

# Cyber Shujaa

## Cloud Security Specialist

### **Assignment 16 -AZ-500: Secure databases**

**NAME: WELDON KIPKIRUI KENEI**

**CS NO: ADC-CSS02-25027**

# Introduction

In this module, we will be equipped with the knowledge and skills needed to plan and implement robust security measures for Azure SQL Database and Azure SQL Managed Instance, ensuring data protection and regulatory compliance.

## Azure SQL Database and SQL Managed Instance security

Azure SQL Database and SQL Managed Instance employ a layered defense-in-depth approach to security, ensuring robust protection for your data. Here are the key security features:

**Network Security:** Firewalls prevent unauthorized access, with IP firewall rules and virtual network service endpoints controlling traffic.

**Authentication:** Supports SQL authentication (username/password) and Microsoft Entra ID authentication, allowing centralized identity management.

**Access Management:** Role-based access control (RBAC) ensures only authorized users can manage databases and servers.

**Threat Protection:** This includes auditing, anomaly detection, and security alerts to monitor and respond to potential threats.

**Data Protection:** Encryption at rest and in transit safeguards sensitive information.

**Compliance:** Helps meet data privacy standards and regulatory requirements.

## Enable Microsoft Entra database authentication

### Authentication Methods

- **SQL Authentication:** Traditional username/password login (less secure).
- **Azure Active Directory (Azure AD) Authentication:** More secure, integrates with Microsoft Entra for centralized identity management, supports Multi-Factor Authentication (MFA), and eliminates password storage in the database.

### Authorization (Permissions & Roles)

- **Server-Level Roles:** Managed via the master database (e.g., `MS\_ServerStateManager` for server-wide permissions).
- **Database-Level Roles:** Predefined roles (e.g., `db\_owner`, `db\_datareader`, `db\_datawriter`) and custom roles for granular access control.
- **Schema-Level Permissions:** Permissions can be granted at the schema level (e.g., `SELECT` on a specific schema).

### Securing Access with Contained Database Users

- **Contained Users:** Users stored within the database (not linked to a server login), improving portability and simplifying management.

- **Microsoft Entra ID-Integrated Users:** Allows direct Entra ID user/group assignments to databases, enabling seamless authentication without SQL logins.

### Best Practices

- Use Entra ID authentication over SQL logins for better security.
- Follow the principle of least privilege —grant only necessary permissions.
- Leverage database roles for easier permission management.

## Enable and monitor database audit

Auditing for Azure Structured Query Language (SQL) Database and Azure Synapse Analytics tracks database events and writes them to an audit log in your Azure storage account, Log Analytics workspace or Event Hubs.

The default auditing policy captures key events like successful and failed logins and query executions. Once auditing is enabled, you can monitor logs using Azure Monitor, Log Analytics, or querying audit records directly. This allows administrators to detect anomalies, investigate security incidents, and maintain regulatory compliance.

## Identify use cases for the Microsoft Purview governance portal

Microsoft Purview provides data governance capabilities for Azure SQL Database and SQL Managed Instance, helping organizations manage, classify, and protect their data. It enables data discovery, allowing users to scan and catalog database assets for better visibility. Additionally, Purview supports sensitivity labeling, ensuring that confidential data is properly classified and secured.

Another key use case is compliance management, where Purview helps organizations meet regulatory requirements like GDPR, HIPAA, and CCPA. It also facilitates access control and auditing, allowing administrators to monitor database activity and enforce security policies.

## Implement data classification of sensitive information by using the Microsoft Purview governance portal

Azure SQL Database and SQL Managed Instance offer data classification capabilities to help organizations identify and protect sensitive information. The classification engine scans databases to detect columns containing potentially sensitive data, allowing administrators to apply sensitivity labels and information types for better security management.

These labels help enforce access control policies and support auditing to monitor interactions with classified data

## Plan and implement a dynamic mask

Dynamic Data Masking (DDM) in Azure SQL Database and SQL Managed Instance helps protect sensitive data by masking it for nonprivileged users. This policy-based security feature ensures that confidential information, such as email addresses or personal identifiers, is hidden in query results while remaining unchanged in the database. Administrators can define masking rules for specific columns, allowing controlled data exposure without modifying the underlying data.

