**What is the use of storage account in Azure?**

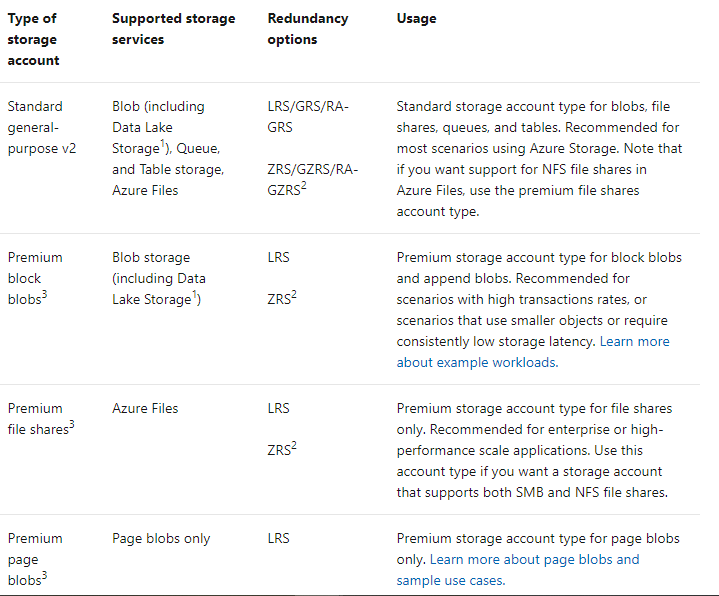
What is Azure Storage?

* Azure Storage is the Azure platform's managed service for providing cloud storage. Azure Storage is composed of several core services and supporting features. It is highly available, secure, durable, scalable, and redundant.
* The word ‘Blob’ expands to a Binary Large Object. Blobs include images, text files, videos and audios. Azure Blob storage is Microsoft's object storage solution for the cloud and it 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.
* With Azure Blob Storage, the files (photos, videos, training documents, etc.), which are known as blobs, are put in containers which function similar to directories. These are then linked to the storage account.
* Azure Files offers fully managed file shares in the cloud that are accessible via the industry standard Server Message Block (SMB) protocol or Network File System (NFS).
* Azure file storage mainly can be used if we want to have a shared drive between two servers or across users. In that case, we will go for Azure file storage.
* We can create an unlimited number of file shares within a storage account. Once we create a file share, then we can create directories, just like folders, and then we can upload files into it. Once we create a file share, we can mount that on any virtual machine, whether it is in Azure or outside.

An **Azure storage account** contains all of your **Azure Storage** data objects: blobs, file shares, queues, tables, and disks.

The data in azure storage account is durable, highly available, secure and scalable.

The **storage account** provides a unique namespace for your **Azure Storage** data that's accessible from anywhere in the world over HTTP or HTTPS.



**What is Azure 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. Unstructured data is data that doesn't adhere to a particular data model or definition, such as text or binary data.

**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](https://docs.microsoft.com/en-us/rest/api/storageservices/blob-service-rest-api), [Azure PowerShell](https://docs.microsoft.com/en-us/powershell/module/az.storage), [Azure CLI](https://docs.microsoft.com/en-us/cli/azure/storage), or an Azure Storage client library. Client libraries are available for different languages, including:

* [.NET](https://docs.microsoft.com/en-us/dotnet/api/overview/azure/storage)
* [Java](https://docs.microsoft.com/en-us/java/api/overview/azure/storage)
* [Node.js](https://github.com/Azure/azure-sdk-for-js/tree/master/sdk/storage)
* [Python](https://docs.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-python)
* [Go](https://github.com/azure/azure-storage-blob-go/)
* [PHP](https://azure.github.io/azure-storage-php/)
* [Ruby](https://azure.github.io/azure-storage-ruby)

**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

**Azure File Service**

The **Azure File Service** lets you access **files** in the **Azure** cloud just like you do on-premises, using the standard SMB protocol.

This lets you build **file** servers on the cloud, lift-and-shift applications directly from an on-premises environment to **Azure**, and enable cloud **file** sharing between multiple applications.

