**Azure storage accounts**

* A storage account provides a unique namespace for your Azure Storage data that's accessible from anywhere in the world over HTTP or HTTPS.
* Data in this account is secure, highly available, durable, and massively scalable.
* When you create your storage account, you’ll start by picking the storage account type.
* The type of account determines the storage services and redundancy options and has an impact on the use cases. Below is a list of redundancy options that will be covered later in this module:
* Locally redundant storage (LRS)
* Geo-redundant storage (GRS)
* Read-access geo-redundant storage (RA-GRS)
* Zone-redundant storage (ZRS)
* Geo-zone-redundant storage (GZRS)
* Read-access geo-zone-redundant storage (RA-GZRS)

# Azure 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 Disks**: Block-level storage volumes for Azure VMs.
* **Azure Tables:** NoSQL table option for structured, non-relational data.

## Benefits of Azure Storage

Azure Storage services offer the following benefits for application developers and IT professionals:

* **Durable and highly available**. Redundancy ensures that your data is safe if transient hardware failures occur. You can also opt to replicate data across data centers or geographical regions for additional protection from local catastrophes or natural disasters. Data replicated in this way remains highly available if an unexpected outage occurs.
* **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.

## Storage account endpoints

* One of the benefits of using an Azure Storage Account is having a unique namespace in Azure for your data.
* In order to do this, every storage account in Azure must have a unique-in-Azure account name.
* The combination of the account name and the Azure Storage service endpoint forms the endpoints for your storage account.

| **Storage service** | **Endpoint** |
| --- | --- |
| Blob Storage | https://<storage-account-name>.blob.core.windows.net |
| Data Lake Storage Gen2 | https://<storage-account-name>.dfs.core.windows.net |
| Azure Files | https://<storage-account-name>.file.core.windows.net |
| Queue Storage | https://<storage-account-name>.queue.core.windows.net |
| Table Storage | https://<storage-account-name>.table.core.windows.net |

## **Azure Blobs**

* Azure Blob storage is an object storage solution for the cloud. It can store massive amounts of data, such as text or binary data.
* Azure Blob storage is unstructured, meaning that there are no restrictions on the kinds of data it can hold.
* Blob storage can manage thousands of simultaneous uploads, massive amounts of video data, constantly growing log files, and can be reached from anywhere with an internet connection.

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.

### **Accessing blob storage**

* 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.

### **Blob storage tiers**

Azure Storage offers different access tiers for your blob storage, helping you store object data in the most cost-effective manner. The available access tiers include:

* **Hot access tier**: Optimized for storing data that is accessed frequently (for example, images for your website).
* **Cool access tier**: Optimized for data that is infrequently accessed and stored for at least 30 days (for example, invoices for your customers).
* **Cold access tier**: Optimized for storing data that is infrequently accessed and stored for at least 90 days.
* **Archive access tier**: Appropriate for data that is rarely accessed and stored for at least 180 days, with flexible latency requirements (for example, long-term backups).

**Azure Files**

* Azure File storage offers fully managed file shares in the cloud that are accessible via the industry standard Server Message Block (SMB) or Network File System (NFS) protocols.
* SMB Azure file shares are accessible from Windows, Linux, and macOS clients.
* NFS Azure Files shares are accessible from Linux or macOS clients.

### **Azure Files key benefits:**

* **Shared access**: Azure file shares support the industry standard SMB and NFS protocols, meaning you can seamlessly replace your on-premises file shares with Azure file shares without worrying about application compatibility.
* **Fully managed**: Azure file shares can be created without the need to manage hardware or an OS. This means you don't have to deal with patching the server OS with critical security upgrades or replacing faulty hard disks.
* **Scripting and tooling**: PowerShell cmdlets and Azure CLI can be used to create, mount, and manage Azure file shares as part of the administration of Azure applications. You can create and manage Azure file shares using Azure portal and Azure Storage Explorer.
* **Resiliency**: Azure Files has been built from the ground up to always be available. Replacing on-premises file shares with Azure Files means you don't have to wake up in the middle of the night to deal with local power outages or network issues.
* **Familiar programmability**: Applications running in Azure can access data in the share via file system I/O APIs. Developers can therefore use their existing code and skills to migrate existing applications. In addition to System IO APIs, you can use Azure Storage Client Libraries or the Azure Storage REST API.

**Azure Queues**

* Azure Queue storage is a service for storing large numbers of messages. Once stored, you can access the messages from anywhere in the world via authenticated calls using HTTP or HTTPS.
* A queue can contain as many messages as your storage account has room for (potentially millions).
* Each individual message can be up to 64 KB in size.
* Queues are commonly used to create a backlog of work to process asynchronously.
* Queue storage can be combined with compute functions like Azure Functions to take an action when a message is received.
* For example, you want to perform an action after a customer uploads a form to your website. You could have the submit button on the website trigger a message to the Queue storage. Then, you could use Azure Functions to trigger an action once the message was received.

**Azure Tables**

* Azure Table storage stores large amounts of structured data.
* Azure tables are ideal for storing structured, non-relational data.
* Azure tables are a NoSQL datastore that accepts authenticated calls from inside and outside the Azure cloud.

## Storage account settings

A storage account defines a policy that applies to all the storage services in the account. For example, you could specify that all the contained services will be stored in the West US datacenter, accessible only over https, and billed to the sales department's subscription.

A storage account defines the following settings:

* **Subscription**: The Azure subscription that's billed for the services in the account.
* **Location**: The datacenter that stores the services in the account.
* **Performance**: Determines the data services you can have in your storage account and the type of hardware disks used to store the data.
  + **Standard** allows you to have any data service (Blob, File, Queue, Table) and uses magnetic disk drives.
  + **Premium** provides more services for storing data. For example, storing unstructured object data as block blobs or append blobs, and specialized file storage used to store and create premium file shares. These storage accounts use solid-state drives (SSD) for storage.
* **Replication**: Determines the strategy used to make copies of your data to protect against hardware failure or natural disaster. At a minimum, Azure automatically maintains three copies of your data within the datacenter associated with the storage account. The minimum replication is called locally redundant storage (LRS), and guards against hardware failure but doesn't protect you from an event that incapacitates the entire datacenter. You can upgrade to one of the other options such as geo-redundant storage (GRS) to get replication at different datacenters across the world.
* **Access tier**: Controls how quickly you're able to access the blobs in a storage account. The Hot access tier is optimized for storing data that's accessed or modified frequently and gives quicker access than Cool, but at increased storage cost. The Cool access tier is optimized for storing data that's infrequently accessed or modified, and has a lower storage cost. Hot access tier applies only to blobs, and serves as the default value for new blobs.
* **Secure transfer required**: A security feature that determines the supported protocols for access. Enabled requires HTTPS, while disabled allows HTTP.
* **Virtual networks**: A security feature that allows inbound access requests only from the virtual network(s) you specify.