

Cyber Shujaa

Cloud Security Specialist

Assignment 16 -AZ-500: Secure storage

NAME: WELDON KIPKIRUI KENEI

CS NO: ADC-CSS02-25027

Introduction

This module will equip us with the knowledge and skills to plan and implement comprehensive security measures for Azure storage resources, safeguarding data integrity, confidentiality, and availability. We will be able to secure data at rest and in transit, manage access control effectively, and implement encryption and data protection measures to ensure the security of critical data assets.

Azure storage

The Azure Storage platform is Microsoft's cloud storage solution for modern data storage scenarios. Azure Storage offers highly available, massively scalable, durable, and secure storage for a variety of data objects in the cloud.

Azure Storage data objects are accessible from anywhere in the world over HTTP or HTTPS via a REST API. Azure Storage also offers client libraries for developers building applications or services with .NET, Java, Python, JavaScript, C++, and Go.

Developers and IT professionals can use Azure PowerShell and Azure CLI to write scripts for data management or configuration tasks. The Azure portal and Azure Storage Explorer provide user interface tools for interacting with Azure Storage.

Azure provides multiple storage options, including Blob Storage for unstructured data, File Storage for fully managed file shares, Queue Storage for messaging between services, Table Storage for NoSQL structured data, and Disk Storage for virtual machine disks.

Configure access control for storage accounts

Azure Storage provides several options for authorizing access to resources, ensuring security and control over data. Role-based access control (RBAC) allows administrators to assign precise permissions using Azure Active Directory (Azure AD), enabling authentication via identities. Shared Access Signatures (SAS) grant temporary, fine-grained access to specific resources without exposing full credentials. Storage account keys provide unrestricted access but should be managed carefully for security.

Additionally, Azure Storage supports Microsoft Entra ID for enhanced identity-based authentication and anonymous public access for select scenarios. These options help balance security, flexibility, and ease of management across different use cases.

Manage the life cycle for storage account access keys

Managing the life cycle of Azure Storage account access keys is crucial for maintaining security and preventing unauthorized access. Regularly rotating access keys minimizes the risk of exposure, and Azure provides two keys per storage account to enable seamless key updates without service disruption. Administrators should use Azure Key Vault to securely store and manage access keys, automating key rotation for enhanced security.

Restricting direct key usage by leveraging Microsoft Entra ID authentication for applications adds another layer of protection. Additionally, auditing key usage through Azure Monitor and logging access activities ensures proactive security management and compliance with best practices.

Select and configure an appropriate method for access to Azure Files

Azure Files supports multiple access methods, including SMB (Server Message Block) for seamless integration with Windows, NFS (Network File System) for Linux-based workloads, and REST APIs for programmatic access.

Authentication options include Microsoft Entra for identity-based access, shared access signatures (SAS) for temporary permissions, and storage account keys for full access (though less secure).

Configuration involves setting up the correct networking, permissions, and security policies to ensure secure, efficient access to Azure Files while meeting operational requirements.

Select and configure an appropriate method for access to Azure Blobs

Azure Blob Storage supports multiple access methods, including REST APIs for direct programmatic access, Azure CLI and PowerShell for administrative operations, and SDKs for integration into applications.

Authentication options include Microsoft Entra ID for identity-based access, Shared Access Signatures (SAS) for temporary permissions, and storage account keys for full control (though less secure).

Proper configuration involves setting up permissions, enabling encryption, implementing network security settings, and defining lifecycle policies to ensure secure and efficient blob storage access.

Select and configure an appropriate method for access to Azure Tables

Azure Storage integrates with Microsoft Entra ID to authenticate and authorize requests to table data using Azure role-based access control (Azure RBAC). A security principal—whether a user, group, or application—first authenticates through Entra ID, obtaining an OAuth 2.0 token to authorize access to the Table service.

Microsoft Entra authorization is recommended over Shared Key for better security and minimal privilege access. Supported in all public regions and national clouds, this method requires storage accounts created using the Azure Resource Manager deployment model. Entra ID authentication enables managed identities for Azure-hosted applications, ensuring secure access.

Azure RBAC roles define permissions at various scopes (table, storage account, resource group, subscription, or management group) to ensure proper access control. Built-in roles such as Storage Table Data Contributor (read/write/delete) and Storage Table Data Reader (read-only) facilitate granular access management.

