

**Lab Manual- Setup and Manage Azure Storage with Blob**

**Prepared for**:

**Date:** 18th Nov 2021

**Prepared by:**

Document Name: Lab Manual **Document Number** AZLabn989

**Contributor:**

Contents

[1. Introduction 2](#_Toc99964002)

[2. Blob Storage 2](#_Toc99964003)

[3. Lab: Creating my First Azure Storage Account 4](#_Toc99964004)

[4. Lab: Creating Blob Container 13](#_Toc99964005)

[5. Lab: Upload Blob (File) 14](#_Toc99964006)

[6. Delete a storage account 16](#_Toc99964007)

[7. Conclusion 17](#_Toc99964008)

## Introduction

The storage account is one of the fundamental pieces of cloud technology. It contains all of our Azure Storage data objects: blobs, files, queues, tables, and disks. Microsoft offers several options to store data on the cloud. Storage can be used for two purposes; one is for actually storing files just as we do on a hard disk. So it becomes where we put our backup files, where we put our database files. In addition, we can store our jpeg or videos in a storage account. The other is as the back end to a Virtual Machine. So actually, many services within Azure need a storage account, sometimes for logging and sometimes as a fundamental piece of the technology. By default, Virtual Machine is run on a managed storage account, which is an abstraction on top of a storage account. We cannot use it to create a Virtual Machine on top of an unmanaged account.

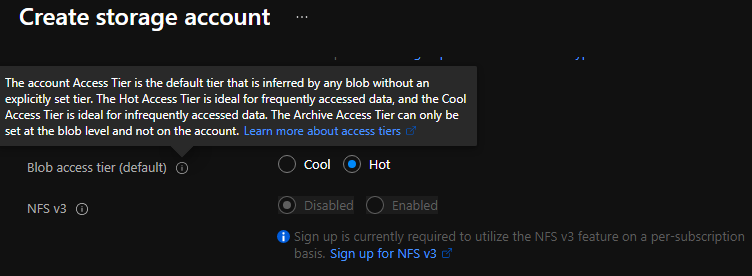
In this article, we will understand how to create an Azure blob storage account and upload a file on the blob.

There are various options available in the Azure Storage Account for storing user data.

## Blob Storage

The word blob is an acronym, which stands for a binary large object. Blobs typically include large files that are unstructured, such as images, video, music files, backup files, etc.

Blob storage can be divided into two access tiers,



**Hot access tier**

Those data, which are accessed frequently, will come under this tier. In addition, the hot storage tier is highly available compared to the cold access tier, like 99.9% as opposed to the 99% of the cold storage tier.

**Cold access tier**

In this tier, data that is not accessed very often will relay. In addition, the cold access tier is cheaper than the hot access tier and as such, you can store more data at a lower cost.

**Note**

We can switch between access tiers at any point if we wish to do so.

**File Storage**

With the help of Azure Files, we can set up highly available network file shares that can be accessed by using the standard Server Message Block (SMB) protocol. That means that multiple VMs can share the same files with both read and write access. We can also read the files using the REST interface or the storage client libraries.

There are multiple common scenarios where Azure File storage is very useful,

1. Store configuration files on a file share and access from multiple VMs.
2. We can keep resource logs, metrics, and crash dumps on a file share and later can be processed or used for analysis.

**Queue Storage**

Queue Storage is somewhat like MSMQ. It allows us to decouple our components and have reliable asynchronous communication. In Azure Queue Storage, the number of queues is only limited by the capacity of the storage account. The Azure Queue service is used to store and retrieve messages. Queues and messages can be created programmatically or using the Storage Explorer tool.

**Table Storage**

Table storage is used to store semi-structured data in a key-value format in a NoSQL datastore. Azure table storage can store petabytes of data, can scale, and is inexpensive. Table storage can be accessed using REST and some of the OData protocols or using the Storage Explorer tool. Azure Table storage is now part of Azure Cosmos DB.

**Disk**

An Azure managed disk is a virtual hard disk (VHD). It is like a physical disk in an on-premises server but virtualized. Azure-managed disks are stored as page blobs, which are random IO storage objects in Azure. We call a managed disk 'managed' because it is an abstraction over page blobs, blob containers, and Azure storage accounts.

Azure offers two types of disk storage,

1. **Managed -** Managed disk has some advantages over unmanaged disks in the sense that disks will be created and managed for us. This is an IaaS offering.
2. **Unmanaged**- With unmanaged disks, we must manage them our self. This means, our virtual hard disks are stored in a storage account as page blobs.

