



Manage Security for Azure SQL Database

Module 4



Learning Units covered in this Module

- Lesson 1: Introduction to Azure SQL Database Security
- Lesson 2: Implement Azure Active Directory Security
- Lesson 3: Manage Logins in Azure SQL Database
- Lesson 4: Implement Firewall Rules and Virtual Networks
- Lesson 5: Implement Transparent Data Encryption
- Lesson 6: Implement Always Encrypted
- Lesson 7: Implement Row Level Security
- Lesson 8: Implement Dynamic Data Masking
- Lesson 9: Implement Auditing for Azure SQL Database
- Lesson 10: Data Discovery and Classification
- Lesson 11: Implement Microsoft Defender for SQL

Lesson 1: Introduction to Azure SQL Database Security

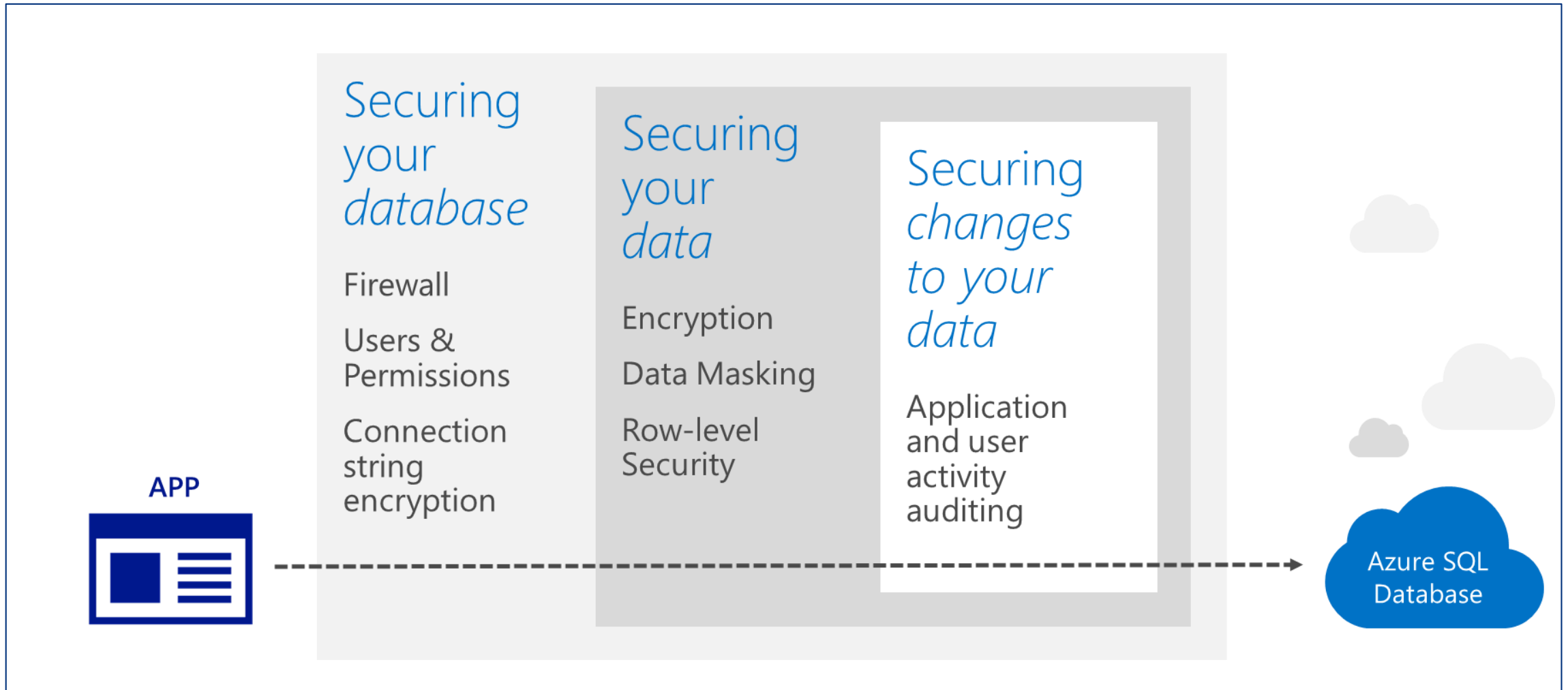
Objectives

After completing this learning, you will be able to:

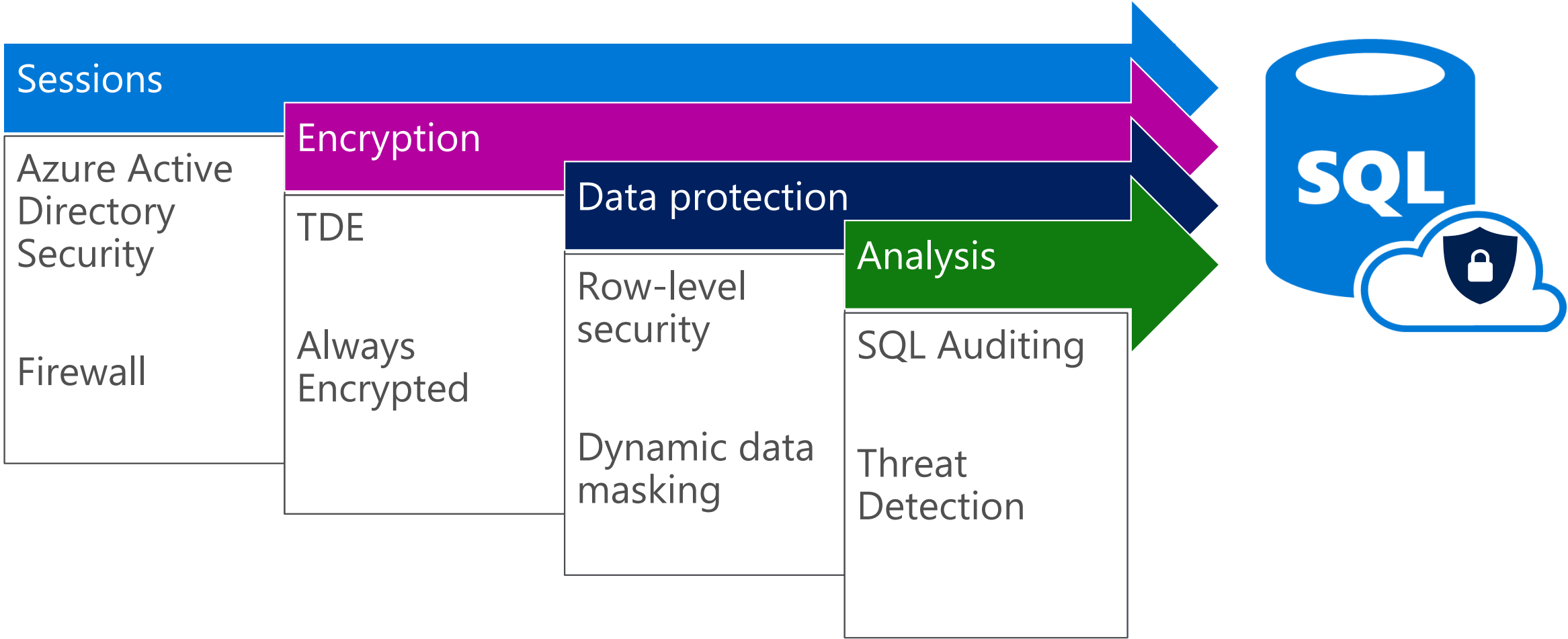
- Know the various options to manage security for an Azure SQL Database.



Azure SQL Database Security Layers



Security Features for Azure SQL DB



The Security Gold Standard



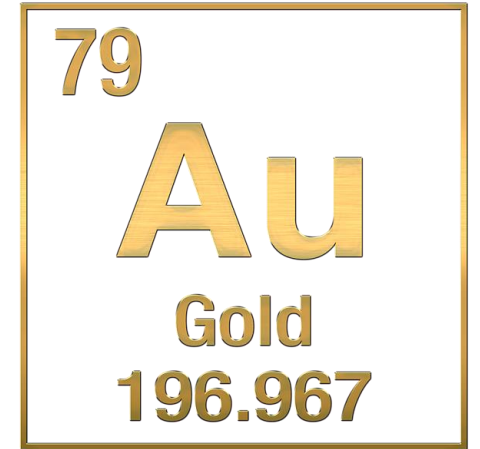
AUTHENTICATION – Verifies who you are



AUTHORIZATION – Assigns what you can do

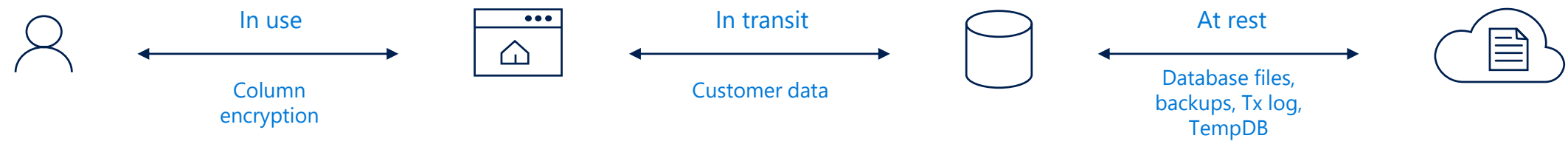


AUDITING – Monitors what you did



Types of data encryption

Data encryption	Encryption technology	Customer value
In transit	Transport Layer Security (TLS) from the client to the server.	Protects data between client and server against snooping and man-in-the-middle attacks. NOTE: Azure SQL Database is phasing out Secure Sockets Layer (SSL) 3.0 and TLS 1.0 in favor of TLS 1.2.
At rest	Transparent Data Encryption (TDE) for Azure SQL Database.	Protects data on the disk. Key management is done by Azure, which makes it easier to obtain compliance.
In use (end-to-end)	Always Encrypted for client-side column encryption.	Data is protected end-to-end, but the application is aware of encrypted columns. This is used in the absence of data masking and TDE for compliance-related scenarios.



Questions?



Knowledge Check

List the security features available for Azure SQL Database.

Name the feature to encrypt the data both at rest and motion.

Lesson 2: Implement Entra ID Security

Objectives

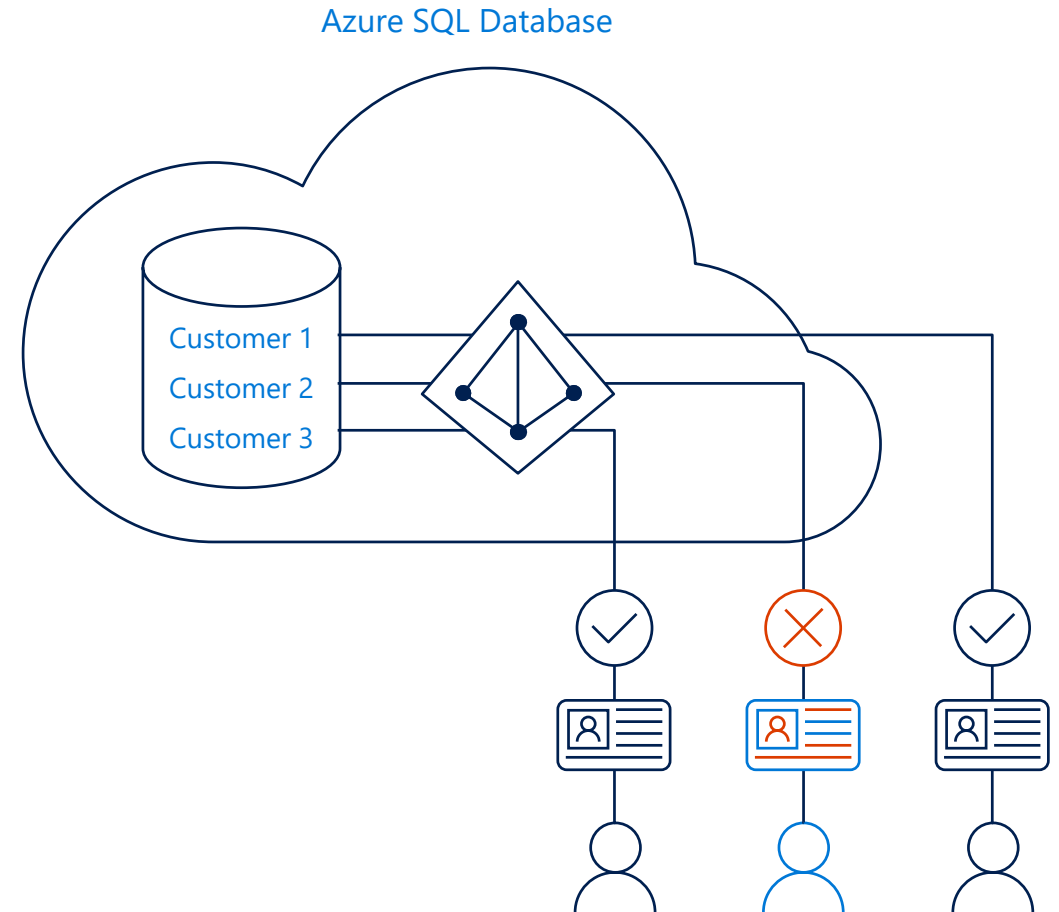
After completing this learning, you will be able to:

- Know how to leverage Azure Active Directory security for authenticating connections to an Azure SQL Database.

