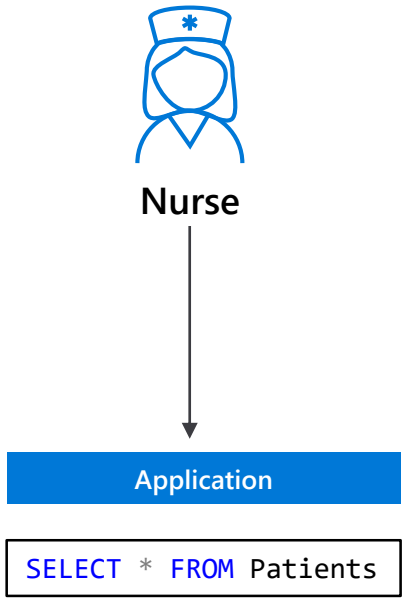
The following examples demonstrate the use of the CREATE SECURITY POLICY syntax

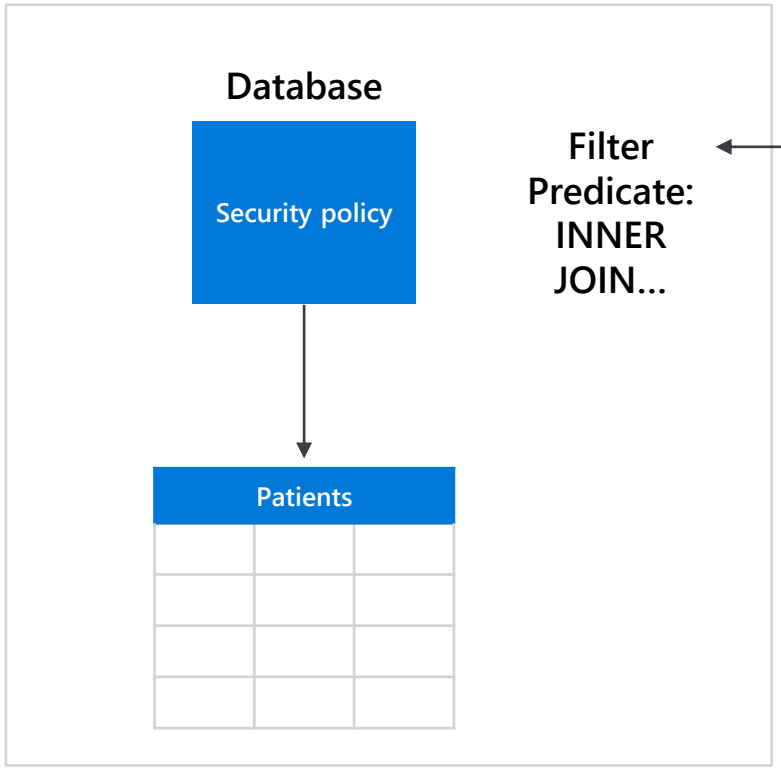
```
-- The following syntax creates a security policy with a filter predicate for the Customer table
CREATE SECURITY POLICY [FederatedSecurityPolicy]
ADD FILTER PREDICATE [rls].[fn_securitypredicate]([CustomerId])
ON [dbo].[Customer];


-- Create a new schema and predicate function, which will use the application user ID stored in CONTEXT_INFO to filter rows.
CREATE FUNCTION rls.fn_securitypredicate (@AppUserId int)
RETURNS TABLE
WITH SCHEMABINDING
AS
RETURN (
SELECT 1 AS fn_securitypredicate_result
WHERE
DATABASE_PRINCIPAL_ID() = DATABASE_PRINCIPAL_ID('dbo') -- application context
AND CONTEXT_INFO() = CONVERT(VARBINARY(128), @AppUserId));
GO
```

Row-level security

Three steps:

1. Policy manager creates filter predicate and security policy in T-SQL, binding the predicate to the patients table.
2. App user (e.g., nurse) selects from Patients table.
3. Security policy transparently rewrites query to apply filter predicate.