Moreover, Azure offers two types of disks,

1. **Premium -** Which means our data will be placed on solid-state disks
2. **Standard**- Where data is placed on regular hard disk drives

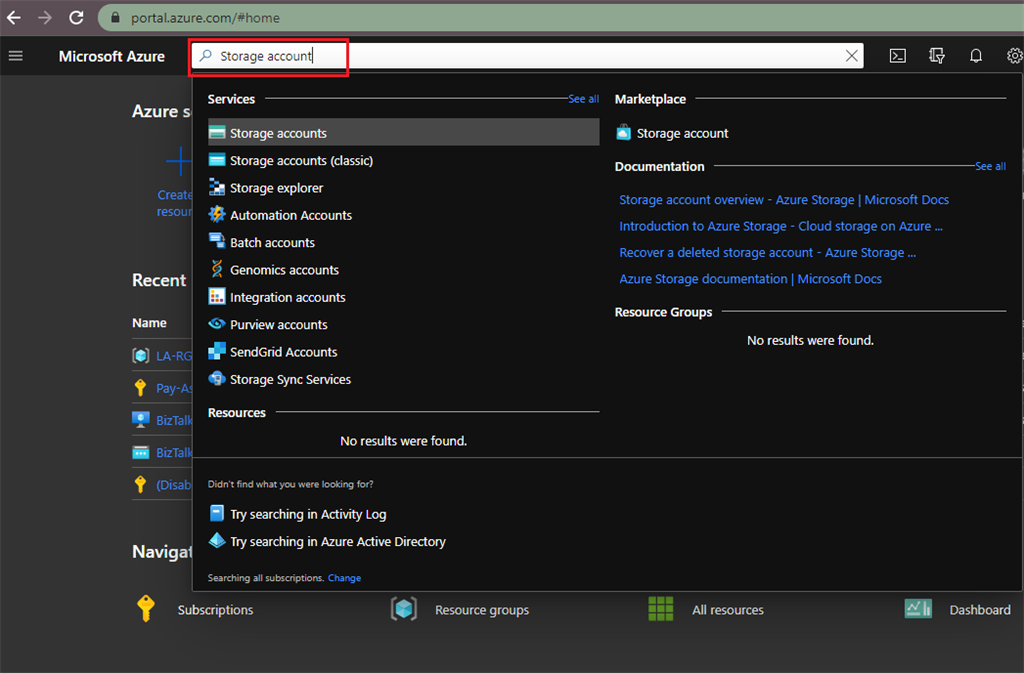
## Lab: Creating my First Azure Storage Account

There are multiple ways to create a storage account,

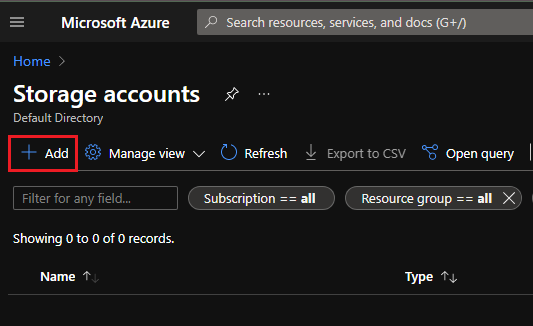
1. Azure portal
2. Azure PowerShell
3. Azure CLI
4. Azure Resource Manager (ARM) Template

In this article, we will use the first option i.e. using the Azure portal. For other options, we will understand in different articles.

* Log in to the [Azure portal](https://portal.azure.com/) ➜ Search Storage Accounts ➜ then select Storage accounts from the search result