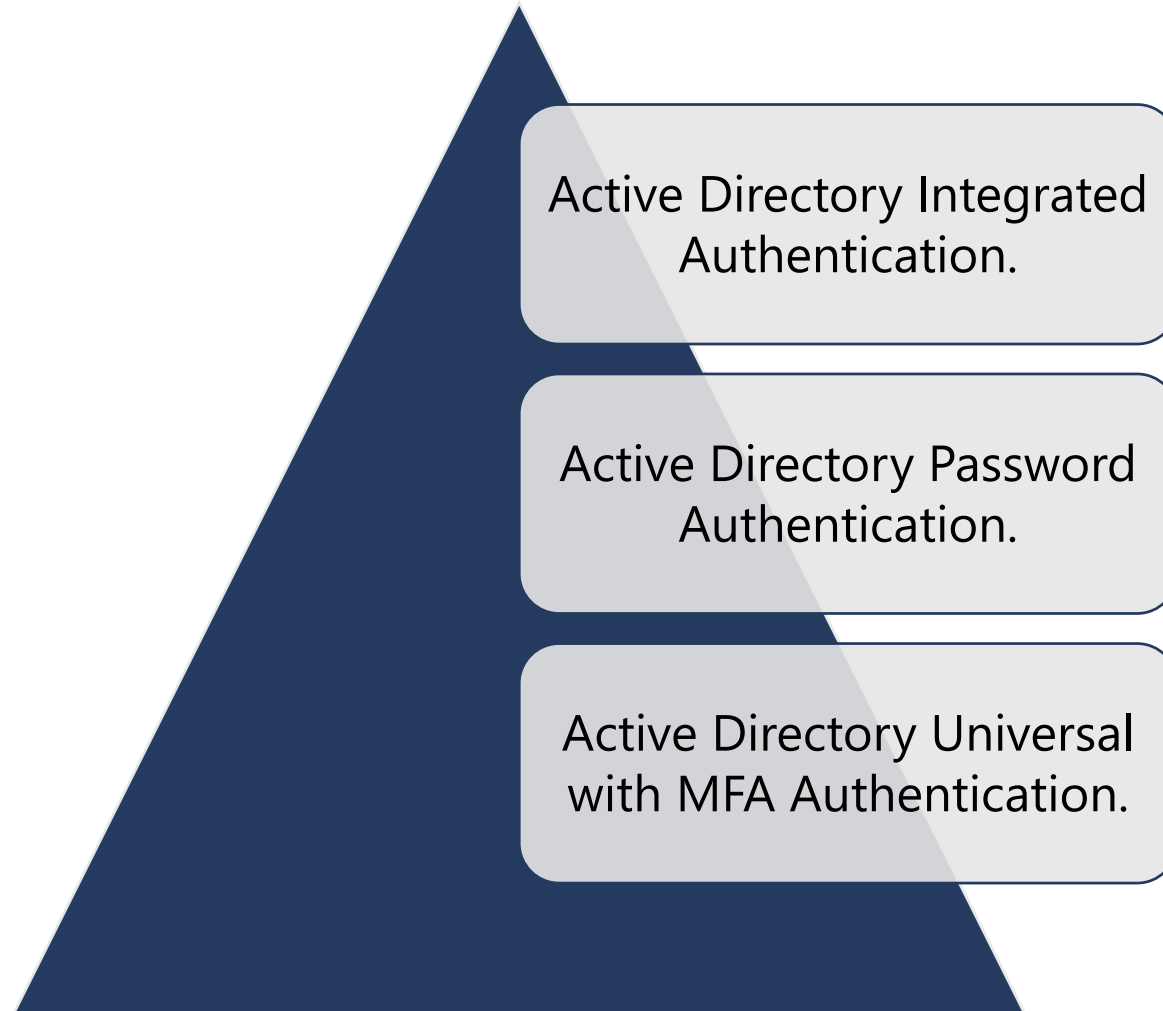


Azure Active Directory(AAD) Security

Azure Active Directory authentication is a mechanism of connecting to Microsoft Azure SQL Database by using identities in Azure Active Directory (Azure AD)



Three Types of AAD Authentication



Benefits of AAD Authentication

Centrally manage user permissions.

Alternative to SQL Server authentication.

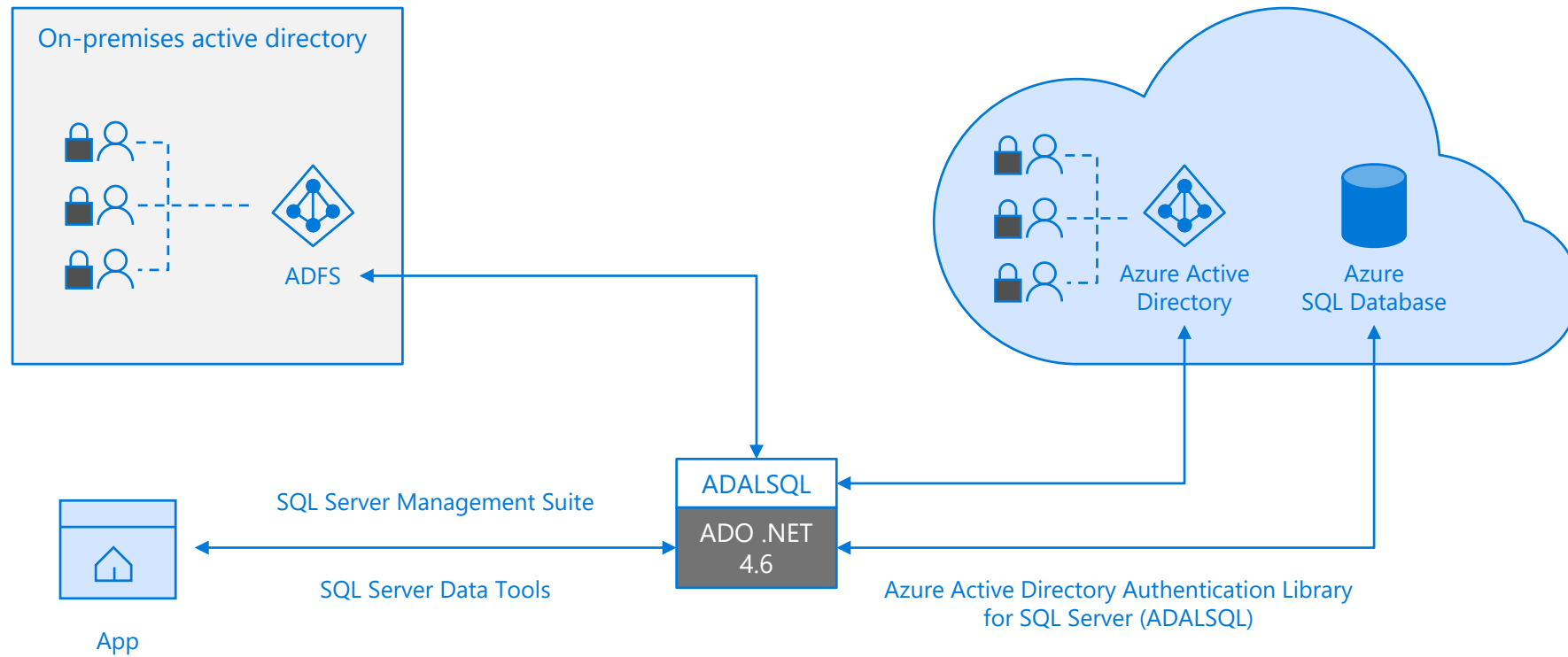
Allows password rotation in a single place.

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

Stops password storing by using integrated Windows authentication and other forms of authentication supported by AAD.

Trust architecture

Azure Active Directory and Azure SQL Database



Demonstration

Implement AAD Authentication

- Connect to Azure Active Directory.
- Connect to Azure SQL DB using SSMS through AAD authentication.



Questions?



Knowledge Check

List three benefits of Azure Activity Directory Authentication.

Can we use Windows authentication for Azure SQL Database?

Lesson 3: Manage Logins in Azure SQL Database

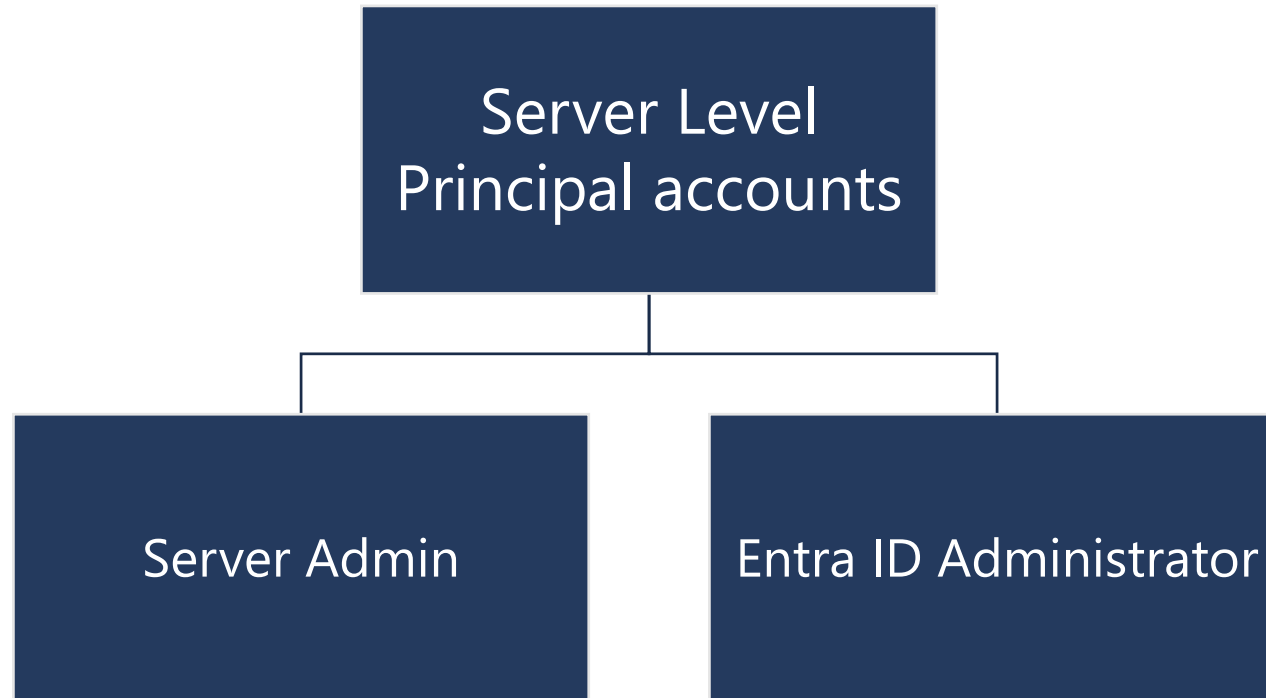
Objectives

After completing this learning, you will be able to:

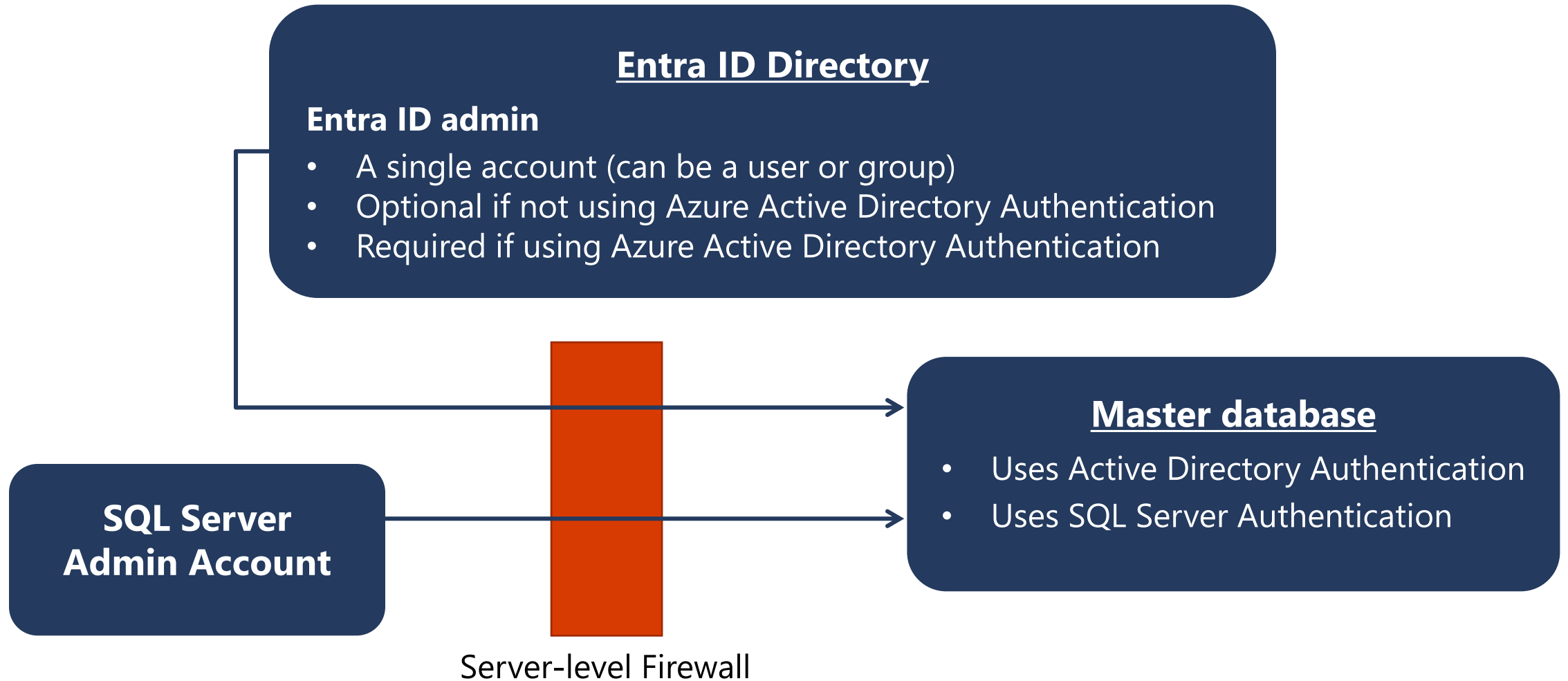
- Manage logins within Azure SQL Database.



Unrestricted Administrative Accounts



Administrator Access Path



Additional Special Roles

Database Creators

- ALTER ROLE dbmanager* ADD MEMBER Mary;
- ALTER ROLE dbmanager* ADD MEMBER [mike@contoso.com];

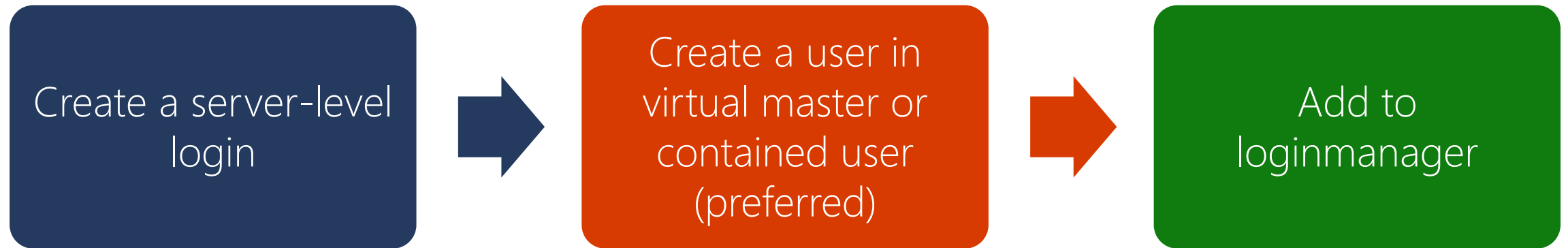


*dbmanager is a database role in virtual master database.

