Introduction to the core Azure Storage services

04/08/2020 • 11 minutes to read • +22

In this article

The Azure Storage platform is Microsoft's cloud storage solution for modern data storage scenarios. Core storage services offer a massively scalable object store for data objects, disk storage for Azure virtual machines (VMs), a file system service for the cloud, a messaging store for reliable messaging, and a NoSQL store. The services are:

- **Durable and highly available.** Redundancy ensures that your data is safe in the event of transient hardware failures. You can also opt to replicate data across datacenters or geographical regions for additional protection from local catastrophe or natural disaster. Data replicated in this way remains highly available in the event of an unexpected outage.
- **Secure.** All data written to an Azure storage account is encrypted by the service. Azure Storage provides you with fine-grained control over who has access to your data.

- **Scalable.** Azure Storage is designed to be massively scalable to meet the data storage and performance needs of today's applications.
- **Managed.** Azure handles hardware maintenance, updates, and critical issues for you.
- **Accessible.** Data in Azure Storage is accessible from anywhere in the world over HTTP or HTTPS. Microsoft provides client libraries for Azure Storage in a variety of languages, including .NET, Java, Node.js, Python, PHP, Ruby, Go, and others, as well as a mature REST API. Azure Storage supports scripting in Azure PowerShell or Azure CLI. And the Azure portal and Azure Storage Explorer offer easy visual solutions for working with your data.

Core storage services

The Azure Storage platform includes the following data services:

- Azure Blobs: A massively scalable object store for text and binary data. Also includes support for big data analytics through Data Lake Storage Gen2.
- Azure Files: Managed file shares for cloud or on-premises deployments.
- Azure Queues: A messaging store for reliable messaging between application components.
- Azure Tables: A NoSQL store for schemaless storage of structured data.
- Azure Disks: Block-level storage volumes for Azure VMs.

Each service is accessed through a storage account. To get started, see Create a storage account.

Example scenarios

The following table compares Files, Blobs, Disks, Queues, and Tables, and shows example scenarios for each.

Feature	Description	When to use

Feature	Description	When to use
Azure Files	Offers fully managed cloud file shares that you can access from anywhere via the industry standard Server Message Block (SMB) protocol. You can mount Azure file shares from cloud or on-premises deployments of Windows, Linux, and macOS.	You want to "lift and shift" an application to the cloud that already uses the native file system APIs to share data between it and other applications running in Azure. You want to replace or supplement on-premises file servers or NAS devices. You want to store development and debugging tools that need to be accessed from many virtual machines.
Azure Blobs	Allows unstructured data to be stored and accessed at a massive scale in block blobs. Also supports Azure Data Lake Storage Gen2 for enterprise big data analytics solutions.	You want your application to support streaming and random access scenarios. You want to be able to access application data from anywhere. You want to build an enterprise data lake on Azure and perform big data analytics.
Azure Disks	Allows data to be persistently stored and accessed from an attached virtual hard disk.	You want to "lift and shift" applications that use native file system APIs to read and write data to persistent disks. You want to store data that is not required to be accessed from outside the virtual machine to which the disk is attached.
Azure Queues	Allows for asynchronous message queueing between application components.	You want to decouple application components and use asynchronous messaging to communicate between them. For guidance around when to use Queue storage versus Service Bus queues, see Storage queues and Service Bus queues - compared and contrasted.

Feature	Description	When to use
Azure Tables	Allow you to store structured NoSQL data in the cloud, providing a key/attribute store with a schemaless design.	You want to store flexible datasets like user data for web applications, address books, device information, or other types of metadata your service requires. For guidance around when to use Table storage versus the Azure Cosmos DB Table API, see Developing with Azure Cosmos DB Table API and Azure Table storage.

Blob storage

Azure Blob storage is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing massive amounts of unstructured 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, 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.

For more information about Blob storage, see Introduction to Blob storage.

Azure Files

Azure Files enables you to set up highly available network file shares that can be accessed by using the standard Server Message Block (SMB) protocol. That means that multiple VMs can share the same files with both read and write access. You can also read the files using the REST interface or the storage client libraries.

One thing that distinguishes Azure Files from files on a corporate file share is that you can access the files from anywhere in the world using a URL that points to the file and includes a shared access signature (SAS) token. You can generate SAS tokens; they allow specific access to a private asset for a specific amount of time.

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 VMs. 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.
- Resource logs, metrics, and crash dumps are just three examples of data that can be written to a file share and processed or analyzed later.

For more information about Azure Files, see Introduction to Azure Files.

Some SMB features are not applicable to the cloud. For more information, see Features not supported by the Azure File service.

Queue storage

The Azure Queue service 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 generally used to store lists of messages to be processed asynchronously.

For example, say 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 would be to use a queue. When the customer finishes their upload, write a message to the queue. Then have an Azure Function retrieve the message from the queue and create the thumbnails. Each of the parts of this processing can be scaled separately, giving you more control when tuning it for your usage.

For more information about Azure Queues, see Introduction to Queues.