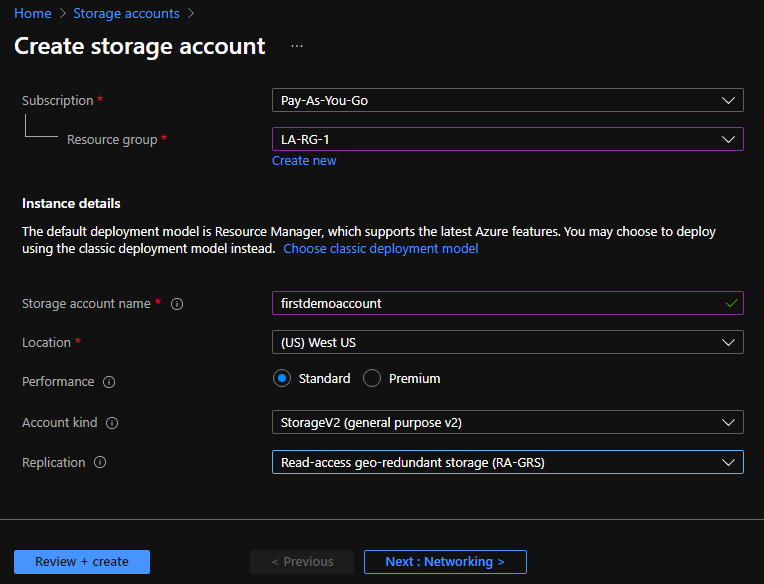
* It will open a Storage account, now we have to click on Add,



Now we have to fill in the mandatory details under the Basic section, like,

1. Select the Subscription from the dropdown.
2. Choose the Resource Group under which this storage account will be created.
3. Provide the Storage account name, it should be unique across Azure and the length must be between 3 and 24 characters. Also, it may include only numbers and lowercase letters.
4. Select the location of the Storage account or use the default location.
5. Select a performance tier. The default tier is Standard.
6. Set the Account kind field to Storage V2 (general-purpose v2).
7. Select replication value, the default replication option is Read-access geo-redundant storage (RA-GRS). It will specify how the storage account will be replicated.

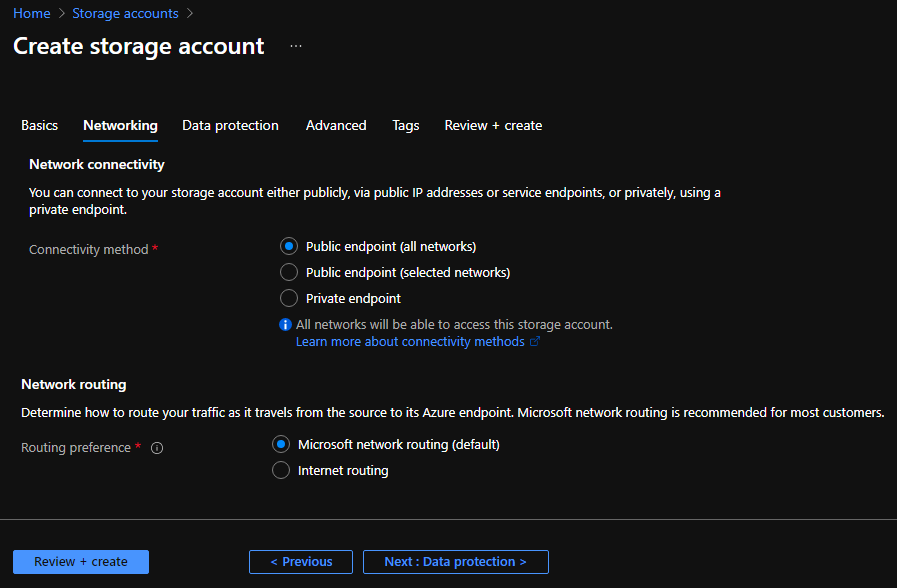
After entering all the values, it will look like below. If you want to keep the default values for the next section then click on 'Review + create', otherwise click on 'Next: Networking >'



* Now we are in the networking section, and here we can change the configuration about Network connectivity and Network routing.

In our case, we will keep the default as it is and not going to change any configuration.

For additional options, we will click on 'Next: Data protection >'.



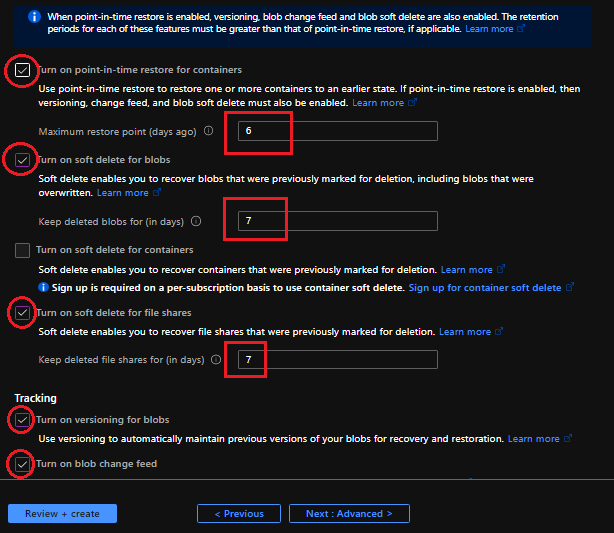
* In the Data protection section, we can define multiple configurations like if want to turn on point-in-time restoration for containers. With help of this configuration, we can restore one or more containers to an earlier state. We have to keep in mind one point here if we are going to enable this 'Point-in-time restore' configuration, then versioning, change feed and blob soft delete must be enabled. The maximum restore point can be 6 days ago.