Additional Special Roles (continued)

Login Managers

- ALTER ROLE loginmanager* ADD MEMBER Mary;
- ALTER ROLE loginmanager* ADD MEMBER [mike@contoso.com];



*loginmanager is a database role in virtual master database.

Non-administrator Users

- Generally, non-administrator accounts do not need access to the virtual master database.
- Create contained database users at the database level.

Options:

Azure Active Directory
authentication
contained database
user.

SQL Server
authentication
contained database
user.

SQL Server
authentication user
based on a SQL Server
authentication login.

Groups and Roles

Azure Active Directory authentication

- Put Azure Active Directory users into an Azure Active Directory group.
- Create a contained database user for the group.
- Place one or more database users into a database role.
- Assign permissions to the database role.

SQL Server authentication

- Create contained database users in the database.
- Place one or more database users into a database role.
- Assign permissions to the database role.

Database Roles

The database roles can be the built-in roles such as:

db_owner

db_ddladmin

db_datawriter

db_datareader

db_denydatawriter

db_denydatareader

Naming Requirements

Certain usernames are not allowed for security reasons. You cannot use the following names:



admin

administrator

guest

root

sa

Demonstration

Connect to an Azure SQL DB using SQL Authentication

- Using SQL Login + SQL User.
- Using Contained Database User.



Questions?



Knowledge Check

Name the two unrestricted admin accounts for Azure SQL Database?

Name the Additional server-level administrative roles for Azure SQL Database?

Lesson 4: Implement Firewall Rules and Virtual Networks

Objectives

After completing this learning, you will be able to:

- Configure firewall rules on server and database level
- Configure virtual networks on your logical SQL Server



Securing your database with firewalls

Initially, all access to your Azure SQL Database server is blocked by the firewall.

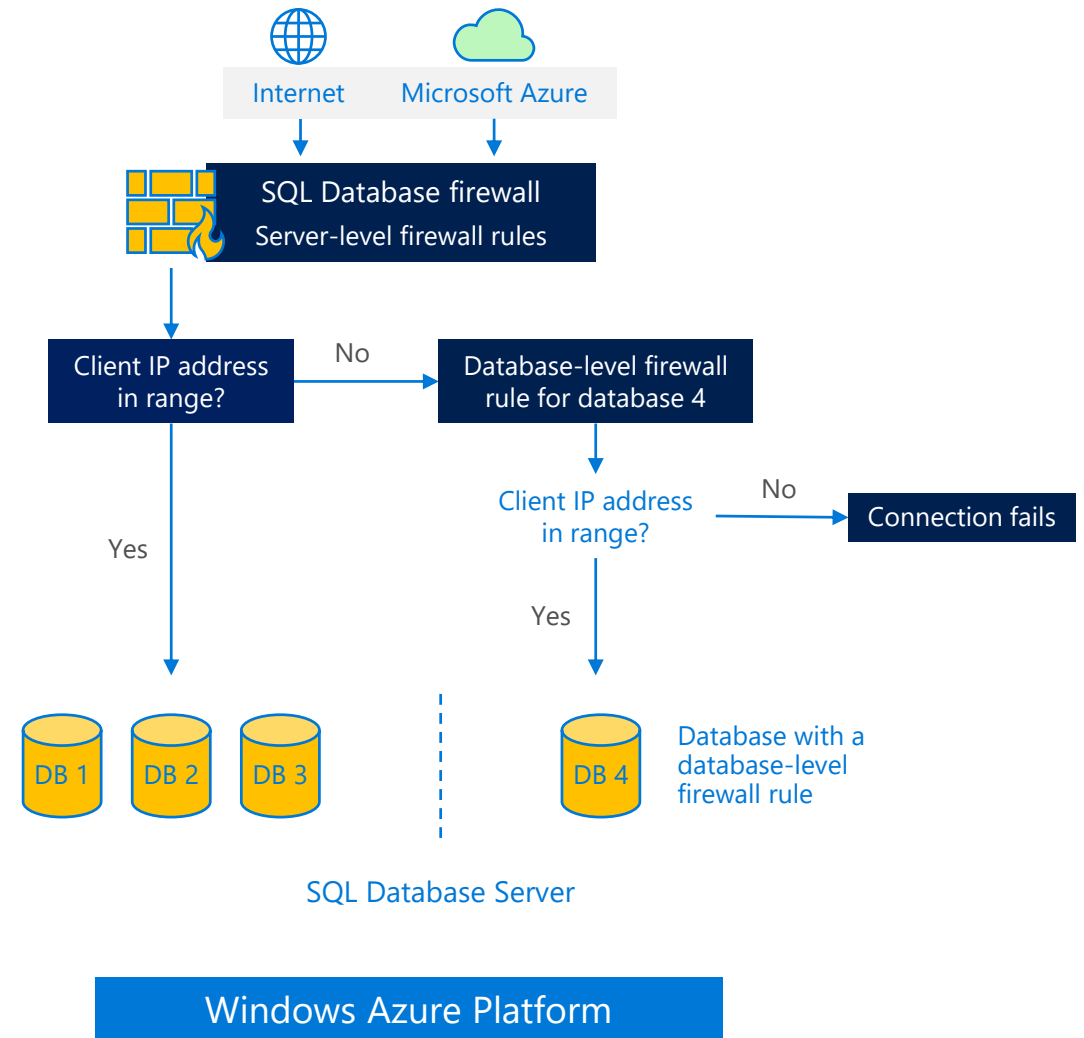
In order to begin using your Azure SQL Database server, you must go to the Management Portal.

Server-level firewall rules enable clients to access all the databases within the same logical server.

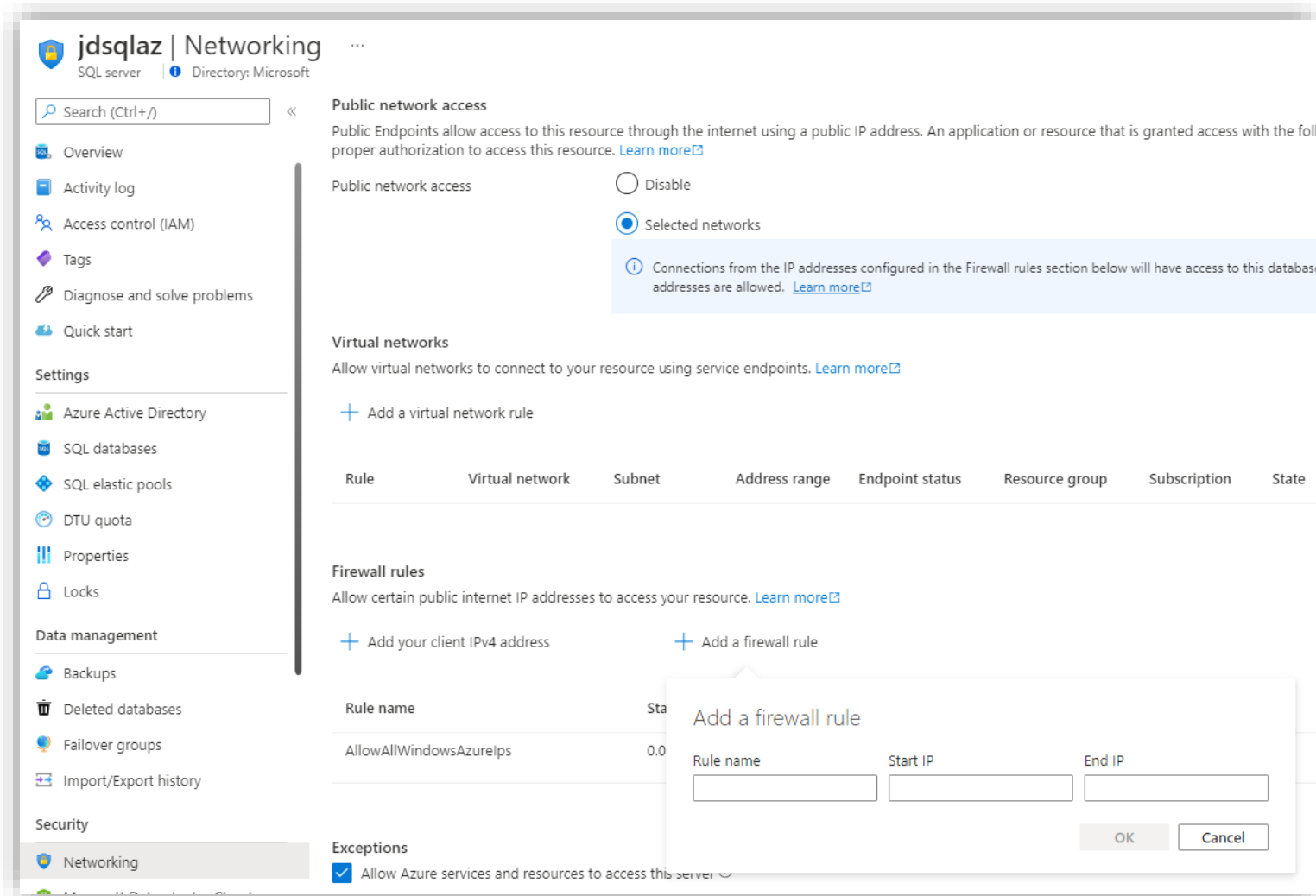
Database-level firewall rules enable clients to access certain databases within the same logical server.

Database-level firewall rules for master and user databases can only be created and managed by using Transact-SQL statements and only after you have configured the first server-level firewall.

Microsoft recommends using database-level firewall rules whenever possible to enhance security and to make your database more portable.



Firewall configuration using portal



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

Enable in the following ways in Azure portal:

- Security -> Networking

Firewall configuration using PowerShell/T-SQL

Manage SQL Database firewall rules using code

- **Windows PowerShell Azure cmdlets**

- Get-AzSqlServerFirewallRule
- New-AzSqlServerFirewallRule
- Set-AzSqlServerFirewallRule
- Remove-AzSqlServerFirewallRule

- **Transact SQL**

- sys.firewall_rules
- sp_set_firewall_rule
- sp_delete_firewall_rule
- sys.database_firewall_rules
- sp_set_database_firewall_rule
- sp_delete_database_firewall_rule

```
# PS Enable Azure connections
```

```
PS C:\>New-AzSqlServerFirewallRule -  
ResourceGroupName "ResourceGroup01" -ServerName  
"Server01" -FirewallRuleName "Rule01" -  
StartIpAddress "192.168.0.198" -EndIpAddress  
"192.168.0.199"
```

```
# PS Allow external IP access to SQL Database
```

```
PS C:\> New-AzureSqlDatabaseServerFirewallRule -  
ServerName "Server01" -RuleName "FirewallRule" -  
StartIpAddress 10.1.1.1 -EndIpAddress 10.1.1.2
```

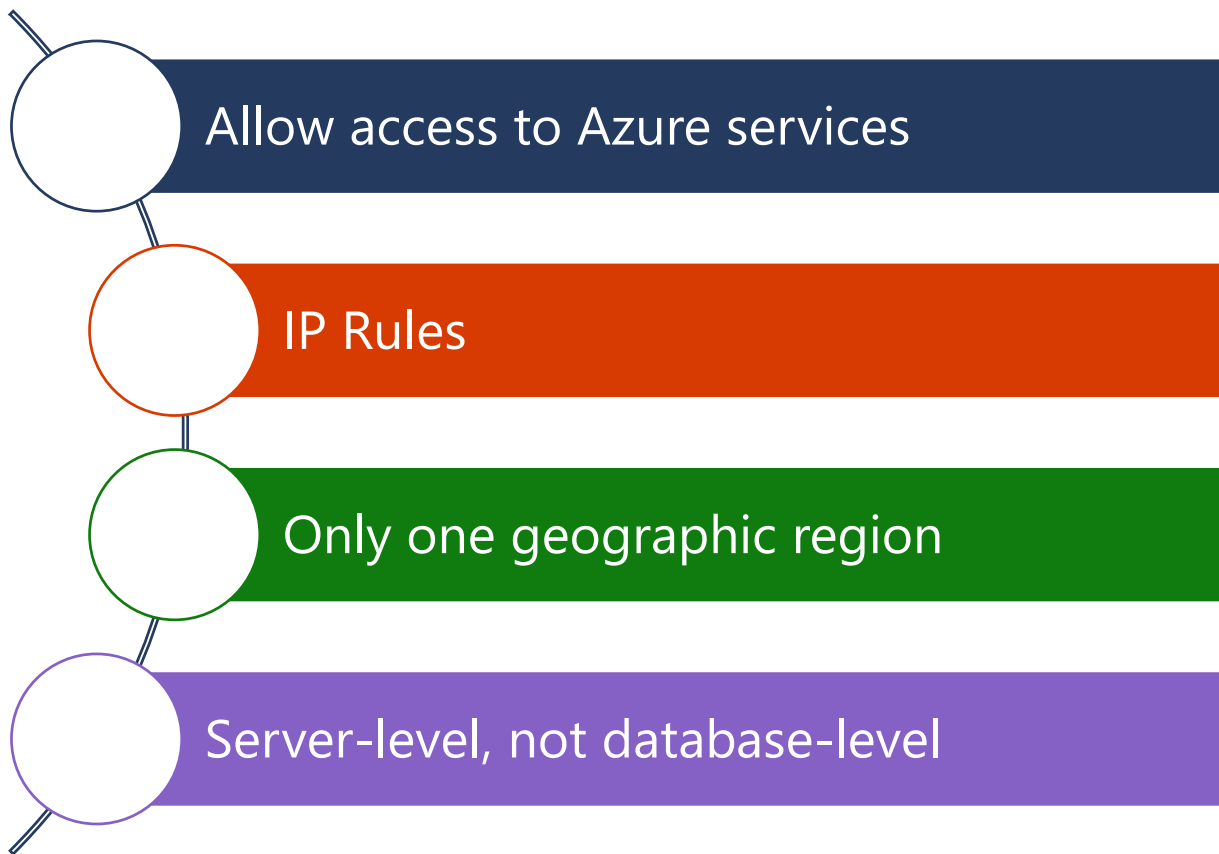
```
-- T-SQL Enable Azure connections
```

```
sp_set_firewall_rule N'Allow Windows Azure',  
'0.0.0.0', '0.0.0.0'
```

```
-- T-SQL Allow external IP access to SQL Database
```

```
sp_set_firewall_rule N'myRule1',  
'12.1.1.1', '12.1.1.2'
```