DDM supports various masking functions, including default masking, partial masking, and custom-defined masks. Users with administrative privileges, such as `db_owner`, can view unmasked data, while others see only the masked version. This feature enhances data security and compliance, preventing unauthorized access to sensitive information, and can be implemented from the portal or using REST APIs.

## Implementing Transparent Data Encryption

Azure SQL Database and SQL Managed Instance use Transparent Data Encryption (TDE) to encrypt data at rest, protecting against unauthorized access to database files, backups, and transaction logs.

TDE employs AES-256 encryption and automatically encrypts the storage layer without requiring changes to the application. By default, Azure manages the encryption key (using a service-managed key), but for greater control, customers can use customer-managed keys (CMKs) stored in Azure Key Vault, enabling compliance with stricter security policies.

TDE is enabled by default in Azure SQL services, ensuring real-time encryption and decryption as data is accessed. It safeguards against threats like stolen storage media or offline attacks.

## Recommend when to use Azure SQL Database Always Encrypted

Always Encrypted is a security feature in Azure SQL Database and SQL Managed Instance that protects sensitive data both at rest and in use through client-side encryption. Unlike Transparent Data Encryption (TDE), which only encrypts data at rest, Always Encrypted ensures data remains encrypted during query processing, preventing database administrators or cloud operators from accessing plaintext data.

The encryption keys are stored separately from the database (either in a client application or Azure Key Vault), and only authorized applications with proper credentials can decrypt the data. This makes it ideal for protecting highly sensitive information like credit card numbers or personal identification data.

The feature supports two encryption types: **deterministic encryption** (which allows equality comparisons and indexing) and **randomized encryption** (more secure but prevents operations on encrypted data). Always Encrypted integrates with tools like SQL Server Management Studio (SSMS) and requires client driver support for secure enclave operations when performing computations on encrypted data.

## Implement Azure SQL firewall

Azure SQL Database and SQL Managed Instance use firewall rules to control network access and prevent unauthorized connections. The firewall blocks all access by default, requiring explicit rules to permit traffic.

There are two types of firewall rules: **server-level rules** (managed through the Azure portal, PowerShell, or CLI) that apply to all databases on a logical server, and **database-level rules** (configured via T-SQL) that provide granular control for individual databases. Server-level rules are useful for administrators, while database-level rules offer flexibility for multi-tenant applications. The firewall integrates with virtual network service endpoints, allowing secure access from specific subnets while keeping the database isolated from public internet exposure.

For enhanced security, Azure SQL supports private endpoints that enable private IP connectivity within a virtual network, completely bypassing public internet access. The firewall also works alongside Entra ID authentication and other security features to create layered protection.

## Conclusion

In this module we learned how to plan and implement robust security measures for Azure SQL Database and Azure SQL Managed Instance, including enabling secure database authentication with Microsoft Enterprise Identity, database auditing, utilizing the Microsoft Purview governance portal for data classification, implementing dynamic masking for sensitive information, Transparent Database Encryption (TDE), and making informed decisions about when to use Azure SQL Database Always Encrypted for client-side data encryption.

<https://learn.microsoft.com/api/achievements/share/en-us/weldonkenei-6817/W2P4JR3N?sharingId=1C66F93EC0530812>


Summary - Training | Microsoft

learn.microsoft.com/en-us/training/modules/security-azure-sql-database-azure-sql-managed-instance/12-summary#completion

GmailGoogle Cloud Skills...PY4E - Python for E...Weldon Keneiweldonhack/YouTubeTo-DoNewsManusAll Bookmarks


Achievements

Ask Learn



AZ-500: Secure compute, storage, and databases

All module assessments passed



Plan and implement security for Azure SQL Database and Azure SQL Managed Instance

Module assessment passed

You have earned 2 achievements!

Congratulations, but what should you do next?

---

First, let's share your achievement