This service can be used for creating files shares on the cloud.

Here the file shares can be accessed via the SMB protocol (Server message block).

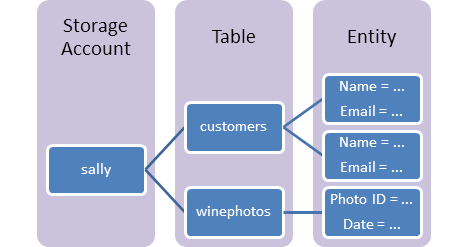
You can mount these files shared from cloud-based or on–premise machines.

The machines could be running on windows.Linux ,macOS.

**Azure Table storage**

**Azure Table storage** is a service that stores non-relational structured data (also known as structured NoSQL data) in the cloud, providing a key/attribute store with a schemaless design.

Because **Table storage** is schemaless, it's easy to adapt your data as the needs of your application evolve.



**Azure Queue**

Azure Queue Storage is a service for storing large numbers of messages.

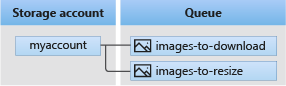
You access messages from anywhere in the world via authenticated calls using HTTP or HTTPS.

A queue message can be up to 64 KB in size. A queue may contain millions of messages, up to the total capacity limit of a storage account.

Queues are commonly used to create a backlog of work to process asynchronously.

## Queue Storage concepts

Queue Storage contains the following components:



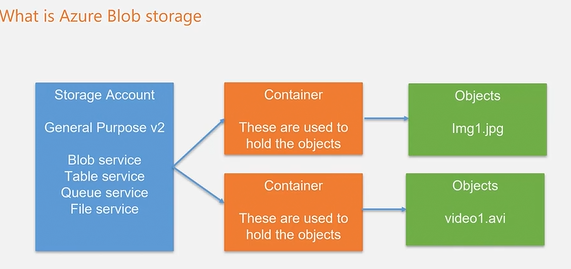
* **URL format:** Queues are addressable using the following URL format:

https://<storage account>.queue.core.windows.net/<queue>

The following URL addresses a queue in the diagram:

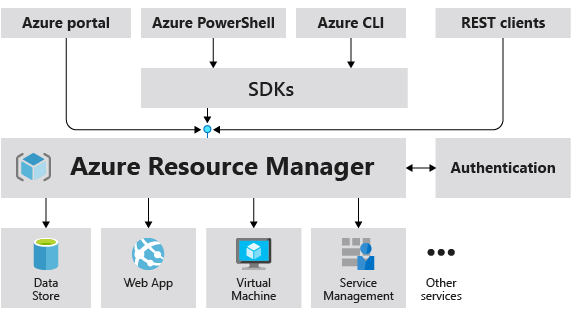
https://myaccount.queue.core.windows.net/images-to-download

* **Storage account:** All access to Azure Storage is done through a storage account. For information about storage account capacity, see [Scalability and performance targets for standard storage accounts](https://docs.microsoft.com/en-us/azure/storage/common/scalability-targets-standard-account?toc=/azure/storage/queues/toc.json).
* **Queue:** A queue contains a set of messages. The queue name **must** be all lowercase. For information on naming queues, see [Naming queues and metadata](https://docs.microsoft.com/en-us/rest/api/storageservices/naming-queues-and-metadata).
* **Message:** A message, in any format, of up to 64 KB. Before version 2017-07-29, the maximum time-to-live allowed is seven days. For version 2017-07-29 or later, the maximum time-to-live can be any positive number, or -1 indicating that the message doesn't expire. If this parameter is omitted, the default time-to-live is seven days.



