

Domain 4: Implement and manage storage in Azure

Configure storage accounts

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/configure-storage-accounts/1-introduction>

Introduction

Completed

Azure Storage is Microsoft's cloud storage solution for modern data storage scenarios.

Suppose you work for a large e-commerce company that needs to store and serve a vast number of product images to its customers. The company wants a scalable and reliable solution that can handle high traffic and ensure data durability. They want to quickly restore data if there's an outage.

In this module, you learn how to configure storage accounts and select appropriate storage types in Azure. The module covers topics such as implementing replication strategies, and configuring secure access to storage.

The goal of this module is to provide Azure Administrators with the knowledge and skills to effectively configure and manage Azure storage accounts.

Learning objectives

In this module, you learn how to:

- Identify features and usage cases for Azure storage accounts.
- Select between different types of Azure Storage and create storage accounts.
- Select a storage replication strategy.
- Configure secure network access to storage endpoints.

Skills measured

The content in the module helps you prepare for [Exam AZ-104: Microsoft Azure Administrator](#).

Prerequisites

- Experience with the Azure portal.
- Familiarity with managing different types of data storage.

2. Implement Azure Storage

<https://learn.microsoft.com/en-us/training/modules/configure-storage-accounts/2-implement-azure-storage>

Implement Azure Storage

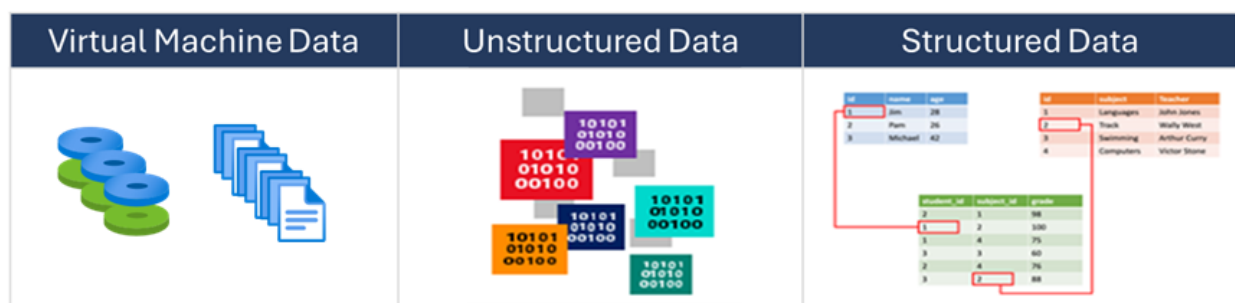
Completed

Azure Storage is Microsoft's cloud storage solution for modern data storage scenarios. Azure Storage offers a massively scalable object store for data objects. It provides a file system service for the cloud, a messaging store for reliable messaging, and a NoSQL store.

Azure Storage is an AI-ready service that you can use to store files, messages, tables, and other types of information. You use Azure Storage for applications like file shares. Developers use Azure Storage for working data. Working data includes websites, mobile apps, and desktop applications. Azure Storage is also used by IaaS virtual machines, and PaaS cloud services.

Things to know about Azure Storage

You can think of Azure Storage as supporting three categories of data: structured data, unstructured data, and virtual machine data. Review the following categories and think about which types of storage are used in your organization.



Category	Description	Storage examples
Virtual machine data	Virtual machine data storage includes disks and files. Disks are persistent block storage for Azure IaaS virtual machines. Files are fully managed file shares in the cloud.	Storage for virtual machine data is provided through Azure managed disks. Data disks are used by virtual machines to store data like database files, website static content, or custom application code. The number of data disks you can add depends on the virtual machine size.

Category	Description	Storage examples
Unstructured data	Unstructured data is the least organized. The format of unstructured data is referred to as <i>nonrelational</i> .	Unstructured data can be stored by using Azure Blob Storage and Azure Data Lake Storage. Blob Storage is a highly scalable, REST-based cloud object store. Azure Data Lake Storage is the Hadoop Distributed File System (HDFS) as a service.
Structured data	Structured data is stored in a relational format that has a shared schema. Structured data is often contained in a database table with rows, columns, and keys. Tables are an autoscaling NoSQL store.	Structured data can be stored by using Azure Table Storage, Azure Cosmos DB, and Azure SQL Database. Azure Cosmos DB is a globally distributed database service. Azure SQL Database is a fully managed database-as-a-service built on SQL.

Things to consider when using Azure Storage

As you think about your configuration plan for Azure Storage, consider these prominent features.

- **Consider durability and availability.** Azure Storage is durable and highly available. Redundancy ensures your data is safe during transient hardware failures. You replicate data across datacenters or geographical regions for protection from local catastrophe or natural disaster. Replicated data remains highly available during an unexpected outage.
- **Consider secure access.** Azure Storage encrypts all data. Azure Storage provides you with fine-grained control over who has access to your data.
- **Consider scalability.** Azure Storage is designed to be massively scalable to meet the data storage and performance needs of modern applications.
- **Consider manageability.** Microsoft Azure handles hardware maintenance, updates, and critical issues for you.
- **Consider data accessibility.** Data in Azure Storage is accessible from anywhere in the world over HTTP or HTTPS. Microsoft provides SDKs for Azure Storage in various languages. You can use .NET, Java, Node.js, Python, PHP, Ruby, Go, and the REST API. Azure Storage supports scripting in Azure PowerShell or the Azure CLI. The Azure portal and Azure Storage Explorer offer easy visual solutions for working with your data.
- **Consider SFTP support.** Blob Storage can use SFTP (SSH File Transfer Protocol), so you can keep using existing SFTP tools to move files directly to and from blobs. To use SFTP, enable hierarchical namespace (HNS). You can turn it on when you create the storage account (Advanced tab) or later under Settings → Configuration.
- **Consider NFSv3 protocol support.** Blob Storage can also be accessed using NFSv3, which lets Linux clients mount a container like an NFS share. NFSv3 can simplify migrations from Linux file workloads to Azure.

- **Consider default authorization preferences.** In the Azure portal, you can enable **Default to Microsoft Entra authorization**. This authentication makes role-based access control (RBAC) the default instead of shared access keys, which can improve security.

3. Explore Azure Storage services

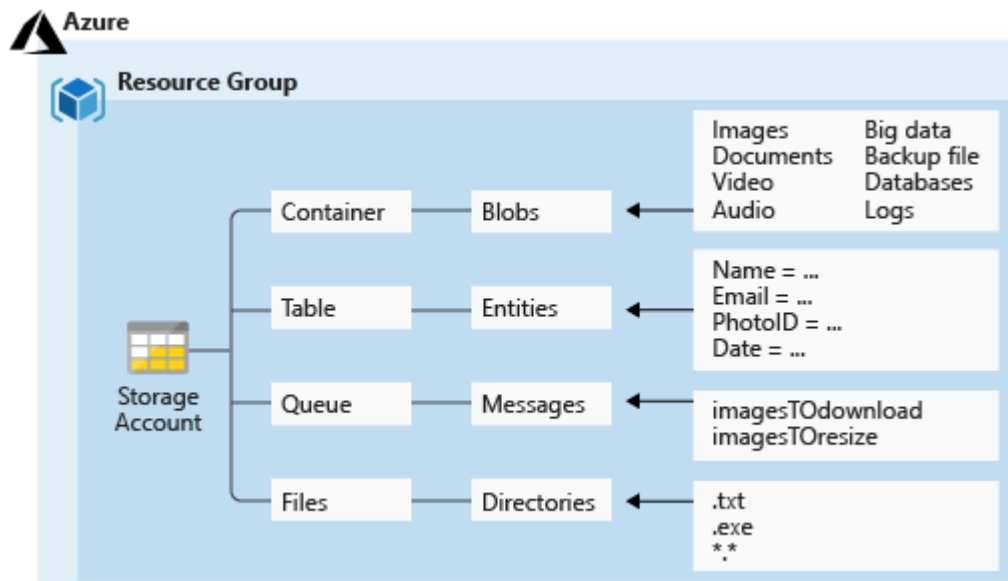
<https://learn.microsoft.com/en-us/training/modules/configure-storage-accounts/3-explore-azure-storage-services>

Explore Azure Storage services

Completed



Let's examine the details of these services.



Azure Blob Storage

[Azure Blob Storage](#) is Microsoft's object storage solution for the cloud. Blob Storage is optimized for storing massive amounts of unstructured or *nonrelational* data, such as text or binary data. Blob

Storage is ideal for:

- Serving images or documents directly to a browser.
- Storing files for distributed access.
- Streaming video and audio.
- Storing data for backup and restore, disaster recovery, and archiving.
- Storing data for analysis by an on-premises or Azure-hosted service.

Objects in Blob Storage can be accessed from anywhere in the world via HTTP or HTTPS. Users or client applications can access blobs via URLs, the Azure Storage REST API, Azure PowerShell, the Azure CLI, or an Azure Storage client library. The storage client libraries are available for multiple languages, including .NET, Java, Node.js, Python, PHP, and Ruby.

Azure Files

[Azure Files](#) enables you to set up highly available network file shares. Shares can be accessed by using the Server Message Block (SMB) protocol and the Network File System (NFS) protocol. Multiple virtual machines can share the same files with both read and write access. You can also read the files by using the REST interface or the storage client libraries.

File shares can be used for many common scenarios:

- Many on-premises applications use file shares. This feature makes it easier to migrate those applications that share data to Azure. If you mount the file share to the same drive letter that the on-premises application uses, the part of your application that accesses the file share should work with minimal, if any, changes.
- Configuration files can be stored on a file share and accessed from multiple virtual machines. Tools and utilities used by multiple developers in a group can be stored on a file share, ensuring that everybody can find them, and that they use the same version.
- Diagnostic logs, metrics, and crash dumps are just three examples of data that can be written to a file share and processed or analyzed later.

The storage account credentials are used to provide authentication for access to the file share. All users who have the share mounted should have full read/write access to the share.

Azure Queue Storage

[Azure Queue Storage](#) is used to store and retrieve messages. Queue messages can be up to 64 KB in size, and a queue can contain millions of messages. Queues are used to store lists of messages to be processed asynchronously.

Consider a scenario where you want your customers to be able to upload pictures, and you want to create thumbnails for each picture. You could have your customer wait for you to create the thumbnails while uploading the pictures. An alternative is to use a queue. When the customer finishes the upload, you can write a message to the queue. Then you can use an Azure Function to retrieve the message from the queue and create the thumbnails. Each of the processing parts can be scaled separately, which gives you more control when tuning the configuration.

Azure Table Storage

[Azure Table storage](#) is a service that stores nonrelational structured data (also known as structured NoSQL data) in the cloud, providing a key/attribute store with a schemaless design. Because Table storage is schemaless, it's easy to adapt your data as the needs of your application evolve. Access to Table storage data is fast and cost-effective for many types of applications, and is typically lower in cost than traditional SQL for similar volumes of data. In addition to the existing Azure Table Storage service, there's a new Azure Cosmos DB Table API offering that provides throughput-optimized tables, global distribution, and automatic secondary indexes.

Things to consider when choosing Azure Storage services

As you think about your configuration plan for Azure Storage, consider the prominent features of the types of Azure Storage and which options support your application needs.

- **Consider storage optimization for massive data.** Azure Blob Storage is optimized for storing massive amounts of unstructured data. Objects in Blob Storage can be accessed from anywhere in the world via HTTP or HTTPS. Blob Storage is ideal for serving data directly to a browser, streaming data, and storing data for backup and restore.
- **Consider storage with high availability.** Azure Files supports highly available network file shares. On-premises apps use file shares for easy migration. By using Azure Files, all users can access shared data and tools. Storage account credentials provide file share authentication to ensure all users who have the file share mounted have the correct read/write access.
- **Consider storage for messages.** Use Azure Queue Storage to store large numbers of messages. Queue Storage is commonly used to create a backlog of work to process asynchronously.
- **Consider storage for structured data.** Azure Table Storage is ideal for storing structured, nonrelational data. It provides throughput-optimized tables, global distribution, and automatic secondary indexes. Because Azure Table Storage is part of Azure Cosmos DB, you have access to a fully managed NoSQL database service for modern app development.

4. Determine storage account types

<https://learn.microsoft.com/en-us/training/modules/configure-storage-accounts/4-determine-storage-account-kinds>

Determine storage account types

Completed

General purpose Azure storage accounts have two basic [types](#): Standard and Premium.



Things to know about storage account types

Standard storage accounts are backed by magnetic hard disk drives (HDD). A standard storage account provides the lowest cost per GB. You can use Standard storage for applications that require bulk storage or where data is infrequently accessed.

Premium storage accounts are backed by solid-state drives (SSD) and offer consistent low-latency performance. You can use Premium storage for Azure virtual machine disks with I/O-intensive applications like databases.

Note

You can't convert a Standard storage account to a Premium storage account or vice versa. You must create a new storage account with the desired type and copy data, if applicable, to a new storage account. All storage account types are encrypted by using Storage Service Encryption (SSE) for data at rest.

Storage account	Supported services	Redundancy options	Recommended usage
Standard general-purpose v2	Blob Storage (including Data Lake Storage), Queue Storage, Table Storage, and Azure Files	LRS, GRS, RA-GRS, ZRS, GZRS, RA-GZRS	Standard storage account for most scenarios, including blobs, file shares, queues, tables, and disks (page blobs).
Premium block blobs	Blob Storage (including Data Lake Storage)	LRS, ZRS	Premium storage account for block blobs and append blobs. Recommended for applications with high transaction rates. Use Premium block blobs if you work with smaller objects or require consistently low storage latency. This storage is designed to scale with your applications.
Premium file shares	Azure Files	LRS, ZRS	Premium storage account for file shares only. Recommended for enterprise or high-performance scale

Storage account	Supported services	Redundancy options	Recommended usage
			applications. Use Premium file shares if you require support for both Server Message Block (SMB) and NFS file shares.
Premium page blobs	Page blobs only	LRS only	Premium high-performance storage account for page blobs only. Page blobs are ideal for storing index-based and sparse data structures, such as operating systems, data disks for virtual machines, and databases.

Note

Administrators managing existing Azure subscriptions may encounter legacy storage account types such as General-purpose v1 (GPv1) and legacy BlobStorage accounts. Microsoft recommends upgrading legacy accounts to General-purpose v2 for access to all current capabilities. Upgrades are supported in-place via the Azure portal, Azure CLI, or PowerShell.

Tip

Before continuing, consider working through the [Create a storage account](#) training module.

5. Determine replication strategies

<https://learn.microsoft.com/en-us/training/modules/configure-storage-accounts/5-determine-replication-strategies>

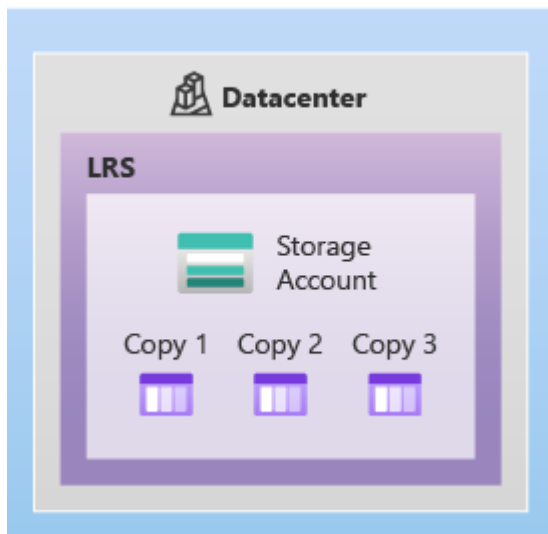
Determine replication strategies

Completed

The data in your Azure storage account is always replicated to ensure durability and high availability. [Azure Storage replication](#) copies your data to protect from planned and unplanned events. These events range from transient hardware failures, network or power outages, massive natural disasters, and so on. You can choose to replicate your data within the same data center, across zonal data centers within the same region, and even across regions. Replication ensures your storage account meets the Service-Level Agreement (SLA) for Azure Storage even if there are failures.