Filter Predicate: INNER JOIN...

Policy manager

```
CREATE FUNCTION dbo.fn_securitypredicate(@wing int)
    RETURNS TABLE WITH SCHEMABINDING AS
    return SELECT 1 as [fn_securitypredicate_result] FROM
        StaffDuties d INNER JOIN Employees e
        ON (d.EmpId = e.EmpId)
        WHERE e.UserSID = SUSER_SID() AND @wing = d.Wing;

CREATE SECURITY POLICY dbo.SecPol
    ADD FILTER PREDICATE dbo.fn_securitypredicate(Wing) ON Patients
    WITH (STATE = ON)
```

```
SELECT * FROM Patients
   SEMIJOIN APPLY dbo.fn_securitypredicate(patients.Wing);
```

```
SELECT Patients.* FROM Patients,
   StaffDuties d INNER JOIN Employees e ON (d.EmpId = e.EmpId)
   WHERE e.UserSID = SUSER_SID() AND Patients.wing = d.Wing;
```

Column-level security

Overview

Control access of specific columns in a database table based on customer's group membership or execution context.

Simplifies the design and implementation of security by putting restriction logic in database tier as opposed to application tier.

Administer via GRANT T-SQL statement.

Both Azure Active Directory (AAD) and SQL authentication are supported.

Column-level security

Three steps:

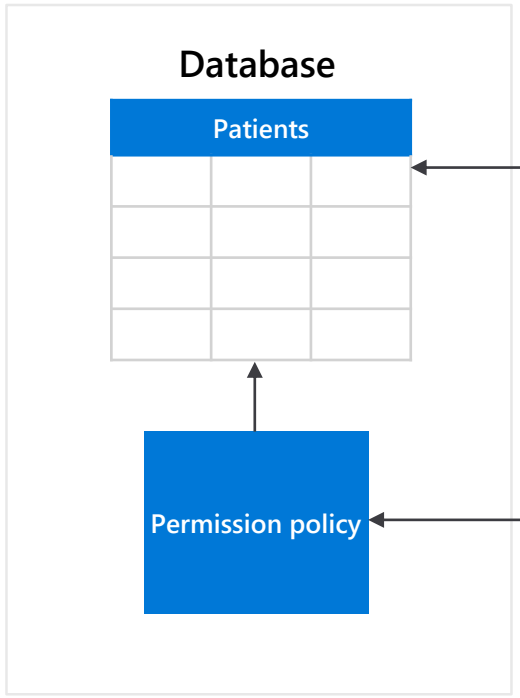
1. Policy manager creates permission policy in T-SQL, binding the policy to the Patients table on a specific group.
2. App user (for example, a nurse) selects from Patients table.
3. Permission policy prevents access on sensitive data.

Nurse

Application

```
SELECT * FROM Membership;

Msg 230, Level 14, State 1, Line 12
The SELECT permission was denied on the column
'SSN' of the object 'Membership', database
'CLS_TestDW', schema 'dbo'.
```

Queries executed as 'Nurse' will fail if they include the SSN column



Policy manager

```
CREATE TABLE Patients (
  PatientID int IDENTITY,
  FirstName varchar(100) NULL,
  SSN char(9) NOT NULL,
  LastName varchar(100) NOT NULL,
  Phone varchar(12) NULL,
  Email varchar(100) NULL
);
```