Next configuration, Turn on soft delete for blobs. With this configuration, we can recover blobs that were previously marked for deletion. Here we can define the number of days to keep deleted blobs.

Similarly, we have other configurations like,

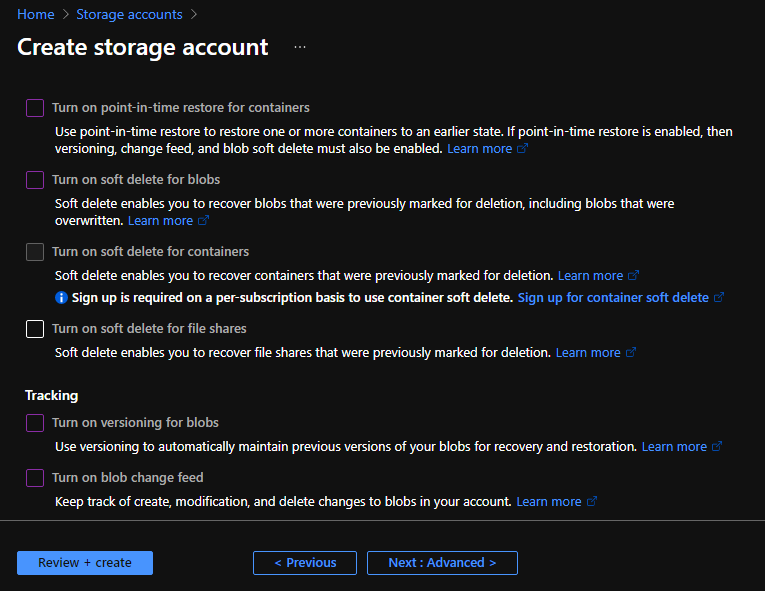
1. Turn on soft delete for containers
2. Turn on soft delete for file shares
3. Turn on versioning for blobs
4. Turn on blob change feed

If you change these setting then it will look like below.



For our scenario, we will keep it as default and not going to change any configuration.

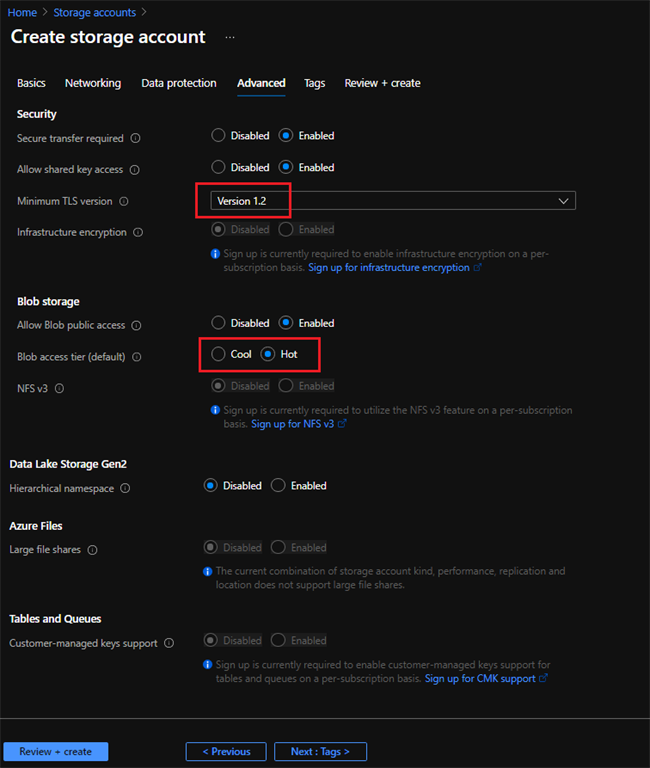
For additional options, we will click on 'Next: Advanced >'.