Locally redundant storage

Primary region

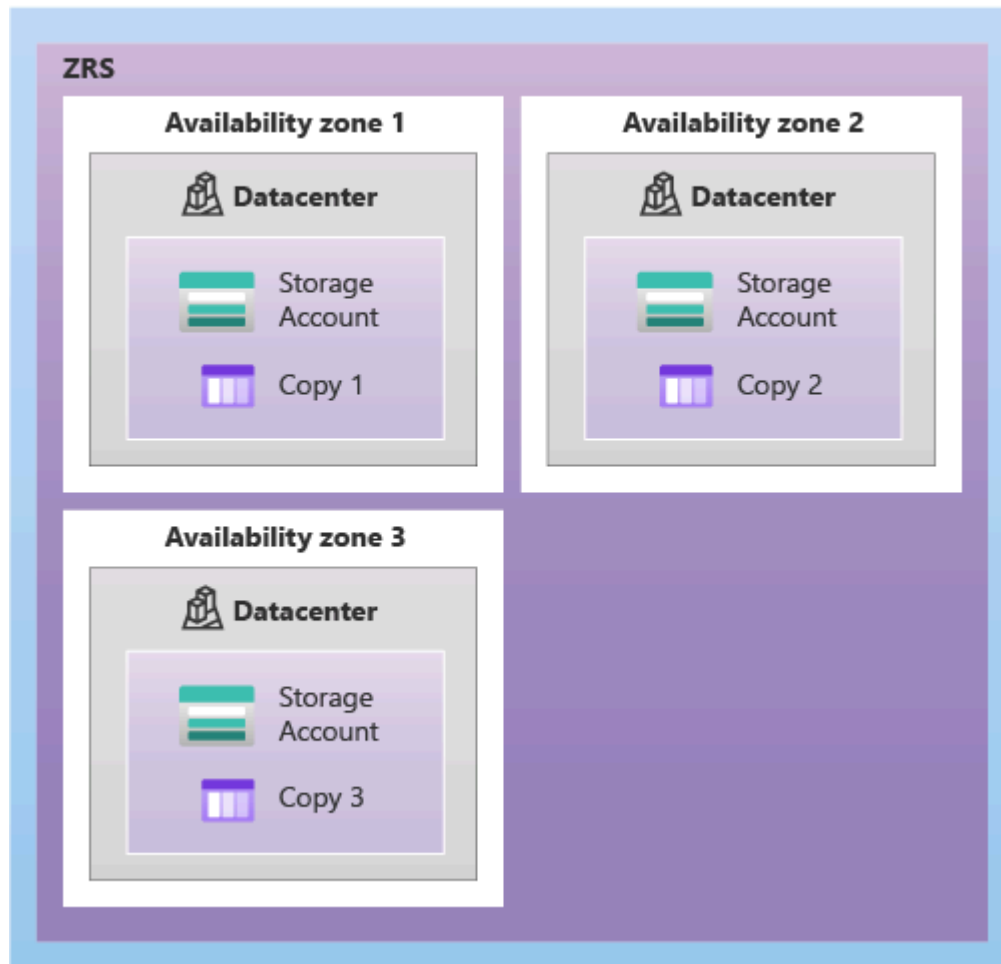


Locally redundant storage is the lowest-cost replication option and offers the least durability compared to other strategies. If a data center-level disaster occurs, such as fire or flooding, all replicas might be lost or unrecoverable. Despite its limitations, LRS can be appropriate in several scenarios:

- Your application stores data that can be easily reconstructed if data loss occurs.
- Your data is constantly changing like in a live feed, and storing the data isn't essential.
- Your application is restricted to replicating data only within a location due to data governance requirements.

Zone redundant storage

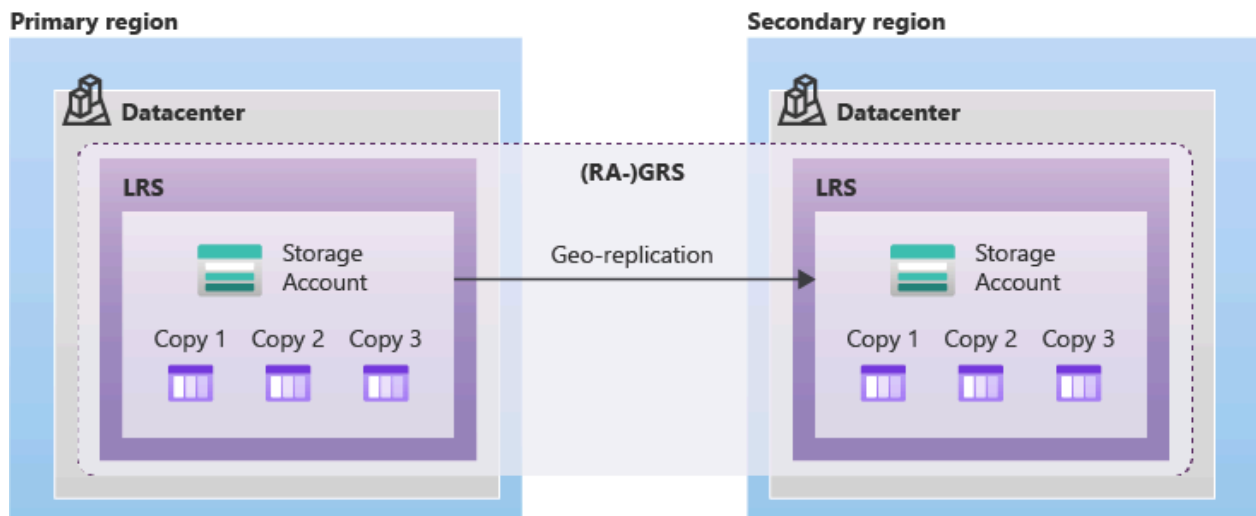
Primary region



Zone redundant storage synchronously replicates your data across three storage clusters in a single region. Each storage cluster is physically separated from the others and resides in its own availability zone. Each availability zone, and the ZRS cluster within it, is autonomous, and has separate utilities and networking capabilities. Storing your data in a ZRS account ensures you can access and manage your data if a zone becomes unavailable. ZRS provides excellent performance and low latency.

- ZRS isn't currently available in all regions.
- Changing to ZRS from another data replication option requires the physical data movement from a single storage stamp to multiple stamps within a region.

Geo-redundant storage



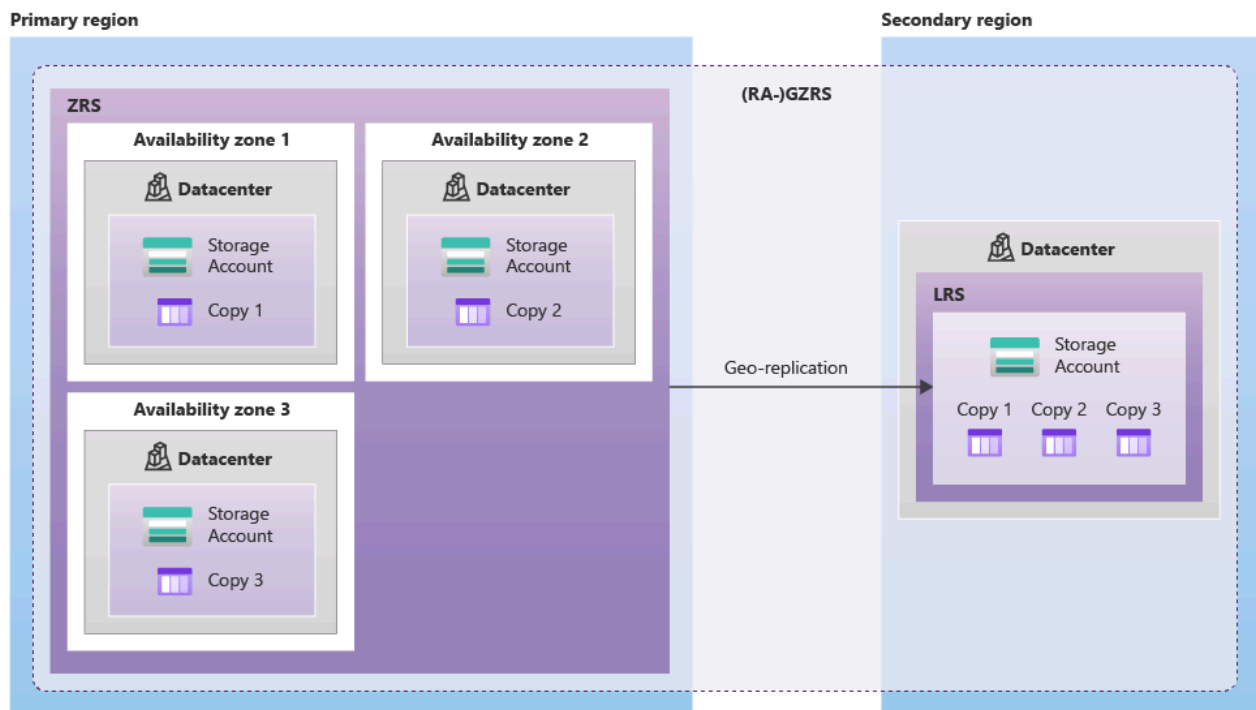
Geo-redundant storage replicates your data to a secondary region (hundreds of miles away from the primary location of the source data). GRS provides a higher level of durability even during a regional outage. GRS is designed to provide at least 99.99999999999999% **(16 9's) durability**. When your storage account has GRS enabled, your data is durable even when there's a complete regional outage or a disaster where the primary region isn't recoverable.

If you implement GRS, you have two related options to choose from:

- **GRS** replicates your data to another data center in a secondary region. The data is available to be read only if Microsoft initiates a failover from the primary to secondary region.
- **Read-access geo-redundant storage (RA-GRS)** is based on GRS. RA-GRS replicates your data to another data center in a secondary region, and also provides you with the option to read from the secondary region. With RA-GRS, you can read from the secondary region regardless of whether Microsoft initiates a failover from the primary to the secondary.

For a storage account with GRS or RA-GRS enabled, all data is first replicated with locally redundant storage. An update is first committed to the primary location and replicated by using LRS. The update is then replicated asynchronously to the secondary region by using GRS. Data in the secondary region uses LRS. Both the primary and secondary regions manage replicas across separate fault domains and upgrade domains within a storage scale unit. The storage scale unit is the basic replication unit within the datacenter. Replication at this level is provided by LRS.

Geo-zone redundant storage



Geo-zone-redundant storage combines the high availability of zone-redundant storage with protection from regional outages as provided by geo-redundant storage. Data in a GZRS storage account is replicated across three Azure availability zones in the primary region, and also replicated to a secondary geographic region for protection from regional disasters. Each Azure region is paired with another region within the same geography, together making a regional pair.

With a GZRS storage account, you can continue to read and write data if an availability zone becomes unavailable or is unrecoverable. Additionally, your data is also durable during a complete regional outage or during a disaster in which the primary region isn't recoverable. GZRS is designed to provide at least 99.99999999999999% (16 9's) durability of objects over a given year. GZRS also offers the same scalability targets as LRS, ZRS, GRS, or RA-GRS. You can optionally enable read access to data in the secondary region with read-access geo-zone-redundant storage (RA-GZRS).

Tip

Microsoft recommends using GZRS for applications that require consistency, durability, high availability, excellent performance, and resilience for disaster recovery. Enable RA-GZRS for read access to a secondary region when there's a regional disaster.

Things to consider when choosing replication strategies

Let's examine the scope of durability and availability for the different replication strategies. The following table describes several key factors during the replication process, including node unavailability within a data center, and whether the entire data center (zonal or nonzonal) becomes unavailable. The table identifies read access to data in a remote, geo-replicated region during region-wide unavailability, and the supported Azure storage account types.

Node in data center unavailable	Entire data center unavailable	Region-wide outage	Read access during region-wide outage
- LRS - ZRS - GRS - RA-GRS - GZRS - RA-GZRS	- ZRS - GRS - RA-GRS - GZRS - RA-GZRS	- GRS - RA-GRS - GZRS - RA-GZRS	- RA-GRS - RA-GZRS

6. Access storage

<https://learn.microsoft.com/en-us/training/modules/configure-storage-accounts/6-access-storage>

Access storage

Completed

Every object you store in Azure Storage has a unique URL address. Your storage account name forms the *subdomain* portion of the URL address. The combination of the subdomain and the domain name, which is specific to each service, forms an endpoint for your storage account.

Let's look at an example. If your storage account name is *mystorageaccount*, default endpoints for your storage account are formed for the Azure services as shown in the following table:

Service	Default endpoint
Container service	<code>// mystorageaccount .blob.core.windows.net</code>
Table service	<code>// mystorageaccount .table.core.windows.net</code>
Queue service	<code>// mystorageaccount .queue.core.windows.net</code>
File service	<code>// mystorageaccount .file.core.windows.net</code>

We create the URL to access an object in your storage account by appending the object's location in the storage account to the endpoint.

For example, to access the *myblob* data in the *mycontainer* location in your storage account, we use the following URL address:

```
// mystorageaccount .blob.core.windows.net/ mycontainer / myblob .
```

Configure custom domains

You can configure a [custom domain](#) to access blob data in your Azure storage account. As we reviewed, the default endpoint for Azure Blob Storage is `\<storage-account-name>.blob.core.windows.net`. If you map a custom domain and subdomain, such as `www.contoso.com`, to the blob or web endpoint for your storage account, your users can use that domain to access blob data in your storage account.

Direct mapping lets you enable a custom domain for a subdomain to an Azure storage account. For this approach, you create a `CNAME` record that points from the subdomain to the Azure storage account.

The following example shows how a subdomain is mapped to an Azure storage account to create a `CNAME` record in the domain name system (DNS):

- Subdomain: `blobs.contoso.com`
- Azure storage account: `\<storage account>.blob.core.windows.net`
- Direct `CNAME` record: `contosoblobs.blob.core.windows.net`

7. Secure storage endpoints

<https://learn.microsoft.com/en-us/training/modules/configure-storage-accounts/7-secure-storage-endpoints>

Secure storage endpoints

Completed

In the Azure portal, each Azure service requires certain steps to configure the service endpoints and restrict network access.

To access these settings for your storage account, you use the **Firewalls and virtual networks** settings. You add the virtual networks that should have access to the service for the account. - This setting restricts access to your storage account from specific subnets on virtual networks or public IPs.

Security + networking

Networking

Azure CDN

Access keys

Firewalls and virtual networks

Custom domain

Save Discard Refresh

Public network access

☒ Enabled from all networks

☐ Enabled from selected virtual networks and IP addresses

☐ Disabled

i All networks, including the internet, can access this storage account. [Learn more](#)

The service endpoints for a storage account provide the base URL for any blob, queue, table, or file object in Azure Storage. Use this base URL to construct the address for any given resource.

Search (Ctrl+ /)

<<

Refresh

Settings

Configuration

Data Lake Gen2 upgrade

Resource sharing (CORS)

Advisor recommendations

Endpoints

Locks

Monitoring

Insights

Alerts

Metrics

Workbooks

Diagnostic settings

Logs

Provisioning state Succeeded

Created 5/5/2022, 8:27:25 PM

Storage account resource ID /subscriptions/ <subscription-id> ..

Blob service

Resource ID /subscriptions/ <subscription-id> ..

Blob service https://storagesamplesdnszone4.z10.blob.storage.a:

File service

Resource ID /subscriptions/ <subscription-id> ..

File service https://storagesamplesdnszone4.z10.file.storage.azl:

Queue service

Resource ID /subscriptions/ <subscription-id> ..

Queue service https://storagesamplesdnszone4.z10.queue.storage:

Things to know about configuring service endpoints

Here are some points to consider about configuring service access settings:

- You can configure the service to allow access to one or more public IP ranges.
- Subnets and virtual networks must exist in the same Azure region or region pair as your storage account.

Important

Be sure to test the service endpoint and verify the endpoint limits access as expected.

Things to know about configuring private endpoints

In addition to service endpoints, Azure Storage supports private endpoints for enhanced security and network isolation. Private endpoints are the recommended approach for production workloads requiring secure access.

A private endpoint uses a private IP address from your virtual network to bring the Azure Storage service into your VNet. All traffic between your VNet and the storage service goes over the Microsoft backbone network, eliminating exposure to the public internet.

Key differences from service endpoints

- Private endpoints assign a private IP from your VNet to the storage account, keeping all traffic within the Microsoft backbone. Use private endpoints for production workloads requiring complete network isolation and compliance requirements
- Service endpoints keep the storage account on its public endpoint but restrict access to specific VNets and subnets. Use service endpoints for development scenarios or when you need simpler configuration with some public internet access

Tip

Learn more with the [Secure and isolate access to Azure resources by using network security groups and service endpoints](#) training module. This module has a sandbox where you can restrict access to Azure Storage by using service endpoints.

8. Module assessment

<https://learn.microsoft.com/en-us/training/modules/configure-storage-accounts/8-knowledge-check>

Module assessment

Completed

9. Summary and resources

<https://learn.microsoft.com/en-us/training/modules/configure-storage-accounts/9-summary-resources>

Summary and resources

Completed

In this module, you learned about Azure Storage and how to create a storage account.