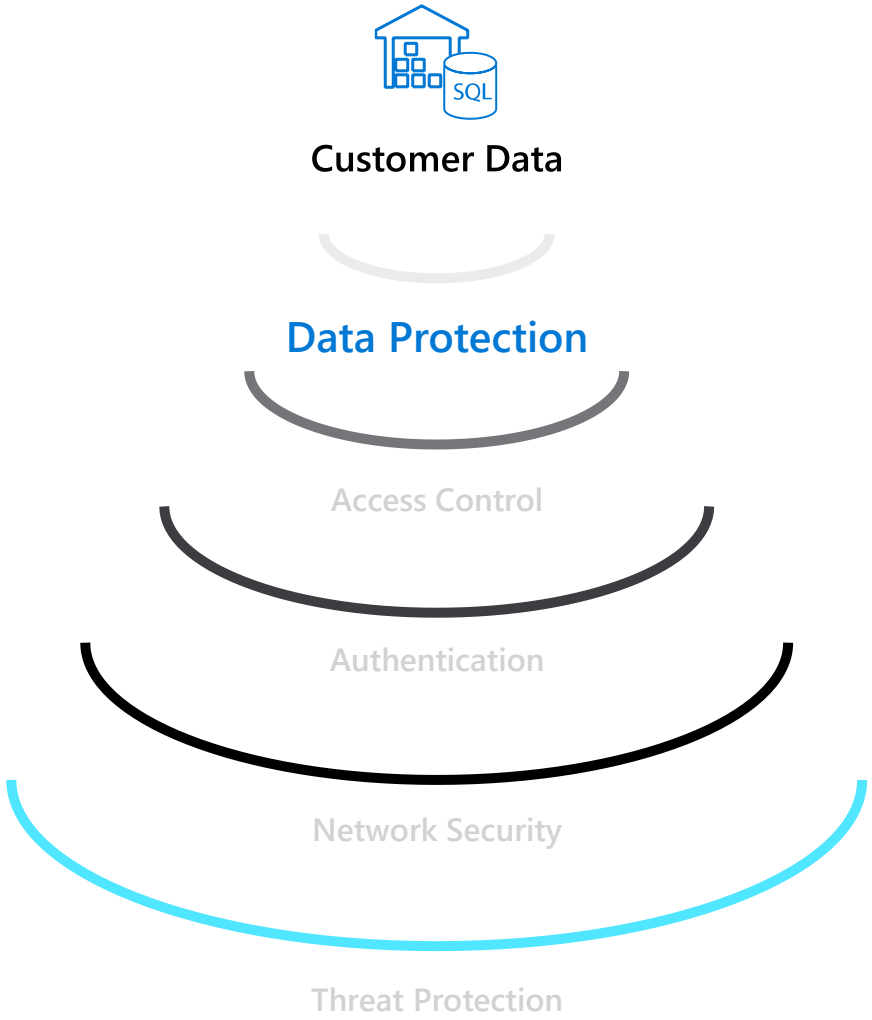
```
GRANT SELECT ON Patients (
  PatientID, FirstName, LastName, Phone, Email
) TO Nurse;
```

Allow 'Nurse' to access all columns except for sensitive SSN column

Data Protection - Business requirements

How do I protect sensitive data against unauthorized (high-privileged) users?

What key management options do I have?