Select and configure an appropriate method for access to Azure Queues

Azure Queues support access via REST APIs, Azure SDKs, Azure CLI, and PowerShell for integration into applications and automation. Authentication options include Microsoft Entra ID for identity-based access with Azure role-based access control (RBAC), Shared Access Signatures (SAS) for temporary, restricted permissions, and storage account keys for full control (though less secure).

Configuration involves setting queue permissions, defining policies for message retention and processing, enabling encryption for data security, and implementing network restrictions to ensure secure access. Proper setup optimizes message flow and security in cloud-based applications.

Select and configure appropriate methods for protecting against data security threats, including soft delete, backups, versioning, and immutable storage

Protecting against data security threats in Azure Storage requires a combination of safeguards, including soft delete, backups, versioning, and immutable storage.

Soft Delete

Soft delete helps recover accidentally deleted blobs, files, or even entire storage accounts by retaining deleted data for a specified retention period. It can be enabled at the blob, file share, and storage account levels to prevent permanent loss.

Backups

Regular backups ensure data recovery from unexpected incidents such as corruption or accidental deletions. Azure Backup provides automated, policy-driven backup solutions, integrating with Azure Storage to create restorable recovery points.

Versioning

Azure Blob Storage supports versioning, automatically keeping previous states of blobs when changes occur. This helps track modifications, revert unintended updates, and recover from overwrites, providing strong data integrity protection.

Immutable Storage

For compliance and security, immutable storage enforces write-once, read-many (WORM) policies, preventing modifications or deletions within a specified retention period. This is useful for regulatory compliance, ensuring audit-proof data protection.

Configure Bring Your Own Key (BYOK)

The process of importing a key generated outside Key Vault is referred to as Bring Your Own Key (BYOK). The process involves generating a key using an on-premises hardware security module (HSM) or a key management service, then importing it into Azure Key Vault.

Once stored in Key Vault, the key can be used for encrypting Azure services like Azure Storage, SQL Database, and Azure Information Protection. As administrators, we must configure access policies to control who can use the key, enable logging and monitoring for security tracking, and ensure key rotation to maintain protection.

Enable double encryption at the Azure Storage infrastructure level

Azure Storage automatically encrypts all data at the service level using 256-bit AES encryption, ensuring strong security compliance. For customers requiring higher assurance, Azure offers double encryption at the infrastructure level, where data is encrypted twice—once at the service level and again at the infrastructure level—using two different encryption algorithms and keys.

This additional layer protects against scenarios where one encryption method or key might be compromised. Infrastructure encryption can be enabled at the time of storage account creation and applies to general-purpose v2, premium block blob, premium page blob, and premium file share accounts. It relies on Microsoft-managed keys and cannot be disabled once configured. Azure Policy provides built-in policies to enforce infrastructure encryption for compliance

Conclusion

In this module, we learned how to plan and implement comprehensive security measures for Azure storage resources, including configuring access control for storage accounts, managing access keys, selecting and configuring appropriate methods for accessing various Azure storage services, protecting against data security threats through techniques like soft delete, backups, versioning, and immutable storage, and enhancing data encryption with Bring Your Own Key (BYOK) and double encryption at the Azure Storage infrastructure level.

<https://learn.microsoft.com/api/achievements/share/en-us/weldonkenei-6817/ZB5D7S62?sharingId=1C66F93EC0530812>


World x Remi x CSS2 x Plan x Summ x Assign x AZ-9 x AZ-10 x Theor x Keny x +

learn.microsoft.com/en-us/training/modules/security-storage/13-summary#completion

Gmail Google Cloud Skills PY4E - Python for E... Weldon Kenei weldonhack/ YouTube To-Do News Manus All Bookmarks

Achievements

Ask Learn



Plan and implement security for storage

Module assessment passed

You have earned an achievement!

Congratulations, but what should you do next?

First, let's share your achievement

You put in the time to learn something new, let your network share in your victory!

[LinkedIn](#) [Email](#) [Twitter](#) [Facebook](#) [Share](#)

Don't lose your momentum, keep learning

Below you will find recommended content to help you along your path!