**Azure Storage**

* In simple words it is Microsoft’s cloud storage. It has wide varieties of different types of cloud storage depending upon the data object to store. We can store text, binary data, files/documents, messaging contents, NoSQL etc.
* Benefits - Azure cloud storage is highly available, secure, tremendous scalable, accessible over HTTP/HTTPS etc.
* Azure Blobs, Azure Files, Azure Queues and Azure tables are the different services coming under the Azure storage services.

Microsoft Azure Blob storage is used to store the unstructured data like text or binary information.

Data which doesn’t hold any data models or definitions.

* Azure cloud provides ability to scale up this storage as need with cost effective features.
* Data stored in the Blob storage can be accessible via its URL over HTTP or HTTPS.
* We can fetch the objects from Blob storage using Azure storage REST API, Azure CLI, Azure PowerShell and by using Azure storage client library which are available in different languages like .NET, Java, Node.JS, Python, PHP and Ruby.

## Azure Storage Types

A storage account is a container that has a group of Azure Storage services together (Azure Blobs, Azure Files, Azure Queues, and Azure Tables).

Azure Storage is a fundamental building block for all of the Azure Services. We can manage them as a group. Once you create the account, we can create any data service inside it and once you delete the account, all the data will be deleted.

A storage account can be created in Azure resources.

An Azure subscription may contain multiple resource groups, where each group contains one or more storage accounts.

Within Azure, there are two types of storage accounts, five types of storage, four levels of data redundancy, and three tiers for storing files.

## What is Blob?

Blob is a service for storing large amounts of unstructured data that can be accessed from anywhere in the world via HTTP or HTTPS." Blob stands for " Binary Large Object ".

It's designed to store large amounts of unstructured text or binary data like virtual hard disks, videos, images, or even log files.

The data can be exposed to the public or stored privately. It scales up or down as your needs change.

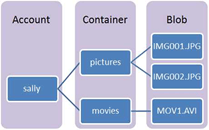
We no longer manage it, we only pay for what we use.

## Why do we use Blob?

* Store any type of unstructured data that includes images, videos, audio, documents, and backups at an exabyte scale. It handles trillions of stored objects, with millions of average requests per second, for customers around the world.
* Blob has Strong Consistency. When an object is changed, it is verified everywhere for superior data integrity, ensuring you always have access to the latest version.
* We can have the flexibility to perform the edits in storage, which can improve your application performance and reduce bandwidth consumption.
* We have many different types of blobs for our flexibility. Automatically we configure Geo-replication options in a single menu, to easily empower global and local access.
* One infrastructure but access worldwide. With regions around the world, it is ideal for streaming and storing media, whether it is live broadcast events or a long-term archive of petabytes of movies and television shows. We can perform secure backup and disaster recovery.

## Blob Service Hierarchy

Every blob service has the following components,



* **Storage Account:** This storage account can be either a General Storage account (V1 or V2) or a Blob Storage Account.
* **Container:**A container contains a group of blobs in which there can exist an unlimited amount of blobs. A mandatory requirement of a container is that its name should always be lowercase.
* **Blob:**A blob is a file of any size and type.

## What is a Container?

A container is a basic structural element for your storage account. It organizes a set of blobs, similar to a directory in a file system. Normally, a storage account can have an unlimited number of containers, and a container can store an unlimited number of blobs. So we first create a storage account then create a container and store our files.

## Types of Blob

We specify the blob type when we create the blob. Once the blob has been created, we cannot change its type, and it can be updated only by using operations appropriate for that blob type, i.e., writing a block or list of blocks to a block blob, appending blocks to an append blob, and writing pages to a page blob.

Blobs can be further sub-categorized into three types.