Virtual Network service endpoints



Create/Update ✕
virtual network rule

Name * ⓘ
SQLDB_Endpoint ✓
provide vnet rule name

Subscription * ⓘ
PFE Subscription ▼

Virtual network * ⓘ
SQLDB_VNet ▼

Subnet name / Address prefix * ⓘ
AzureSQLDB / 10.0.1.0/24 ▼

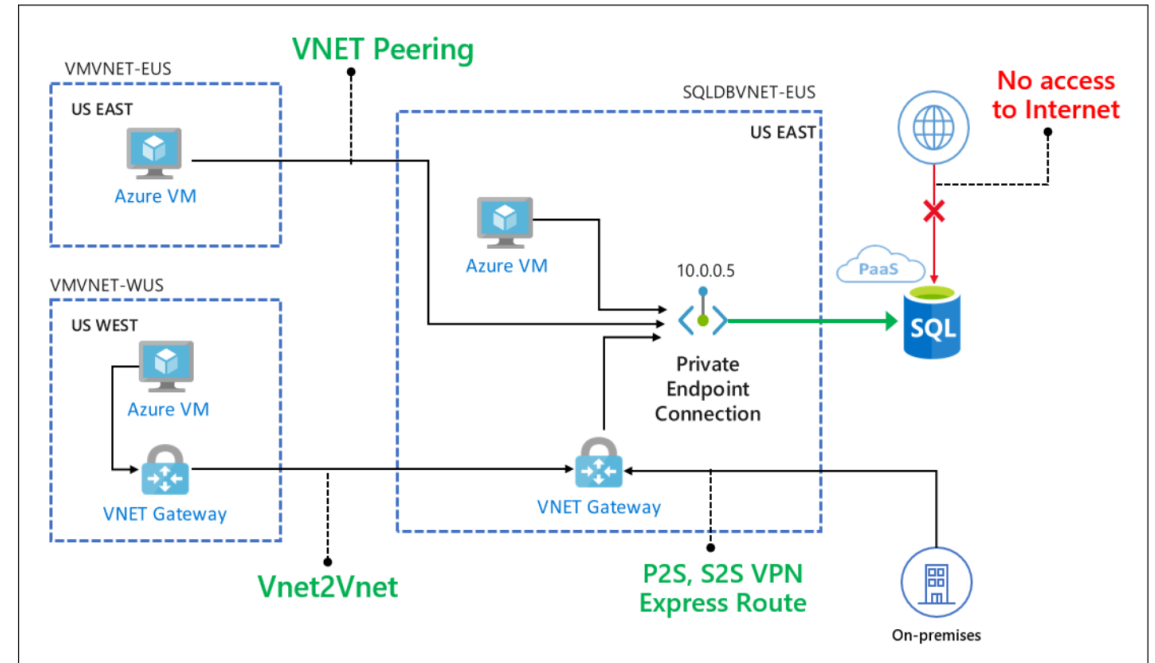
Virtual network	Service endpoint status
SQLDB_VNet/AzureSQLDB	Enabled

Private Link* for Azure SQL Database

Connection via a private endpoint, that is a private IP address within a specific VNet and Subnet.

Enable cross-premises access to the private endpoint using ExpressRoute, private peering, or VPN tunneling.

Subsequently all access via public endpoint can be disabled and not need to use the IP-based firewall.



*Private Link is currently in preview.

Questions?



Knowledge Check

True or False? Initially, all access to your Azure SQL Database server is blocked by the firewall?

Can you use the Azure Portal to configure database-level firewall rules?

Why should you use Virtual Network Service Endpoints?

Lesson 5: Implement Transparent Data Encryption

Objectives

After completing this learning, you will be able to:

- Know how to secure data at rest using Transparent Data Encryption.



Understanding TDE Functionality

Data is encrypted at rest.

Encryption keys are managed by Azure.

Performs real-time I/O encryption and decryption of the data at the page level.

Each page is decrypted when it's read into memory and then encrypted before being written to disk.

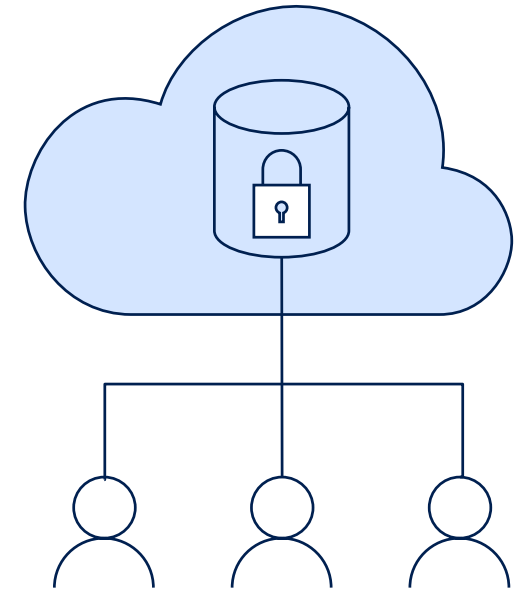
TDE is enabled for all newly deployed Azure SQL Databases.

No need for application change.

Support for equality operations (including joins) on encrypted data.

Bring Your Own Key (BYOK) supported.

SQL Database



Encryption Keys

Service-managed transparent data encryption

- The database encryption key is protected by a built-in server certificate.
- Unique for each server.
- Primary and geo-secondary database are protected by the primary database's parent server key.
- Microsoft automatically rotates these certificates at least every 90 days.

Bring Your Own Key

- Take control over your transparent data encryption keys and control who can access them and when.
- Azure Key Vault.
- You set the asymmetric key at the server level, and all databases under that server inherit it.
- You can control key management tasks such as key rotations and key vault permissions.

Enable TDE Using Azure Portal

Home > Azure SQL > jdsqcentral | SQL databases > SalesDB (jdsqcentral/SalesDB)

SalesDB (jdsqcentral/SalesDB) | Data Encryption

SQL database

Search Save Discard Feedback

- Overview
- Activity log
- Tags
- Diagnose and solve problems
- Query editor (preview)
- Resource visualizer
- Settings
- Data management
- Integrations
- Power Platform
- Security
 - Auditing
 - Ledger
 - Data Discovery & Classification
 - Dynamic Data Masking
 - Microsoft Defender for Cloud
 - Identity
 - Data Encryption**

Transparent data encryption

Always encrypted

Transparent data encryption encrypts your databases, backups, and logs at rest without any changes to your application. To enable encryption, go to each database. [Learn more](#)

Data encryption **ON** OFF

Encryption status ✓ Encrypted

Once database level customer-managed key is selected, switching to server level encryption key is only possible if the server is configured with service managed key.

Transparent data encryption ⓘ

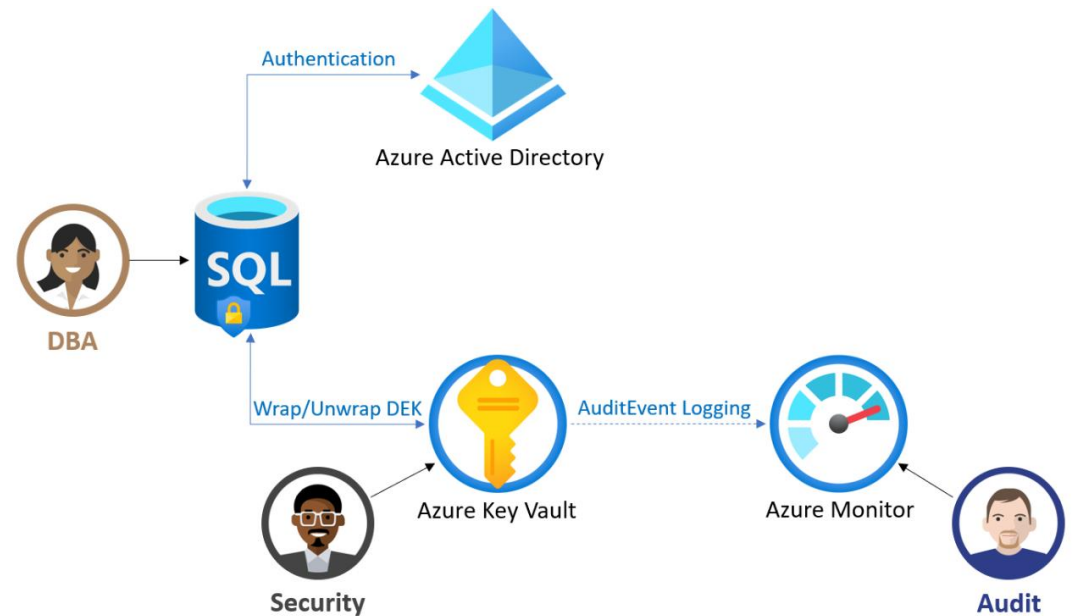
- ☒ Server level encryption key
- ☐ Database level customer-managed key (CMK)

TDE with customer-managed key (BYOK)

You are responsible for and in a full control of a key lifecycle management (key creation, upload, rotation, deletion), key usage permissions, and auditing of operations on keys.

The key used for encryption of the Database Encryption Key (DEK), called TDE protector, is a customer-managed asymmetric key stored in a customer-owned and customer-managed Azure Key Vault (AKV), a cloud-based external key management system.

TDE protector is set at the logical server level and is inherited by all encrypted databases associated with that server.



Demonstration

Implement TDE using Azure Portal and T-SQL Code

- Enable TDE With Bring Your Own Key using Azure Portal.



Questions?



Knowledge Check

Does TDE encrypt the data in motion?

What kind of application changes are required to use TDE?

Which 2 types of Encryption Keys can be used for TDE?

Lesson 6: Implement Always Encrypted

Objectives

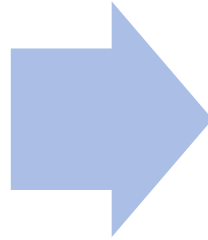
After completing this learning, you will be able to:

- Know how to secure data at rest and in motion using Always encrypted.



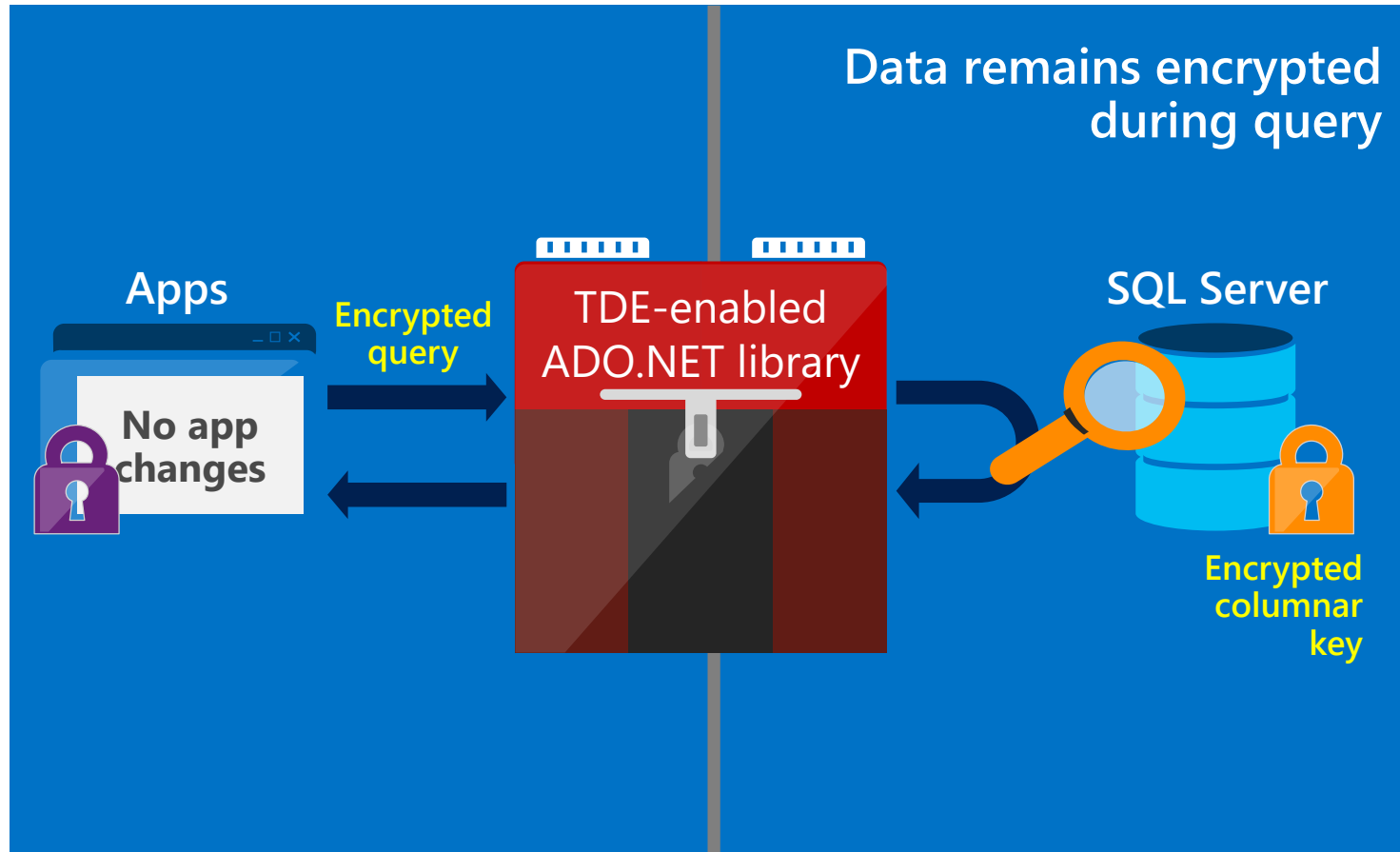
Always Encrypted

Always Encrypted allows clients to encrypt sensitive data inside client applications and never reveal the encryption keys to SQL Database.



As a result, Always Encrypted provides a separation between those who own the data (and can view it) and those who manage the data (but should have no access).