The main takeaways from this module are:

- Azure Storage provides a range of storage options for different types of data, including virtual machine data, unstructured data, and structured data.
- There are different types of storage accounts available, each with its own features and pricing model. It's important to consider the specific requirements of your application when choosing the right storage account type.
- Azure Storage offers four data services: Azure Blob Storage, Azure Files, Azure Queue Storage, and Azure Table Storage. Each service is optimized for different types of data and has its own use cases and benefits.
- Replication is an important consideration for ensuring data durability and high availability. Azure Storage offers different replication strategies to choose from based on your requirements.
- Configuring custom domains and secure endpoints allow you to access and secure your storage account in Azure.

Learn more with Copilot

Copilot can assist you in configuring Azure infrastructure solutions. Copilot can compare, recommend, explain, and research products and services where you need more information. Open a Microsoft Edge browser and choose Copilot (top right) or navigate to copilot.microsoft.com. Take a few minutes to try these prompts and extend your learning with Copilot.

- What is an Azure storage account? What type of storage accounts are available?
- Explain for a nontechnical person Azure data redundancy for storage accounts.

Learn more with Azure documentation

- [Storage account overview](#). This article is your starting point for learning about Azure storage accounts.
- [Azure storage redundancy](#). This article reviews how to tradeoff cost and availability when selecting a redundancy option.

Learn more with self-paced training

- [Create an Azure storage account](#). Learn how to create an Azure Storage account with the correct options for your business needs.
- [Design and implement private access to Azure Services](#). Learn how to implement private access to Azure Services with Azure Private Link, and virtual network service endpoints.

Configure Azure Blob Storage

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/configure-blob-storage/1-introduction>

Introduction

Completed

Azure Blob Storage is an AI-ready service for storing large amounts of unstructured object data. Unstructured data is data that doesn't adhere to a particular data model or definition, such as text or binary data.

In this module, your media company has an extensive library of video clips that are accessed thousands of times a day. The company relies on you to configure Blob Storage for the video data. You plan to use access tiers to reduce cost and improve performance. You're developing a lifecycle management strategy for the older videos. Your plan also includes configuring object replication for failover.

Learning objectives

In this module, you will:

- Understand the purpose and benefits of Azure Blob Storage.
- Create and configure Azure Blob Storage accounts.
- Manage containers and blobs within Azure Blob Storage.
- Optimize blob storage performance and scalability.
- Implement lifecycle management policies to automate data movement and deletion.
- Determine the best pricing plans for your Azure Blob Storage.

Skills measured

The content in the module helps you prepare for [Exam AZ-104: Microsoft Azure Administrator](#).

Prerequisites

Here are some common prerequisites that can be beneficial for understanding and successfully completing this module.

- Basic understanding of cloud computing: Familiarity with cloud computing concepts, such as virtualization, scalability, and pay-as-you-go pricing models, can provide a foundation for understanding how Azure Blob Storage fits into the broader cloud ecosystem.
- Knowledge of Azure fundamentals: Having a basic understanding of Microsoft Azure services and concepts, such as Azure Resource Manager, Azure Storage Accounts, and Azure Virtual Networks, can help you navigate and configure blob storage effectively.
- Understand fundamental storage concepts like file systems, directories, files, and data replication can be beneficial when working with blob storage.
- Experience with Azure portal or Azure CLI: Familiarity with the Azure portal (web-based management interface) or Azure CLI (command-line interface) can help you navigate and configure blob storage resources efficiently.
- Basic programming or scripting skills: While not always required, having some knowledge of programming or scripting languages like PowerShell or Python can be advantageous when automating blob storage configuration tasks.

2. Implement Azure Blob Storage

<https://learn.microsoft.com/en-us/training/modules/configure-blob-storage/2-implement>

Implement Azure Blob Storage

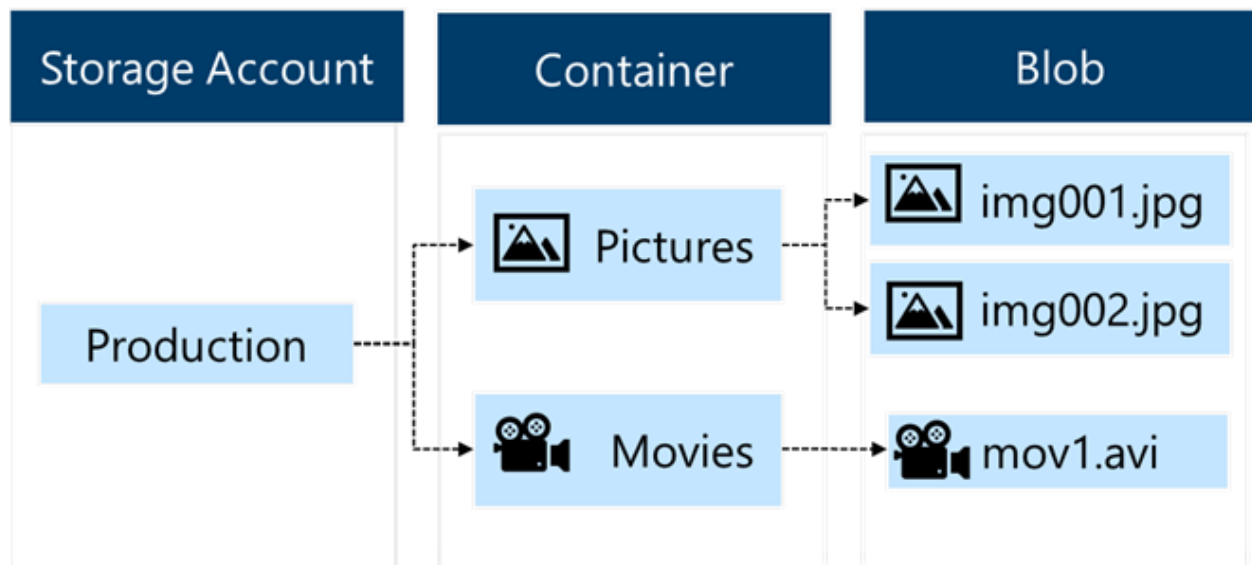
Completed

[Azure Blob Storage](#) is a service that stores unstructured data in the cloud as objects or blobs. Blob stands for Binary Large Object. Blob Storage is also referred to as *object storage* or *container storage*.



Things to know about Azure Blob Storage

Let's examine some configuration characteristics of Blob Storage.



- Blob Storage can store any type of text or binary data. Some examples are text documents, images, video files, and application installers.
- Blob Storage uses three resources to store and manage your data:
 - An Azure storage account
 - Containers in an Azure storage account
 - Blobs in a container
- To implement Blob Storage, you configure several settings:
 - Blob container options.
 - Blob types and upload options.
 - Blob Storage access tiers.
 - Blob lifecycle rules.
 - Blob object replication options.

Things to consider when implementing Azure Blob Storage

There are many common uses for Blob Storage. Consider the following scenarios and think about your own data needs:

- **Consider browser uploads.** Use Blob Storage to serve images or documents directly to a browser.
- **Consider distributed access.** Blob Storage can store files for distributed access, such as during an installation process.
- **Consider streaming data.** Stream video and audio by using Blob Storage.

- **Consider archiving and recovery.** Blob Storage is a great solution for storing data for backup and restore, disaster recovery, and archiving.
- **Consider application access.** You can store data in Blob Storage for analysis by an on-premises or Azure-hosted service.

3. Create blob containers

<https://learn.microsoft.com/en-us/training/modules/configure-blob-storage/3-create-blob-containers>

Create blob containers

Completed

Azure Blob Storage uses a container resource to group a set of blobs. A blob can't exist by itself in Blob Storage. A blob must be stored in a container resource.

Things to know about containers and blobs

Let's look at the configuration characteristics of containers and blobs.

- All blobs must be in a container.
- Containers organize your blob storage.
- A container can store an unlimited number of blobs.
- An Azure storage account can contain an unlimited number of containers.
- You must create a storage container before you can begin to upload data.

Configure a container

In the Azure portal, you configure settings to create a container for an Azure storage account. As you review these details, consider how you might organize containers in your storage account.

The image shows the 'New container' dialog box in the Azure portal. It has a title bar with a plus icon and the text 'Container' and 'Change access level'. Below the title bar, the text 'New container' is displayed. There are two main input fields: 'Name' with a red asterisk and 'Public access level' with an information icon. The 'Name' field contains the text 'container01' and has a green checkmark on the right. The 'Public access level' field is a dropdown menu currently showing 'Private (no anonymous access)'. A blue arrow points from this dropdown to a separate window showing the full list of options: 'Private (no anonymous access)', 'Blob (anonymous read access for blobs only)', and 'Container (anonymous read access for containers and blobs)'. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

- **Name:** Enter a name for your container. The name must be unique within the Azure storage account.
 - The name can contain only lowercase letters, numbers, and hyphens.
 - The name must begin with a letter or a number.
 - The minimum length for the name is three characters.
 - The maximum length for the name is 63 characters.
- **Public access level:** The access level specifies whether the container and its blobs can be accessed publicly. By default, container data is private and visible only to the account owner. There are three access level choices:
 - **Private:** (Default) Prohibit anonymous access to the container and blobs.
 - **Blob:** Allow anonymous public read access for the blobs only.
 - **Container:** Allow anonymous public read and list access to the entire container, including the blobs.

Important

The Blob and Container access levels have no effect unless the storage account's **Allow Blob anonymous access** setting is enabled. When disabled, all containers remain private regardless of their individual access level settings. Microsoft recommends keeping anonymous access disabled at the account level unless serving public content scenarios.

4. Assign blob access tiers

Assign blob access tiers

Completed

Azure Storage supports several [access tiers](#) for blob data. These tiers include Hot, Cool, Cold, and Archive. Each access tier is optimized to support a particular pattern of data usage.

Things to know about blob access tiers

Let's examine characteristics of the blob access tiers.

Hot tier

The Hot tier is optimized for frequent reads and writes of objects in the Azure storage account. A good usage case is data that is actively being processed. The hot tier has the highest storage costs, but the lowest access costs.

Cool tier

The Cool tier is optimized for storing large amounts of infrequently accessed data. This tier is intended for data that remains in the Cool tier for at least 30 days. A usage case for the Cool tier is short-term backup and disaster recovery datasets and older media content. This content shouldn't be viewed frequently, but it needs to be immediately available. Storing data in the Cool tier is more cost-effective. The cool tier has lower storage costs and higher access costs compared to the hot tier.

Cold tier

The Cold tier is also optimized for storing large amounts of infrequently accessed data. This tier is intended for data that can remain in the tier for at least 90 days. The cold tier has lower storage costs and higher access costs compared to the cool tier.

Archive tier

The Archive tier is an offline tier that's optimized for data that can tolerate several hours of retrieval latency. Data must remain in the Archive tier for at least 180 days or be subject to an early deletion charge. Data for the Archive tier includes secondary backups, original raw data, and legally required compliance information. This tier is the most cost-effective option for storing data. Accessing data is more expensive in the Archive tier than accessing data in the other tiers.

To access the blob's content, you can rehydrate it to the hot, cool, or cold tier using two methods: **Copy Blob** (recommended - creates a new blob in an online tier) or **Set Blob Tier** (changes tier in place). Both methods support Standard priority (up to 15 hours) or High priority (within 1 hour for objects under 10 GB, at higher cost). Use High priority for urgent data retrieval in disaster recovery scenarios.

Compare access tiers

The access options for Azure Blob Storage offer a range of features and support levels to help you optimize your storage costs. As you compare the features and support, think about which access options can best support your application needs.

Comparison	Hot access tier	Cool access tier	Cold access tier	Archive access tier
Availability	99.9%	99%	99%	99%
Availability (RA-GRS reads)	99.99%	99.9%	99.9%	99.9%
Latency (time to first byte)	milliseconds	milliseconds	milliseconds	hours
Minimum storage duration	N/A	30 days	90 days	180 days

5. Add blob lifecycle management rules

<https://learn.microsoft.com/en-us/training/modules/configure-blob-storage/5-add-blob-lifecycle-management-rules>

Add blob lifecycle management rules

Completed

Every data set has a unique lifecycle. Early in the lifecycle, users tend to access some of the data in the set, but not all of the data. As the data set ages, access to all of the data in the set tends to dramatically reduce. Some data set stays idle in the cloud and is rarely accessed. Some data expires within a few days or months after creation. Other data is actively read and modified throughout the data set lifetime.

Azure Blob Storage supports [lifecycle management](#) for data sets. It offers a rich rule-based policy for GPv2 accounts and Premium block blob accounts. Legacy Blob Storage accounts are also supported, but GPv2 is recommended for new deployments. You can use lifecycle policy rules to transition your data to the appropriate access tiers, and set expiration times for the end of a data set's lifecycle.



Things to know about lifecycle management

You can use Azure Blob Storage lifecycle management policy rules to accomplish several tasks.

- Transition blobs to a cooler storage tier (Hot to Cool, Hot to Cold, Hot to Archive, Cool to Cold, Cool to Archive, Cold to Archive) to optimize for performance and cost.
- Delete current versions of a blob, previous versions of a blob, or blob snapshots at the end of their lifecycles.
- Automatically transition blobs from Cool back to Hot when accessed. This setting optimizes for unpredictable access patterns without early deletion charges.
- Apply rules to an entire storage account, to select containers, or to a subset of blobs using name prefixes or blob index tags as filters.

Business scenario

Consider a scenario where data is frequently accessed in the early stages of the lifecycle, but only occasionally after two weeks. After the first month, the data set is rarely accessed. In this scenario, the Hot tier of Blob Storage is best during the early stages. Cool tier storage is most appropriate for occasional access. Archive tier storage is the best option after the data ages over a month. To achieve this transition, lifecycle management policy rules are available to move aging data to cooler tiers.

Configure lifecycle management policy rules

In the Azure portal, you create lifecycle management policy rules for your Azure storage account by specifying several settings. For each rule, you create **If - Then** block conditions to transition or expire data based on your specifications. As you review these details, consider how you can set up lifecycle management policy rules for your data sets.

Add a rule ...

✓ Details 2 Base blobs 3 Filter set

Lifecycle management uses your rules to automatically move blobs to cooler tiers or to delete them. If you create multiple rules, the associated actions must be implemented in tier order (from hot to cool storage, then archive, then deletion).

If

Base blobs were *

☒ Last modified

☐ Created

More than (days ago) *

Enter a value

↓

Then

Delete the blob

Move to cool storage
For infrequently accessed data that you want to keep on cool storage for at least 30 days.

Move to cold storage
For rarely accessed data that you want to keep for at least 90 days.

Move to archive storage
Use if you don't need online access and want to keep the object for 180 days or longer.

Delete the blob
Deletes the object per the specified conditions.

- **If:** The **If** clause sets the evaluation clause for the policy rule. When the **If** clause evaluates to true, the **Then** clause is executed. Use the **If** clause to set the time period to apply to the blob data. The lifecycle management feature checks if the data is accessed or modified according to the specified time.
 - **More than (days ago):** The number of days to use in the evaluation condition.
- **Then:** The **Then** clause sets the action clause for the policy rule. When the **If** clause evaluates to true, the **Then** clause is executed. Use the **Then** clause to set the transition action for the blob data. The lifecycle management feature transitions the data based on the setting.
 - **Move to cool storage:** The blob data is transitioned to Cool tier storage.
 - **Move to cold storage:** The blob data is transitioned to Cold tier storage.
 - **Move to archive storage:** The blob data is transitioned to Archive tier storage.
 - **Delete the blob:** The blob data is deleted.

By designing policy rules to adjust storage tiers in respect to the age of data, you can design the least expensive storage options for your needs.

Tip

Expand your knowledge in the [Manage the Azure Blob storage lifecycle](#) training module.

