Microsoft

Manage Security for Azure SQL Database

Module 4

Learning Units covered in this Module

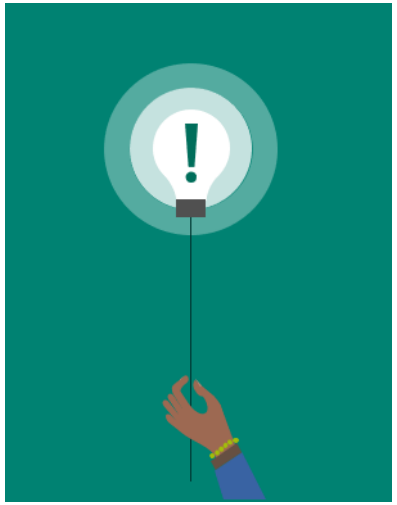
- Lesson 1: Introduction to Azure SQL Database Security
- Lesson 2: Implement Entra ID Security
- Lesson 3: Manage Logins in Azure SQL Database
- Lesson 4: Implement Firewall Rules and Virtual Networks
- Lesson 5: Implement Auditing for Azure SQL Database
- Lesson 6: Implement Ledger for Azure SQL Database
- Lesson 7: Data Discovery and Classification
- Lesson 8: Implement Microsoft Defender for SQL

Lesson 1: Introduction to Azure SQL Database Security

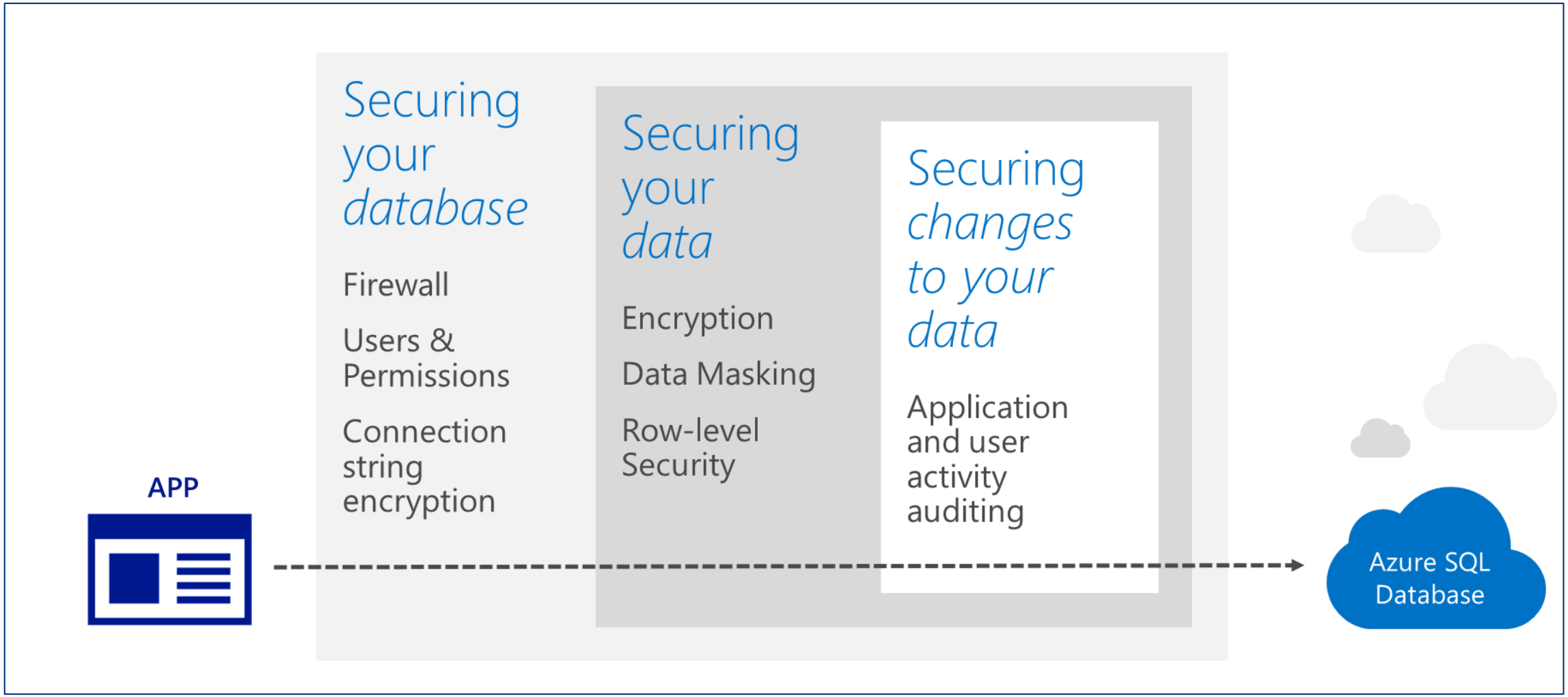
Objectives

After completing this learning, you will be able to:

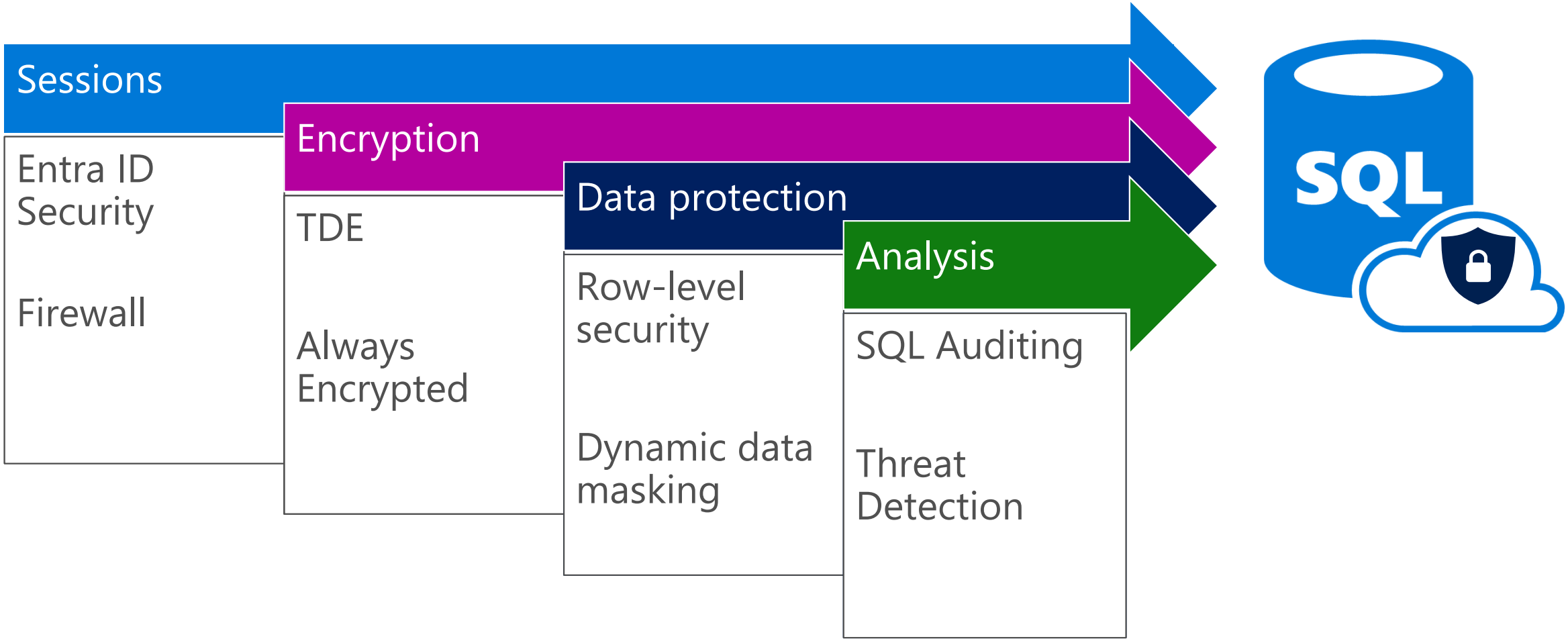
- Know the various options to manage security for an Azure SQL Database.



Azure SQL Database Security Layers



Security Features for Azure SQL DB

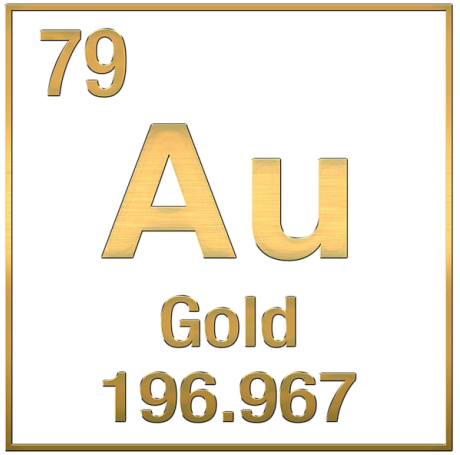


The Security Gold Standard

AUTHENTICATION – Verifies who you are

AUTHORIZATION – Assigns what you can do

AUDITING – Monitors what you did



Questions?

Knowledge Check

List the security features available for Azure SQL Database.

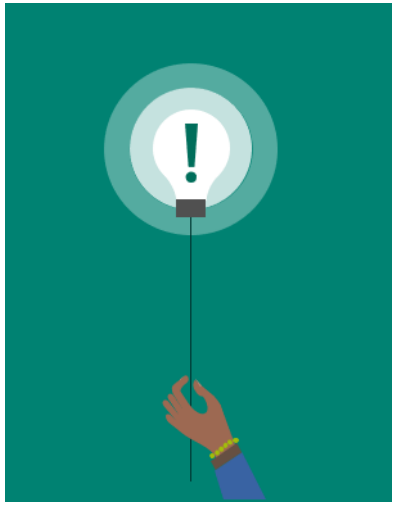
Name the feature to encrypt the data both at rest and motion.