Dynamic Data Masking

Overview

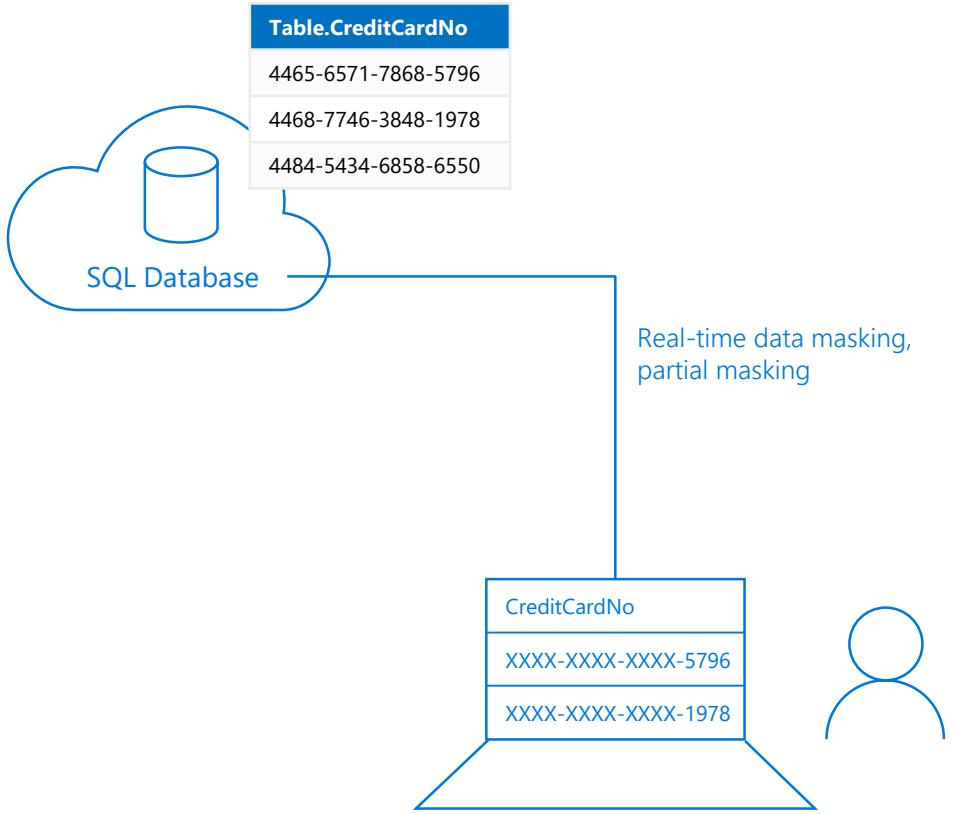
Prevent abuse of sensitive data by hiding it from users

Easy configuration in new Azure Portal

Policy-driven at 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, such as full or partial, for various sensitive data categories (credit card numbers, SSN, etc.)



Dynamic Data Masking

Three steps

1. Security officer defines dynamic data masking policy in T-SQL over sensitive data in the Employee table. The security officer uses the built-in masking functions (default, email, random)

2. The app-user selects from the Employee table

3. The dynamic data masking policy obfuscates the sensitive data in the query results for non-privileged users

Security officer

1

```
ALTER TABLE [Employee]
ALTER COLUMN [SocialSecurityNumber]
ADD MASKED WITH (FUNCTION = 'DEFAULT()')

ALTER TABLE [Employee]
ALTER COLUMN [Email]
ADD MASKED WITH (FUNCTION = 'EMAIL()')

ALTER TABLE [Employee]
ALTER COLUMN [Salary]
ADD MASKED WITH (FUNCTION = 'RANDOM(1,20000)')

GRANT UNMASK to admin1
```

2

Business app

```
SELECT [First Name],
       [Social Security Number],
       [Email],
       [Salary]
FROM   [Employee]
```

3

Non-masked data (admin login)

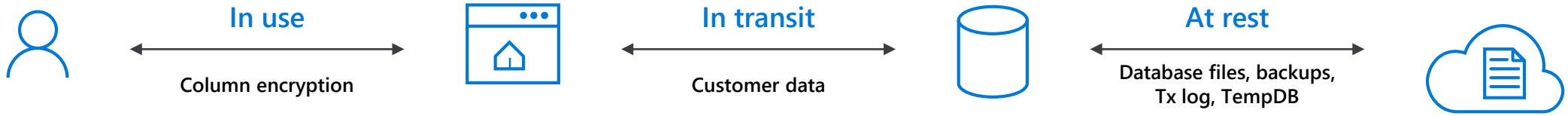
	First Name	Social Security Num...	Email	Salary
1	LILA	758-10-9637	lila.barnett@comcast.net	1012794
2	JAMIE	113-29-4314	jamie.brown@ntlworld.com	1025713
3	SHELLEY	550-72-2028	shelley.lynn@charter.net	1040131
4	MARCELLA	903-94-5665	marcella.estrada@comcast.net	1040753
5	GILBERT	376-79-4787	gilbert.juarez@verizon.net	1041308