* In the Advanced section, we can define the Security, Blob storage, Data Lake Storage Gen2, Azure Files & Table, and Queues configuration.

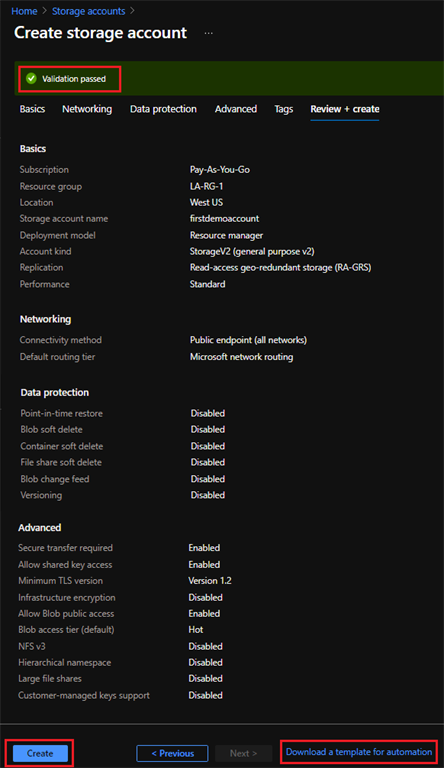
Please check the configuration in the below image, for our scenario I am keeping the default values.

The next option is Tags; we will bypass this section because here you can just define the tags and their values. Now we will click on 'Review + create'.

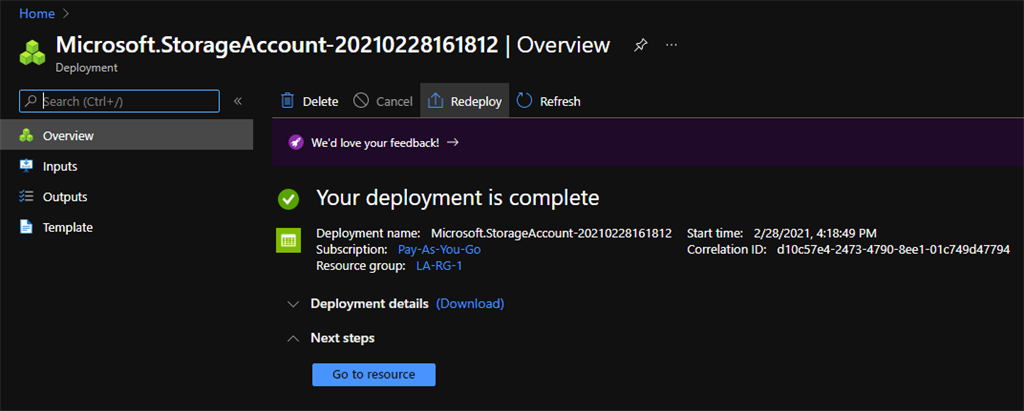


* Once we click on 'Review + Create', it will review your storage account settings and give the result of the review. In case all configurations are properly defined, we will get the message "Validation Passed". In addition 'Create' button will appear. Clicking on that button will create the Storage account. In case of validation failed then it will give you the reason why and how to fix that issue.

Here, we will get a link to download a template that we can use for automation.

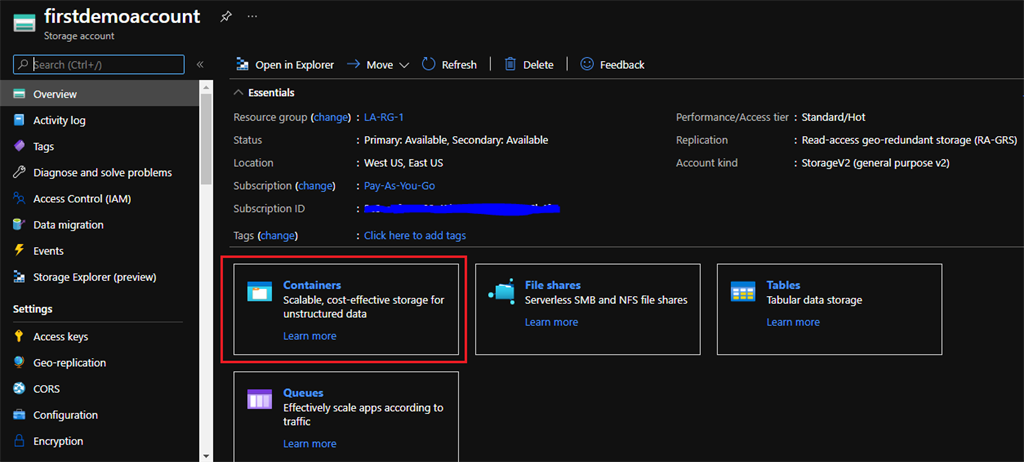


* It will take hardly a minute or two to complete the deployment. Once deployment completed below screen will come, then we can click on 'Go to resource'