Understanding Always Encrypted Functionality



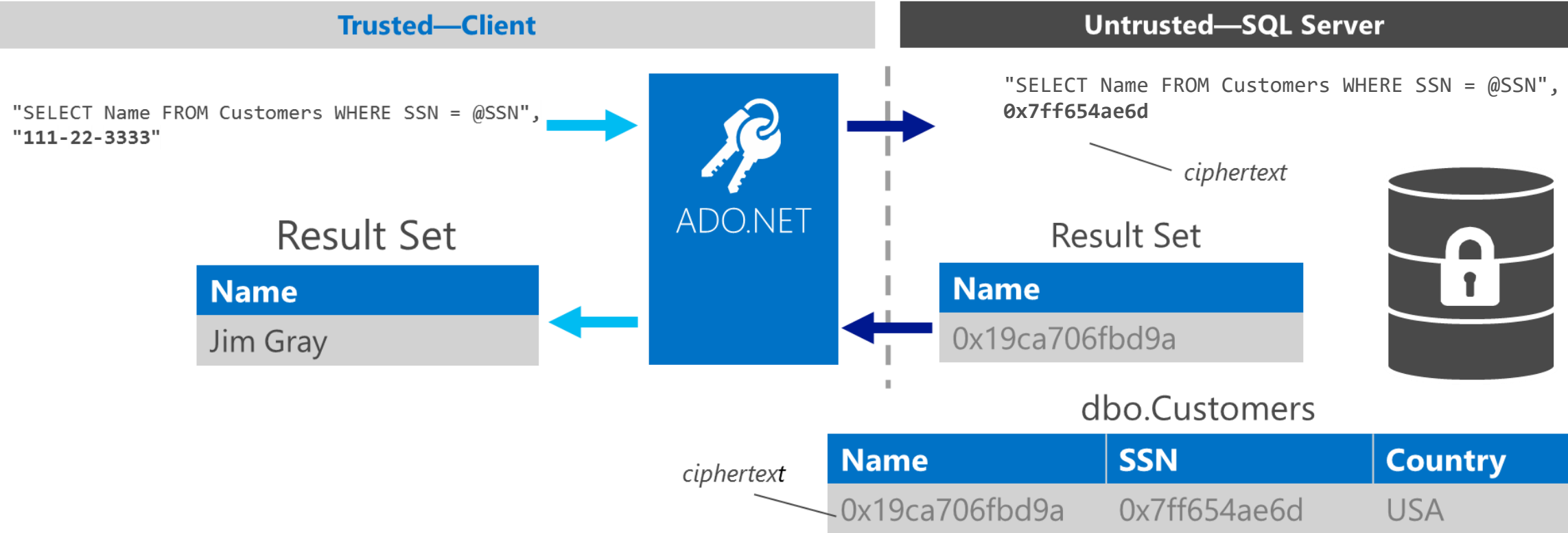
Capability

- Transparent client-side encryption, while SQL Server executes T-SQL queries on encrypted data.

Benefits

- Sensitive data remains encrypted and query-able at all times.
- Unauthorized users never have access to data or keys.
- No changes to applications are necessary.

Understanding Always Encrypted Functionality (Contd.)



Encryption Methodologies

Two types of encryption
are available:



```
graph TD; A[Two types of encryption are available:] --> B[Randomized encryption]; A --> C[Deterministic encryption];
```

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

Deterministic encryption
uses method that always
generates the same
encrypted value for any
given plain text value.

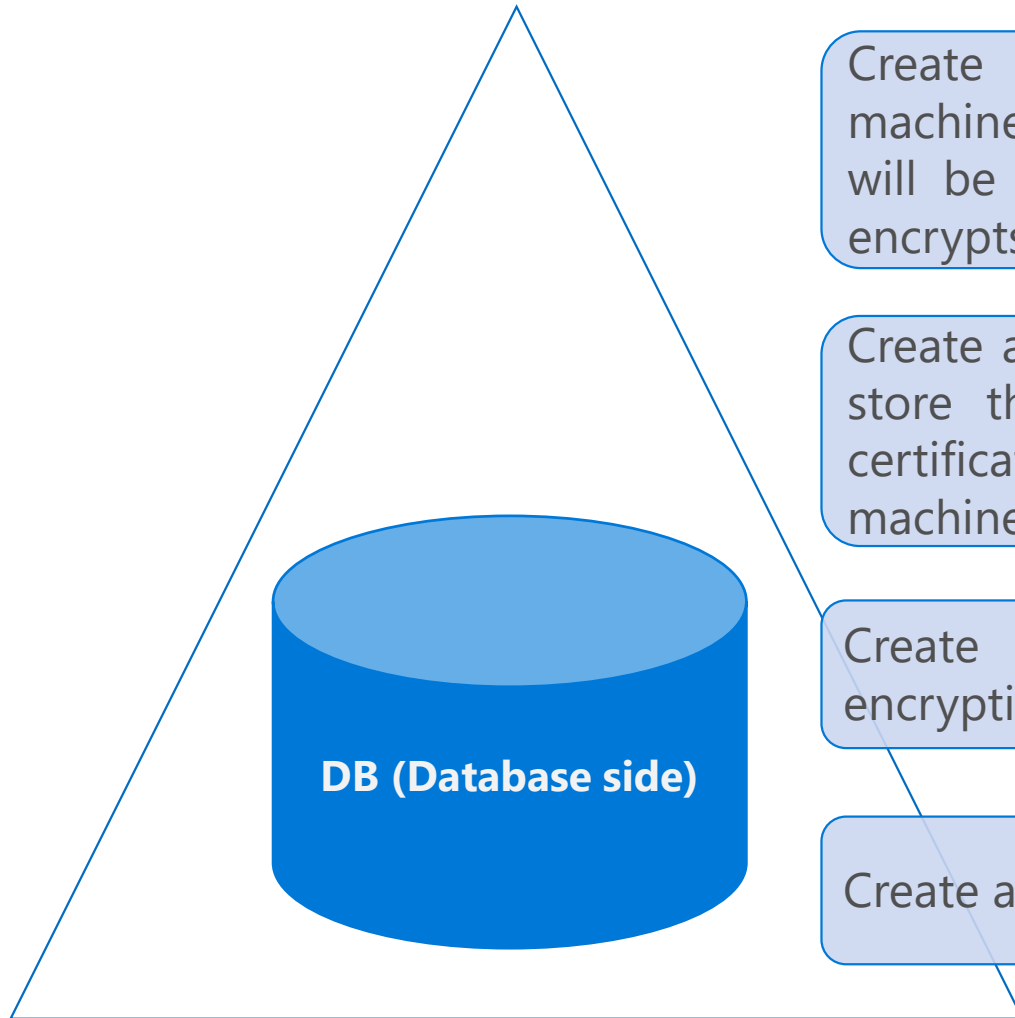
Randomized encryption

- `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 quality**.
- **Comparison** (for example, in WHERE clauses and joins, distinct, group by).

Enabling Always Encrypted on Azure SQL DB



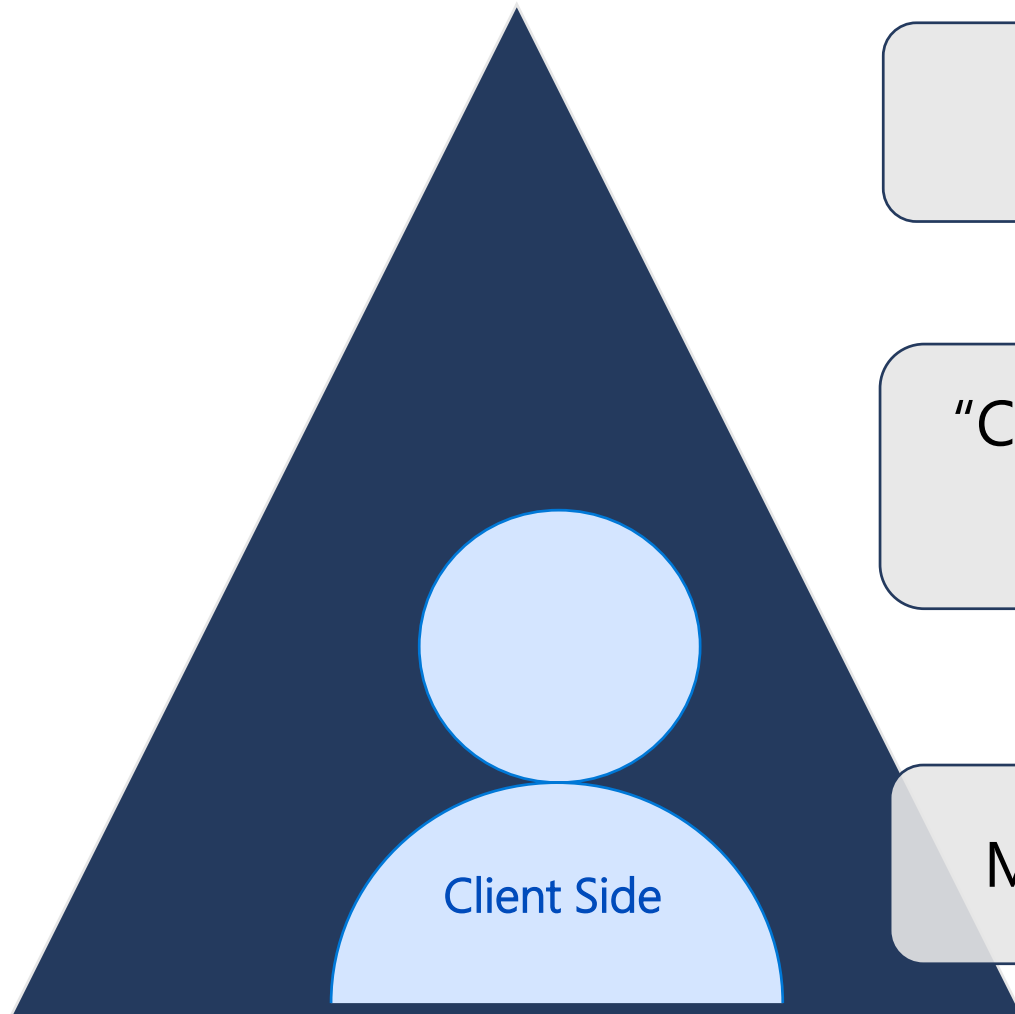
Create a local, self-signed certificate on the development machine, which will act as a Column Master Key (CMK). The CMK will be used to protect Column Encryption Keys (CEK), which encrypts the sensitive data.

Create a CMK store definition object in the database, which will store the information about the location of the CMK. The certificate will never be copied to the database or SQL Server machine.

Create a Column Encryption Key (CEK). You use column encryption keys to encrypt data in database columns.

Create a table with encrypted columns.

Enabling Always Encrypted on Azure SQL DB (contd.)



Same local stored certificate.

"Column Encryption Setting=Enabled;"
in connection string.

Make sure you use the correct driver.

Demonstration

Enable Always Encrypted

- Enable Always Encrypted
- Select data through Application.



Implement Always Encrypted

- **Exercise 1:** Implement Always Encrypted on Azure SQL Database.
- **Exercise 2:** Use the .Net Client App to Test Always Encrypted.



Questions?



Knowledge Check

Can a DBA see Always Encrypted data?

What are the 4 steps that you need to perform to enable Always Encrypted?

Lesson 7: Implement Row Level Security

Objectives

After completing this learning, you will be able to:

- Know how to control access to the data using Row Level Security (RLS).



Row Level Security (RLS)

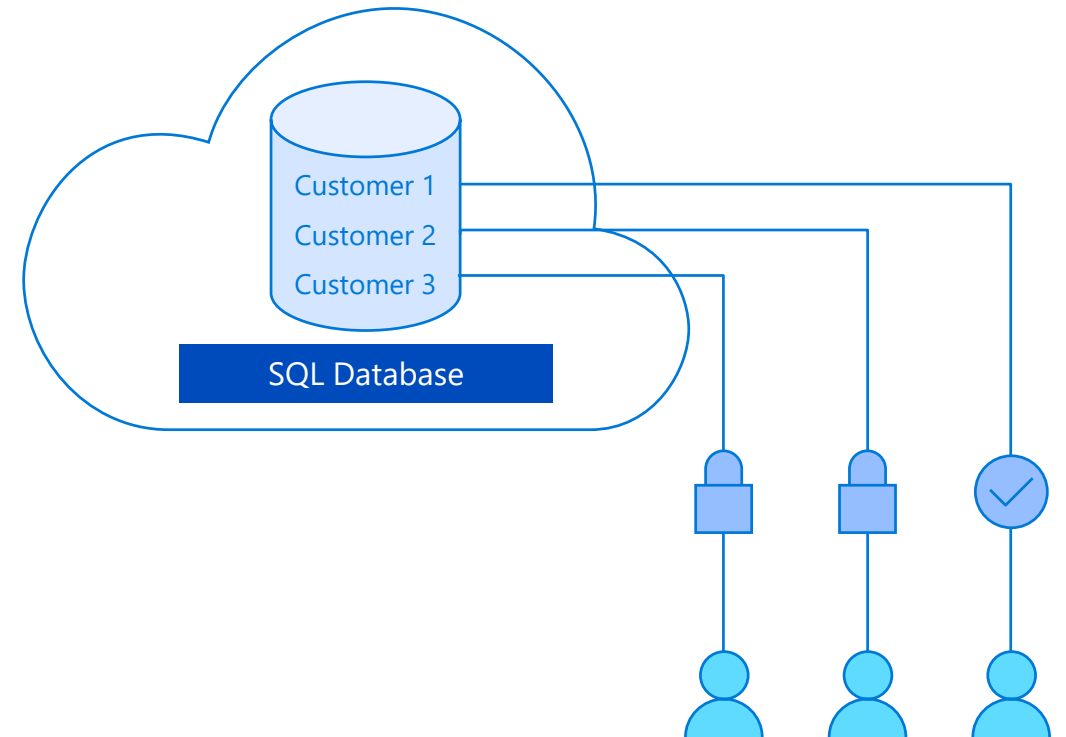
Row-Level Security enables customers to control access to rows in a database table based on the characteristics of the user executing a query.