Masked data (admin1 login)

	First Name	Social Security Number	Email	Salary
1	LILA	XXX-XX-XX37	lXX@XXXX.net	8940
2	JAMIE	XXX-XX-XX14	jXX@XXXX.com	19582
3	SHELLEY	XXX-XX-XX28	sXX@XXXX.net	3713
4	MARCELLA	XXX-XX-XX65	mXX@XXXX.net	11572
5	GILBERT	XXX-XX-XX87	gXX@XXXX.net	4487

Types of data encryption

Data Encryption	Encryption Technology	Customer Value
In transit	Transport Layer Security (TLS) from the client to the server TLS 1.2	Protects data between client and server against snooping and man-in-the-middle attacks
At rest	Transparent Data Encryption (TDE) for Azure SQL Data Warehouse	Protects data on the disk User or Service Managed key management is handled by Azure, which makes it easier to obtain compliance



Transparent data encryption (TDE)

Overview

All customer data encrypted at rest

TDE performs real-time I/O encryption and decryption of the data and log files.

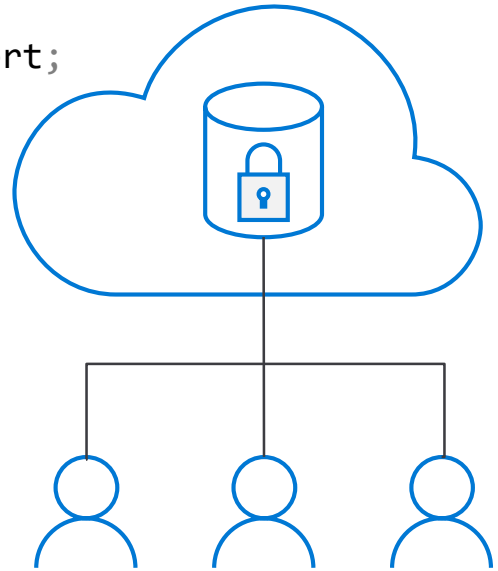
Service OR User managed keys.

Application changes kept to a minimum.

Transparent encryption/decryption of data in a TDE-enabled client driver.

Compliant with many laws, regulations, and guidelines established across various industries.

```
USE master;
GO
CREATE MASTER KEY ENCRYPTION BY PASSWORD = '<UseStrongPasswordHere>';
go
CREATE CERTIFICATE MyServerCert WITH SUBJECT = 'My DEK Certificate';
go
USE MyDatabase;
GO
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM = AES_128
ENCRYPTION BY SERVER CERTIFICATE MyServerCert;
GO
ALTER DATABASE MyDatabase
SET ENCRYPTION ON;
GO
```



Transparent data encryption (TDE)