Lesson 2: Implement Entra ID Security

Objectives

After completing this learning, you will be able to:

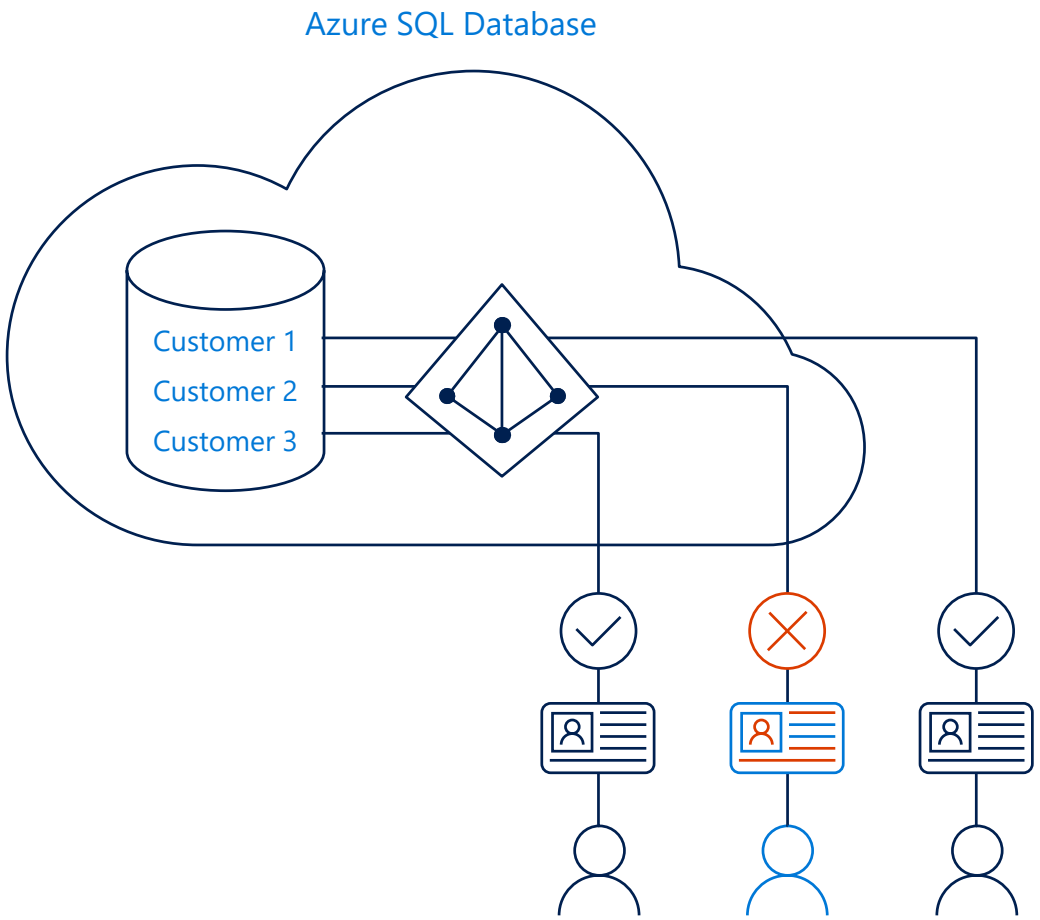
- Know how to leverage Entra ID security for authenticating connections to an Azure SQL Database.



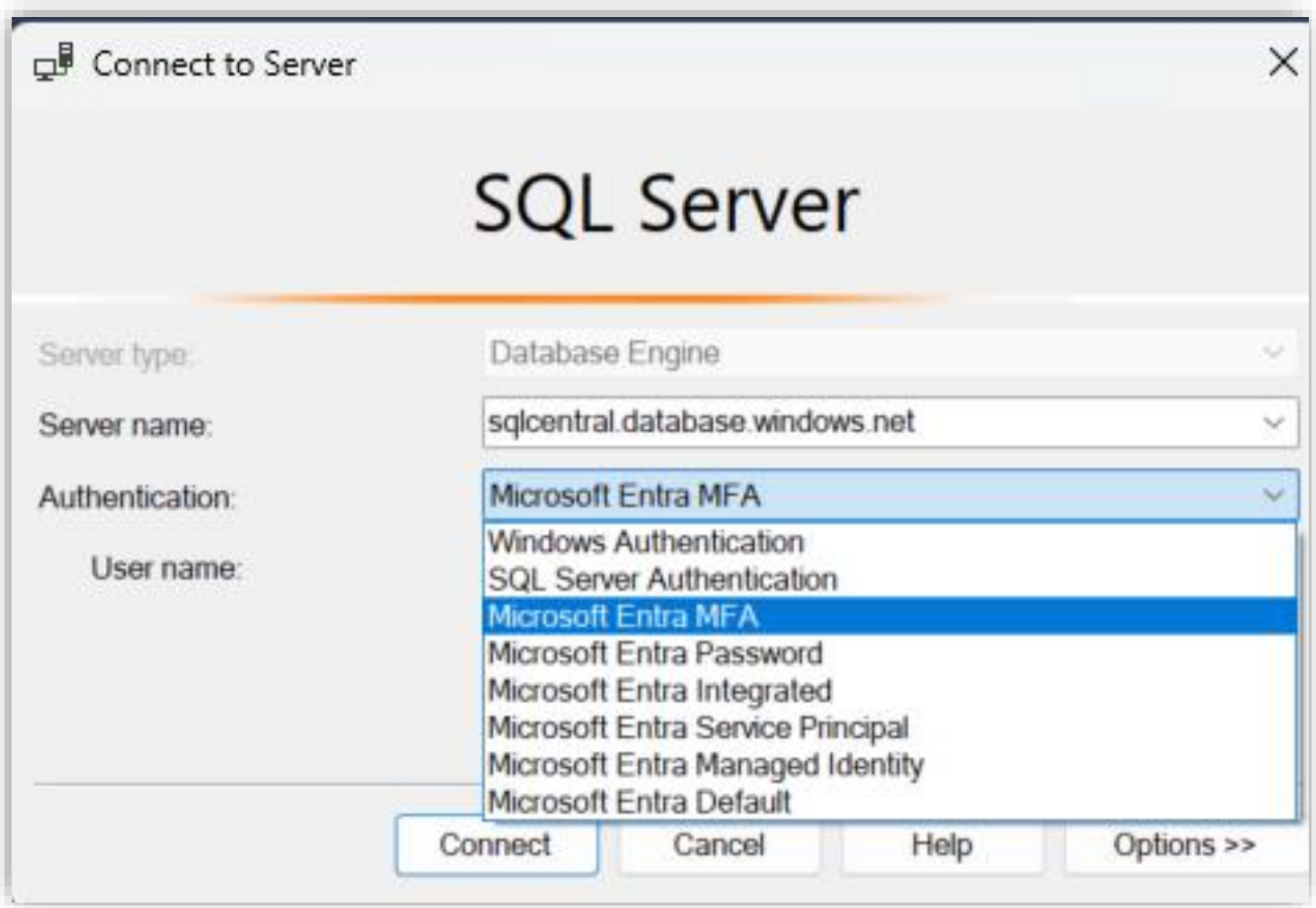
Entra ID Security

Entra ID is now called Entra ID authentication.

Authentication is a mechanism of connecting to Microsoft Azure SQL Database by using identities in Entra ID



Types of Entra ID Authentication



Benefits of Entra Authentication

Centrally manage user permissions.

Alternative to SQL Server authentication.

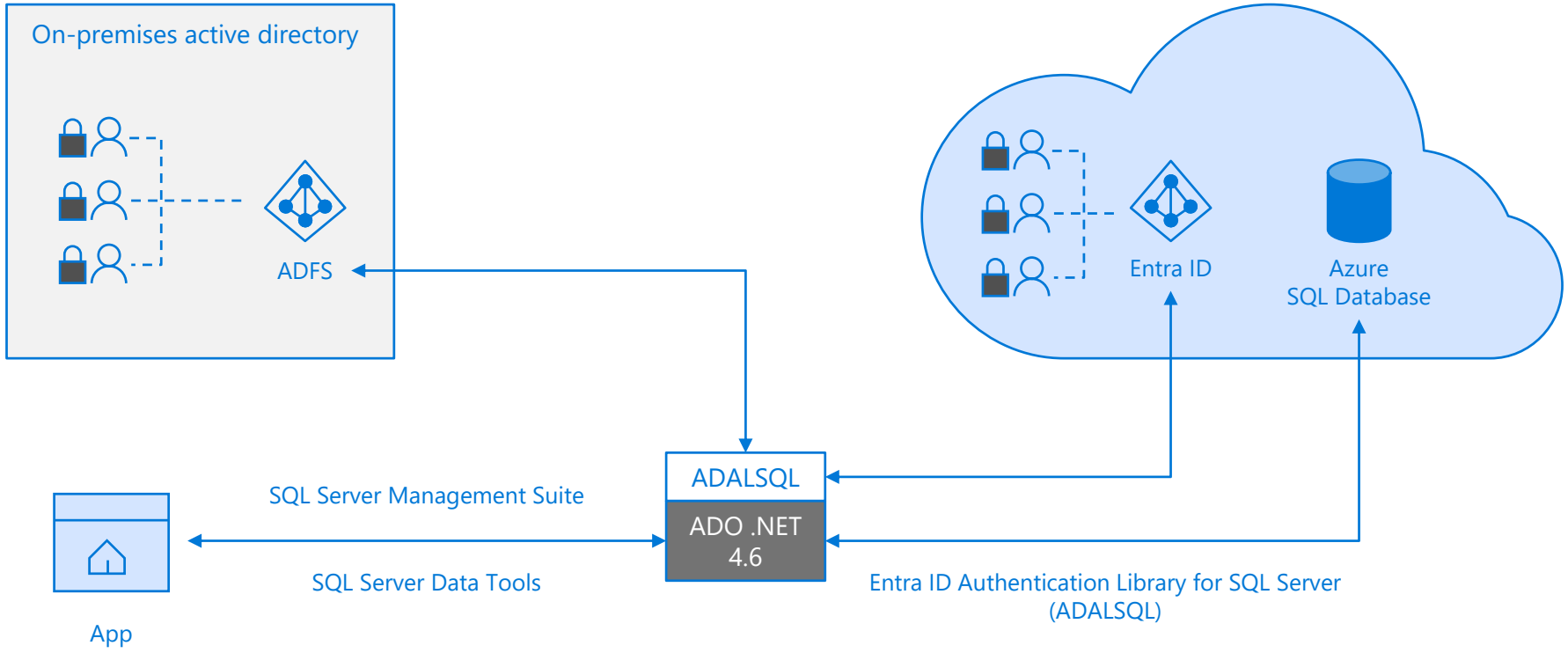
Allows password rotation in a single place.

Enables management of database permissions using external Entra ID groups.

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

Trust architecture

Entra ID and Azure SQL Database



Demonstration

Implement Entra ID Authentication

- Connect to Entra ID.
- Connect to Azure SQL DB using SSMS through Entra ID authentication.

Questions?

Knowledge Check

List three benefits of Azure Activity Directory Authentication.

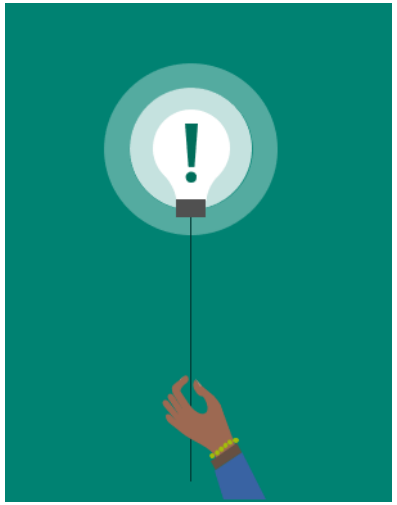
Can we use Windows authentication for Azure SQL Database?

