

# Explore Azure Storage services

4 minutes

Azure Storage offers four data services that can be accessed by using an Azure storage account:

- **Azure Blob Storage (containers):** A massively scalable object store for text and binary data.
- **Azure Files:** Managed file shares for cloud or on-premises deployments.
- **Azure Queue Storage:** A messaging store for reliable messaging between application components.
- **Azure Table Storage:** A service that stores non-relational structured data (also known as structured NoSQL data).

Let's examine the details of these services.

## Azure Blob Storage (containers)

Azure Blob Storage is Microsoft's object storage solution for the cloud. Blob Storage is optimized for storing massive amounts of unstructured or *non-relational* 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.

### Note

You can access data from Azure Blob Storage **by using the NFS protocol**.

## 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 Cosmos DB)

Azure Table Storage is a fully managed NoSQL database service for modern app development. As a fully managed service, Azure Cosmos DB takes database administration off your hands with automatic management, updates, and patching. It also handles capacity management with cost-effective serverless and automatic scaling options that respond to application needs to match capacity with demand.

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, non-relational 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.

---

## Next unit: Determine storage account types

Continue >

How are we doing? ☆ ☆ ☆ ☆ ☆