6. Determine blob object replication

<https://learn.microsoft.com/en-us/training/modules/configure-blob-storage/6-determine-blob-object-replication>

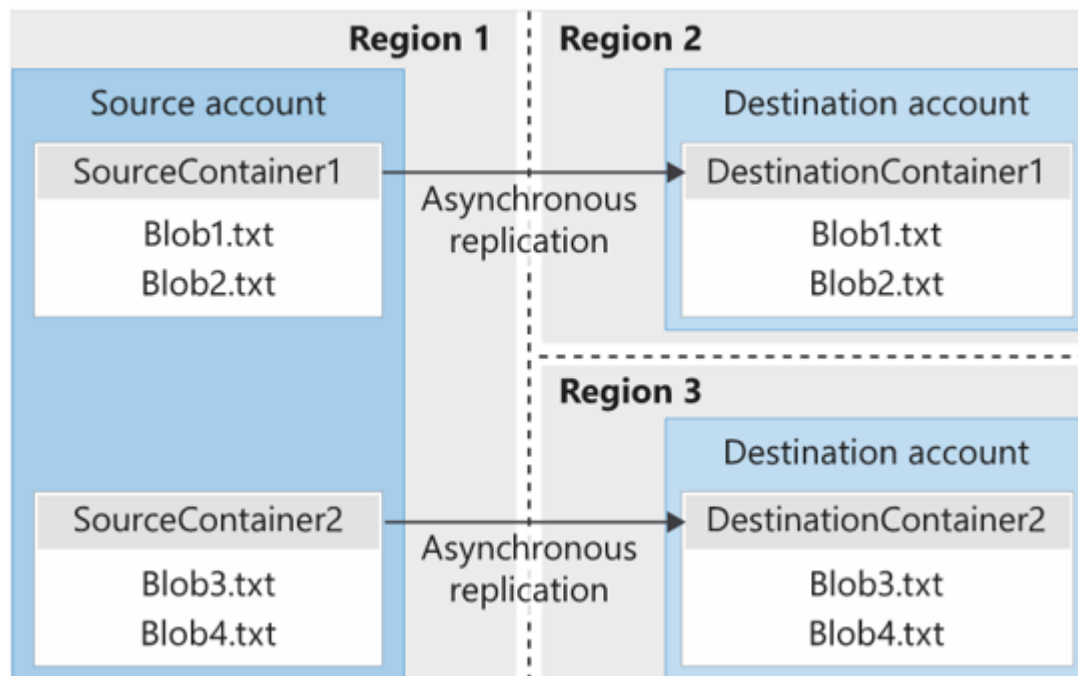
Determine blob object replication

Completed

[Object replication](#) copies blobs in a container asynchronously according to policy rules that you configure.



Replication includes the blob content, metadata properties, and versions. The following illustration shows an example of asynchronous replication of blob containers between regions.



Things to know about blob object replication

There are several considerations to keep in mind when planning your configuration for blob object replication.

- Object replication requires that [Blob versioning](#) is enabled on both the source and destination accounts. When blob versioning is enabled, you can access earlier versions of a blob. This access lets you recover your modified or deleted data.
- Object replication doesn't support blob snapshots. Any snapshots on a blob in the source account aren't replicated to the destination account.
- Object replication is supported when the source and destination accounts are in the Hot, Cool, or Cold tier. The source and destination accounts can be in different tiers.
- When you configure object replication, you create a replication policy that specifies the source Azure storage account and the destination storage account.
- A replication policy includes one or more rules that specify a source container and a destination container. The policy identifies the blobs in the source container to replicate.

Things to consider when configuring blob object replication

There are many benefits to using blob object replication. Consider the following scenarios and think about how replication can be a part of your Blob Storage strategy.

- **Consider latency reductions.** Minimize latency with blob object replication. You can reduce latency for read requests by enabling clients to consume data from a region that's in closer physical proximity.

- **Consider efficiency for compute workloads.** Improve efficiency for compute workloads by using blob object replication. With object replication, compute workloads can process the same sets of blobs in different regions.
- **Consider data distribution.** Optimize your configuration for data distribution. You can process or analyze data in a single location and then replicate only the results to other regions.
- **Consider costs benefits.** Manage your configuration and optimize your storage policies. After your data is replicated, you can reduce costs by moving the data to the Archive tier by using lifecycle management policies.

7. Manage blobs

<https://learn.microsoft.com/en-us/training/modules/configure-blob-storage/7-upload-blobs>

Manage blobs

Completed

A blob can be any type of data and any size file. Azure Storage offers three types of blobs: *block blob*, *page blob*, and *append blob*.

Things to know about blob types

Let's take a closer look at the characteristics of blob types.

- **Block blobs.** A block blob consists of blocks of data that are assembled to make a blob. Most Blob Storage scenarios use block blobs. Block blobs are ideal for storing text and binary data in the cloud, like files, images, and videos. The block blob type is the default type for a new blob. When you're creating a new blob, if you don't choose a specific type, the new blob is created as a block blob.
- **Append blobs.** An append blob is similar to a block blob because the append blob also consists of blocks of data. The blocks of data in an append blob are optimized for *append* operations. Append blobs are useful for logging scenarios, where the amount of data can increase as the logging operation continues.
- **Page blobs.** A page blob can be up to 8 TB in size. Page blobs are more efficient for frequent read/write operations. Azure Virtual Machines uses page blobs for operating system disks and data disks.

Note

After you create a blob, you can't change its type.

Things to consider when managing blob storage

You can use the portal to upload and manage blobs. This option is good for a few files. After you identify the files to upload, you choose the blob type and block size, and the container folder. You also set the access tier and the encryption scope.

Upload blob

☐ Overwrite if files already exist

^ Advanced

Blob type ⓘ

Block blob^

Block blobPage blobAppend blob

☒ Upload .vhd files as page blobs (recommended)

Block size ⓘ

4 MiBv

Access tier ⓘ

Hot (Inferred)v

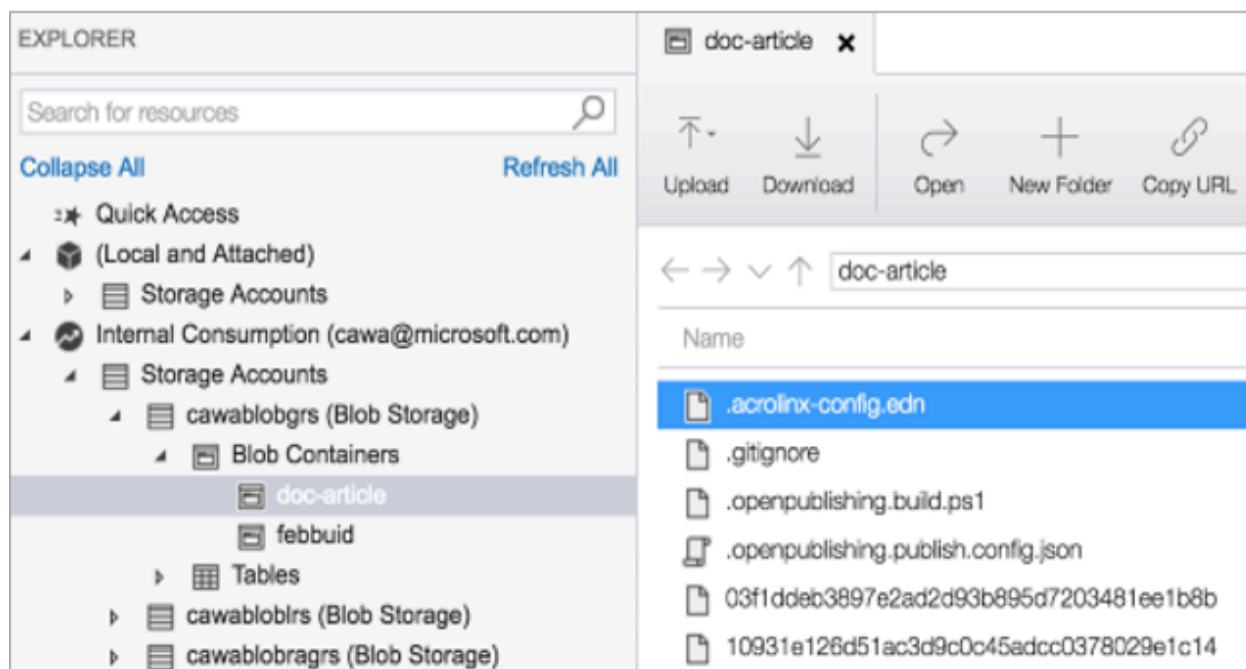
Upload to folder

Encryption scope

☒ Use existing default container scope☐ Choose an existing scope

For larger numbers of files, it's best to use a tool. Review the following options and consider which tools would suit your configuration needs.

- **Azure Storage Explorer.** Upload, download, and manage blobs, files, queues, and tables, as well as Azure Data Lake Storage entities and managed disks. You can also view, edit, and manage resources, preview data, and configure storage permissions and access controls.



- **AzCopy**. An easy-to-use command-line tool for Windows and Linux. You can copy data to and from Blob Storage, across containers, and across storage accounts.
- **Azure Data Box Disk**. A service for transferring on-premises data to Blob Storage when large datasets or network constraints make uploading data over the wire unrealistic. You can use Azure Data Box Disk to request solid-state disks (SSDs) from Microsoft. You can copy your data to those disks and ship them back to Microsoft to be uploaded into Blob Storage.

8. Determine Blob Storage pricing

<https://learn.microsoft.com/en-us/training/modules/configure-blob-storage/8-determine-storage-pricing>

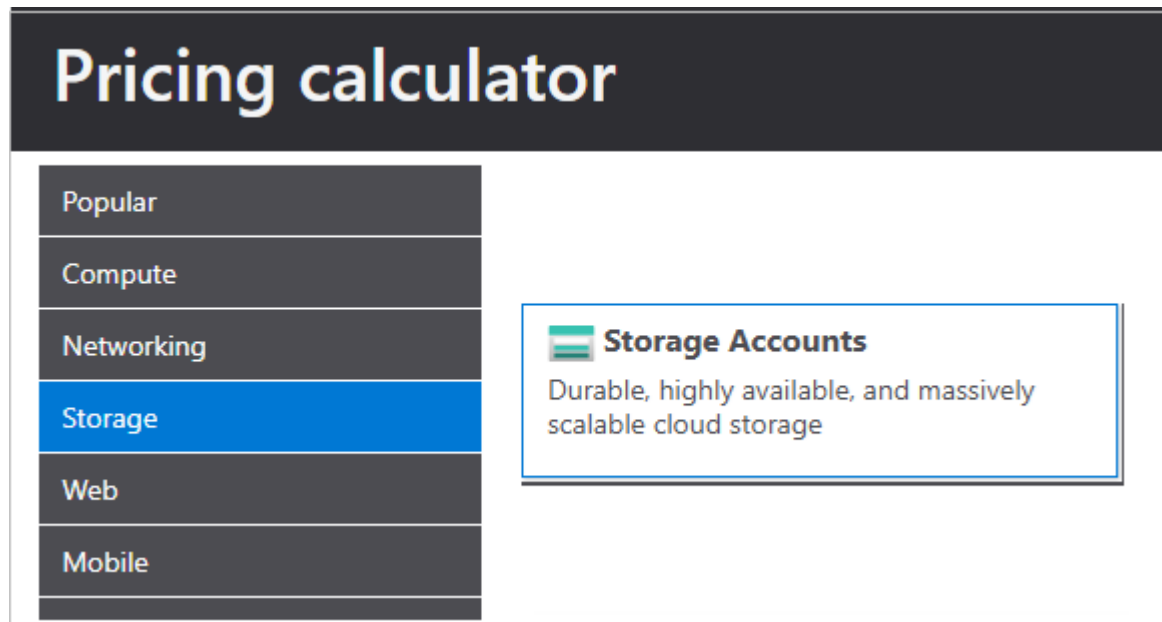
Determine Blob Storage pricing

Completed

Understanding your access patterns and correlating them with your durability and availability needs helps you to best manage your Azure Blob Storage costs. The primary tool for estimating these costs is the Azure pricing calculator. The pricing tool can calculate migration, monthly estimates, and future pricing estimates based on the workload-driven input that you specify. In general, the cost of block blob storage depends on:

- Volume of data stored per month.
- Quantity and types of operations performed, along with any data transfer costs.
- Data redundancy option selected.

You can use the Azure Pricing Calculator to estimate your storage costs.



Things to know about pricing for Blob Storage

Review the following billing considerations for an Azure storage account and Blob Storage.

- **Performance tiers.** The Blob Storage tier determines the amount of data stored and the cost for storing that data. As the performance tier gets cooler, the per-gigabyte cost decreases.
- **Data access costs.** Data access charges increase as the tier gets cooler. For data in the Cool, Cold, and Archive tiers, you're billed a per-gigabyte data access charge for read actions.
- **Transaction costs.** There's a per-transaction charge for all tiers. The charge increases as the tier gets cooler.
- **Geo-replication data transfer costs.** This charge only applies to accounts that have geo-replication configured. Geo-replication data transfer incurs a per-gigabyte charge.
- **Outbound data transfer costs.** Outbound data transfers incur billing for bandwidth usage on a per-gigabyte basis. This billing is consistent with general-purpose Azure storage accounts.
- **Changes to the storage tier.** If you change the account storage tier from Cool to Hot, you incur a charge equal to reading all the data existing in the storage account. Changing the account storage tier from Hot to Cool incurs a charge equal to writing all the data into the Cool tier (GPv2 accounts only).

9. Exercise - Provide storage for the public website

Exercise - Provide storage for the public website

Completed

Exercise scenario

The company website supplies product images, videos, marketing literature, and customer success stories. Customers are located worldwide and demand is rapidly expanding. The content is mission-critical and requires low latency load times. It's important to keep track of the document versions and to quickly restore documents if they're deleted.

Job skills

- Create a storage account with high availability.
- Ensure the storage account has anonymous public access.
- Create a blob storage container for the website documents.
- Enable soft delete so files can be easily restored.
- Enable blob versioning.

Note

Estimated time: 30 minutes. To complete this exercise, you need an [Azure subscription](#).

Launch the exercise, and follow the instructions. When finished, be sure to return to this page so you can continue learning.

[Launch Exercise](#)

10. Module assessment

Module assessment

Completed

11. Summary and resources

<https://learn.microsoft.com/en-us/training/modules/configure-blob-storage/11-summary-resources>

Summary and resources

Completed

In this module, you learned about Azure Blob Storage and how to configure it. You discovered that Blob Storage is Microsoft's object storage solution for the cloud. You learned Azure blob storage is optimized for storing massive amounts of unstructured data like text or binary files. You explored the features of Blob Storage and its use cases. You also learned how to configure Blob Storage, including choosing the appropriate access tiers to reduce cost and improve performance. And, you learned about creating a lifecycle management strategy, and configuring object replication for failover.

The main takeaways from this module are:

- Azure Blob Storage is a powerful solution for storing unstructured data in the cloud, such as text documents, images, and videos.
- Blob Storage offers different access tiers (Hot, Cool, Cold, and Archive) to optimize performance and cost based on the usage patterns of your data.
- You can configure lifecycle management policies to automatically transition data between access tiers and set expiration times for data.
- Object replication allows you to asynchronously copy blobs between containers in different regions, providing redundancy and reducing latency for read requests.

Learn more with Copilot

Copilot can assist you in configuring Azure infrastructure solutions. Copilot can compare, recommend, explain, and research products and services where you need more information. Open a Microsoft Edge browser and choose Copilot (top right) or navigate to copilot.microsoft.com. Take a few minutes to try these prompts and extend your learning with Copilot.

- What are common administration tasks associated with Azure blob storage?
- How is Azure blob storage priced?

Learn more with Azure documentation

- [Azure Blob Storage documentation](#) - Microsoft Azure's official documentation provides comprehensive information on configuring and managing blob storage. You can find detailed guides, tutorials, and examples to help you navigate through different aspects of blob storage configuration.
- [Azure Blob Storage Concepts](#) - This article provides an overview of the key concepts related to Azure Blob Storage, including storage accounts, containers, and blobs. It explains how to create and manage these entities and covers various configuration options.
- [Azure Blob Storage Security](#) - Understanding the security aspects of blob storage is crucial for proper configuration. This article explores authentication, authorization, and encryption options available in Azure Blob Storage. It also covers best practices for securing your blob storage resources.
- [Azure Blob Storage Performance and Scalability](#) - This article delves into performance considerations when configuring blob storage. The module covers the storage account type, and optimizing data transfer.
- [Azure Blob Storage Lifecycle Management](#) - Blob storage lifecycle management allows you to automate the movement and deletion of data based on predefined rules. This article explains how to configure and manage lifecycle policies to optimize storage costs and improve data management.

Configure Azure Storage security

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/configure-storage-security/1-introduction>

Introduction

Completed

Azure Storage provides a comprehensive set of security capabilities that work together to enable developers to build secure applications.

In this module, your company is storing sensitive data in Azure Storage, including personal information. The data is used internally and by external application developers. You're responsible for ensuring the data is secure for all users. You're tasked with providing configuration solutions to grant secure access to the information.

Learning objectives

In this module, you learn how to:

- Configure a shared access signature, including the uniform resource identifier (URI) and SAS parameters.
- Configure Azure Storage encryption.
- Implement customer-managed keys.
- Recommend opportunities to improve Azure Storage security.

Skills measured

The content in the module helps you prepare for [Exam AZ-104: Microsoft Azure Administrator](#).

Prerequisites

- Familiarity with navigating the Azure portal.