Lesson 3: Manage Logins in Azure SQL Database

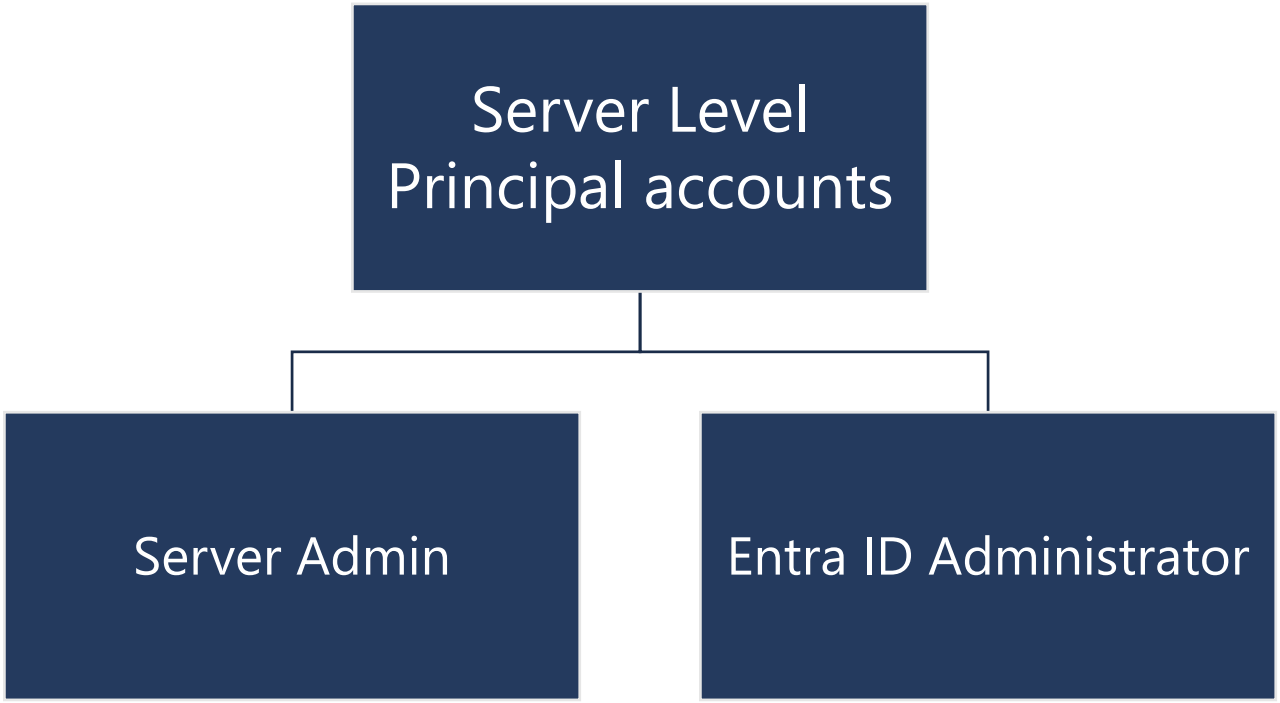
Objectives

After completing this learning, you will be able to:

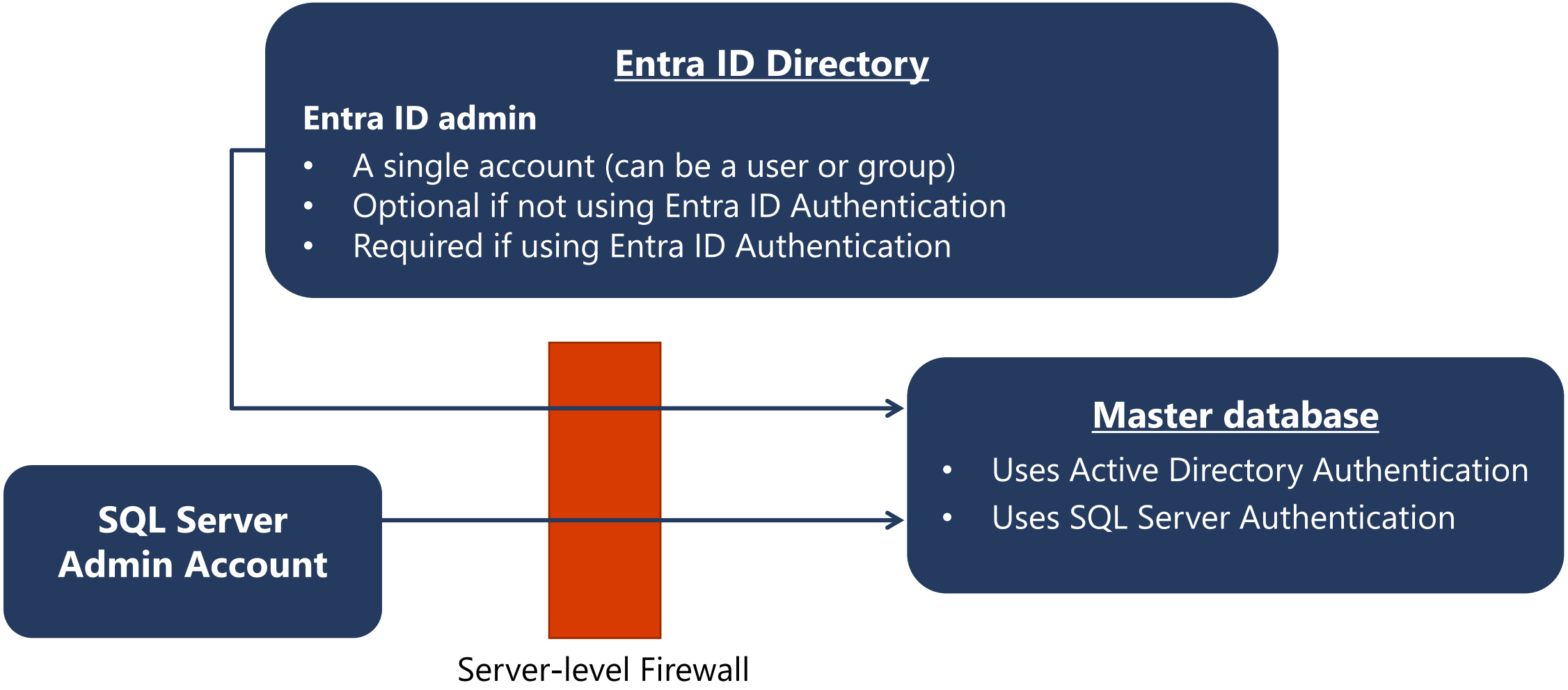
- Manage logins within Azure SQL Database.



Unrestricted Administrative Accounts



Administrator Access Path



Additional Special Roles

Database Creators

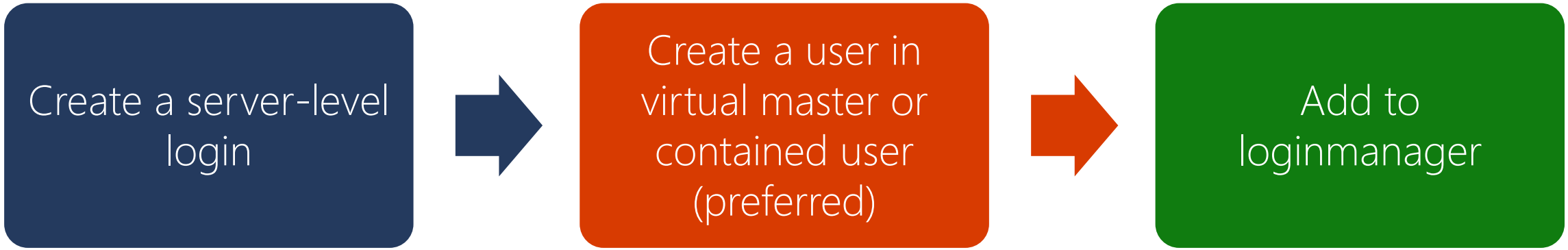
- ALTER ROLE dbmanager* ADD MEMBER Mary;
- ALTER ROLE dbmanager* ADD MEMBER [mike@contoso.com];

*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:

Entra ID authentication contained database user.	SQL Server authentication contained database user.	SQL Server authentication user based on a SQL Server authentication login.

Groups and Roles

Entra ID authentication	SQL Server authentication
• Put Entra ID users into an Entra ID 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.	• 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?

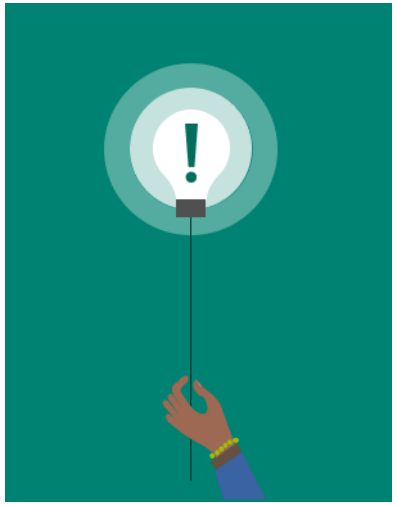
Name the Additional server-level administrative roles for Azure SQL Database?

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.

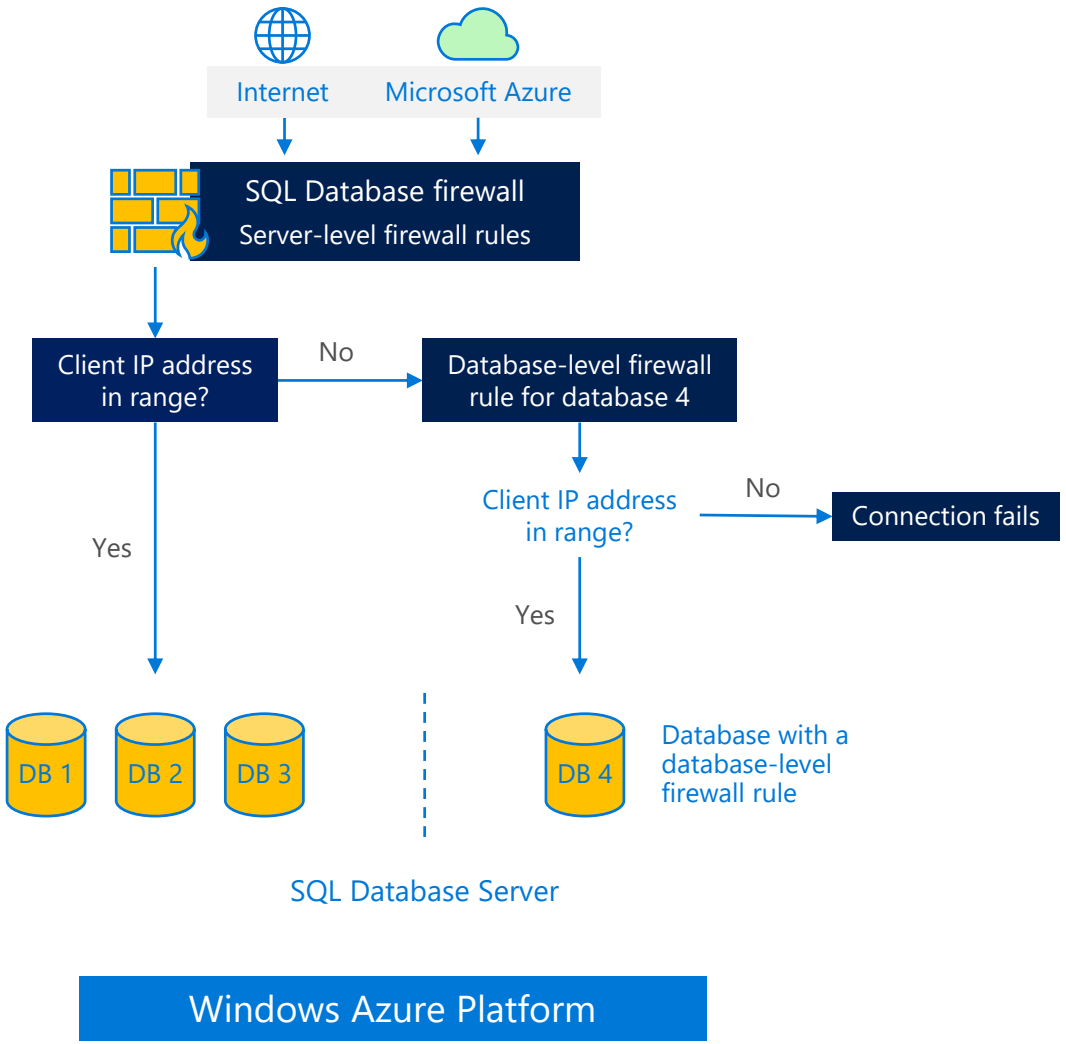
In order to begin using your Azure SQL Database server, you must go to the Management Portal.