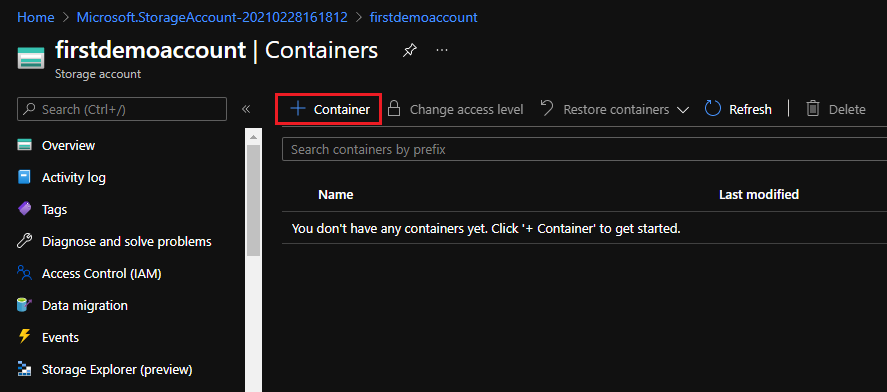


## Lab: Creating Blob Container

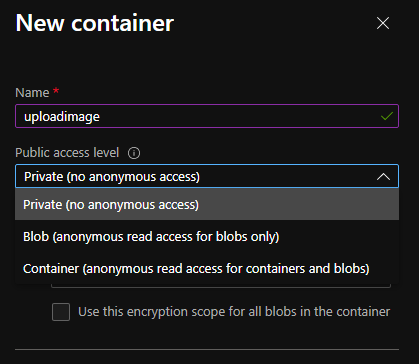
* It will open our newly created Storage account. Once it opens then we have to create a container where we can upload the file. Click on '**Containers**' and it will open to create a new container,



* Click on '+ Container' where we will upload the sample file.

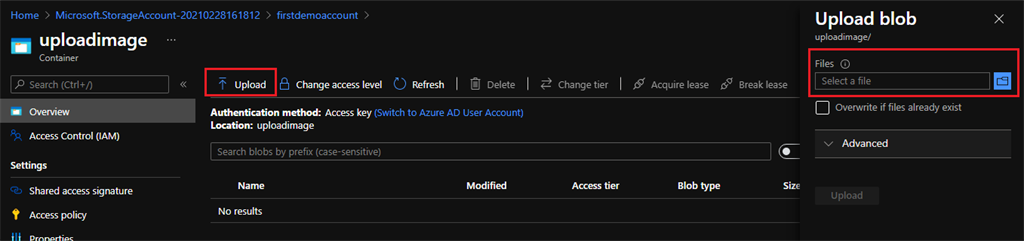


* Enter the name of the container and select the Public access level based on our need. For our scenario, we will keep this configuration as it is.

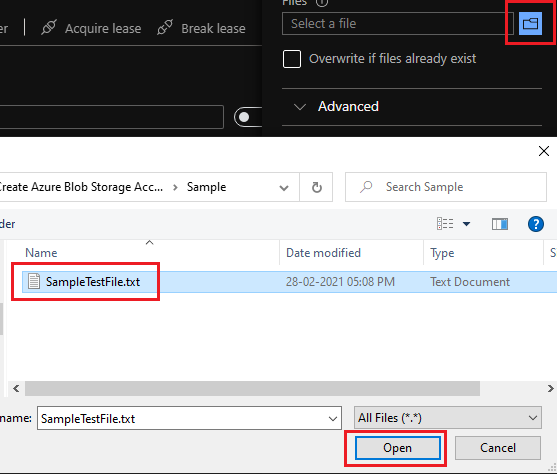


## Lab: Upload Blob (File)

* Click on **upload** and it will open a new panel where we will see a placeholder to select a file. In addition, we can see a checkbox that will be helpful to overwrite the already existing file.