2. Review Azure Storage security strategies

<https://learn.microsoft.com/en-us/training/modules/configure-storage-security/2-review-strategies>

Review Azure Storage security strategies

Completed

Administrators use different strategies to ensure their data is secure. Common approaches include encryption, authentication, authorization, and user access control with credentials, file permissions, and private signatures. Azure Storage offers a suite of security capabilities based on common strategies to help you secure your data.

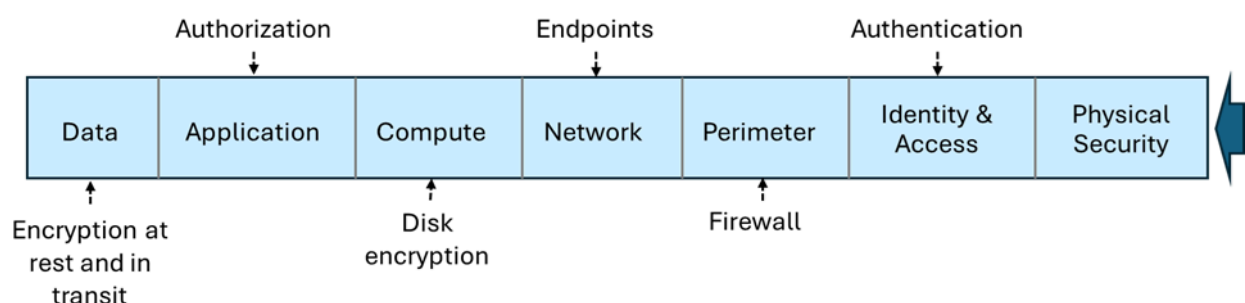


Note

The video refers to Active Directory, which is now branded as Microsoft Entra ID.

Things to know about Azure Storage security strategies

Let's look at some characteristics of Azure Storage security. As you go through this module, consider defense in depth. How can you apply storage security features to this concept?



- **Encryption at rest.** Storage Service Encryption (SSE) with a 256-bit Advanced Encryption Standard (AES) cipher encrypts all data written to Azure Storage. When you read data from

Azure Storage, Azure Storage decrypts the data before returning it. This process incurs no extra charges and doesn't degrade performance. Encryption at rest includes encrypting virtual hard disks (VHDs) with Azure Disk Encryption. This encryption uses BitLocker for Windows images, and uses dm-crypt for Linux.

- **Encryption in transit.** You can configure your storage account to only accept requests from secure connections by setting the **Secure transfer required** property for the storage account. Existing accounts should explicitly disallow TLS 1.0 and 1.1, which are deprecated.
- **Encryption models.** Azure supports various encryption models, including server-side encryption that uses service-managed keys, customer-managed keys in Key Vault, or customer-managed keys on customer-controlled hardware. With client-side encryption, you can manage and store keys on-premises or in another secure location.
- **Authorize requests.** For optimal security, Microsoft recommends using Microsoft Entra ID with managed identities to authorize requests against blob, queue, and table data, whenever possible. Authorization with Microsoft Entra ID and managed identities provides superior security and ease of use over Shared Key authorization.
- **RBAC.** RBAC ensures that resources in your storage account are accessible only when you want them to be, and to only those users or applications whom you grant access. Assign RBAC roles scoped to an Azure storage account.
- **Storage analytics.** Azure Storage Analytics performs logging for a storage account. You can use this data to trace requests, analyze usage trends, and diagnose issues with your storage account.

Tip

The Microsoft [storage cloud security benchmark](#) provides recommendations on how you can secure your cloud storage solutions.

Things to consider when using authorization security

Review the following strategies for authorizing requests to Azure Storage. Think about what security strategies would work for your Azure Storage.

Authorization strategy	Description
Microsoft Entra ID	Microsoft Entra ID is Microsoft's cloud-based identity and access management service. With Microsoft Entra ID, you can assign fine-grained access to users, groups, or applications by using role-based access control.
Shared Key	Access is authorized with an account access key. The key can be the primary or secondary access key. To enforce Entra ID authorization, disable the Shared Key at the storage account level.

Authorization strategy	Description
Shared access signatures	A SAS delegates access to a particular resource in your Azure storage account with specified permissions and for a specified time interval.
Anonymous access to containers and blobs	Anonymous public access is disabled by default on new storage accounts. Microsoft recommends keeping anonymous access disabled for accounts containing sensitive data.

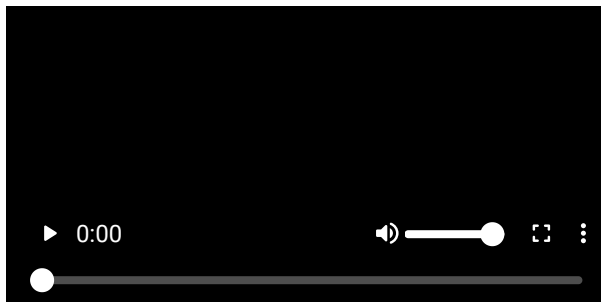
3. Create shared access signatures

<https://learn.microsoft.com/en-us/training/modules/configure-storage-security/3-create-shared-access-signatures>

Create shared access signatures

Completed

A [shared access signature \(SAS\)](#) is a uniform resource identifier (URI) that grants restricted access rights to Azure Storage resources. SAS is a secure way to share your storage resources without compromising your account keys.



You can provide a SAS to clients who shouldn't have access to your storage account key. By distributing a SAS URI to these clients, you grant them access to a resource for a specified period of time. You'd typically use a SAS for a service where users read and write their data to your storage account.

- A *user delegation SAS* is secured with Microsoft Entra credentials and also by the permissions specified for the SAS. A user delegation SAS is supported for Blob Storage and Data Lake Storage,
- An *account-level SAS* to allow access to anything that a service-level SAS can allow, plus other resources and abilities. For example, you can use an account-level SAS to allow the ability to create file systems.

- A *service-level SAS* to allow access to specific resources in a storage account. You'd use this type of SAS, for example, to allow an app to retrieve a list of files in a file system, or to download a file.
- A *stored access policy* can provide another level of control when you use a service-level SAS on the server side. You can group SASs and provide other restrictions by using a stored access policy.

Recommendations for managing risks

Let's look at some recommendations that can help mitigate risks when working with a SAS.

Recommendation	Description
Always use HTTPS for creation and distribution	If a SAS is passed over HTTP and intercepted, an attacker can intercept and use the SAS. These <i>man-in-the-middle</i> attacks can compromise sensitive data or allow for data corruption by the malicious user.
Reference stored access policies where possible	Stored access policies give you the option to revoke permissions without having to regenerate the Azure storage account keys. Set the storage account key expiration date far in the future.
Set near-term expiry times for an unplanned SAS	If a SAS is compromised, you can mitigate attacks by limiting the SAS validity to a short time. This practice is important if you can't reference a stored access policy. Near-term expiration times also limit the amount of data that can be written to a blob by limiting the time available to upload to it.
Require clients automatically renew the SAS	Require your clients to renew the SAS well before the expiration date. By renewing early, you allow time for retries if the service providing the SAS is unavailable.
Plan carefully for the SAS start time	If you set the start time for a SAS to now, then due to clock skew (differences in current time according to different machines), failures might be observed intermittently for the first few minutes. In general, set the start time to at least 15 minutes in the past. Or, don't set a specific start time, which causes the SAS to be valid immediately in all cases. The same conditions generally apply to the expiry time. You might observe up to 15 minutes of clock skew in either direction on any request. For clients that use a REST API version earlier than 2012-02-12, the maximum duration for a SAS that doesn't reference a stored access policy is 1 hour. Any policies that specify a longer term fail.
Define minimum access permissions for resources	A security best practice is to provide a user with the minimum required privileges. If a user only needs read access to a single entity, then grant them read access to that single entity, and not read/write/delete access to all entities. This practice also helps lessen the damage if a SAS is compromised because the SAS has less power in the hands of an attacker.
Validate data written by using a SAS	When a client application writes data to your Azure storage account, keep in mind there can be problems with the data. If your application requires validated or authorized data, validate the data after written, but before used. This practice also protects against corrupt or malicious data being written to your account, either by a user who properly acquired the SAS, or by a user exploiting a leaked SAS.

Recommendation	Description
Don't assume a SAS is always the correct choice	In some scenarios, the risks associated with a particular operation against your Azure storage account outweigh the benefits of using a SAS. For such operations, create a middle-tier service that writes to your storage account after performing business rule validation, authentication, and auditing. Also, sometimes it's easier to manage access in other ways. If you want to make all blobs in a container publicly readable, you can make the container Public, rather than providing a SAS to every client for access.

4. Identify URI and SAS parameters

<https://learn.microsoft.com/en-us/training/modules/configure-storage-security/4-identify-uri-sas-parameters>

Identify URI and SAS parameters

Completed

When you create your shared access signature (SAS), a uniform resource identifier (URI) is created by using parameters and tokens. The URI consists of your Azure Storage resource URI and the SAS token.

Things to know about URI definitions

Let's look at an example URI definition and examine the parameters. This example creates a service-level SAS that grants read and write permissions to a blob. Consider how you might configure the parameters to support your Azure Storage resources.

```
https://myaccount.blob.core.windows.net/?restype=service&comp=properties&sv=2015-04
```

Parameter	Example	Description
Resource URI	<code>https://myaccount. blob .core.windows.net/ ? restype= service &comp=properties</code>	Defines the Azure Storage endpoint and other

Parameter	Example	Description
		parameters. This example defines an endpoint for Blob Storage and indicates that the SAS applies to service-level operations. When the URI is used with <code>GET</code> , the Storage properties are retrieved. When the URI is used with <code>SET</code> , the Storage properties are configured.
Storage version	<code>sv =2015-04-05</code>	For Azure Storage version 2012-02-12 and later, this parameter indicates the version to use. This example indicates that version 2015-04-05 (April 5, 2015) should be used.
Storage service	<code>ss =bf</code>	Specifies the Azure Storage to which the SAS applies. This example indicates that the SAS applies to Blob Storage and Azure Files.
Start time	<code>st =2015-04-29T22%3A18%3A26Z</code>	(Optional) Specifies the start time for the SAS in UTC time. This example sets the start time as April 29, 2015 22:18:26 UTC. If you want the SAS to be valid immediately, omit the start time.
Expiry time	<code>se =2015-04-30T02%3A23%3A26Z</code>	Specifies the expiration time for the SAS in UTC time. This example sets the expiry time as April 30, 2015 02:23:26 UTC.
Resource	<code>sr =b</code>	Specifies which resources are accessible via the SAS. This example specifies that the accessible resource is in Blob Storage.
Permissions	<code>sp =rw</code>	Lists the permissions to grant. This example grants access to read and write operations.

Parameter	Example	Description
IP range	<code>sip =168.1.5.60-168.1.5.70</code>	Specifies a range of IP addresses from which a request is accepted. This example defines the IP address range 168.1.5.60 through 168.1.5.70.
Protocol	<code>spr =https</code>	Specifies the protocols from which Azure Storage accepts the SAS. This example indicates that only requests by using HTTPS are accepted.
Signature	<code>sig =F%6GRVAZ5Cdj2Pw4tgU7IL STkWgn7bUkkAg8P6HESXwmf%4B</code>	Specifies that access to the resource is authenticated by using a Hash-Based Message Authentication Code (HMAC) signature. The signature is computed with a key using the SHA256 algorithm, and encoded by using Base64 encoding.

Tip

Continue your learning with the [Implement shared access signatures](#) training modules.

5. Determine Azure Storage encryption

<https://learn.microsoft.com/en-us/training/modules/configure-storage-security/5-determine-storage-service-encryption>

Determine Azure Storage encryption

Completed

Azure Storage encryption for data at rest protects your data by ensuring your organizational security and compliance commitments are met. The encryption and decryption processes happen automatically. Because your data is secured by default, you don't need to modify your code or applications.

When you create a storage account, Azure generates two 512-bit storage account access keys for that account. These keys can be used to authorize access to data in your storage account via Shared Key authorization, or via SAS tokens that are signed with the shared key.

Microsoft recommends that you use Azure Key Vault to manage your access keys, and that you regularly rotate and regenerate your keys. Azure Key Vault supports automatic key rotation policies, allowing you to define rotation schedules (for example, every 90 days) that rotate keys automatically. You can also manually rotate your keys when needed.

Things to know about Azure Storage encryption

Examine the following characteristics of Azure Storage encryption.

- Data is automatically encrypted before written to Azure storage.
- Data is automatically decrypted when retrieved.
- Azure Storage encryption, encryption at rest, decryption, and key management are transparent to users.
- All data written to Azure Storage is encrypted through 256-bit advanced encryption standard (AES) encryption. AES is one of the strongest block ciphers available.
- Azure Storage encryption is enabled for all new and existing storage accounts and can't be disabled.

Configure Azure Storage encryption

In the Azure portal, you configure Azure Storage encryption by specifying the encryption type. You can manage the keys yourself, or you can have the keys managed by Microsoft. Consider how you might implement Azure Storage encryption for your storage security.



211795eastus | Encryption

Storage account

Encryption

Encryption scopes

Storage service encryption protects your data at rest. Azure Storage encrypts your data as it's written in our datacenters, and automatically decrypts it for you as you access it.

Please note that after enabling Storage Service Encryption, only new data will be encrypted, and any existing files in this storage account will retroactively get encrypted by a background encryption process.

[Learn more about Azure Storage encryption](#)

Encryption selection

Enable support for customer-managed keys ⓘ

Blobs and files only

Infrastructure encryption ⓘ

Disabled

Encryption type

☒ Microsoft-managed keys

☐ Customer-managed keys

- **Infrastructure encryption.** [Infrastructure encryption](#) can be enabled for the entire storage account, or for an encryption scope within an account. When infrastructure encryption is enabled for a storage account or an encryption scope, data is encrypted twice—once at the service level and once at the infrastructure level—with two different encryption algorithms and two different keys.
- **Platform-managed keys.** Platform-managed keys (PMKs) are encryption keys generated, stored, and managed entirely by Azure. Customers don't interact with PMKs. The keys used for Azure Data Encryption-at-Rest, for instance, are PMKs by default.
- **Customer-managed keys.** Customer managed keys (CMK), on the other hand, are keys read, created, deleted, updated, and/or administered by one or more customers. Keys stored in a customer-owned key vault or hardware security module (HSM) are CMKs. Bring Your Own Key (BYOK) is a CMK scenario in which a customer imports (brings) keys from an outside storage location. This topic is discussed in more detail on the next page.

6. Create customer-managed keys

Create customer-managed keys

Completed

For your Azure Storage security solution, you can use Azure Key Vault to manage your encryption keys. The Azure Key Vault APIs can be used to generate encryption keys. You can also create your own encryption keys and store them in a key vault.



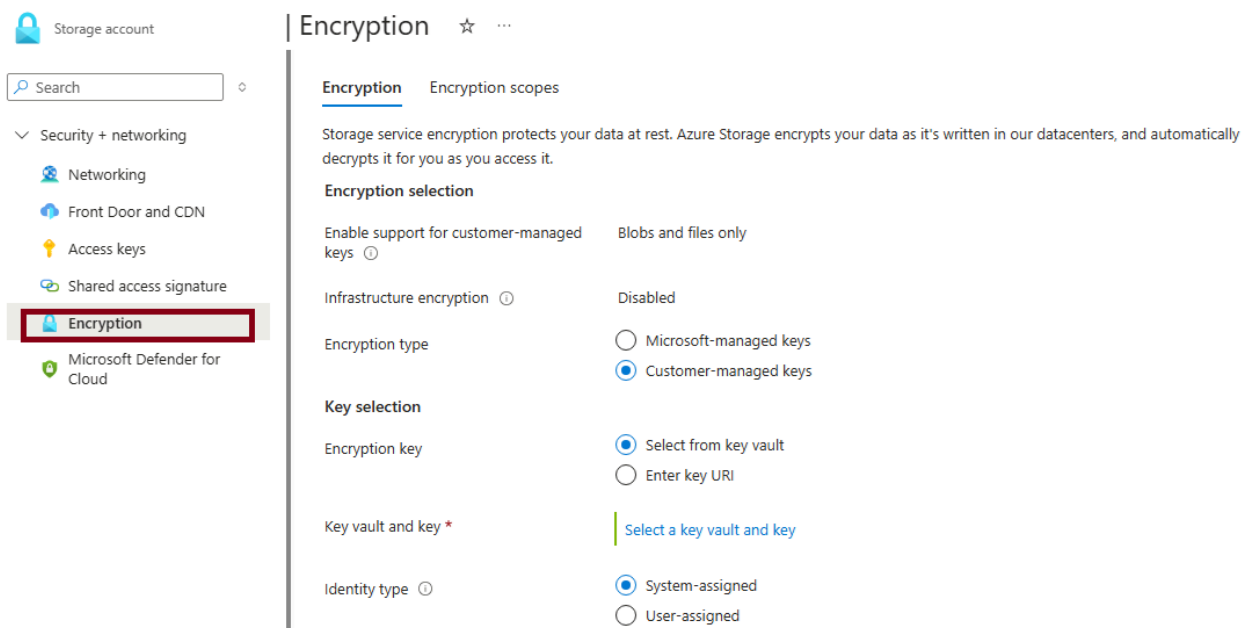
Things to know about customer-managed keys

Consider the following characteristics of customer-managed keys.