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

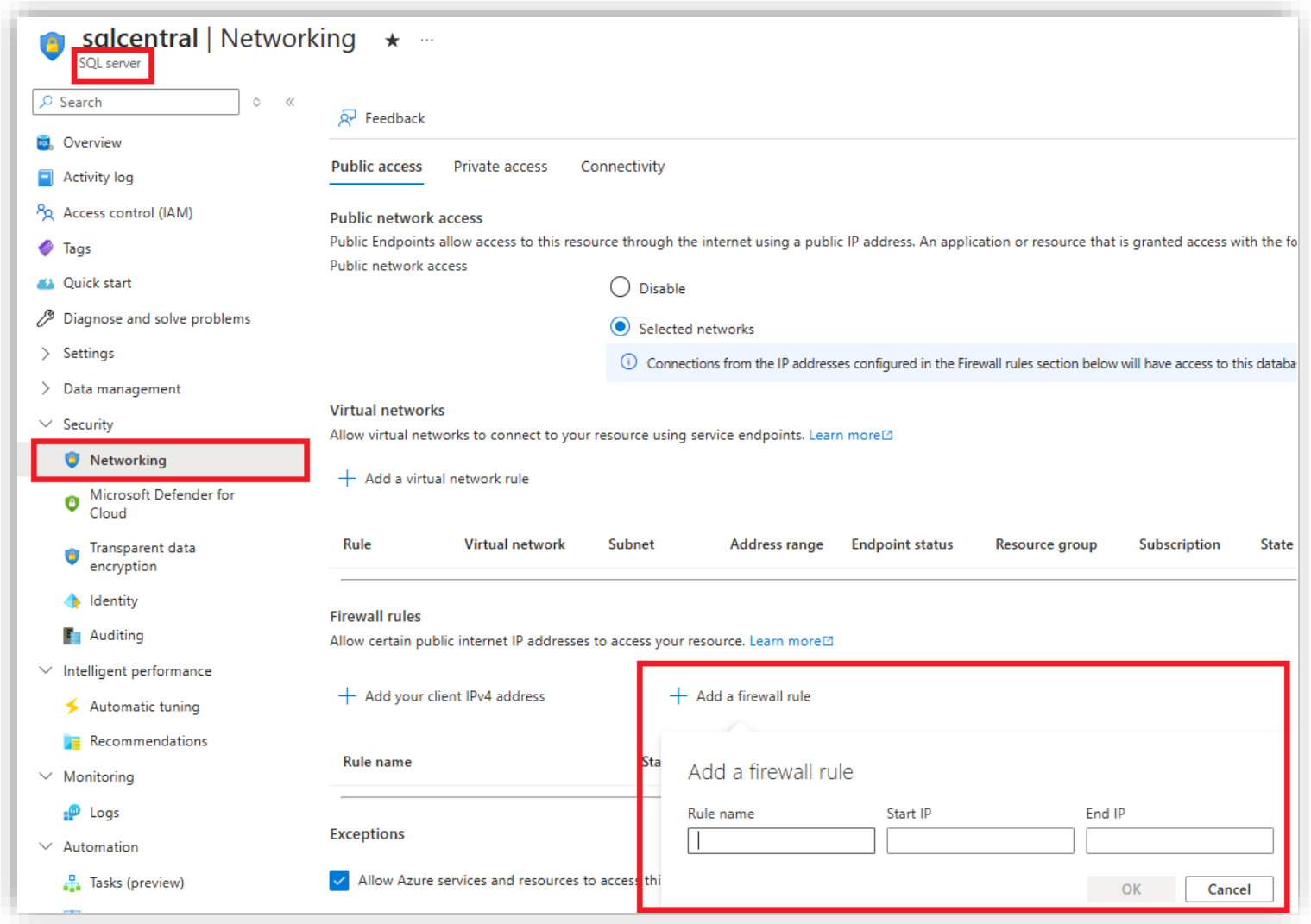
Database-level firewall rules enable clients to access certain databases within the same logical server.