Table storage

Azure Table storage is now part of Azure Cosmos DB. To see Azure Table storage documentation, see the Azure Table Storage Overview. In addition to the existing Azure Table storage service, there is a new Azure Cosmos DB Table API offering that provides throughput-optimized tables, global distribution, and automatic secondary indexes. To learn more and try out the new premium experience, see Azure Cosmos DB Table API.

For more information about Table storage, see Overview of Azure Table storage.

Disk storage

An Azure managed disk is a virtual hard disk (VHD). You can think of it like a physical disk in an on-premises server but, virtualized. Azure-managed disks are stored as page blobs, which are a random IO storage object in Azure. We call a managed disk 'managed' because it is an abstraction over page blobs, blob containers, and Azure storage accounts. With managed disks, all you have to do is provision the disk, and Azure takes care of the rest.

For more information about managed disks, see Introduction to Azure managed disks.

Types of storage accounts

Azure Storage offers several types of storage accounts. Each type supports different features and has its own pricing model. For more information about storage account types, see Azure storage account overview.

Secure access to storage accounts

Every request to Azure Storage must be authorized. Azure Storage supports the following authorization methods:

- **Azure Active Directory (Azure AD) integration for blob and queue data.** Azure Storage supports authentication and authorization with Azure AD for the Blob and Queue services via role-based access control (RBAC). Authorizing requests with Azure AD is recommended for superior security and ease of use. For more information, see Authorize access to Azure blobs and queues using Azure Active Directory.
- **Azure AD authorization over SMB for Azure Files.** Azure Files supports identity-based authorization over SMB (Server Message Block) through either Azure Active Directory Domain Services (Azure AD DS) or on-premises Active Directory Domain Services (preview). Your domain-joined Windows VMs can access Azure file shares using Azure AD credentials. For more information, see Overview of Azure Files

identity-based authentication support for SMB access and Planning for an Azure Files deployment.

- **Authorization with Shared Key.** The Azure Storage Blob, Files, Queue, and Table services support authorization with Shared Key. A client using Shared Key authorization passes a header with every request that is signed using the storage account access key. For more information, see Authorize with Shared Key.
- **Authorization using shared access signatures (SAS).** A shared access signature (SAS) is a string containing a security token that can be appended to the URI for a storage resource. The security token encapsulates constraints such as permissions and the interval of access. For more information, see Using Shared Access Signatures (SAS).
- **Anonymous access to containers and blobs.** A container and its blobs may be publicly available. When you specify that a container or blob is public, anyone can read it anonymously; no authentication is required. For more information, see Manage anonymous read access to containers and blobs.

Encryption

There are two basic kinds of encryption available for the core storage services. For more information about security and encryption, see the Azure Storage security guide.

Encryption at rest

Azure Storage encryption protects and safeguards your data to meet your organizational security and compliance commitments. Azure Storage automatically encrypts all data prior to persisting to the storage account and decrypts it prior to retrieval. The encryption, decryption, and key management processes are transparent to users. Customers can also choose to manage their own keys using Azure Key Vault. For more information, see Azure Storage encryption for data at rest.

Client-side encryption

The Azure Storage client libraries provide methods for encrypting data from the client library before sending it across the wire and decrypting the response. Data encrypted via client-side encryption is also encrypted at rest by Azure Storage. For more information about client-side encryption, see Client-side encryption with .NET for Azure Storage.

Redundancy

To ensure that your data is durable, Azure Storage stores multiple copies of your data. When you set up your storage account, you select a redundancy option. For more information, see Azure Storage redundancy.

Transfer data to and from Azure Storage

You have several options for moving data into or out of Azure Storage. Which option you choose depends on the size of your dataset and your network bandwidth. For more information, see Choose an Azure solution for data transfer.

Pricing

When making decisions about how your data is stored and accessed, you should also consider the costs involved. For more information, see Azure Storage pricing.

Storage APIs, libraries, and tools

You can access resources in a storage account by any language that can make HTTP/HTTPS requests. Additionally, the core Azure Storage services offer programming libraries for several popular languages. These libraries simplify many aspects of working with Azure Storage by handling details such as synchronous and asynchronous invocation, batching of operations, exception management, automatic retries, operational behavior, and so forth. Libraries are currently available for the following languages and platforms, with others in the pipeline:

Azure Storage data API and library references

- Azure Storage REST API
- Azure Storage client library for .NET
- Azure Storage client library for Java/Android
- Azure Storage client library for Node.js
- Azure Storage client library for Python
- Azure Storage client library for PHP
- Azure Storage client library for Ruby
- Azure Storage client library for C++

Azure Storage management API and library references

- Storage Resource Provider REST API
- Storage Resource Provider Client Library for .NET
- Storage Service Management REST API (Classic)

Azure Storage data movement API and library references

- Storage Import/Export Service REST API
- Storage Data Movement Client Library for .NET

Tools and utilities

- Azure PowerShell Cmdlets for Storage
- Azure CLI Cmdlets for Storage
- AzCopy Command-Line Utility
- Azure Storage Explorer is a free, standalone app from Microsoft that enables you to work visually with Azure Storage data on Windows, macOS, and Linux.
- Azure Storage Client Tools
- Azure Developer Tools

Next steps

To get up and running with core Azure Storage services, see Create a storage account.