* **Block Blobs:** The most common use cases for blob storage will involve Block Blobs. These blobs are ideal for storing documents and text or binary media files. 50,000 blocks of up to 100 MB each can be stored in a single block blob which totals a size of more than 4750 GB or 4.75 TB. (50000\*100MB). It stores text and binary data, up to about 4.7 TB. The block blobs mean that a large file may be broken up into blocks, then may be uploaded or downloaded separately, in any sequence and then re-associated with each other, in the proper sequence. Each block within a blob is identified by a Block ID. With a block blob, you can upload multiple blocks in parallel to decrease upload time.
* **Append blobs:**Append blobs are similar to Block blobs, but support appending operations and are designed for log files. When you modify an append blob, blocks are added to the end of the blob only, via the append block operation. The difference between append blobs and block blobs is their storage capacity. This blob can only store up to 4MB of data, unlike 100MB in block blobs. Therefore, append blocks are limited to the storage capacity.
* **Page Blobs:** Page blobs are designed for frequent read/write operations. It can store about 8 TB of data. It writes a range of pages to an existing page blob available. It will not create a new blob. Within Azure, there are two-page blob categories like Standard, used for virtual machines with an average amount of read/write operations, and Premium, used for virtual machines for intensive read/write operations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Block Blob | Page Blob | Append Blob |  |  |
| Collections of blocks and each Block is identifiable by a Block ID | Collection of pages that are optimized for random read-write operations. | Optimized for Append operations |  |  |
| Used for streaming Sequential Data like Video | Used for non-sequential Read and Write. | Used for activities like Logging |  |  |
| Each Block can be up to 4 MB | A page can be up to 512 bytes | Each Block can be up to 4 MB |  |  |
| Up to 50,000 Blocks can be created. | No limitations on the number of Pages created | Up to 50,000 Blocks can be created. |  |  |
| Maximum Size 195 GB | Maximum size 1TB | Maximum size of 195 GB |  |  |
| Multiple clients writing to the same blob is not possible | Multiple clients writing to the same blob is not possible | Multiple clients writing to the same blob is possible |  |  |

## Differentiate blob storage types

## Public access Level

By default, a container and any blobs may be accessed only by the owner of the storage account. To give anonymous users read permissions to a container and its blobs, you can set the container permissions to allow public access. We can configure a container with the following permissions:

* **Private:**This is the default for all new containers. The container and its blobs can be accessed only by the storage account owner
* **Blob:**Blobs within the container can be read by an anonymous user, but container data is not available.
* **Container:** All container and blob data can be read by anonymous request.

| **Recommendation** | **Benefit** |
| --- | --- |
| [Pack small files into larger files](https://learn.microsoft.com/en-us/azure/storage/blobs/access-tiers-best-practices" \l "pack-small-files-before-moving-data-to-cooler-tiers) before moving them to cooler tiers. You can use file formats such as TAR or ZIP. | Cooler tiers have higher data transfer costs. By having fewer large files, you can reduce the number of operations required to transfer data. |
| Use standard-priority rehydration when rehydrating blobs from archive storage. Use high-priority rehydration only for emergency data restoration situations. For more information, see [Rehydrate an archived blob to an online tier](https://learn.microsoft.com/en-us/azure/storage/blobs/archive-rehydrate-to-online-tier) | High-priority rehydration from the archive tier can lead to higher-than-normal bills. |
| Reduce the cost of using resource logs by choosing the appropriate log storage location and by managing log-retention periods. If you only plan to query logs occasionally (for example, querying logs for compliance auditing), consider sending resource logs to a storage account instead of sending them to an Azure Monitor Logs workspace. You can use a serverless query solution such as Azure Synapse Analytics to analyze logs. For more information, see [Optimize cost for infrequent queries](https://learn.microsoft.com/en-us/azure/storage/blobs/blob-storage-monitoring-scenarios" \l "optimize-cost-for-infrequent-queries). Use lifecycle management policies to delete or archive logs. | Storing resource logs in a storage account for later analysis can be a cheaper option. Using lifecycle management policies to manage log retention in a storage account prevents large numbers of logs files building up over time, which can lead to unnecessary capacity charges. |
| If you enable versioning, use a lifecycle management policy to automatically delete old blob versions. | Every write operation to a blob creates a new version. This increases capacity costs. You can keep costs in check by removing versions that you no longer need. |
| If you enable versioning, then place blobs that are frequently overwritten into an account that doesn't have versioning enabled. | Every time a blob is overwritten, a new version is added which leads to increased storage capacity charges. To reduce capacity charges, store frequently overwritten data in a separate storage account with versioning disabled. |
| If you enable soft delete, then place blobs that are frequently overwritten into an account that doesn't have soft delete enabled. Set retention periods. Consider starting with a short retention period to better understand how the feature affects your bill. The minimum recommended retention period is seven days. | Every time a blob is overwritten, a new snapshot is created. The cause of increased capacity charges might be difficult to access because the creation of these snapshots doesn't appear in logs. To reduce capacity charges, store frequently overwritten data in a separate storage account with soft delete disabled. A retention period keeps soft-deleted blobs from piling up and adding to the cost of capacity. |
| Enable SFTP support only when it's used to transfer data. | Enabling the SFTP endpoint incurs an hourly cost. By thoughtfully disabling SFTP support, and then enabling it as needed, you can avoid passive charges from accruing in your account. |
| Disable any encryption scopes that aren't needed to avoid unnecessary charges. | Encryptions scopes incur a per month charge. |