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'
```

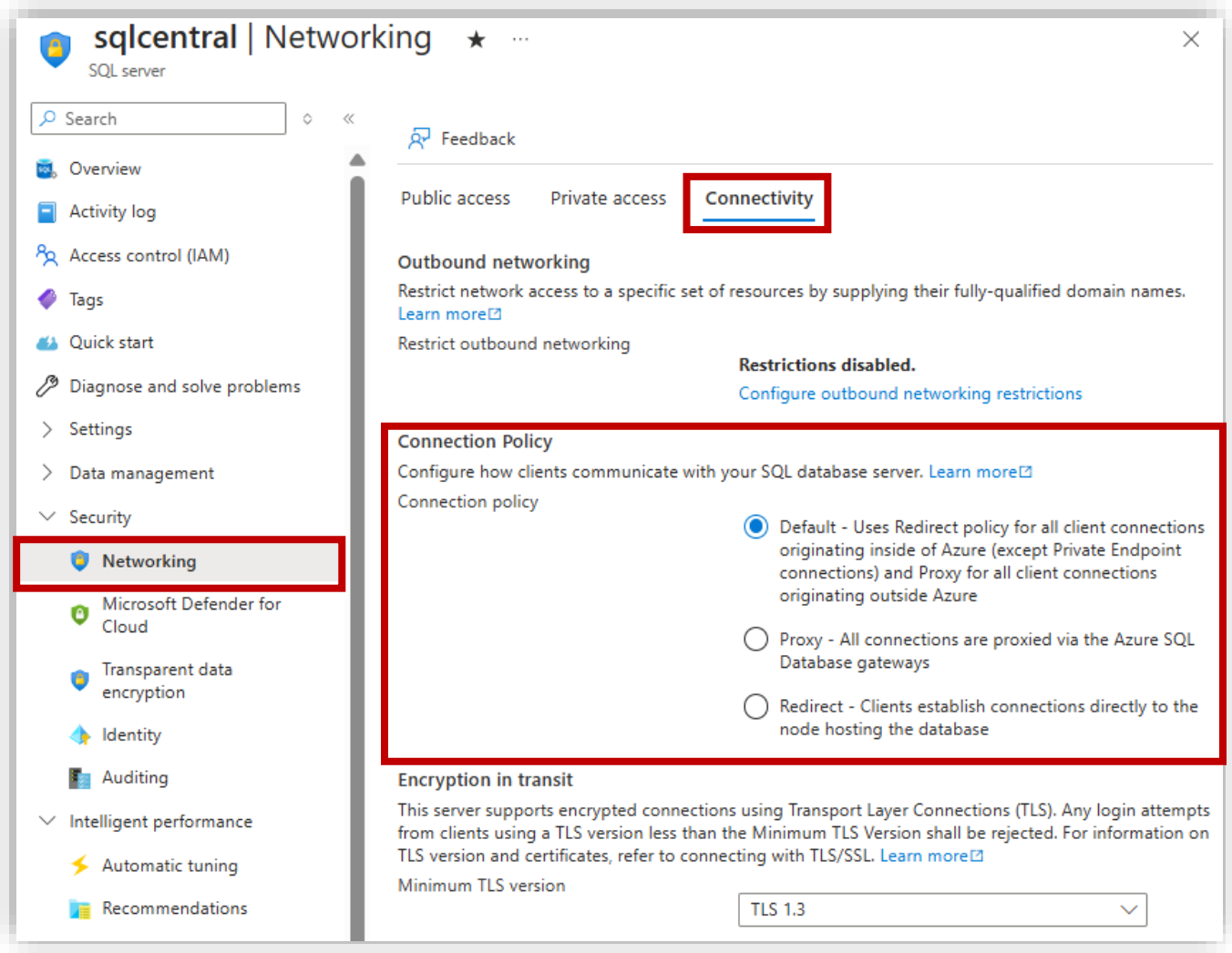
```
-- T-SQL Allow external IP access to SQL Database
sp_set_firewall_rule N'myRule1', '12.1.1.1','12.1.1.2'
```

Connection Policy

Redirect (recommended): Clients establish connections directly to the node hosting the database, leading to reduced latency and improved throughput.

Proxy: In this mode, all connections are proxied via the Azure SQL Database gateways, leading to increased latency and reduced throughput.

Default: This is the connection policy in effect on all servers after creation unless you explicitly alter the connection policy to either Proxy or Redirect.



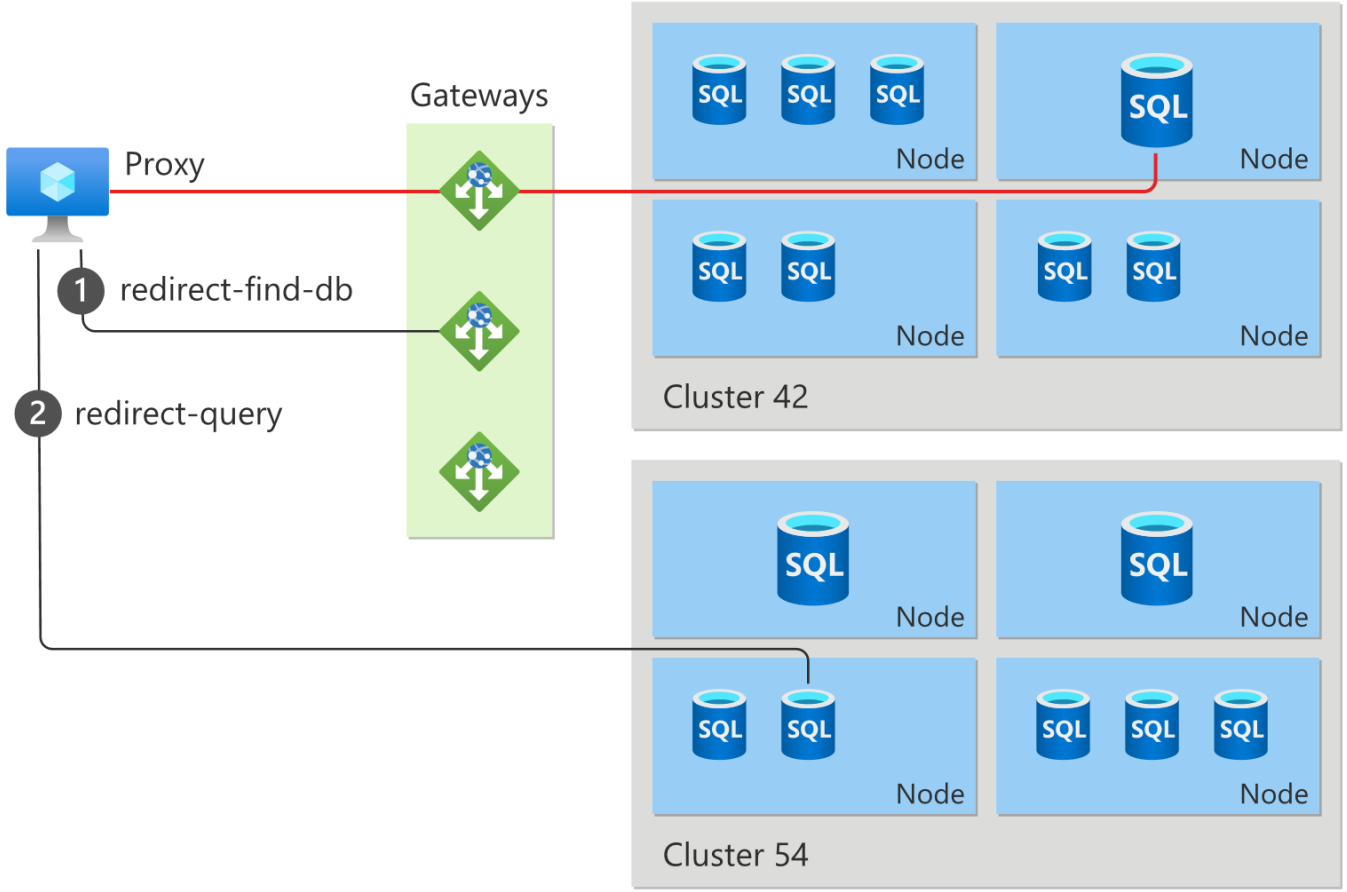
Connection Policy

The following diagram provides a high-level overview of the connectivity architecture

Clients connect to the gateway that has a public IP address and listens on port 1433.

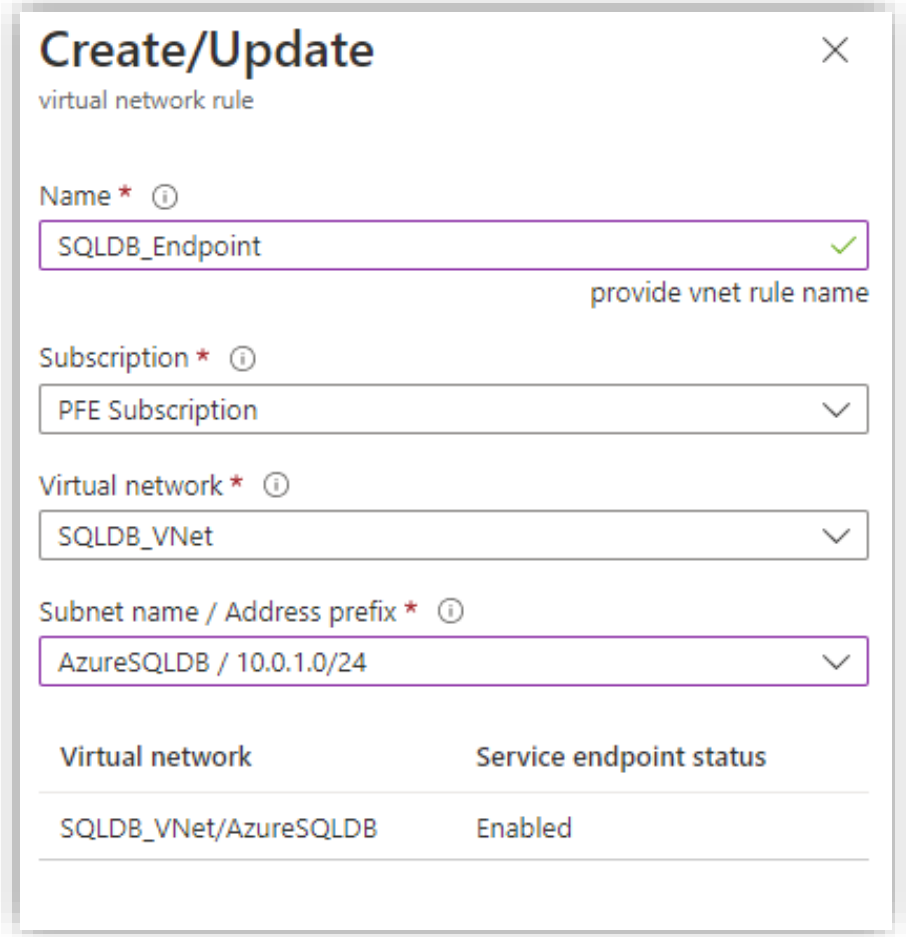
Depending on the effective connection policy, the gateway redirects or proxies the traffic to the correct database cluster.

Inside the database cluster, traffic is forwarded to the appropriate database.



Virtual Network service endpoints

- Allow access to Azure services
- IP Rules
- Only one geographic region
