Introduction to Azure Blob storage

06/24/2020 • 4 minutes to read • +7

In this article

Azure Blob storage is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing massive amounts of unstructured data. Unstructured data is data that doesn't adhere to a particular data model or definition, such as text or binary data.

About Blob storage

Blob storage is designed for:

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

Users or client applications can access objects in Blob storage via HTTP/HTTPS, from anywhere in the world. Objects in Blob storage are accessible via the Azure Storage REST API, Azure PowerShell, Azure CLI, or an Azure Storage client library. Client libraries are available for different languages, including:

- .NET
- Java
- Node.js
- Python
- Go
- PHP
- Ruby

About Azure Data Lake Storage Gen2

Blob storage supports Azure Data Lake Storage Gen2, Microsoft's enterprise big data analytics solution for the cloud. Azure Data Lake Storage Gen2 offers a hierarchical file system as well as the advantages of Blob storage, including:

- Low-cost, tiered storage
- High availability
- Strong consistency
- Disaster recovery capabilities

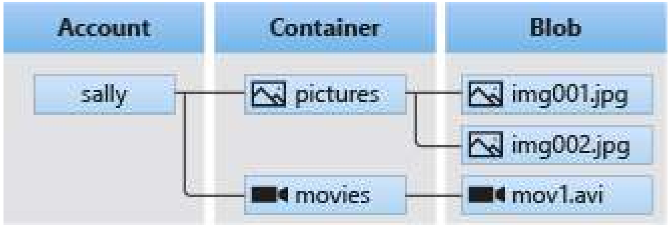
For more information about Data Lake Storage Gen2, see Introduction to Azure Data Lake Storage Gen2.

Blob storage resources

Blob storage offers three types of resources:

- The **storage account**
- A **container** in the storage account
- A **blob** in a container

The following diagram shows the relationship between these resources.



Storage accounts

A storage account provides a unique namespace in Azure for your data. Every object that you store in Azure Storage has an address that includes your unique account name. The combination of the account name and the Azure Storage blob endpoint forms the base address for the objects in your storage account.

For example, if your storage account is named *mystorageaccount*, then the default endpoint for Blob storage is:

Copy

```
http://mystorageaccount.blob.core.windows.net
```

To create a storage account, see Create a storage account. To learn more about storage accounts, see Azure storage account overview.

Containers

A container organizes a set of blobs, similar to a directory in a file system. A storage account can include an unlimited number of containers, and a container can store an unlimited number of blobs.

> **Note**
>
> The container name must be lowercase. For more information about naming containers, see **Naming and Referencing Containers, Blobs, and Metadata**.

Blobs

Azure Storage supports three types of blobs:

- **Block blobs** store text and binary data. Block blobs are made up of blocks of data that can be managed individually. Block blobs store up to about 4.75 TiB of data. Larger block blobs are available in preview, up to about 190.7 TiB
- **Append blobs** are made up of blocks like block blobs, but are optimized for append operations. Append blobs are ideal for scenarios such as logging data from virtual machines.
- **Page blobs** store random access files up to 8 TB in size. Page blobs store virtual hard drive (VHD) files and serve as disks for Azure virtual machines. For more information about page blobs, see Overview of Azure page blobs

For more information about the different types of blobs, see Understanding Block Blobs, Append Blobs, and Page Blobs.

Move data to Blob storage

A number of solutions exist for migrating existing data to Blob storage:

- **AzCopy** is an easy-to-use command-line tool for Windows and Linux that copies data to and from Blob storage, across containers, or across storage accounts. For more information about AzCopy, see Transfer data with the AzCopy v10.

- The **Azure Storage Data Movement library** is a .NET library for moving data between Azure Storage services. The AzCopy utility is built with the Data Movement library. For more information, see the reference documentation for the Data Movement library.
- **Azure Data Factory** supports copying data to and from Blob storage by using the account key, a shared access signature, a service principal, or managed identities for Azure resources. For more information, see Copy data to or from Azure Blob storage by using Azure Data Factory.
- **Blobfuse** is a virtual file system driver for Azure Blob storage. You can use blobfuse to access your existing block blob data in your Storage account through the Linux file system. For more information, see How to mount Blob storage as a file system with blobfuse.
- **Azure Data Box** service is available to transfer on-premises data to Blob storage when large datasets or network constraints make uploading data over the wire unrealistic. Depending on your data size, you can request Azure Data Box Disk, Azure Data Box, or Azure Data Box Heavy devices from Microsoft. You can then copy your data to those devices and ship them back to Microsoft to be uploaded into Blob storage.
- The **Azure Import/Export service** provides a way to import or export large amounts of data to and from your storage account using hard drives that you provide. For more information, see Use the Microsoft Azure Import/Export service to transfer data to Blob storage.

Next steps

- Create a storage account
- Scalability and performance targets for Blob storage

Is this page helpful?

👍 Yes 👎 No