- By creating your own keys (referred to as *customer-managed keys*), you have more flexibility and greater control.
- You can create, disable, audit, rotate, and define access controls for your encryption keys.
- Customer-managed keys can be used with Azure Storage encryption. You can use a new key or an existing key vault and key. The Azure storage account and the key vault must be in the same region, but they can be in different subscriptions.
- Customer-managed keys are stored in a customer-owned Azure Key Vault or Azure Key Vault Managed HSM. Managed HSM provides FIPS 140-2 Level 3 validation for organizations with the highest compliance requirements.

Configure customer-managed keys

In the Azure portal, you can configure customer-managed encryption keys. You can create your own keys, or you can have the keys managed by Microsoft. Consider how you might use Azure Key Vault to create your own customer-managed encryption keys.



- **Encryption type:** Choose how the encryption key is managed: by Microsoft or by yourself (customer).
- **Encryption key:** Specify an encryption key by entering a URI, or select a key from an existing key vault.

Tip

Expand your understanding of storage security in the [Plan and implement security for storage](#) training module.

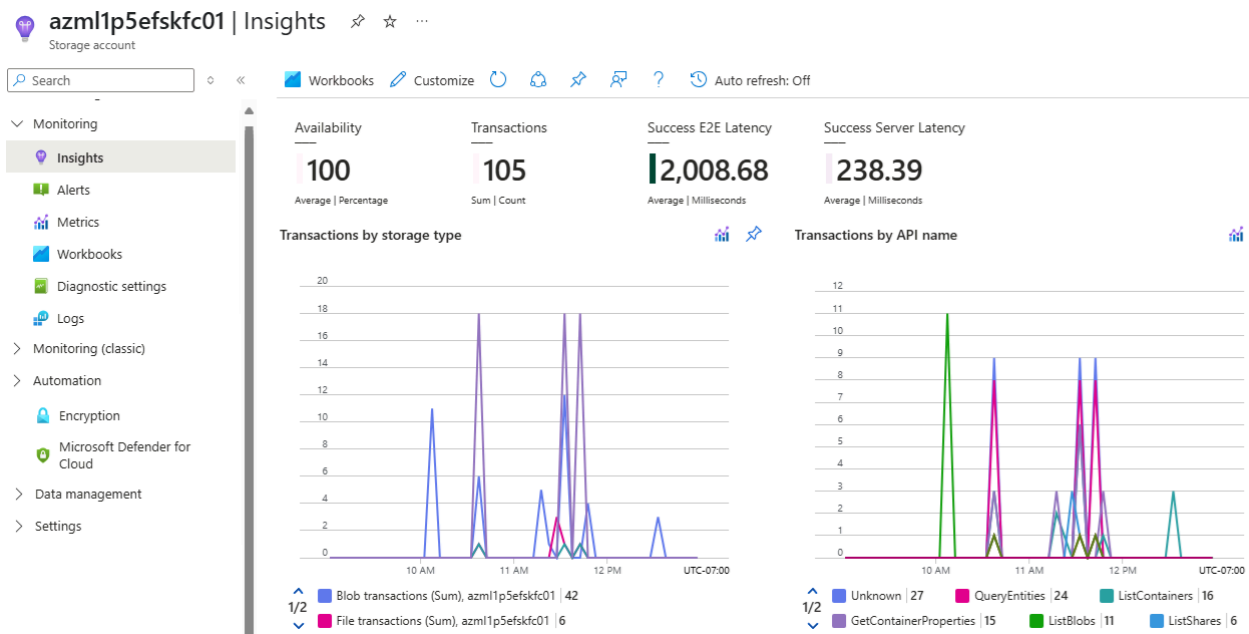
7. Apply Azure Storage security best practices

<https://learn.microsoft.com/en-us/training/modules/configure-storage-security/7-apply-best-practices>

Apply Azure Storage security best practices

Completed

[Storage insights](#) provides comprehensive monitoring of your Azure Storage accounts. Storage Insights delivers a unified view of your Azure Storage services performance, capacity, and availability.



What are the benefits of Storage insights?

- **Detailed Metrics and Logs.** Azure Storage Insights offers detailed metrics, logs, and diagnostic information that enhance visibility into storage operations. Insights helps in monitoring key performance indicators (KPIs) such as latency, throughput, capacity utilization, and transactions.
- **Enhanced Security and Compliance.** By using Azure Storage Insights, you can ensure enhanced security and compliance. It provides actionable insights and alerts that help in swiftly identifying and resolving security issues.
- **Role-Based Access Control (RBAC).** Azure Storage Insights integrates with Azure's security features, including role-based access control (RBAC), Microsoft Entra ID, connection strings, and access control list (ACL) permissions. RBAC ensures secure access to your data and resources.
- **Unified View.** It delivers a unified view of your Azure Storage services' performance, capacity, and availability, which is crucial for maintaining the security and efficiency of your storage accounts.

When to use Storage Insights

- **Real-Time Monitoring.** Azure Storage Insights enables real-time monitoring of storage accounts, allowing you to track usage trends, monitor performance, and set up alerts for any anomalies.
- **Security Auditing.** It aids in security auditing by providing comprehensive monitoring and detailed logs, which are essential for ensuring compliance and identifying any security issues.
- **Health Analysis and Optimization.** The tool helps in health analysis and optimization of storage accounts, ensuring security and optimal performance.

When to use Microsoft Defender for Storage

While Storage Insights provides passive monitoring and historical analysis, Microsoft Defender for Storage offers proactive threat detection for active security threats.

Key capabilities

- **Malware scanning.** Automatically scans blob uploads for malware and viruses.
- **Sensitive data threat detection.** Identifies when personally identifiable information (PII) or credentials are stored inappropriately.
- **Activity-based threat detection.** Monitors for unusual access patterns, suspicious download volumes, and hash reputation analysis.

Microsoft Defender for Storage complements Storage Insights by providing active threat detection rather than reactive monitoring and historical reporting.

8. Exercise: Manage Azure Storage

<https://learn.microsoft.com/en-us/training/modules/configure-storage-security/8-simulation-storage>

Exercise: Manage Azure Storage

Completed

Lab scenario

In this lab, you learn to create storage accounts for Azure blobs and Azure files. You learn to configure and secure blob containers. You also learn to use Storage Browser to configure and secure Azure file shares.

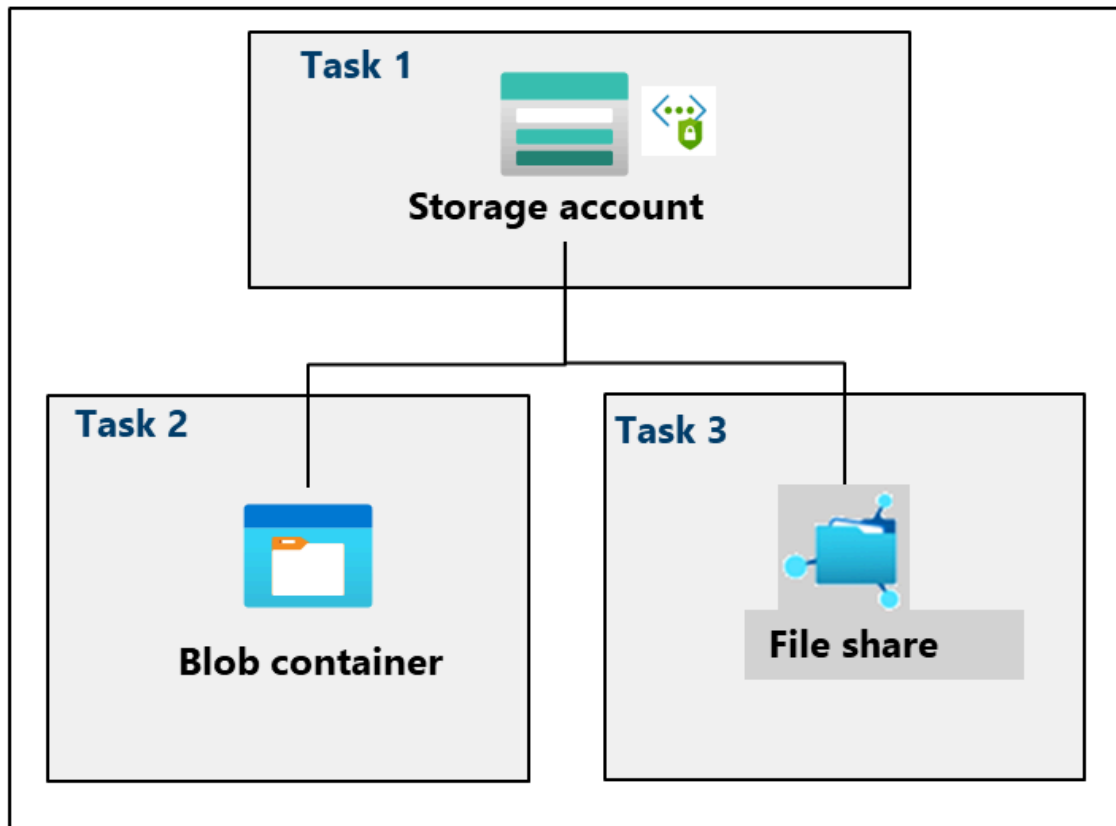
Note

This lab covers storage accounts, blobs, and files. As you go through the steps, consider the security features you learned about.

Architecture diagram



az104-07-rg7



Job skills

- Create and configure a storage account.
- Create and configure secure blob storage.
- Create and configure secure Azure file storage.

Note

Estimated timing: 50 minutes. To complete this exercise, you need an [Azure subscription](#).

Launch the exercise, and follow the instructions. When finished, be sure to return to this page so you can continue learning.

[Launch Exercise](#)

9. Module assessment

Module assessment

Completed

10. Summary and resources

<https://learn.microsoft.com/en-us/training/modules/configure-storage-security/10-summary-resources>

Summary and resources

Completed

Azure Administrators must be familiar with how to configure storage security.

In this module, you examined several options for securing Azure Storage. You discovered how to configure shared access signatures (SAS), including the uniform resource identifier (URI) and SAS parameters. You reviewed how to implement customer-managed keys and define stored access policies to configure Azure Storage encryption. You explored opportunities for improving your Azure Storage security solution.

Learn more with Copilot

Copilot can assist you in configuring Azure infrastructure solutions. Copilot can compare, recommend, explain, and research products and services where you need more information. Open a Microsoft Edge browser and choose Copilot (top right) or navigate to copilot.microsoft.com. Take a few minutes to try these prompts and extend your learning with Copilot.

- What are the different ways to secure Azure storage? Provide usage case examples.
- How do I configure an Azure Shared Access Signature?

Learn more with documentation

- Grant [limited access to Azure Storage resources with shared access signatures](#).
- Read about [Azure Storage encryption for data at rest](#).

- Create a [SAS](#) for your Azure storage account.
- Create a [service-level SAS](#).
- Construct a [user delegation SAS](#).
- Use [customer-managed keys](#) for Azure Storage encryption.

Learn more with self-paced training

- Secure your [Azure storage account](#).
- Implement [Azure Storage security](#).

Configure Azure Files

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/configure-azure-files-file-sync/1-introduction>

Introduction

Completed

Azure Files offers fully managed file shares in the cloud that are accessible via industry standard protocols. Azure File Sync is a service that allows you to cache several Azure Files shares on an on-premises Windows Server or cloud virtual machine.

In this module, your company has a large repository of organizational documents. Offices are located in different geographical regions, and users need the most current versions of the documents. You're researching how to implement Azure Files shares to provide a central location for the documents.

Learning objectives

In this module, you learn how to:

- Identify storage for file shares.
- Compare file shares to blob storage.
- Configure Azure file shares, file share snapshots, and soft delete.
- Use Azure Storage Explorer to access your file share.

Skills measured

The content in the module helps you prepare for [Exam AZ-104: Microsoft Azure Administrator](#).

Prerequisites

- Familiarity with shared file systems.
- Familiarity with navigating the Azure portal.

2. Compare storage for file shares and blob data

<https://learn.microsoft.com/en-us/training/modules/configure-azure-files-file-sync/2-compare-files-to-blobs>

Compare storage for file shares and blob data

Completed

[Azure Files](#) offers fully managed file shares in the cloud. You can access Azure file shares by using the Server Message Block (SMB), Network File System (NFS), and HTTP protocols. Clients can connect to Azure file shares from Windows, Linux, and macOS devices.

Things to know about Azure Files

Here are some characteristics of Azure files:

- **Serverless deployment.** An Azure file share is a PaaS offering of a fully managed file share that doesn't require any infrastructure. You don't need to take care of any VMs, operating systems, or updates.
- **Almost unlimited storage.** A single Azure file share can store up to 100 terabytes (TiB) of files, and a file can be up to 4 TiB in size. The files are organized in a hierarchical folder structure in the same way as on on-premises file servers.
- **Data encryption.** The data on an Azure file share is encrypted at rest in an Azure datacenter and in transit on a network.
- **Access from anywhere.** By default, clients can access Azure file shares from anywhere if they have internet connectivity.
- **Integration into an existing environment.** You can control access to Azure file shares by using Microsoft Entra identities or AD DS identities that are synced to Microsoft Entra ID. This helps ensure that users can have the same experience accessing an Azure file share as when they access an on-premises file server.
- **Previous versions and backups.** You can create Azure file share snapshots that integrate with the Previous Versions feature in File Explorer. You can also use Azure Backup to back up Azure file shares.
- **Data redundancy.** Azure file share data replicates to multiple locations in the same Azure datacenter or across many Azure datacenters. The replication setting of the Azure storage

account that includes the file share controls the data redundancy.

Things to consider when using Azure Files

There are many common scenarios for using Azure Files. As you review the following suggestions, think about how Azure Files can provide solutions for your organization.

- **Consider replacement and supplement options.** Replace or supplement traditional on-premises file servers or NAS devices by using Azure Files.
- **Consider global access.** Directly access Azure file shares by using most operating systems, such as Windows, macOS, and Linux, from anywhere in the world.
- **Consider lift and shift support.** *Lift and shift* applications to the cloud with Azure Files for apps that expect a file share to store file application or user data.
- **Consider using Azure File Sync.** Replicate Azure file shares to Windows Servers by using Azure File Sync. You can replicate on-premises or in the cloud for performance and distributed caching of the data where it's being used. We take a closer look at Azure File Sync in a later unit.
- **Consider shared applications.** Store shared application settings such as configuration files in Azure Files.
- **Consider diagnostic data.** Use Azure Files to store diagnostic data such as logs, metrics, and crash dumps in a shared location.
- **Consider tools and utilities.** Azure Files is a good option for storing tools and utilities that are needed for developing or administering Azure VMs or cloud services.

Compare Azure Files to Azure Blob Storage

It's important to understand when to use Azure Files to store data in file shares rather than using Azure Blob Storage to store data as blobs. The following table compares different features of these services and common implementation scenarios.

Azure Files (file shares)	Azure Blob Storage (blobs)
Azure Files provides the SMB and NFS protocols, client libraries, and a REST interface that allows access from anywhere to stored files.	Azure Blob Storage provides client libraries and a REST interface that allows unstructured data to be stored and accessed at a massive scale in block blobs.
- Files in an Azure Files share are true directory objects. - Data in Azure Files is accessed through file shares across multiple virtual machines.	- Blobs in Azure Blob Storage are a flat namespace. - Blob data in Azure Blob Storage is accessed through a container.

Azure Files (file shares)	Azure Blob Storage (blobs)
Azure Files is ideal to lift and shift an application to the cloud that already uses the native file system APIs. Share data between the app and other applications running in Azure. Azure Files is a good option when you want to store development and debugging tools that need to be accessed from many virtual machines.	Azure Blob Storage is ideal for applications that need to support streaming and random-access scenarios. Azure Blob Storage is a good option when you want to be able to access application data from anywhere.

3. Manage Azure file shares

<https://learn.microsoft.com/en-us/training/modules/configure-azure-files-file-sync/3-manage-file-shares>

Manage Azure file shares

Completed

Azure Files offers two industry-standard file system protocols for mounting Azure file shares: the Server Message Block (SMB) protocol and the Network File System (NFS) protocol. Azure file shares don't support both the SMB and NFS protocols on the same file share, although you can create SMB and NFS Azure file shares within the same storage account.