- Server-level, not database-level

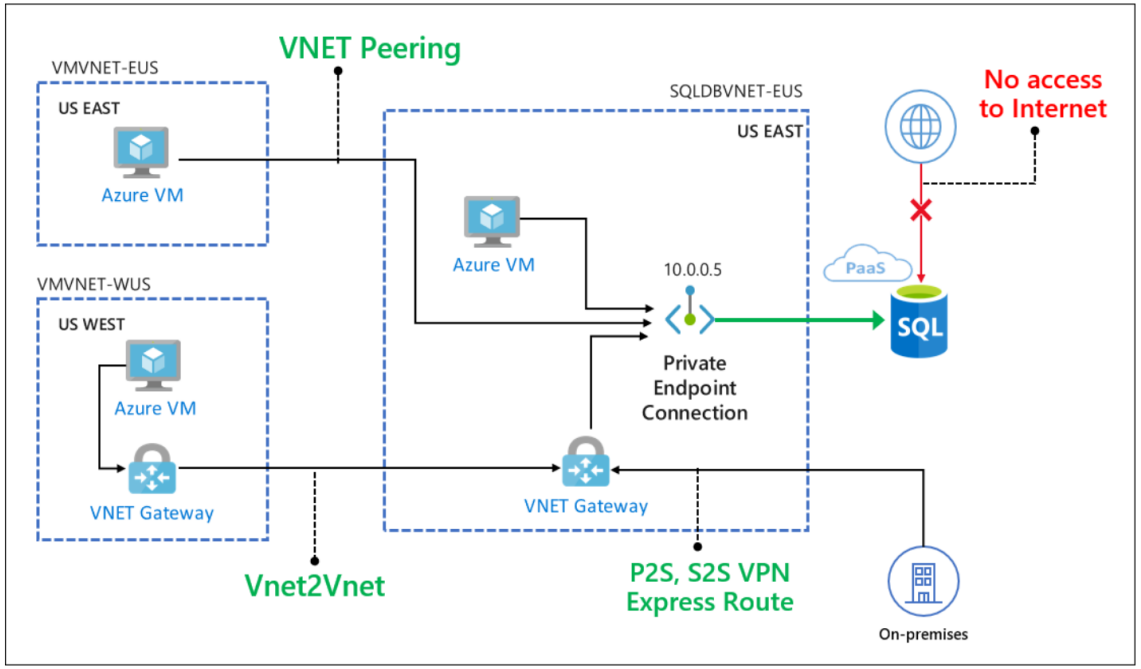


Private Link* for Azure SQL Database

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?

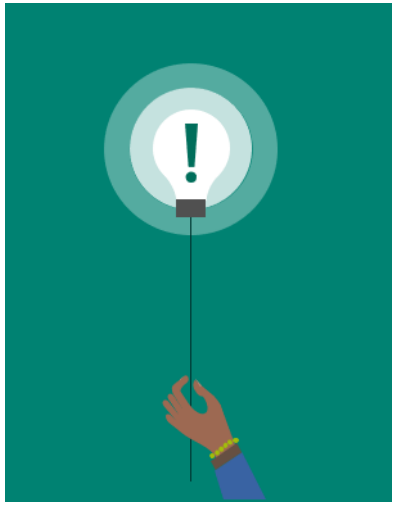
Why should you use Virtual Network Service Endpoints?

Lesson 5: Implement 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.

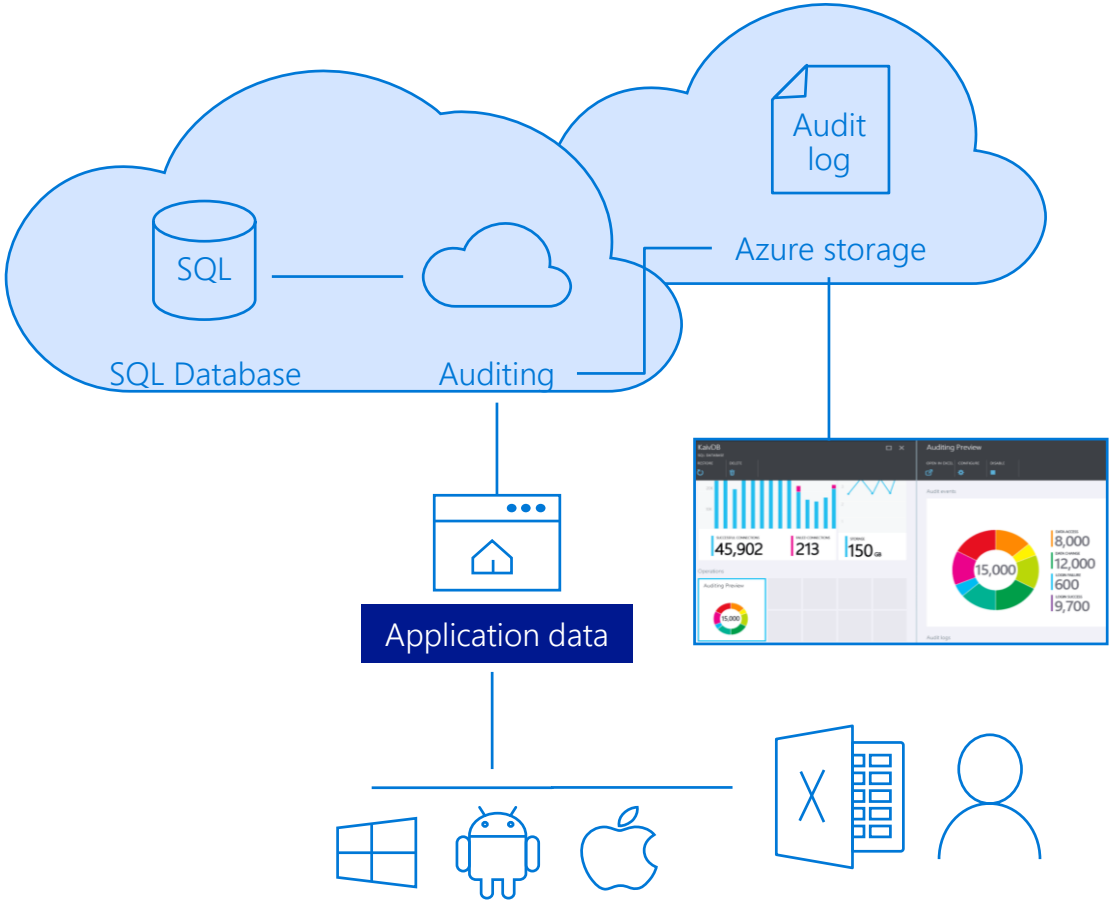
Dashboard views in portal for at-a-glance insights.

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?

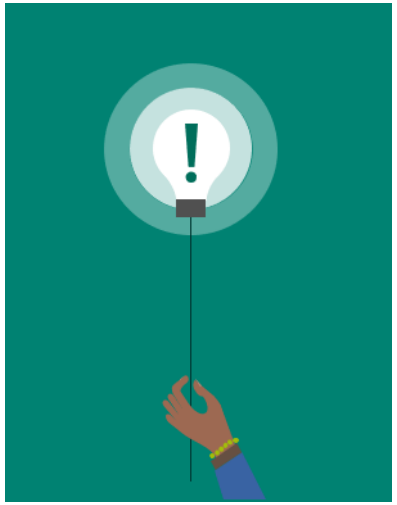
Which tools can you use to analyze the Audit Logs?

Lesson 6: Implement Ledger for Azure SQL Database

Objectives

After completing this learning, you will be able to:

- Understand what Ledger for SQL Server is and how to configure it.



Security enhancements - Ledger for SQL Server

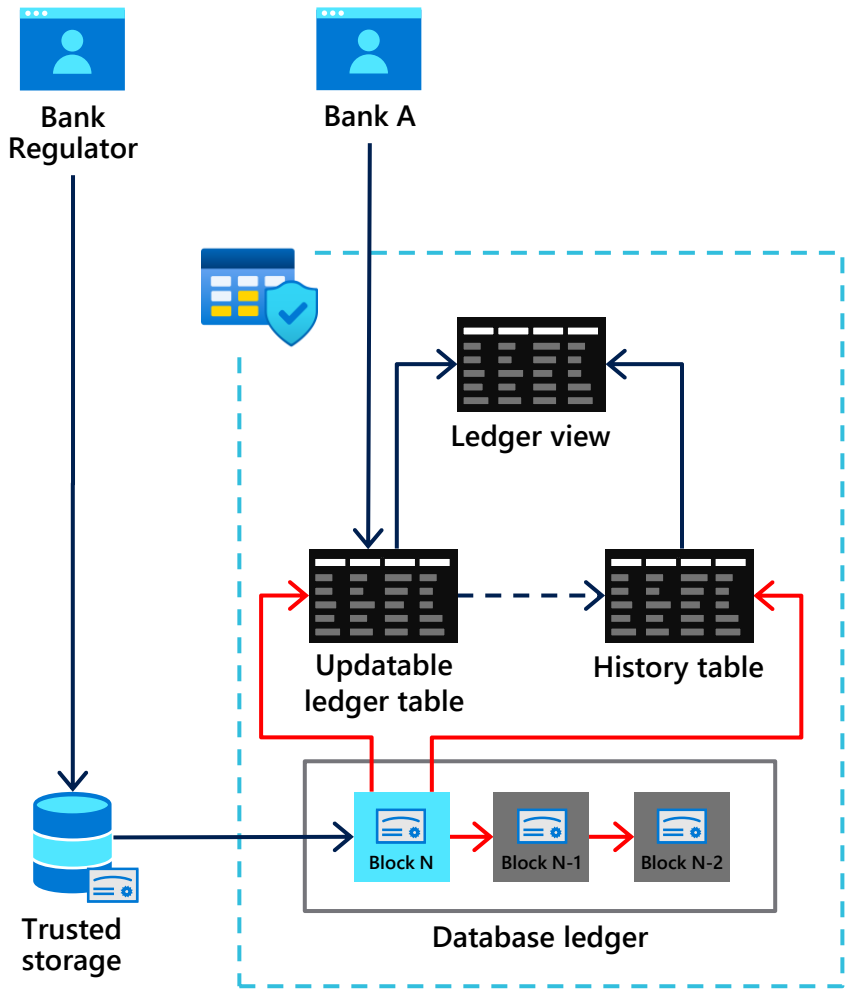
Ledger for SQL Server -The ledger feature provides tamper-evidence capabilities in your database. You can cryptographically attest to other parties, such as auditors or other business parties, that your data hasn't been tampered with.

Ledger for SQL Server

Tamper-evidence track record of data over time

Challenge: I want the power of blockchain in a centralized system like SQL Server

- Use a cryptographically hashed ledger detect tampering by malicious actors