Key Vault

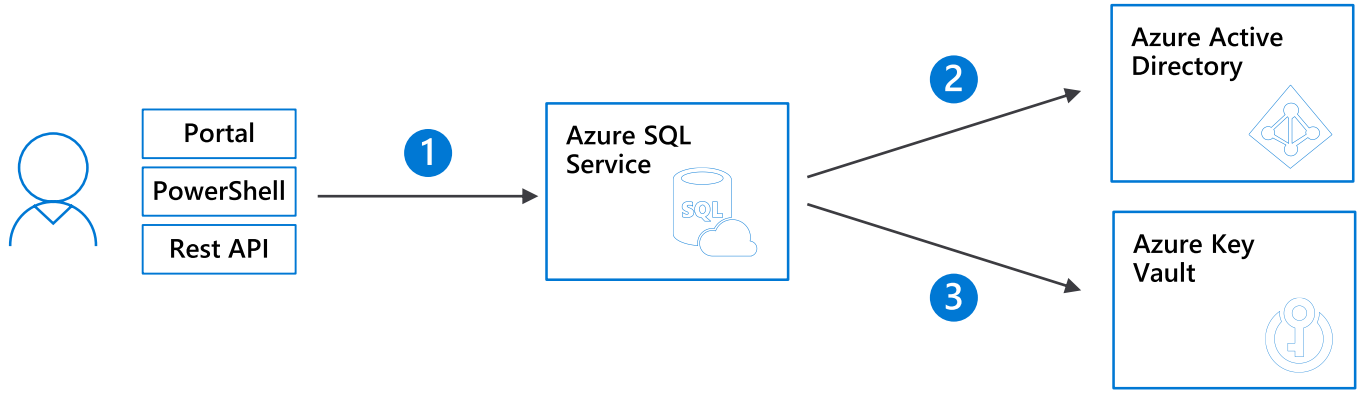
Benefits with User Managed Keys

Assume more control over who has access to your data and when.

Highly available and scalable cloud-based key store.

Central key management that allows separation of key management and data.

Configurable via Azure Portal, PowerShell, and REST API.



1. The Key Vault admin grants vault access to the SQL Database server using its unique Azure Active Directory (AD) identity
2. The server uses its Azure AD identity to authenticate with Azure AD for access to your Key Vault
3. The server sends get, wrap key, and unwrap key request to the asymmetric key in key Vault for database encryption key protection.

Industry-leading security

Category	Feature	SQL Data Warehouse	Amazon Redshift	Snowflake	Google Big Query
Data Protection	Data In Transit	Yes	Yes	Yes	No
	Data encryption at rest (Service & User Managed Keys)	Yes	Yes	Yes	Yes
	Data In Use (Always Encrypted)	No	No	No	No
	Data Discovery and Classification	Yes	No	No	No
Access Control	Native Row Level Security	Yes	No	No	No
	Table and View Security (GRANT / DENY)	Yes	Yes	Yes	Yes
	Column Level Security	Yes	No	No	No
Authentication	SQL Authentication	Yes	Yes	Yes	No
	Native Azure Active Directory	Yes	No	No	No
	Integrated Security	Yes	Yes	Yes	Yes
	Multi-Factor Authentication	Yes	Yes	Yes	Yes
Network Security	Virtual Network (VNET)	Yes	Yes	Yes	Yes
	SQL Firewall (server)	Yes	Yes	Yes	No
	Integration with ExpressRoute	Yes	No	No	No
Threat Protection	SQL Threat Detection	Yes	Yes	No	No
	SQL Auditing	Yes	Yes	No	Yes
	Vulnerability Assessment	Yes	Yes	No	No

Bringing parity between Azure SQL Database and Azure SQL Data Warehouse

*Release timelines subject to change

Category	Capability	Azure SQL Database	Azure SQL Data Warehouse
Data Protection	Transparent Data Encryption (TDE) – Service, User-managed keys	Generally Available	Generally Available
	Column-level Encryption	Generally Available	CY2019H2
	Always Encrypted	Generally Available	Future
	Dynamic Data Masking	Generally Available	CY2019H2
	Data Discovery and Classification	Public Preview	Coming soon (Mar'2019)
Access Control	Row-level security	Generally Available	Generally Available
	Column-level Security	Generally Available	Generally Available
Authentication	SQL Authentication	Generally Available	Generally Available
	Azure Active Directory Authentication (w/ MFA)	Generally Available	Generally Available
Network Security	Virtual Network (VNet) – Service Endpoints	Generally Available	Generally Available
	Virtual Network (VNet) – Private Link	Public Preview (CY2019H1)	Public Preview (CY2019H1)
	SQL Firewall (server- and database-level)	Generally Available	Generally Available
Threat Protection	SQL Threat Detection	Generally Available	Generally Available
	SQL Auditing	Generally Available	Generally Available
	Vulnerability Assessment	Generally Available	Generally Available