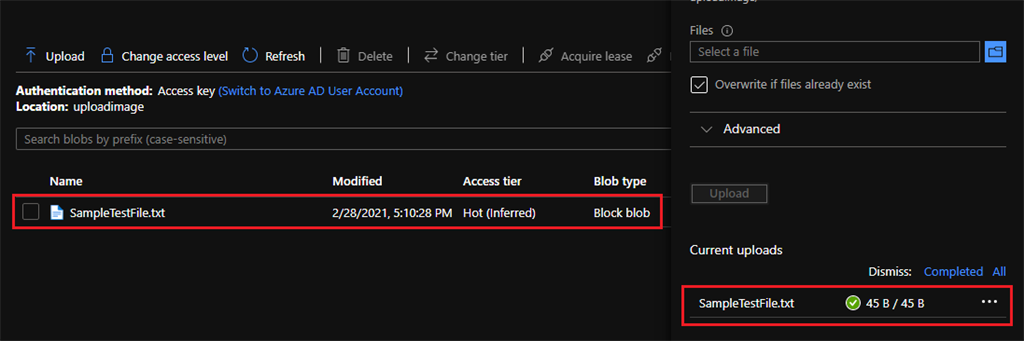


* Now, click on the folder icon and it will open a pop-up to select the file from the folder. Select the file and click on open. Once it selected then click on 'Upload'.



* Once the file uploaded successfully, we can see the file available in the container along with other details.

Please verify the below screen to check the status of the uploaded file on the container.



## Delete a storage account

Deleting a storage account deletes the entire account, including all data in the account, and cannot be undone.

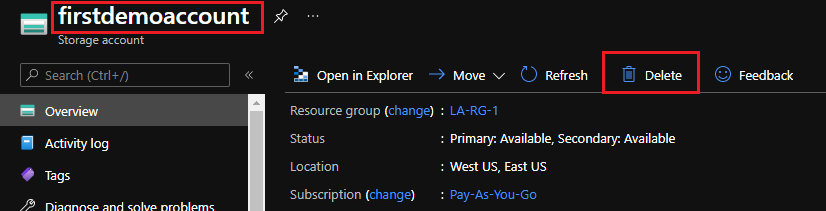
There are again multiple ways to delete the existing storage account

1. Azure portal
2. Azure PowerShell
3. Azure CLI
4. Azure Resource Manager (ARM) Template

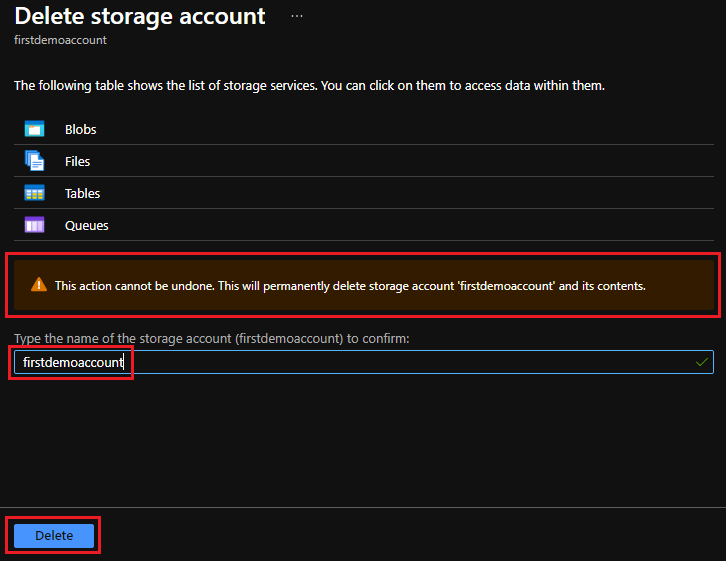
In this article, we will use the first option i.e. using the Azure portal. For other options, we will understand in different articles.

Log in to the [Azure portal](https://portal.azure.com/) ➜ Search Storage Accounts ➜ then select Storage accounts, which want to delete

Once we open the storage account, we will have an option to delete it. Click on 'Delete'.



It will open the Delete storage account; here we have to type the name of the storage account to confirm. Then click on 'Delete'.



**Caution**

This action cannot be undone. This will permanently delete the storage account and its contents.

## Conclusion

There are several advantages to using Azure storage irrespective of type. Azure storage is easily scalable, extremely flexible, and relatively low in cost depending on the options we choose.