Understanding RLS Functionality

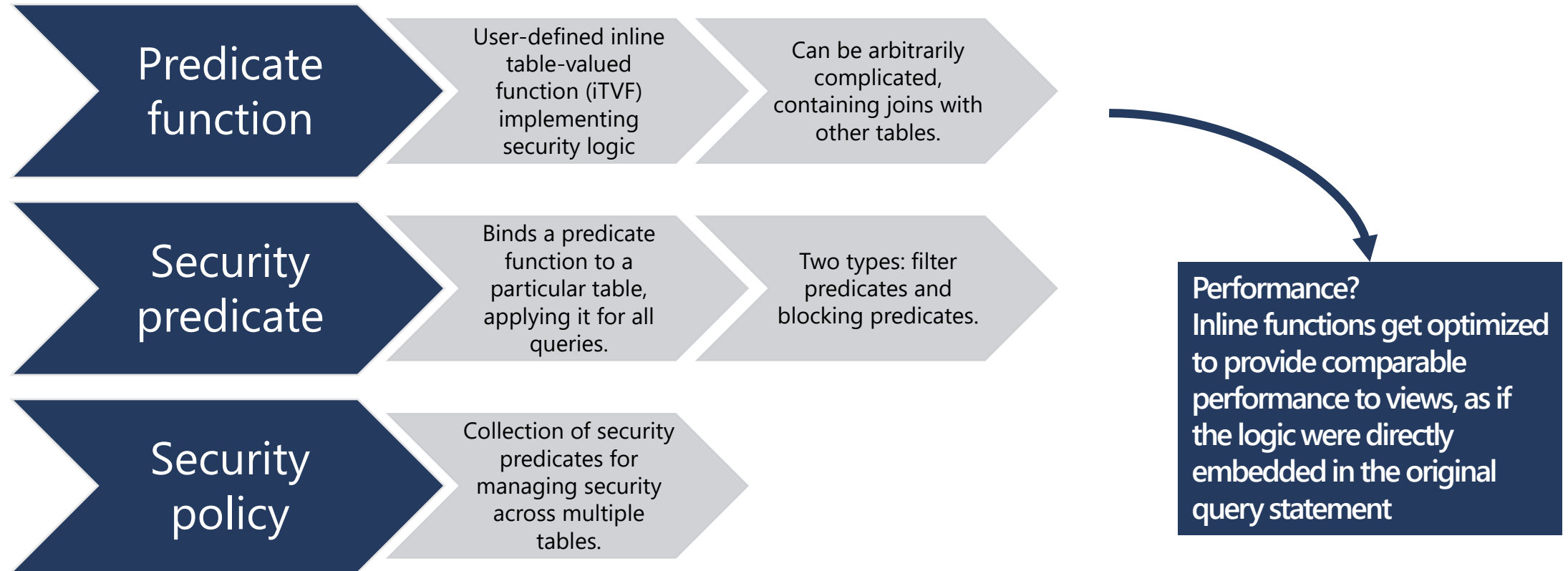
Fine-grained access control over specific rows in a database table.

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

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



RLS Implementation details



```
CREATE SECURITY POLICY mySecurityPolicy
ADD FILTER PREDICATE dbo.fn_securitypredicate(wing, startTime, endTime)
ON dbo.patients
```

Demonstration

Implement RLS using T-SQL Code

- Enable RLS using T-SQL.



Implement Row Level Security

- **Exercise 1:** Implement Row Level Security on Azure SQL Database.



Questions?



Knowledge Check

What is the purpose of the predicate function?

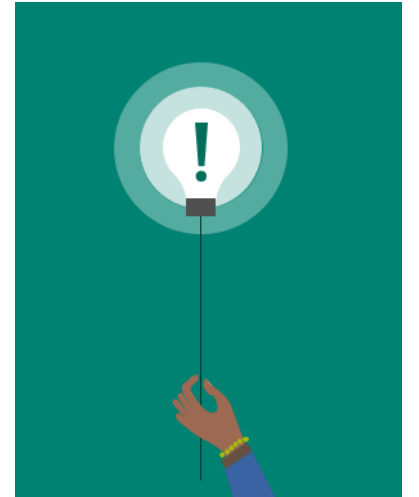
Why do we need row level security (RLS)?

Lesson 8: Implement Dynamic Data Masking

Objectives

After completing this learning, you will be able to:

- Know how to mask the critical data using Dynamic Data Masking.

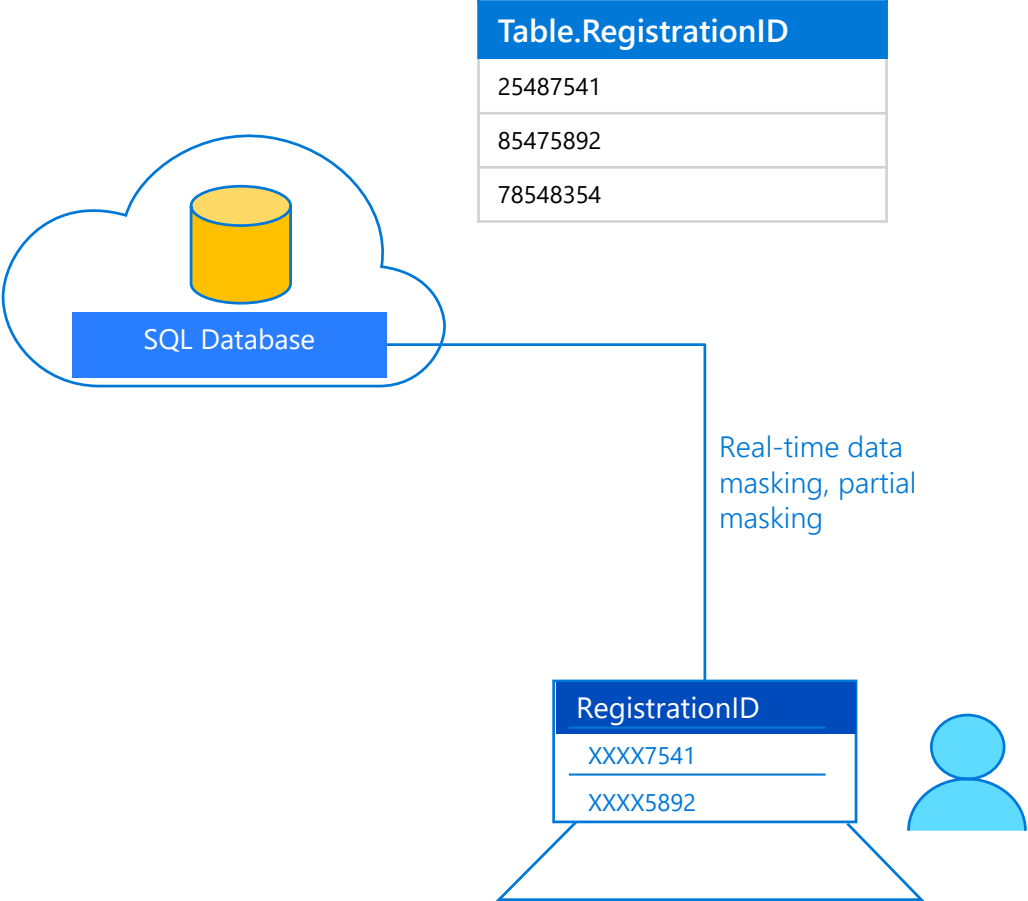


Dynamic Data Masking

Dynamic Data Masking is a policy-based security feature that helps to limit the exposure of data in a database by returning masked data to non-privileged users who run queries over designated database fields.

Understanding Dynamic Data Masking Functionality

- 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.).



Enable Dynamic Data Masking on Azure SQL DB

Security officer defines dynamic data masking policy in T-SQL over sensitive data in the Employee table.

The app user selects from the Employee table.

The dynamic data masking policy obfuscates the sensitive data in the query results.



Business app

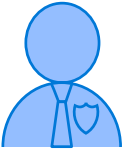


```
ALTER TABLE dbo.Employee
ALTER COLUMN [FirstName] ADD MASKED WITH (FUNCTION = 'partial(1,
"xxx", 2)')

ALTER TABLE dbo.Employee
ALTER COLUMN [EMAIL] ADD MASKED WITH (FUNCTION = 'email()')

ALTER TABLE dbo.Employee
ALTER COLUMN [Salary] ADD MASKED WITH (FUNCTION =
'random(2000,20000)')

GRANT UNMASK to admin1
```



Security officer

```
SELECT EmployeeID,
FirstName, MiddleInitial,
LastName, EMAIL, Salary
FROM [Employee]
```

Other Login

	EmployeeID	FirstName	MiddleInitial	LastName	EMAIL	Salary
1	1	Lydia	L	Kelley	Lydia.Kelley@contoso.com	57353
2	2	Julia	T	James	Julia.James@contoso.com	138286
3	3	Chester	D	Dixon	Chester.Dixon@contoso.com	117503
4	4	Darla	D	Faulkner	Darla.Faulkner@contoso.com	74581
5	5	Danny	S	Velasquez	ngyacn17@contoso.com	94116

Admin Login

	EmployeeID	FirstName	MiddleInitial	LastName	EMAIL	Salary
1	1	Lxxxia	L	Kelley	LXXX@XXXX.com	19530
2	2	Jxxxia	T	James	JXXX@XXXX.com	7376
3	3	Cxxxer	D	Dixon	CXXX@XXXX.com	17612
4	4	Dxxxla	D	Faulkner	DXXX@XXXX.com	7050
5	5	Dxxxny	S	Velasquez	nXXX@XXXX.com	12530

Demonstration

Implement Dynamic Data masking T-SQL Code

- Enable Dynamic Data masking using T-SQL.



Questions?



Knowledge Check

What's the purpose of Dynamic Data Masking?

List two different masking rules in Dynamic Data Masking?

Lesson 9: Implement Auditing for Azure SQL Database

Objectives

After completing this learning, you will be able to:

- Know how you can configure Auditing on Azure SQL Database.



SQL Auditing

SQL Auditing tracks database events and writes them to an audit log in your Azure storage account, Log Analytics workspace or Event Hubs.

Helps you maintain regulatory compliance, understand database activity, and gain insight into discrepancies and anomalies that could indicate business concerns or suspected security violations.

Enables and facilitates adherence to compliance standards, although it doesn't guarantee compliance.

SQL Auditing (continued)

Gain insight into database events and streamline compliance-related tasks.

Configurable to track and log database activity.

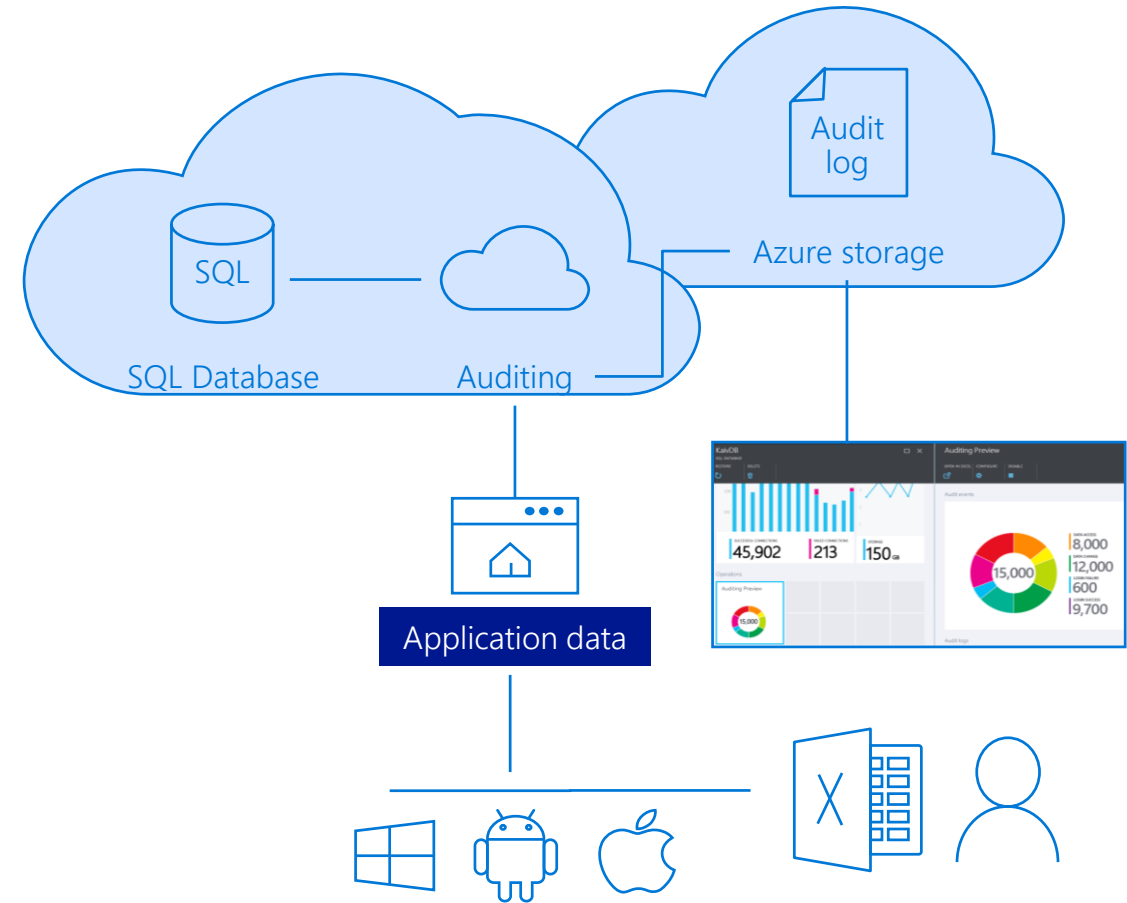
Dashboard views in portal for at-a-glance insights.

Audit logs reside Azure Storage Account, Log Analytics or Event Hub.

Available in Basic, Standard, Premium and Managed Instance.

The default auditing policy includes:

- BATCH_COMPLETED_GROUP
- SUCCESSFUL_DATABASE_AUTHENTICATION_GROUP
- FAILED_DATABASE_AUTHENTICATION_GROUP



Analyze audit logs and reports

Azure Monitor logs

- Azure portal

Event Hub

- Avro Tools or similar tools

Azure storage account

- Azure Storage Explorer
- Azure portal
- Power BI
- SQL Server Management Studio (SSMS)
- PowerShell

Demonstration

Implement Auditing for Azure SQL Database

- Enable auditing for Azure SQL Database using Azure portal.



Questions?



Knowledge Check

Which 3 action groups are configured by default when you enable auditing?

Where are the auditing records stored?

Which tools can you use to analyze the Audit Logs?