Types of Azure file shares

Azure Files supports two storage tiers: premium and standard. Standard file shares are created in general purpose (GPv2) storage accounts, while premium file shares are created in FileStorage storage accounts. The two storage tiers have the attributes described in the following table.

Storage tier	Performance	Storage account type	Redundancy options	Billing model	Use cases
Premium	SSD-backed, consistent low latency	FileStorage	LRS, ZRS	Provisioned (pay for capacity reserved)	High-performance workloads requiring low latency
Transaction Optimized	HDD-backed, standard performance	General-purpose v2 (GPv2)	LRS, GRS, RA-GRS, ZRS, GZRS, RA-GZRS	Pay-as-you-go	High-transaction workloads,

Storage tier	Performance	Storage account type	Redundancy options	Billing model	Use cases
					frequently accessed data
Hot	HDD-backed, standard performance	General-purpose v2 (GPv2)	LRS, GRS, RA-GRS, ZRS, GZRS, RA-GZRS	Pay-as-you-go	General-purpose team shares and collaborative workloads
Cool	HDD-backed, standard performance	General-purpose v2 (GPv2)	LRS, GRS, RA-GRS, ZRS, GZRS, RA-GZRS	Pay-as-you-go	Cost-efficient online archive and backup scenarios

Note

Transaction Optimized, Hot, and Cool are all Standard (HDD-based) tiers with different pricing structures optimized for specific access patterns. Premium tier uses SSD storage with provisioned billing (you pay for the capacity you reserve), while Standard tiers use pay-as-you-go billing.

Types of authentication

There are three main authentications methods that Azure Files supports.

Authentication method	Description
Identity-based authentication over SMB	SMB identity-based authentication supports three Active Directory sources: On-premises AD DS, Microsoft Entra Domain Services, and Microsoft Entra Kerberos. Once your Active Directory source is selected, assign Azure RBAC roles to users who need access to the file share.
Access key	An access key is an older and less flexible option. An Azure storage account has two access keys that can be used when making a request to the storage account, including to Azure Files. Access keys are static and provide full control access to Azure Files. Access keys should be secured and not shared with users, because they bypass all access control restrictions. A best practice is to avoid sharing storage account keys and use identity-based authentication whenever possible.
A Shared Access Signature (SAS) token	SAS is a dynamically generated Uniform Resource Identifier (URI) that's based on the storage access key. SAS provides restricted access rights to an Azure storage account. Restrictions include allowed permissions, start and expiry time, allowed IP addresses from where requests can be sent, and allowed protocols. With Azure Files, a SAS token is only used to provide REST API access from code.

Creating SMB Azure file shares (classic)

Classic Azure file shares live inside a storage account, so they follow the same limits as that account. You can choose between two storage tiers: SSD (premium) and HDD (standard).

SSD file shares are great when you need fast, consistent performance with low latency—usually in the single digit milliseconds. HDD shares are more budget friendly and work well for general purpose storage.

If you need SMB access, make sure to create your file share inside a storage account. SMB file shares let you pick from several access tiers, including transaction optimized, hot, and cool.

Home > Storage center | File shares

New file share ...

Basics Backup Review + create

Name *

Access tier *

Transaction optimized



Premium

Premium file shares are only available for premium file storage accounts.

Transaction optimized

Lowest transaction cost pricing for transaction-heavy workloads that don't need the low latency offered by premium file shares. Recommended while migrating data to Azure Files.

Hot

Balanced storage and transaction pricing for workloads that have a good measure of both.

Cool

Most cost-efficient storage pricing for storage-intensive workloads.

Performance

Maximum IO/s ⓘ 20000

Maximum capacity 100 TiB

Note

When connecting over SMB, don't forget that traffic uses port 445. Many ISPs block port 445 outbound, which is the most common connectivity issue when mounting Azure file shares from on-premises environments.

Important

[File shares \(preview\)](#) are now generally available that don't require an Azure storage account. This option provides simplified management for scenarios where you only need file shares without other storage services.

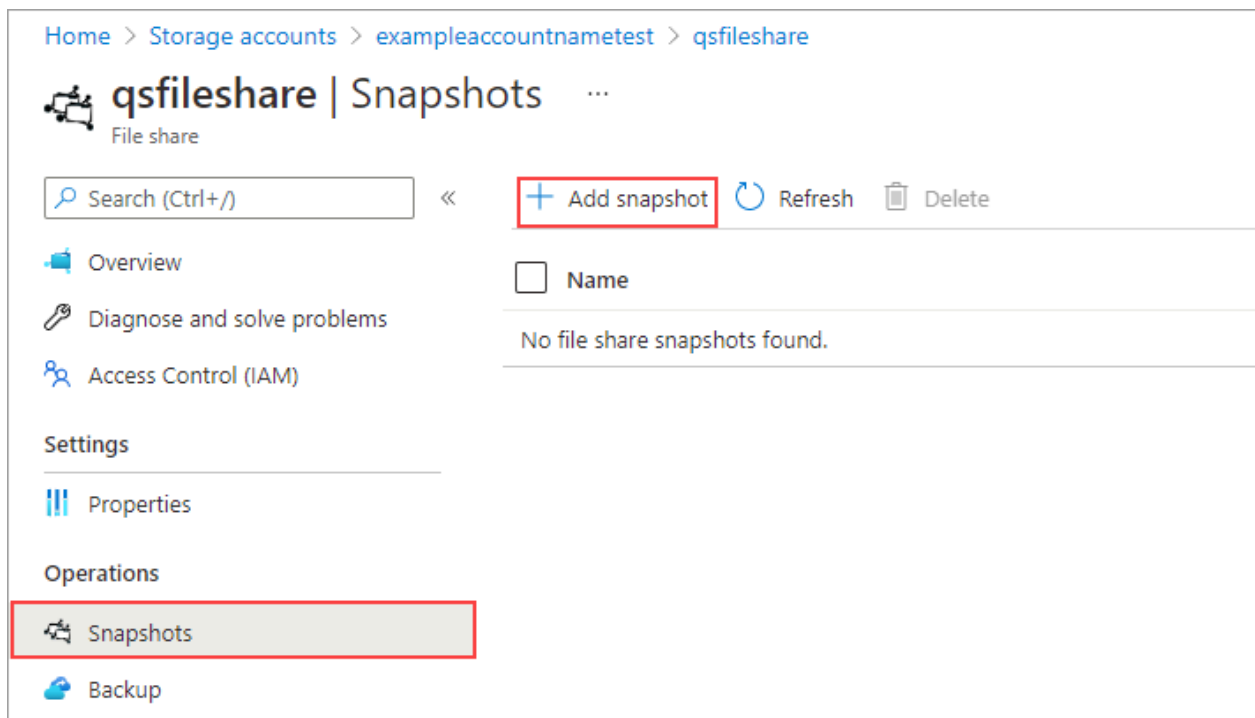
4. Create file share snapshots

<https://learn.microsoft.com/en-us/training/modules/configure-azure-files-file-sync/4-create-file-share-snapshots>

Create file share snapshots

Completed

Azure Files provides the capability to take share [snapshots of file shares](#). Share snapshots provide point-in-time copies of your Azure file shares that protect against accidental deletion and enable recovery from application errors.



Things to know about file share snapshots

- Snapshots are incremental, read-only point-in-time copies at the share level.
- To reduce time and cost only captures from the last snapshot.
- Same experience for SMB and NFS shares in all Azure public regions.

- Snapshot adds a unique timestamp to the share URI.
- Uses the shares redundancy settings.
- Up to 200 snapshots per file share for low-RPO recovery points.
- Snapshots persist until deleted. Deleting the share deletes all snapshots.
- Azure Backup can lease snapshots to help prevent accidental deletion.
- Restore a file, folder, or full share; full restore requires only the latest snapshot.

Things to consider when using file share snapshots

File share snapshots can help you protect and recover your data. As you review the benefits, consider where snapshots fit into your Azure Files setup.

Benefit	Description
Protect against application error and data corruption	File-share workloads constantly read and write data. If a misconfiguration, bad deployment, or software bug overwrites or corrupts data, a snapshot lets you roll the share back to a known-good point in time. Take a snapshot before releasing new code so you have a clean restore point if something goes wrong.
Protect against accidental deletions or unintended changes	If a file is changed, snapshots give you a quick way to restore an earlier version. Use snapshots to roll back to the last good copy when something unexpected happens.
Support backup and recovery	Create snapshots on a schedule to build a backup history for your file share. Keeping prior versions makes it easier to meet audit needs and recover data after mistakes or a broader outage.

For automated snapshot creation or integration with existing scripts, PowerShell and Azure CLI provide programmatic access to snapshot operations. Both tools support adding metadata to snapshots and can be scheduled through Azure Automation, GitHub Actions, or any continuous integration system.

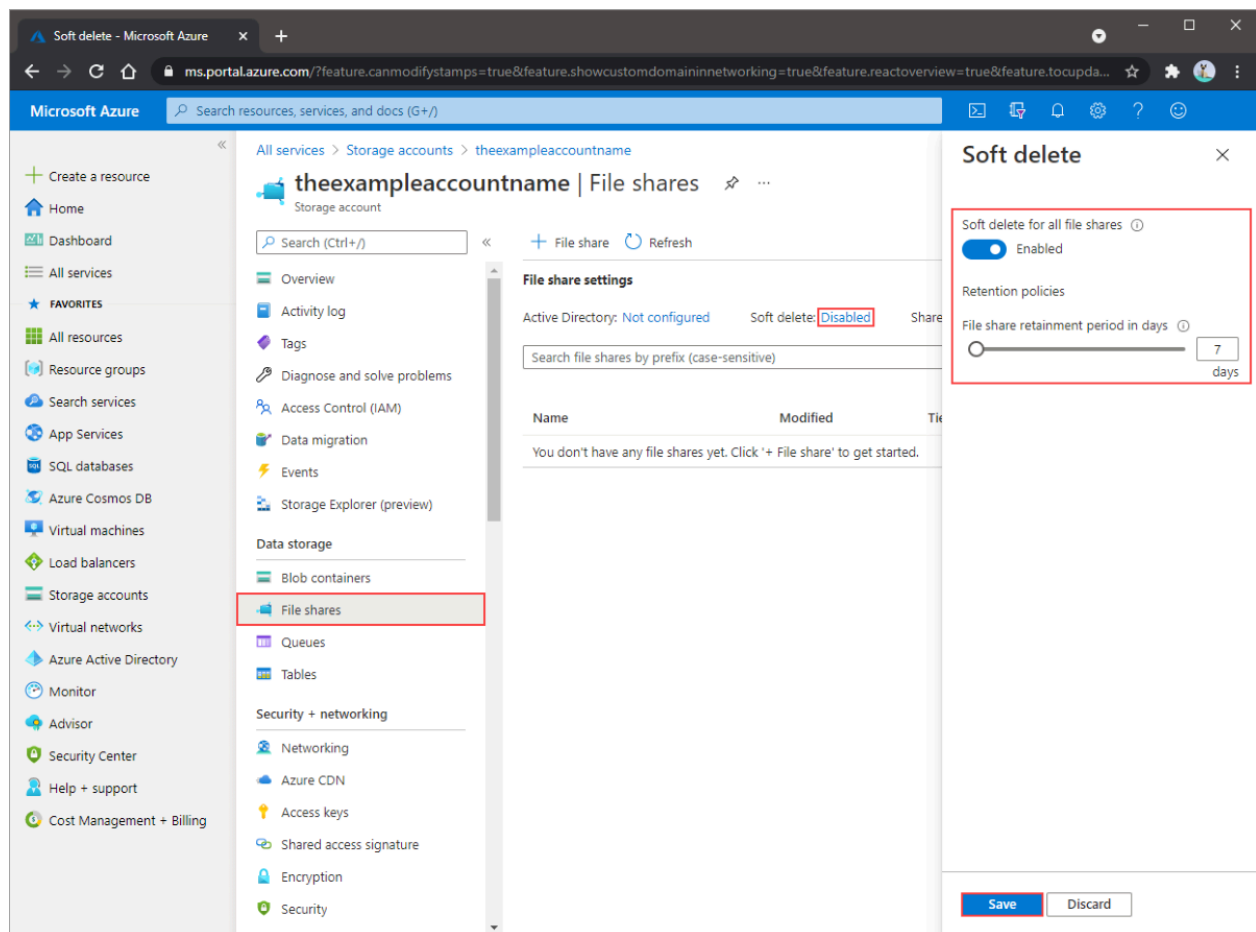
5. Implement soft delete for Azure Files

<https://learn.microsoft.com/en-us/training/modules/configure-azure-files-file-sync/5-implement-file-sync>

Implement soft delete for Azure Files

Completed

Azure Files offers [soft delete for file shares](#). Soft delete lets you recover deleted files and file shares.



Things to know about soft delete for Azure Files

Let's take a look at the characteristics of soft delete for Azure Files.

- Soft delete for file shares is enabled at the storage account level.
- Soft delete transitions content to a soft deleted state instead of being permanently erased.
- Soft delete lets you configure the retention period. The retention period is the amount of time that soft deleted file shares are stored and available for recovery.
- Soft delete provides a retention period between 1 and 365 days.
- Soft delete can be enabled on either new or existing file shares.

Things to consider when using soft delete for Azure Files

There are many advantages to using soft delete for Azure Files. Consider the following scenarios, and think about how you can use soft delete.

- **Recover from accidental data loss.** You can recover deleted or corrupted data with soft delete.

- **Upgrade scenarios.** Use soft delete to restore to a known good state after a failed upgrade attempt.
- **Ransomware protection.** Use soft delete to recover data without paying ransom to cybercriminals.
- **Long-term retention.** Use soft delete to comply with data retention requirements.
- **Business continuity.** Use soft delete to prepare your infrastructure to be highly available for critical workloads.

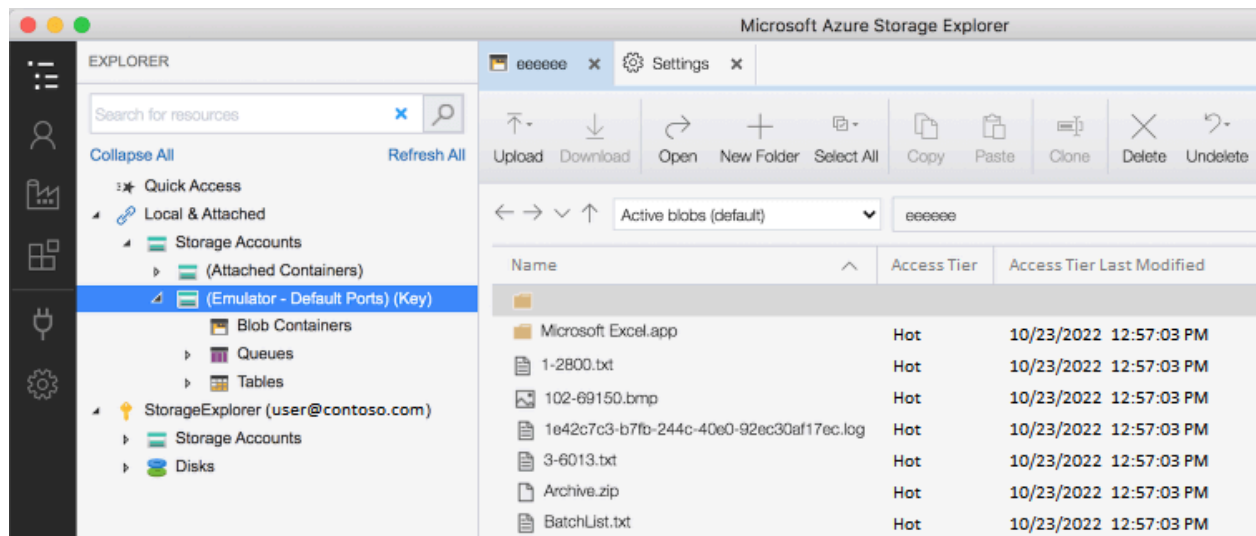
6. Use Azure Storage Explorer

<https://learn.microsoft.com/en-us/training/modules/configure-azure-files-file-sync/6-identify-components>

Use Azure Storage Explorer

Completed

[Azure Storage Explorer](#) is a standalone application that makes it easy to work with Azure Storage data on Windows, macOS, and Linux. With Azure Storage Explorer, you can access multiple accounts and subscriptions, and manage all your Storage content.