**Azure Resources Groups**

**Azure Resources Groups** are logical collections of virtual machines, storage accounts, virtual networks, web apps, databases, and/or database servers. Typically, users will **group** related **resources** for an application, divided into **groups** for production and non-production — but you can subdivide further as needed



* **resource** - A manageable item that is available through Azure. Virtual machines, storage accounts, web apps, databases, and virtual networks are examples of resources. Resource groups, subscriptions, management groups, and tags are also examples of resources.
* **resource group** - A container that holds related resources for an Azure solution. The resource group includes those resources that you want to manage as a group. You decide which resources belong in a resource group based on what makes the most sense for your organization.
* **resource provider** - A service that supplies Azure resources. For example, a common resource provider is Microsoft.Compute, which supplies the virtual machine resource. Microsoft.Storage is another common resource provider.
* **Resource Manager template** - A JavaScript Object Notation (JSON) file that defines one or more resources to deploy to a resource group, subscription, management group, or tenant. The template can be used to deploy the resources consistently and repeatedly.
* **declarative syntax** - Syntax that lets you state "Here is what I intend to create" without having to write the sequence of programming commands to create it. The Resource Manager template is an example of declarative syntax. In the file, you define the properties for the infrastructure to deploy to Azure.

## Resource groups

There are some important factors to consider when defining your resource group:

* All the resources in your resource group should share the same lifecycle. You deploy, update, and delete them together. If one resource, such as a server, needs to exist on a different deployment cycle it should be in another resource group.
* Each resource can exist in only one resource group.
* You can add or remove a resource to a resource group at any time.
* You can move a resource from one resource group to another group. For more information.
* The resources in a resource group can be located in different regions than the resource group.
* When creating a resource group, you need to provide a location for that resource group. You may be wondering, "Why does a resource group need a location? And, if the resources can have different locations than the resource group, why does the resource group location matter at all?" The resource group stores metadata about the resources. When you specify a location for the resource group, you're specifying where that metadata is stored. For compliance reasons, you may need to ensure that your data is stored in a particular region.

If the resource group's region is temporarily unavailable, you can't update resources in the resource group because the metadata is unavailable. The resources in other regions will still function as expected, but you can't update them. For more information about building reliable applications

A resource group can be used to scope access control for administrative actions. To manage a resource group, you can assign.  
You can [apply tags](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/tag-resources) to a resource group. The resources in the resource group don't inherit those tags.

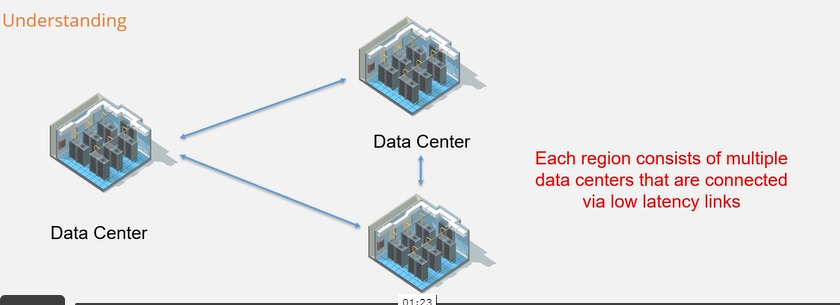
* A resource can connect to resources in other resource groups. This scenario is common when the two resources are related but don't share the same lifecycle. For example, you can have a web app that connects to a database in a different resource group.
* When you delete a resource group, all resources in the resource group are also deleted. For information about how Azure Resource Manager orchestrates those deletions, see [Azure Resource Manager resource group and resource deletion](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/delete-resource-group).
* You can deploy up to 800 instances of a resource type in each resource group. Some resource types are [exempt from the 800 instance limit](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/resources-without-resource-group-limit). For more information, see [resource group limits](https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/azure-subscription-service-limits#resource-group-limits).
* Some resources can exist outside of a resource group. These resources are deployed to the [subscription](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/deploy-to-subscription), [management group](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/deploy-to-management-group), or [tenant](https://docs.microsoft.com/en-us/azure/azure-resource-manager/templates/deploy-to-tenant). Only specific resource types are supported at these scopes.

**What is Region and Availability Zone?**

Azure has regions across the world which can be used to host resources.

These are different geographic locations situated across the world.

You can host resources in Azure in any one of the available regions.



Availability Zones:

* These are physically separate locations within an Azure region.
* Each Availability zone is made up of one or more datacentres.
* Each is equipped with independent power, cooling and networking.
* We can use availability zones to run mission-critical applications with high availability and low-latency replication.