Lesson 10: Data Discovery and Classification

Objectives

After completing this learning, you will be able to:

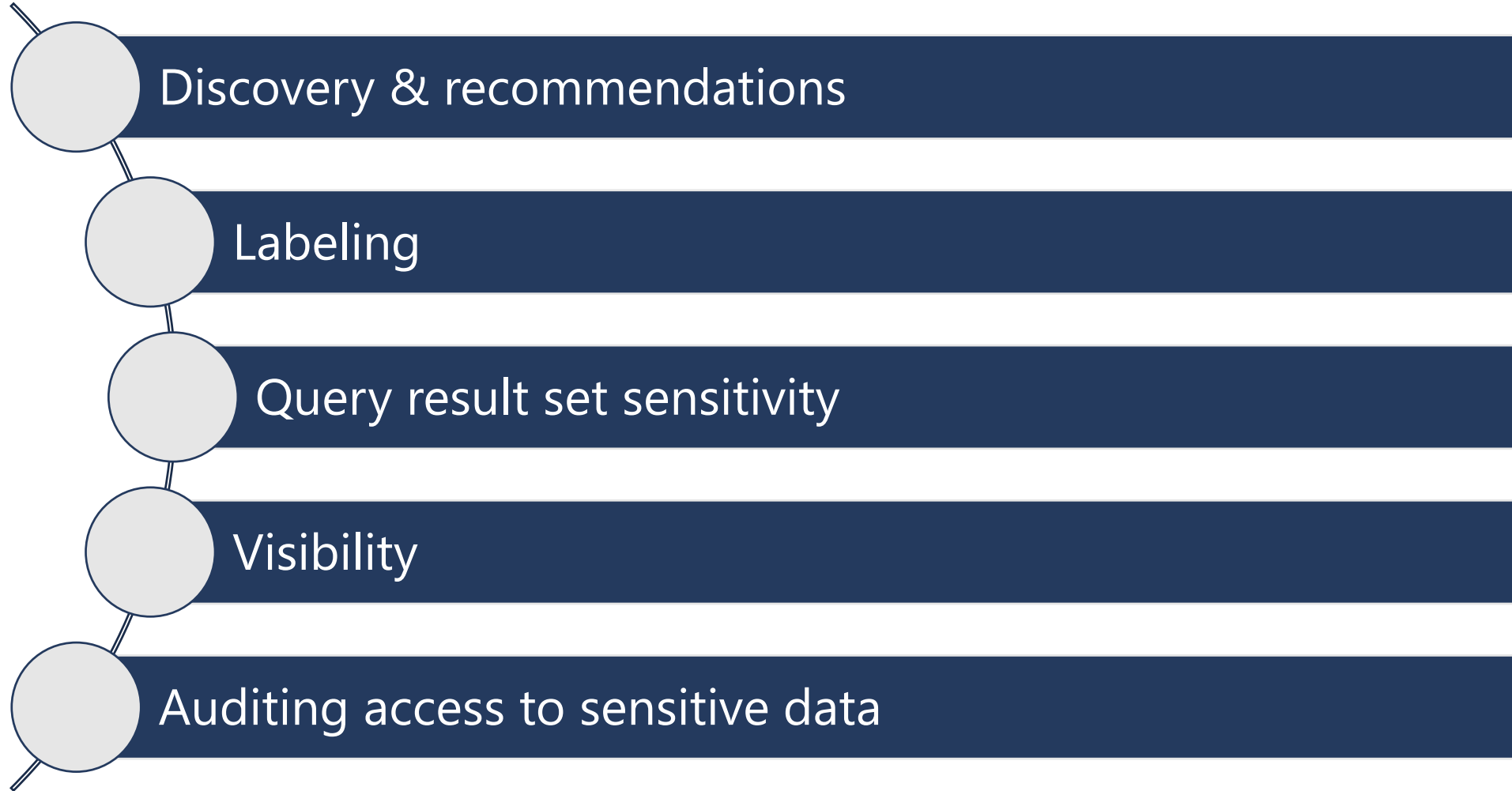
- Know how to discover, classify, label & protect the sensitive data in your databases



Data Discovery and Classification

Data discovery & classification provides advanced capabilities built into Azure SQL Database for **discovering, classifying, labeling & protecting** the sensitive data in your databases.

Data Discovery and Classification (continued)



Demonstration

Data Discovery and Classification

- Classify your SQL Database.



Lesson 11: Microsoft Defender for SQL

Objectives

After completing this learning, you will be able to:

- Know how to proactively identify security threats like SQL Injection or anomalous SQL login by enabling threat detection
- Know how to discover, track, and help you remediate potential database vulnerabilities



Microsoft Defender for SQL

Formerly known as Advanced Data Security (ADS),

Microsoft Defender for SQL provides a set of advanced SQL security capabilities, including:

- [Advanced Threat Protection](#) detects anomalous activities indicating unusual and potentially harmful attempts to access or exploit your database. It continuously monitors your database for suspicious activities, and it provides immediate security alerts on potential vulnerabilities, Azure SQL injection attacks, and anomalous database access patterns. Advanced Threat Protection alerts provide details of the suspicious activity and recommend action on how to investigate and mitigate the threat.
- [Vulnerability Assessment](#) is an easy-to-configure service that can discover, track, and help you remediate potential database vulnerabilities. It provides visibility into your security state, and it includes actionable steps to resolve security issues and enhance your database fortifications.

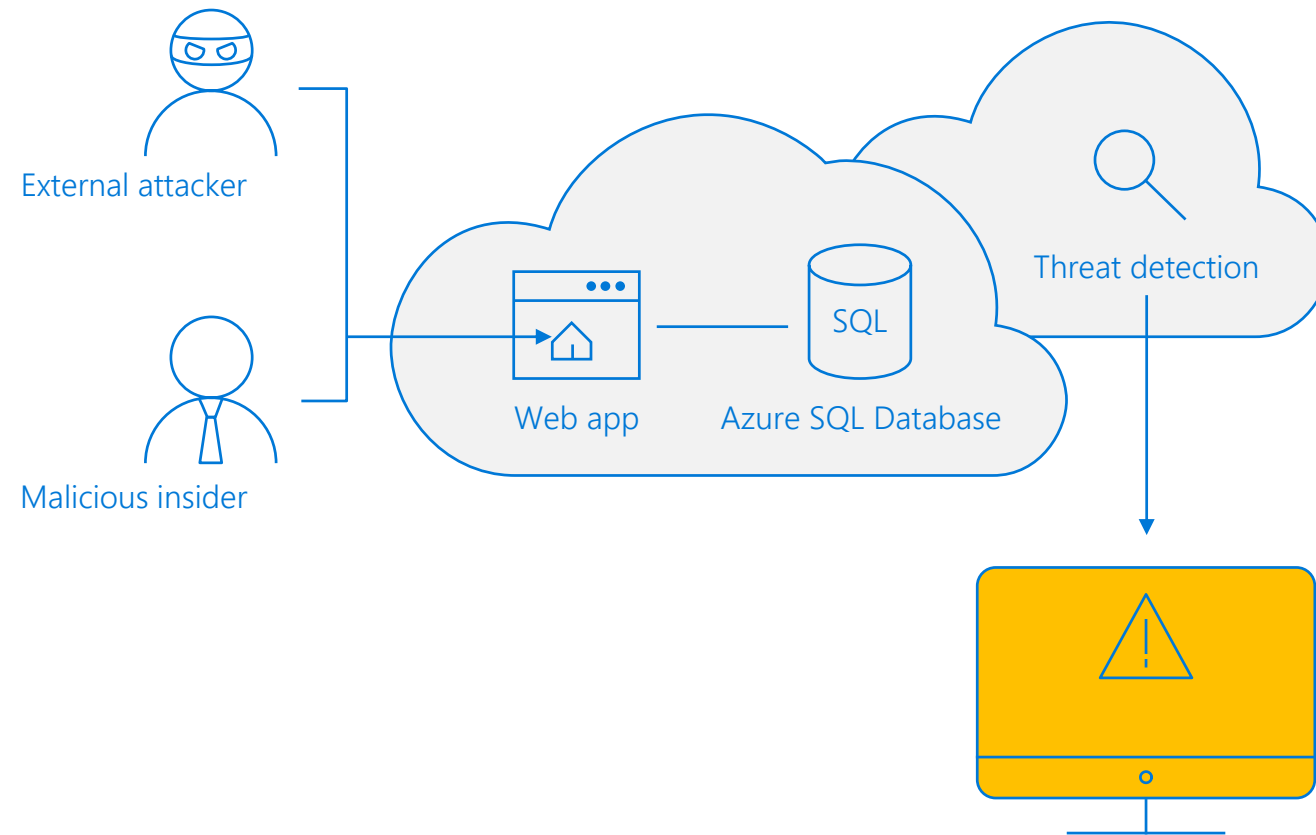
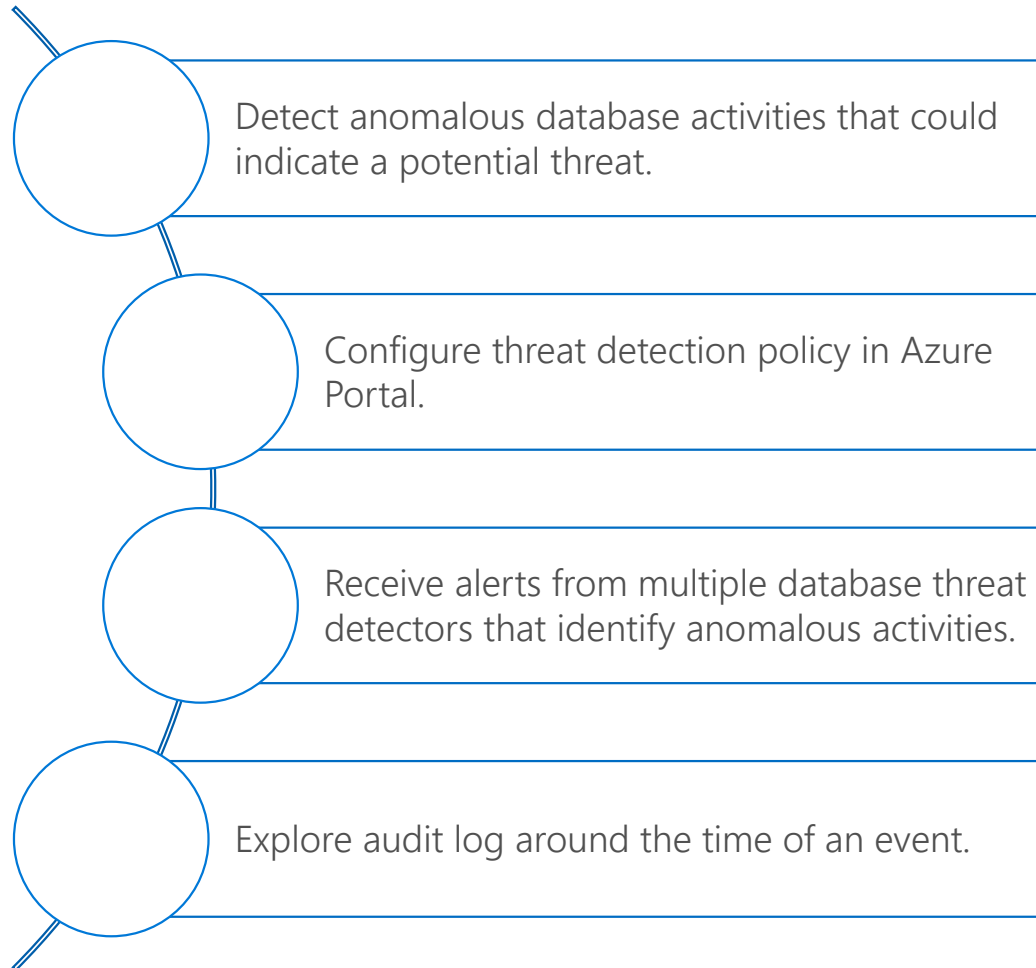
Advanced Threat Detection

Advanced Threat Protection for single and pooled databases detects anomalous activities indicating unusual and potentially harmful attempts to access or exploit databases.

Advanced Threat Protection can identify:

- Potential SQL injection, Access from unusual location or data center.
- Access from unfamiliar principal or potentially harmful application.
- Brute force SQL credentials.

Advanced Threat Detection (continued)



Enable Microsoft Defender for SQL

Security

Auditing

Ledger

Data Discovery & Classification

Dynamic Data Masking

Microsoft Defender for Cloud

Transparent data encryption

Recommendations

0

Security alerts

0

Findings

--

Microsoft Defender for SQL: **Disabled**

Microsoft Defender for SQL

[Azure Defender for SQL](#) helps you strengthen your security posture, identify and manage security vulnerabilities and protect against threats on your SQL servers.

You are invited to a 30-day trial, free of charge. After the trial ends, you will be charged \$15/Server/Month

Enable Microsoft Defender for SQL

Configure Microsoft Defender for SQL for Email Alerts

Recommendations

Security alerts

Findings

Microsoft Defender for SQL: Enabled at the subscription-level (Configure)

0

0

--

Recommendations

Defender for Cloud continuously monitors the configuration of your SQL Servers to identify potential security vulnerabilities and recommends actions to mitigate them.

✓

✓

✓

No recommendations to display

There are no security recommendations for this resource

View all recommendations in Defender for Cloud

Security incidents and alerts

Defender for Cloud uses advanced analytics and global threat intelligence to alert you to malicious activity. Alerts displayed below are from the past 21 days.

Check for alerts on this resource in Microsoft Defender for Cloud >

Server settings

jdsqldemo

Save Discard Feedback

MICROSOFT DEFENDER FOR SQL

ON OFF

Microsoft Defender for SQL costs 15 USD/server/month. It includes Vulnerability Assessment and Advanced Threat Protection. We invite you to a trial period for the first 30 days, without charge.

VULNERABILITY ASSESSMENT SETTINGS

Subscription

PFE Subscription

Select Subscription

Storage account

sqlvambkz6jggwgzv2

Select Storage account

Periodic recurring scans

ON OFF

Scans will be triggered automatically once a week. In most cases, it will be on the day Vulnerability Assessment has been enabled and saved. A scan result summary will be sent to the email addresses you provide.

Send scan reports to

SQLDBA@adventureworks.com

Also send email notification to admins and subscription owners

ADVANCED THREAT PROTECTION SETTINGS

Advanced Threat Protection for SQL alerts emails are sent by Defender for Cloud.


Add your contact details to the subscription's email settings in Defender for Cloud.

Enable Auditing for better threats investigation experience

Review Microsoft Defender Email Alerts

Microsoft

Azure SQL database

 Potential exploitation of application code vulnerability to SQL Injection was detected on database samplecrmwedemo. This may indicate a SQL Injection attack on database 'samplecrmwedemo'.