- Built into SQL Server with T-SQL
- Establish digital trust in a centralized system using blockchain technology.
- Attest to other parties that data integrity has not been compromised
- Automatic digest storage



Ledger Tables – Updatable and Append-Only

Updatable Ledger Tables are standard SQL tables which allow updates and deletes

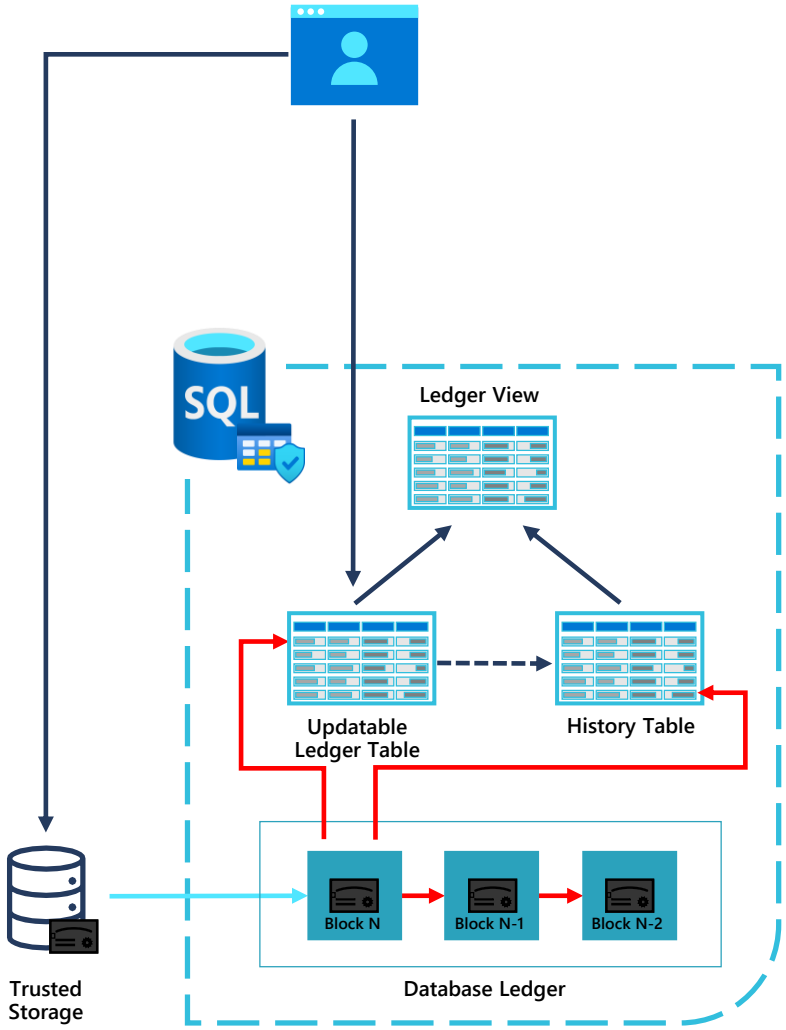
The history of rows that have been updated or deleted are preserved in the history table and easy-to-query Ledger View

Integrity of the updatable and history tables are maintained through cryptographic links from the Database Ledger

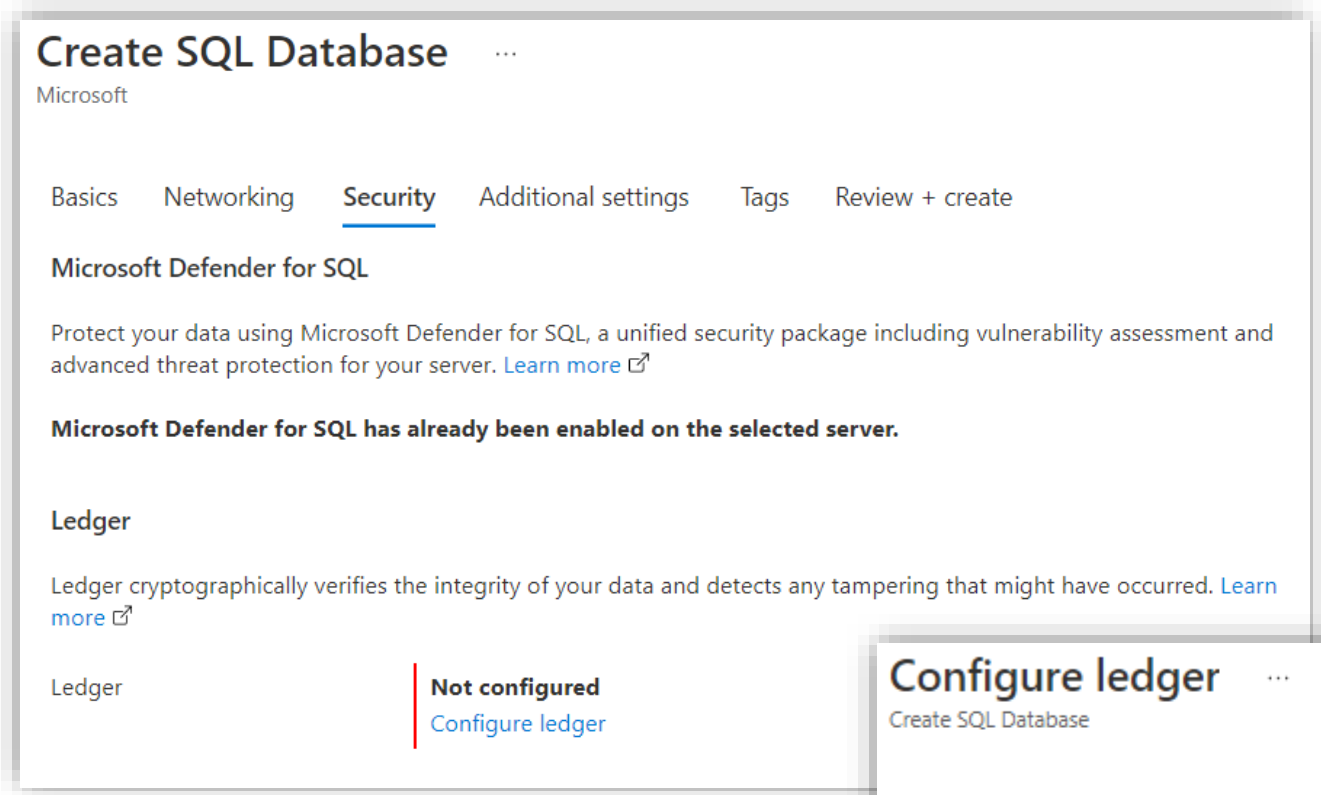
System periodically uploads digital receipts to a customer-configured trusted storage service

Customer can use digital receipts to verify the integrity of data in Ledger tables

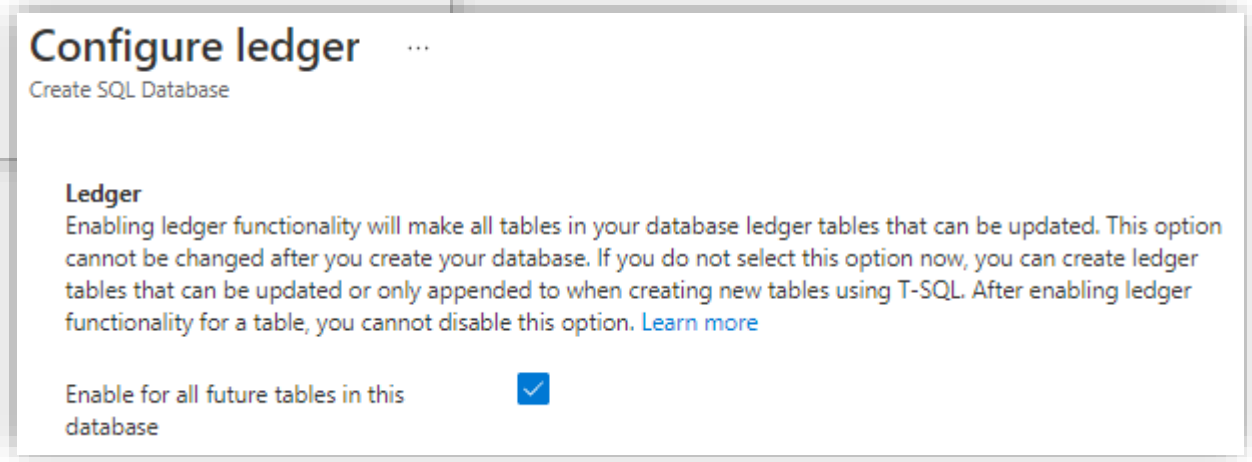
Append-Only Ledger Tables block UPDATE/DELETE at the API and remove the need for a history table



Configure Ledger in Azure SQL Database



Enabling the ledger functionality at the database level will make all tables in this database updatable ledger tables. This option cannot be changed after the database is created.



Creating an Account Balance Updatable Ledger Table