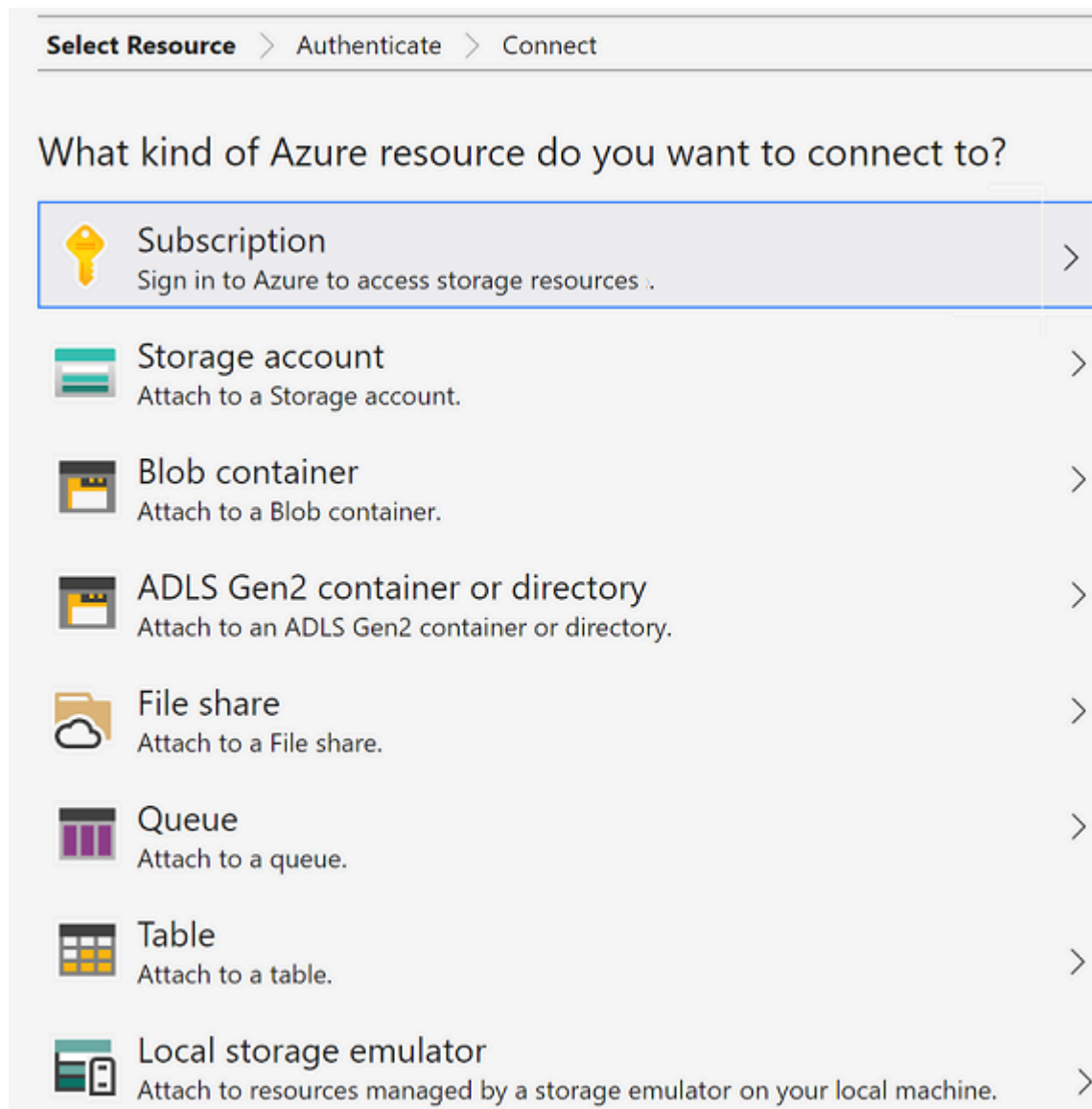
Things to know about Azure Storage Explorer

Azure Storage Explorer has the following characteristics.

- Azure Storage Explorer requires both management (Azure Resource Manager) and data layer permissions to allow full access to your resources. You need Microsoft Entra ID permissions to

access your storage account, the containers in your account, and the data in the containers.

- Azure Storage Explorer lets you connect to different storage accounts.
 - Connect to storage accounts associated with your Azure subscriptions.
 - Connect to storage accounts and services that are shared from other Azure subscriptions.
 - Connect to and manage local storage by using the Azure Storage Emulator.



Things to consider when using Azure Storage Explorer

Azure Storage Explorer supports many scenarios for working with storage accounts in Azure. As you review these options, think about which scenarios apply to your Azure Storage implementation.

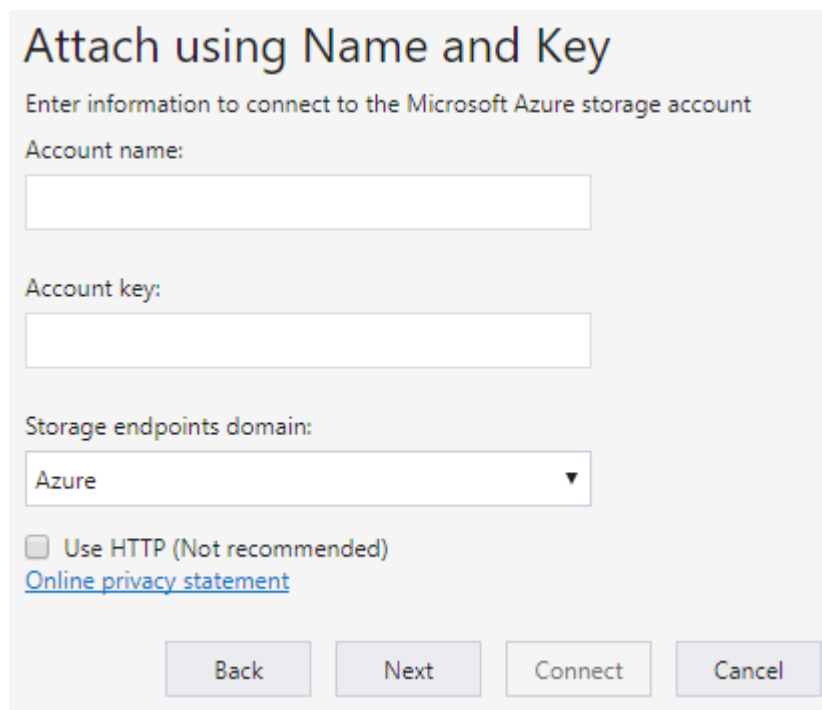
Scenario	Description
Connect to an Azure subscription	Manage storage resources that belong to your Azure subscription.

Scenario	Description
Work with local development storage	Manage local storage by using the Azure Storage Emulator.
Attach to external storage	Manage storage resources that belong to another Azure subscription or that are under national Azure clouds by using the storage account name, key, and endpoints. This scenario is described in more detail in the next section.
Attach a storage account with a SAS	Manage storage resources that belong to another Azure subscription by using a shared access signature (SAS).
Attach a service with a SAS	Manage a specific Azure Storage service (blob container, queue, or table) that belongs to another Azure subscription by using a SAS.

Attach to external storage account

Azure Storage Explorer lets you attach to external storage accounts so storage accounts can be easily shared.

To create the connection, you need the external storage **Account name** and **Account key**. In the Azure portal, the account key is called **key1**.



The screenshot shows a dialog box titled "Attach using Name and Key". Below the title is the instruction "Enter information to connect to the Microsoft Azure storage account". There are three input fields: "Account name:" (a text box), "Account key:" (a text box), and "Storage endpoints domain:" (a dropdown menu currently showing "Azure"). Below these fields is a checkbox labeled "Use HTTP (Not recommended)" and a link for "Online privacy statement". At the bottom are four buttons: "Back", "Next", "Connect", and "Cancel".

To use a storage account name and key from a national Azure cloud, use the **Storage endpoints domain** drop-down menu to select **Other**, and then enter the custom storage account endpoint domain.

Access keys

Access keys provide access to the entire storage account. You're provided two access keys so you can maintain connections by using one key while regenerating the other.

Important

Store your access keys securely. We recommend regenerating your access keys regularly.

When you regenerate your access keys, you must update any Azure resources and applications that access this storage account to use the new keys. This action doesn't interrupt access to disks from your virtual machines.

7. Consider Azure File Sync

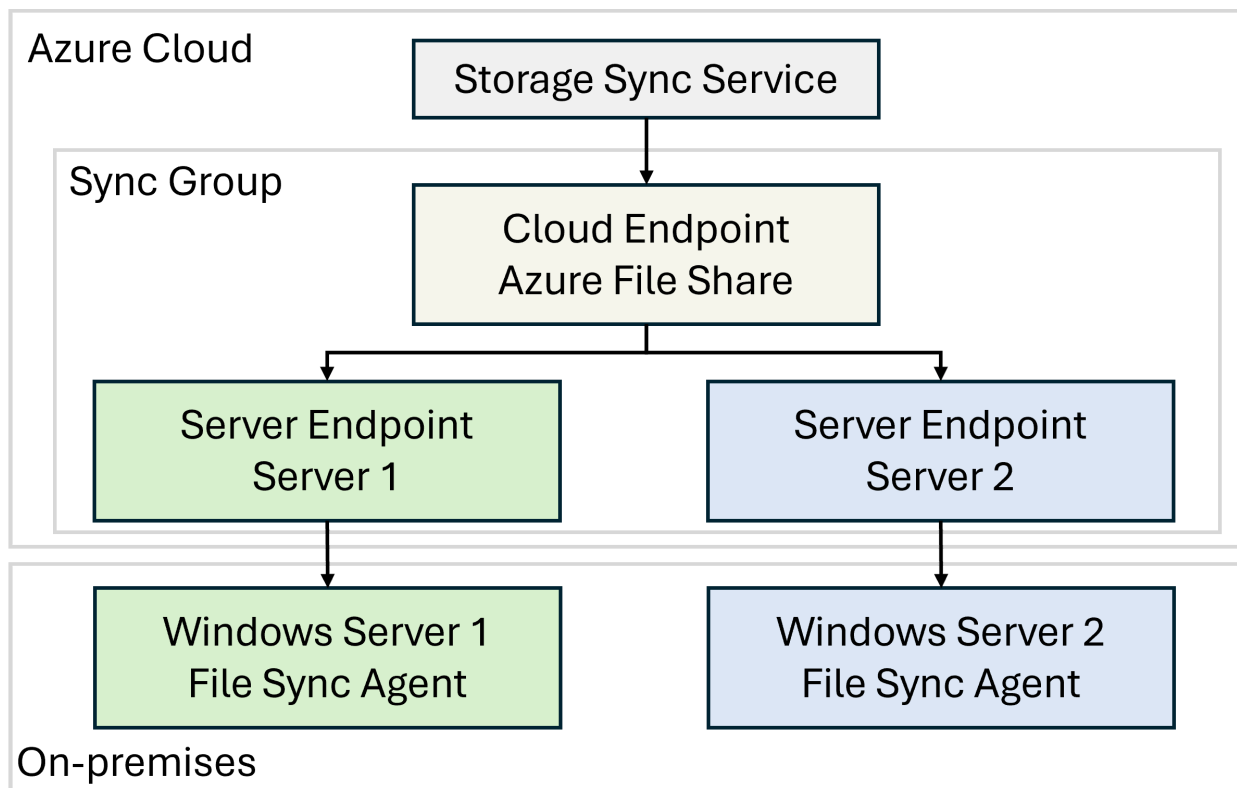
<https://learn.microsoft.com/en-us/training/modules/configure-azure-files-file-sync/7-deploy-azure-file-sync>

Consider Azure File Sync

Completed

[Azure File Sync](#) enables you to cache several Azure Files shares on an on-premises Windows Server or cloud virtual machine. You can use Azure File Sync to centralize your organization's file shares in Azure Files, while keeping the flexibility, performance, and compatibility of an on-premises file server.

Azure File Sync consists of five main components that work together to synchronize files between on-premises Windows Servers and Azure file shares.



- The **Storage Sync Service** is the primary Azure resource responsible for managing file synchronization. It can support up to 100 sync groups, operates within a single Azure region, and allows for up to 99 registered Windows Servers.
- The **sync group** establishes the synchronization setup, containing one cloud endpoint (Azure file share) and up to 50 server endpoints. Server endpoints are specific NTFS paths on registered Windows Servers, but cannot be on the system volume, and cloud tiering is not supported there.
- The **cloud endpoint** is an Azure file share that participates in the sync group. Only one cloud endpoint is allowed per sync group.
- The **server endpoint** is a path on a registered Windows Server that syncs with the cloud endpoint. The server endpoint must be an NTFS-formatted volume, and can't be a system volume.
- The **Azure File Sync Agent** is installed on each Windows Server. The agent is a background Windows service for sync operations and management tasks.

Things to know about Azure File Sync

Let's take a look at the characteristics of Azure File Sync.

- Azure File Sync transforms Windows Server into a quick cache of your Azure Files shares.
- You can use any protocol that's available on Windows Server to access your data locally with Azure File Sync, including SMB, NFS, and FTPS.

- Azure File Sync supports as many caches as you need around the world.
- There is a maximum of 100 sync groups per Storage Sync Service, and 50 server endpoints per sync group.

Things to consider when using Azure File Sync

There are many advantages to using Azure File Sync. Consider the following scenarios, and think about how you can use Azure File Sync with your Azure Files shares.

- **Consider application lift and shift.** Use Azure File Sync to move applications that require access between Azure and on-premises systems. Provide write access to the same data across Windows Servers and Azure Files.
- **Consider support for branch offices.** Support your branch offices that need to back up files by using Azure File Sync. Use the service to set up a new server that connects to Azure storage.
- **Consider backup and disaster recovery.** After you implement Azure File Sync, Azure Backup backs up your on-premises data. Restore file metadata immediately and recall data as needed for rapid disaster recovery.
- **Consider file archiving with cloud tiering.** Azure File Sync stores only recently accessed data on local servers. Implement cloud tiering so older data moves to Azure Files.

8. Knowledge check

<https://learn.microsoft.com/en-us/training/modules/configure-azure-files-file-sync/8-knowledge-check>

Knowledge check

Completed

9. Summary and resources

<https://learn.microsoft.com/en-us/training/modules/configure-azure-files-file-sync/9-summary-resources>

Summary and resources

Completed

Azure Administrators are familiar with Azure Files and the Azure File Sync agent. They know how to implement fully managed file shares in the cloud by using industry standard protocols. They understand how to use Azure File Sync to cache Azure Files shares on an on-premises Windows Server or cloud virtual machine.

In this module, you learned when to use Azure Files and how the service compares to Azure Blob Storage. You also reviewed Azure Files features such as snapshots and soft delete. You learned how Azure File Sync can be used with on-premises data stores. You also were introduced to Azure Storage Explorer.

The main takeaways for this module are:

- Azure Files provides the SMB and NFS protocols, client libraries, and a REST interface that allows access from anywhere to stored files.
- Azure Files is ideal to lift and shift an application to the cloud that already uses the native file system APIs. Share data between the app and other applications running in Azure.
- Azure Files offers two industry-standard file system protocols for mounting Azure file shares: the Server Message Block (SMB) protocol and the Network File System (NFS) protocol.
- Azure Files offers two types of file shares: standard and premium. The premium tier stores data on modern solid-state drives (SSDs), while the standard tier uses hard disk drives (HDDs).
- File share snapshots capture a point-in-time, read-only copy of your data.
- Soft delete allows you to recover your deleted file share.
- Azure Storage Explorer is a standalone application that makes it easy to work with stored data on Windows, macOS, and Linux.
- Azure File Sync enables you to cache file shares on an on-premises Windows Server or cloud virtual machine.

Learn more with Copilot

Copilot can assist you in configuring Azure infrastructure solutions. Copilot can compare, recommend, explain, and research products and services where you need more information. Open a Microsoft Edge browser and choose Copilot (top right) or navigate to copilot.microsoft.com. Take a few minutes to try these prompts and extend your learning with Copilot.

- What are Azure Files and how are they different from Azure blob storage?
- What are some common configuration and administration tasks for Azure Files?

Learn more with documentation

- [Azure Files documentation](#). This page is your starting point for all things related to Azure Files.
- [Azure File Sync documentation](#). This page is your starting point for all things related to Azure File Sync.

Learn more with self-paced training

- [Implement a hybrid file server infrastructure](#). In this module, you learn to deploy Azure File Sync and use Storage Migration Services to migrate file servers to Azure.
- [Guided Project - Azure Files and Azure Blobs](#). In this module, you practice storing business data securely by using Azure Blob Storage and Azure Files. The lab combines both learning and practical experience.