[View recent SQL alerts](#)

Activity details

Severity: High

Subscription ID: DS-THREATDETECTION_DEMO_TOMERR_R&D_60843

Subscription Name: DS-THREATDETECTION_DEMO_TOMERR_R&D_60843

Server: samplecrmwedemo

Database: samplecrmwedemo

IP address: 10.10.10.10

Principal Name: de*****

Application: .Net SqlClient Data Provider

Date: May 13, 2018 12:09:12 UTC

Threat ID: 1

Potential causes: Defect in application code constructing SQL statements; application code doesn't sanitize user input and was exploited to inject malicious SQL statements.

Investigation steps: [View the vulnerable SQL statement](#)

Remediation steps: [Read more about SQL Injection threat and how to fix the vulnerable application code.](#)

Security alerts

samplecrmwedemo

Filter Security Center

1 Thu

Severity	Description	Count	Detected by
High	Potential SQL Injection	3	Microsoft
Medium	Potential SQL Brute Force attempt	1	Microsoft
Medium	Attempted logon by a potentially harmful application	1	Microsoft
Medium	A possible vulnerability to SQL Injection	1	Microsoft
Medium	Logon from an unusual location	1	Microsoft
Medium	Logon by an unfamiliar principal	1	Microsoft

Potential SQL Injection

samplecrmwedemo

[Learn more](#)

General information

DESCRIPTION: Potential SQL Injection was detected on your database samplecrmwedemo on server ronmatwedemo

DETECTION TIME: Sunday, 13 May 2018, 3:09:12 pm

SEVERITY: High

STATE: Active

ATTACKED RESOURCE: samplecrmwedemo

SUBSCRIPTION: Microsoft

DETECTED BY: Microsoft

ACTION TAKEN: Detected

ENVIRONMENT: Azure

RESOURCE TYPE: SQL Server

SERVER: samplecrmwedemo

DATABASE: samplecrmwedemo

IP ADDRESS: 10.10.10.10

PRINCIPAL NAME: dev1

APPLICATION: .Net SqlClient Data Provider

VULNERABLE STATEMENT: SELECT * FROM sql_users WHERE username = ''OR 1 = 1 --' AND password = 'dfafdfafdf'

THREAT ID: 1

Remediation steps

INVESTIGATION STEPS: [View the vulnerable SQL statement](#)

REMEDIATION STEPS: [Read more about SQL Injection threat and how to fix the vulnerable application code.](#)

Review Recommendations and Alerts

Recommendations

Defender for Cloud continuously monitors the configuration of your SQL Servers to identify potential security vulnerabilities and recommends actions to mitigate them.



No recommendations to display

There are no security recommendations for this resource

[View all recommendations in Defender for Cloud](#)

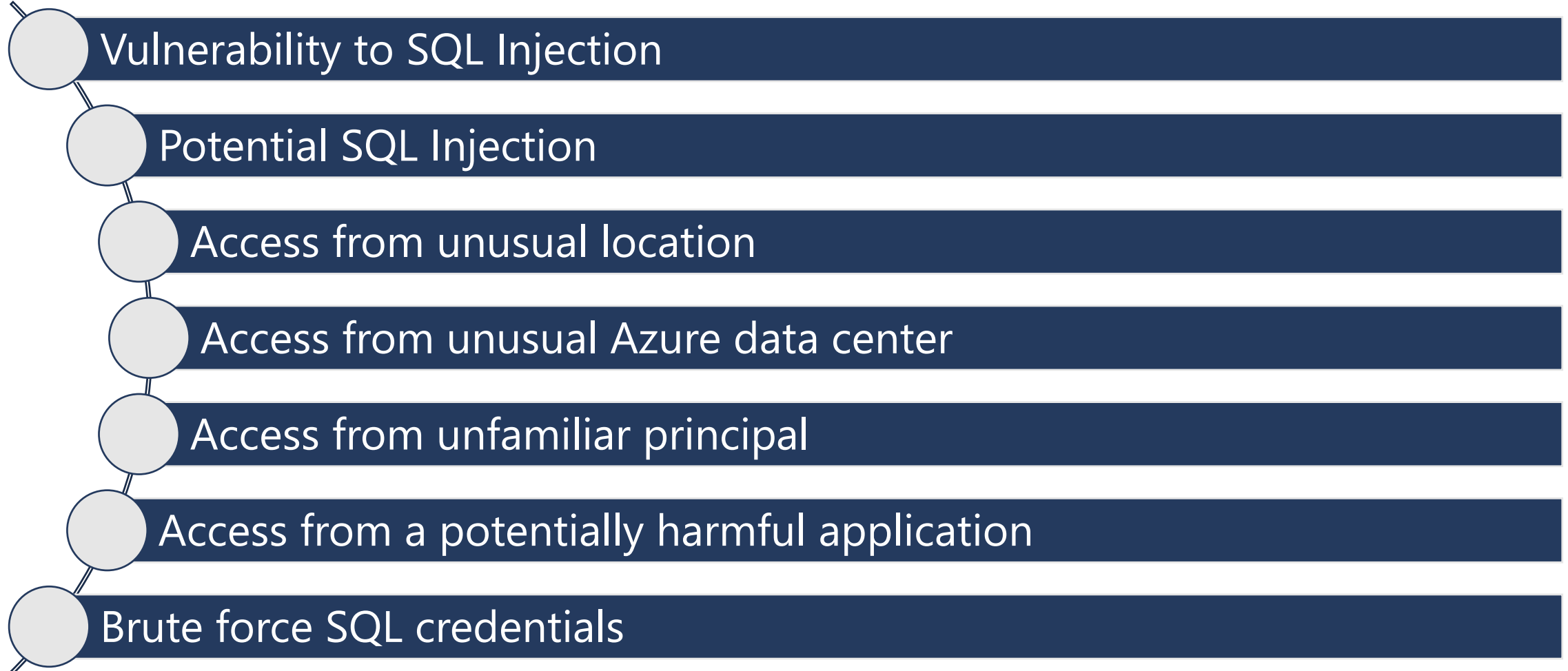
Security incidents and alerts

Defender for Cloud uses advanced analytics and global threat intelligence to alert you to malicious activity. Alerts displayed below are from the past 21 days.



[Check for alerts on this resource in Microsoft Defender for Cloud >](#)

Azure SQL Database Threat Detection Alerts



Demonstration

Microsoft Defender for for Azure SQL Database

- Enable Threat Detection for Azure SQL Database.



SQL Vulnerability Assessment

SQL Vulnerability Assessment is an easy to configure service that can **discover, track, and help you remediate potential database vulnerabilities**. Use it to **proactively** improve your database security.

SQL Vulnerability Assessment (continued)

Get visibility

Discover sensitive data and potential security holes.

Remediate

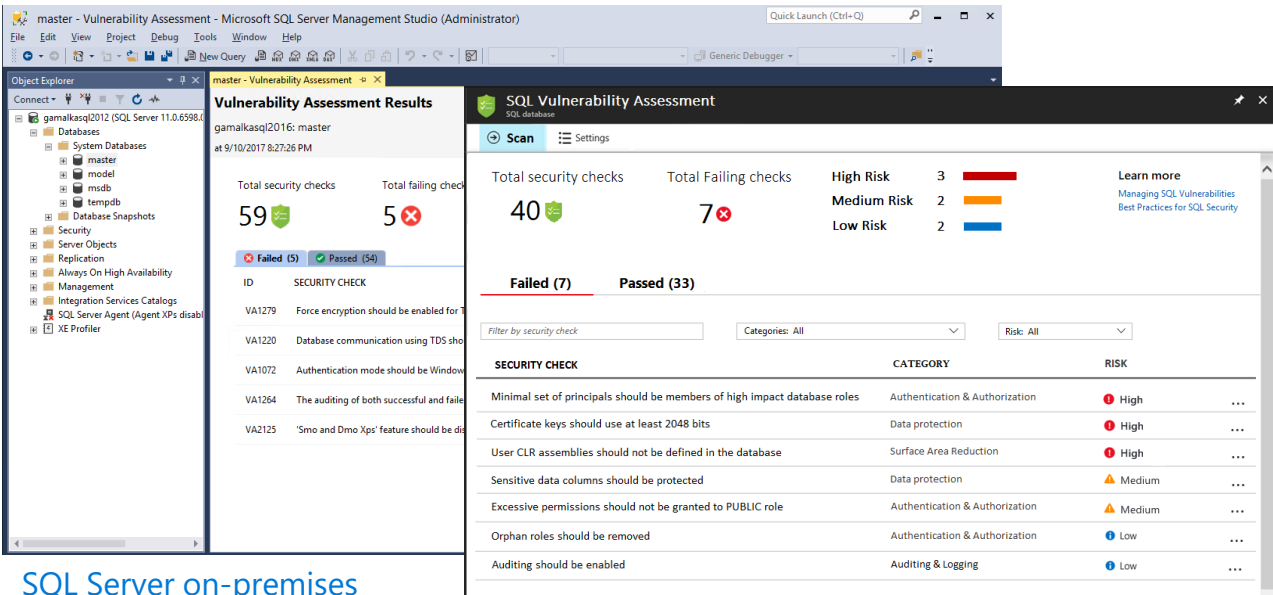
Actionable remediation and security hardening steps.

Customize

Baseline policy tuned to your environment, allowing you to focus on deviations.

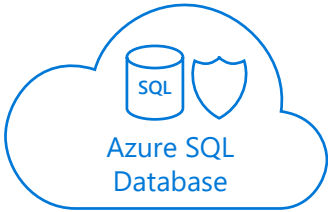
Report

Pass internal or external audits to facilitate compliance.



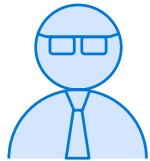
SQL Server on-premises

Azure SQL Database



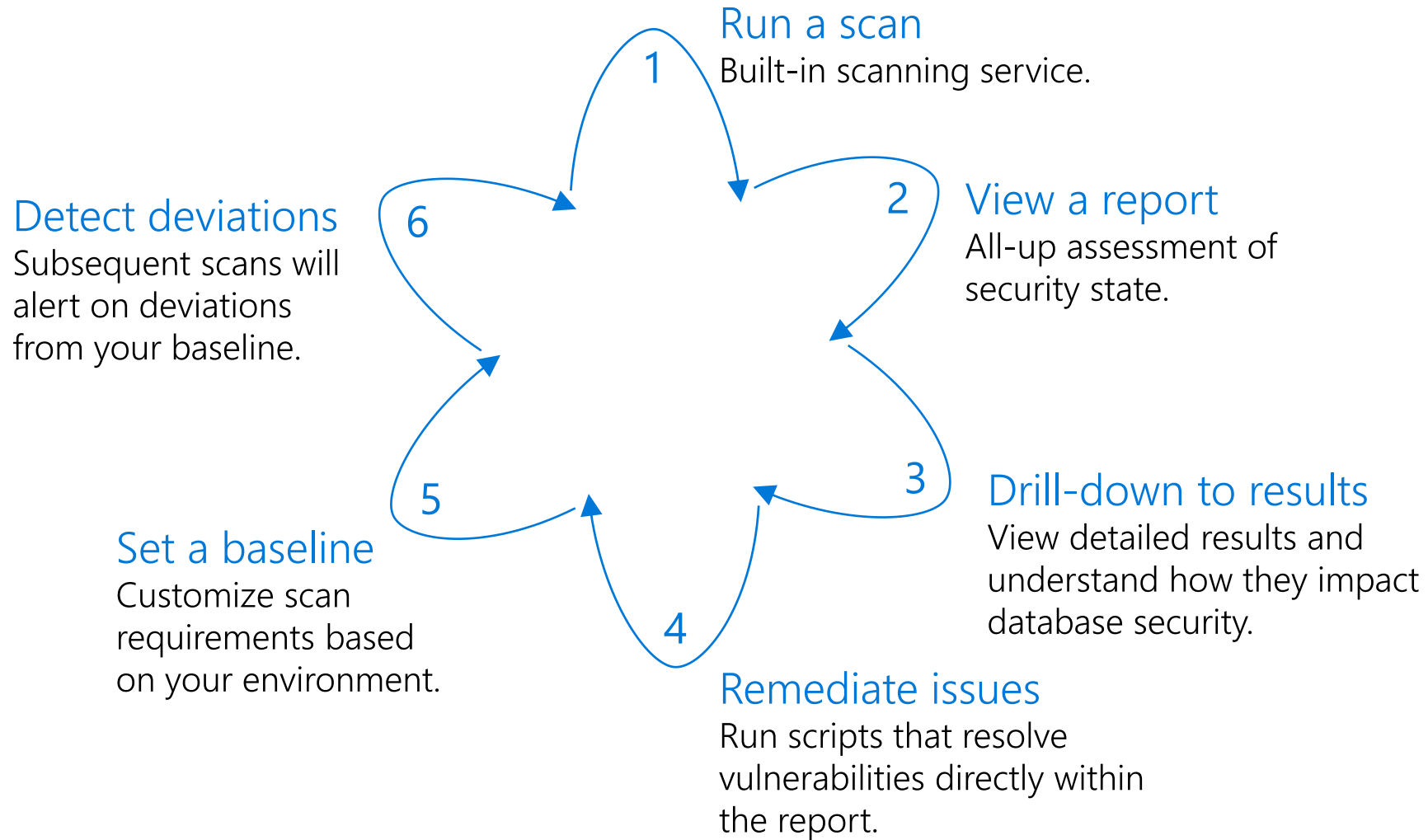
Vulnerability Assessment

Identifies, tracks, and resolves SQL security vulnerabilities



Developer/DBA

Using Vulnerability Assessment



Demonstration

Vulnerability Assessment

- Run a scan, review the report and set a baseline.



Vulnerability Assessment

- **Exercise 1:** Run a scan, review the report and set a baseline.



Questions?



Knowledge Check

List one important event type captured in threat detection.

Where are the threat detection records stored?

What are the steps to implement a Vulnerability Assessment?

Module Summary

Introduction to Azure SQL Database Security

Implement Azure Active Directory Security

Manage Logins in Azure SQL Database

Implement Firewall Rules and Virtual Networks

Implement Transparent Data Encryption

Implement Always Encrypted

Implement Row Level Security

Implement Dynamic Data Masking

Implement Auditing for Azure SQL Database

Implement Microsoft Defender for SQL