```sql
CREATE SCHEMA [Account];
GO
CREATE TABLE [Account].[Balance]
([CustomerID] INT NOT NULL PRIMARY KEY CLUSTERED,
    [LastName] VARCHAR (50) NOT NULL,
    [FirstName] VARCHAR (50) NOT NULL,
    [Balance] DECIMAL (10,2) NOT NULL)
WITH (SYSTEM_VERSIONING = ON (HISTORY_TABLE = [Account].[BalanceHistory]),
 LEDGER = ON);
```

	ledger_table_name	history_table_name	ledger_view_name
1	Account.Balance	Account.MSSQL_LedgerHistoryFor_1525580473	Account.Balance_Ledger

Viewing the Account Balance Updatable Ledger Table

```sql
SELECT ts.[name] + '.' + t.[name] AS [ledger_table_name]
, hs.[name] + '.' + h.[name] AS [history_table_name]
, vs.[name] + '.' + v.[name] AS [ledger_view_name]
FROM sys.tables AS t
JOIN sys.tables AS h ON (h.[object_id] = t.[history_table_id])
JOIN sys.views v ON (v.[object_id] = t.[ledger_view_id])
JOIN sys.schemas ts ON (ts.[schema_id] = t.[schema_id])
JOIN sys.schemas hs ON (hs.[schema_id] = h.[schema_id])
JOIN sys.schemas vs ON (vs.[schema_id] = v.[schema_id])
WHERE t.[name] = 'Balance';
```

	ledger_table_name	history_table_name	ledger_view_name
1	Account.Balance	Account.MSSQL_LedgerHistoryFor_1525580473	Account.Balance_Ledger

Add 4 Accounts In 2 Separate Transactions

Tx1: Add Nick with an opening balance of $50

Tx2: Add John, Joe and Mary

1. Each transaction has it's own unique transaction ID
2. Tx2 modified 3 rows, each tracked with a ledger sequence number

Updatable ledger table

Results Messages

	CustomerID	LastName	FirstName	Balance	ledger_start_transaction_id (1)	ledger_end_transaction_id	ledger_start_sequence_number (2)	ledger_end_sequence_number
1	1	Jones	Nick	50.00	999	NULL	0	NULL
2	2	Smith	John	500.00	1002	NULL	0	NULL
3	3	Smith	Joe	30.00	1002	NULL	1	NULL
4	4	Michaels	Mary	200.00	1002	NULL	2	NULL

Update Nick's Balance From $50 To $100

Applies to: Azure SQL Database, Managed Instance preview

Updatable ledger table – Nick's balance is now $100

	CustomerID	LastName	FirstName	Balance	ledger_start_transaction_id	ledger_end_transaction_id	ledger_start_sequence_number	ledger_end_sequence_number
1	1	Jones	Nick	100.00	1055	NULL	0	NULL
2	2	Smith	John	500.00	1002	NULL	0	NULL
3	3	Smith	Joe	30.00	1002	NULL	1	NULL
4	4	Michaels	Mary	200.00	1002	NULL	2	NULL

History Table – Shows the historical value of row containing Nick's opening balance

	CustomerID	LastName	FirstName	Balance	ledger_start_transaction_id	ledger_end_transaction_id	ledger_start_sequence_number	ledger_end_sequence_number
1	1	Jones	Nick	50.00	999	1055	0	1

Ledger View – Shows Nick's update as a delete followed but a subsequent insert

	CustomerID	LastName	FirstName	Balance	ledger_transaction_id	ledger_sequence_number	ledger_operation_type_id	ledger_operation_type_desc
1	1	Jones	Nick	50.00	999	0	1	INSERT
2	2	Smith	John	500.00	1002	0	1	INSERT
3	3	Smith	Joe	30.00	1002	1	1	INSERT
4	4	Michaels	Mary	200.00	1002	2	1	INSERT
5	1	Jones	Nick	50.00	1055	1	2	DELETE
6	1	Jones	Nick	100.00	1055	0	1	INSERT

Demonstration

Use SQL Server Ledger to track changes to a table.

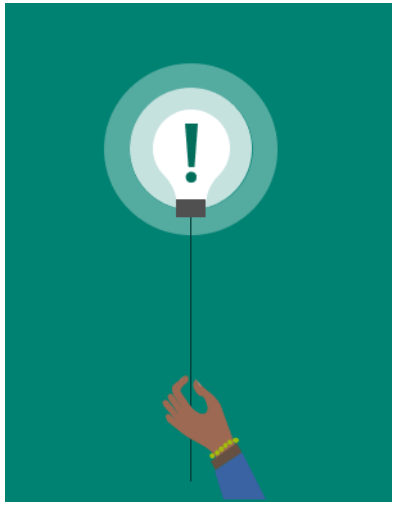
Questions?

Lesson 7: 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)

- Discovery & recommendations
- Labeling
- Query result set sensitivity
- Visibility
- Auditing access to sensitive data

Demonstration

Data Discovery and Classification

- Classify your SQL Database.

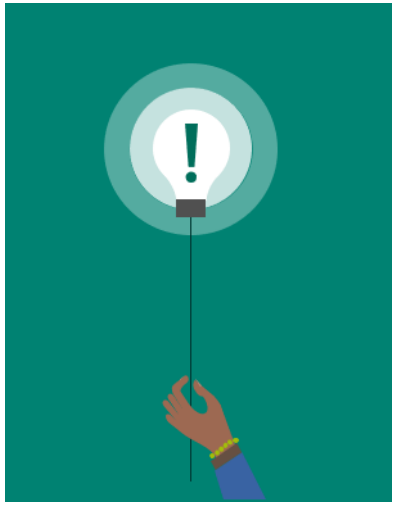
Questions?

Lesson 8: 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

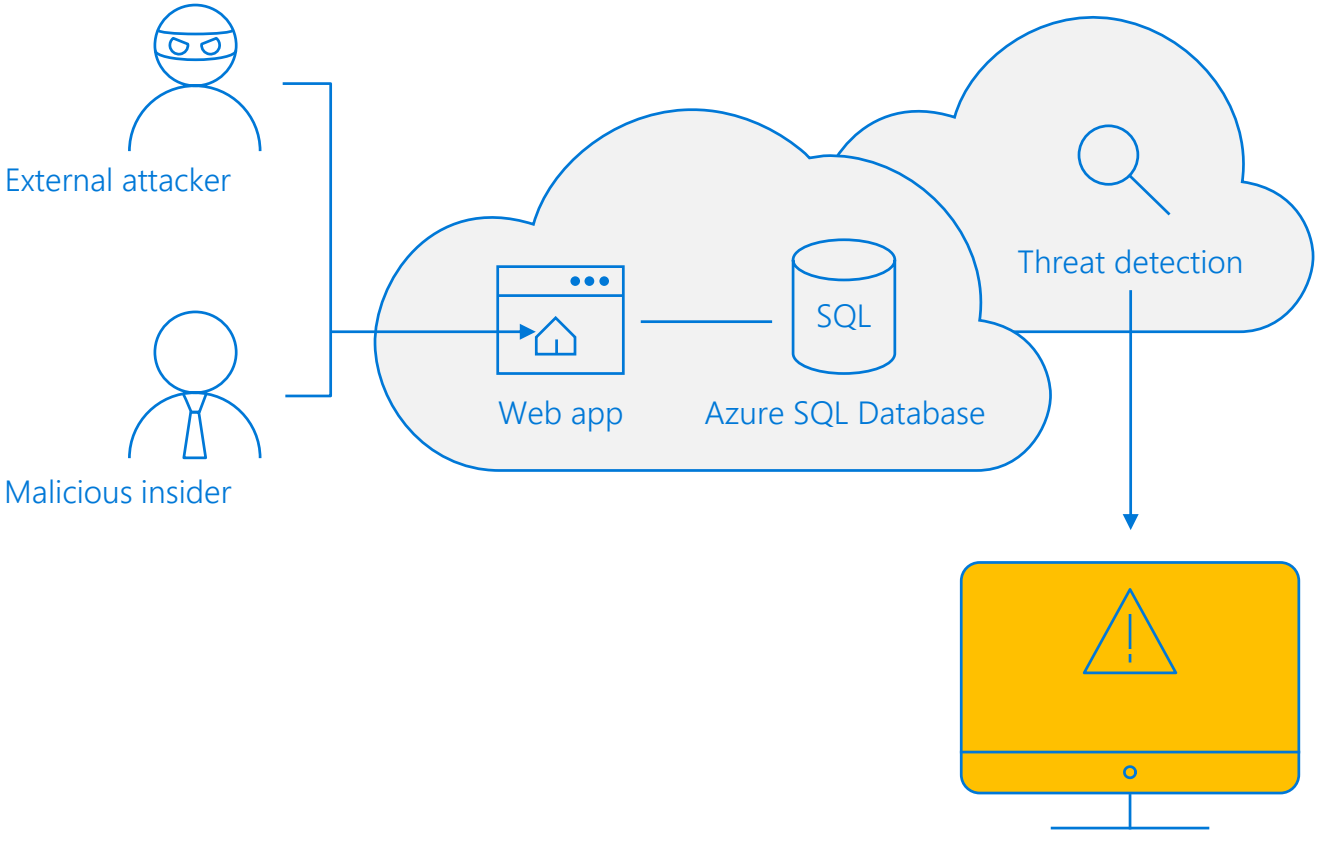
Advanced Threat Protection for single and pooled databases detects anomalous activities indicating unusual and potentially harmful attempts to access or exploit databases.

Advanced Threat Protection can identify:

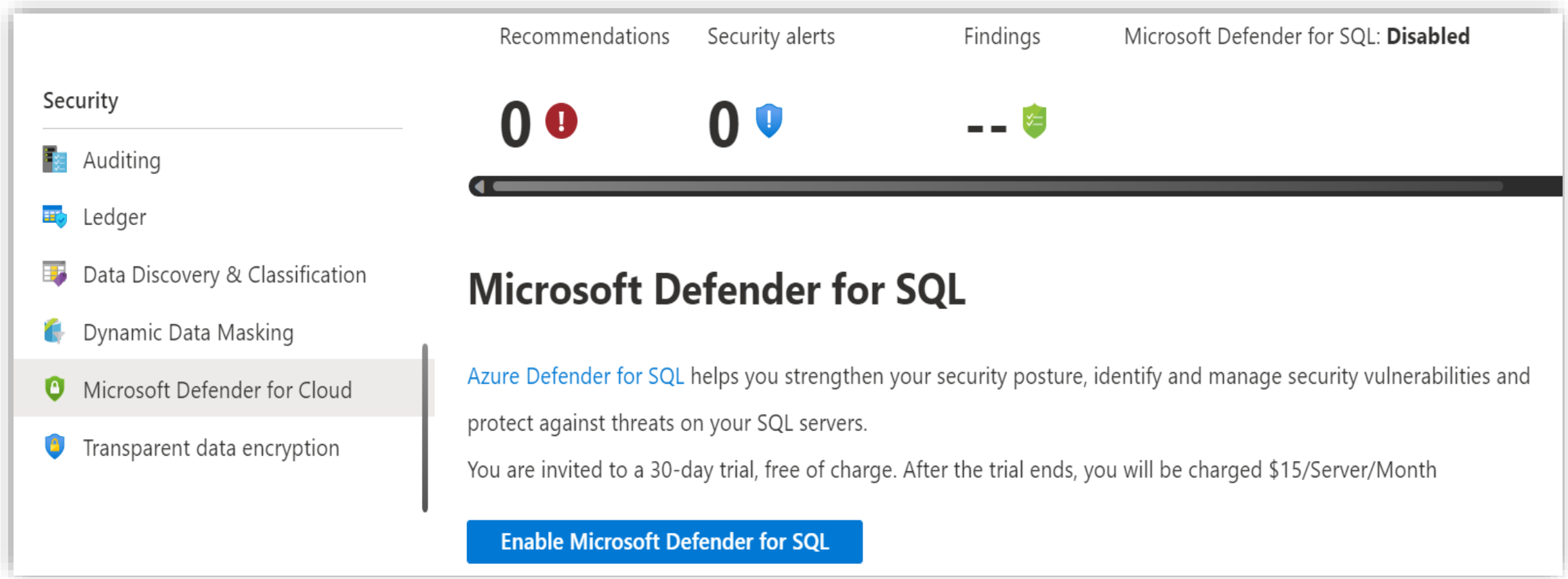
- Potential SQL injection, Access from unusual location or data center.
- Access from unfamiliar principal or potentially harmful application.
- Brute force SQL credentials.

Advanced Threat Detection (continued)

- Detect anomalous database activities that could indicate a potential threat.
- Configure threat detection policy in Azure Portal.
- Receive alerts from multiple database threat detectors that identify anomalous activities.
- Explore audit log around the time of an event.



Enable Microsoft Defender for SQL



Configure Microsoft Defender for SQL for Email Alerts

Recommendations 0

Security alerts 0

Findings --

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

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 detecte may indicate a SQL Injection attack on da 'samplecrmwedemo'.

View recent SQL alerts

Activity details

Severity	High
Subscription ID	
Subscription Name	DS-THREATDETECTION_DEMO_TOMERR_R&D_60843
Server	
Database	
IP address	
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

HIGH SEVERITY 3
MEDIUM SEVERITY 3

DESCRIPTION	COUNT	DETECTED BY	SEVERITY
Potential SQL Injection	3		High
Potential SQL Brute Force attempt	1	Microsoft	High
Attempted logon by a potentially harmful application	1	Microsoft	High
A possible vulnerability to SQL Injection	1	Microsoft	Medium
Logon from an unusual location	1	Microsoft	Medium
Logon by an unfamiliar principal	1	Microsoft	Medium

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	
DETECTED BY	Microsoft
ACTION TAKEN	Detected
ENVIRONMENT	Azure
RESOURCE TYPE	SQL Server
SERVER	
DATABASE	
IP ADDRESS	
PRINCIPAL NAME	dev1
APPLICATION	.Net SqlClient Data Provider
VULNERABLE STATEMENT	SELECT * FROM sqli_users WHERE username = ' 'OR 1 = 1 --' AND password = 'dfdfdfdfdf'
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

- Vulnerability to SQL Injection
- Potential SQL Injection
- Access from unusual location
- Access from unusual Azure data center
- Access from unfamiliar principal
- Access from a potentially harmful application
- Brute force SQL credentials

Demonstration

Microsoft Defender for for Azure SQL Database

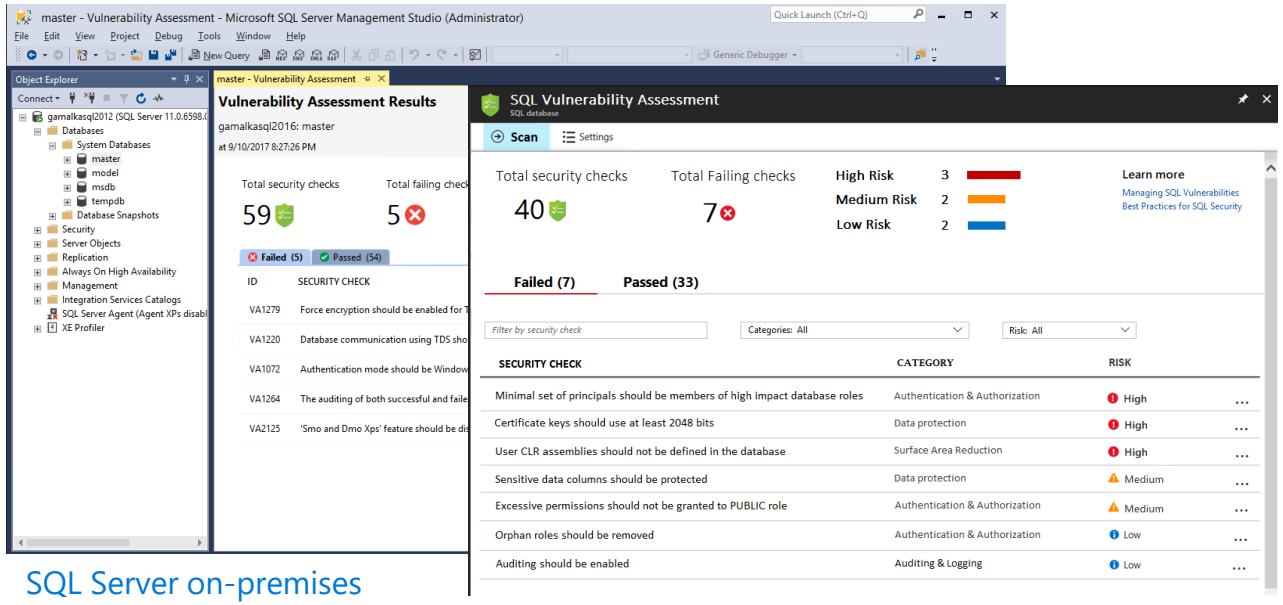
- Enable Threat Detection for Azure SQL Database.

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.

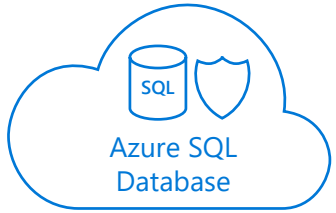
SQL Vulnerability Assessment (continued)

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

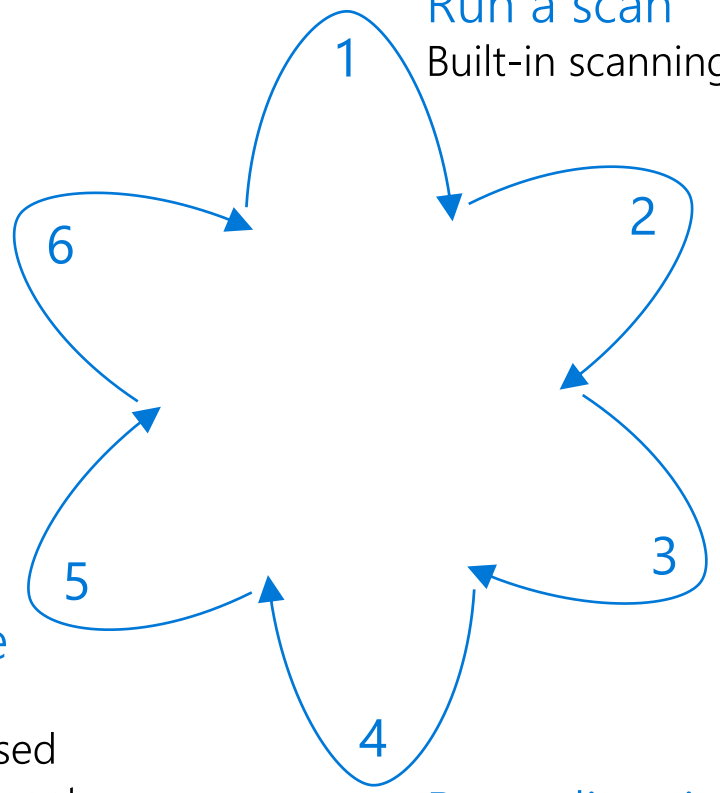


Vulnerability Assessment

Identifies, tracks, and resolves SQL security vulnerabilities

Developer/DBA

Using Vulnerability Assessment



1. **Run a scan**
   Built-in scanning service.
2. **View a report**
   All-up assessment of security state.
3. **Drill-down to results**
   View detailed results and understand how they impact database security.
4. **Remediate issues**
   Run scripts that resolve vulnerabilities directly within the report.
5. **Set a baseline**
   Customize scan requirements based on your environment.
6. **Detect deviations**
   Subsequent scans will alert on deviations from your baseline.

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 Entra ID 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

Microsoft