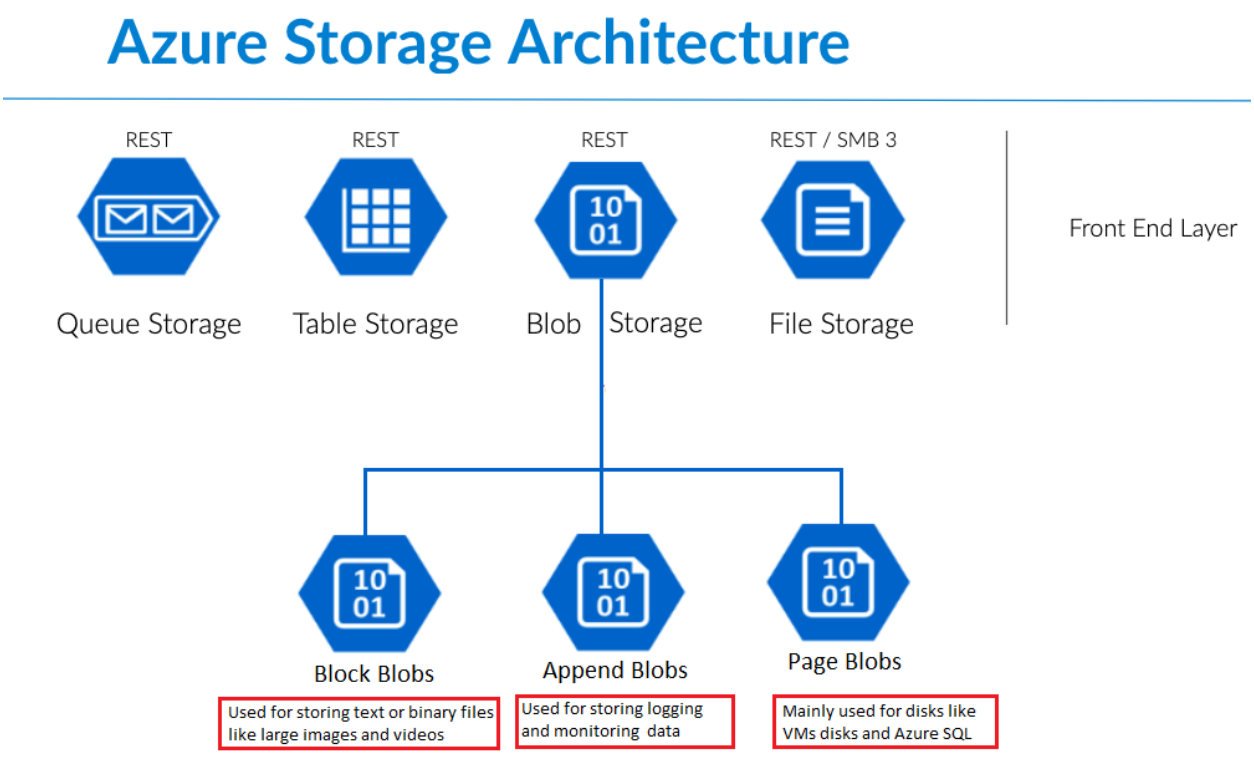
Contents

What is Azure Storage Account

The Azure Storage platform is Microsoft's cloud storage solution for modern data storage scenarios. Core storage services offer a massively scalable object store for data objects, disk storage for Azure virtual machines (VMs), a file system service for the cloud, a messaging store for reliable messaging, and a NoSQL store. The storage account provides a unique namespace (URL) for your Azure Storage data that is accessible from anywhere in the world over HTTP or HTTPS.



Azure Storage offers several types of storage accounts. Each type supports different features and has its own pricing model. Consider these differences before you create a storage account to determine the type of account that is best for your applications.

1. **BlobStorage** - Legacy Blob-only storage accounts. Use general-purpose v2 accounts instead when possible. It only supports two types of blob
    a. Block Blob
    b. Append Blob
2. **Storage (General-purpose v1)** - Legacy account type for blobs, files, queues, and tables. Use general-purpose v2 accounts instead when possible. It supports all three types of blob
    a. Block Blob
    b. Append Blob
    c. Page Blob
3. **StorageV2 (General-purpose v2)** - Basic storage account type for blobs, files, queues, and tables. Recommended for most scenarios using Azure Storage.

StorageV2 (General Storage v2) account which takes the features of the **BlobStorage** accounts and combines them with the **Storage (General-purpose v1)**, plus tiering

StorageV2 (General-purpose v2) = Storage (General-purpose v1) + BlobStorage

Azure storage offers different access tiers, which allow you to store blob object data in the most cost-effective manner. The available access tiers

1. **Hot**- Lowest access rates, most expensive per GB capacity.
2. **Cool** - Still low latency, but cheap per GB capacity at higher access rate and stored for at least 30 days. Data in the cool access tier can tolerate slightly lower availability, but still requires high durability, retrieval latency, and throughput characteristics like hot data. For cool data, a slightly lower availability service-level agreement (SLA) and higher access costs compared to hot data are acceptable trade-offs for lower storage costs.
3. **Archive** - The cheapest per GB capacity but it takes up to 15 hours to move a blob back to cool/hot where it can be accessed again.

Differences between BlobStorage and Storage (General-purpose v1)

	BlobStorage	Storage (General-purpose v1)
Storage Service	Only support **blobs**	It supports **blobs**, **files**, **queues** and **tables**
Blobs types	Only supports **block** and **append** blobs	It supports **block**, **append** & **page** blobs
Access tier	Support both **hot** and **cool** access tier	Only support **hot** access tier

Storage Account core storage services

Azure Storage Account provide these core storage services

1. **Blobs Storage**
2. **Tables Storage**
3. **Files Storage**
4. **Queues Storage**

Blob storage

A massively scalable object store for text and binary data. Azure Storage offers three types of blob storage

1. Block Blobs
2. Append Blobs
3. Page blobs

Block Blobs

Block blobs are composed of blocks and are ideal for storing text or binary files, and for uploading large files efficiently. Blocks of block blob can be managed individually. Block blobs store up to about 4.75 TB of data.

Append Blobs

Append blobs are also made up of blocks, but they are optimized for append operations, making them ideal for logging scenarios. An append blob is comprised of blocks and is optimized for append operations. When you modify an append blob, blocks are added to the end of the blob only, via the Append Block operation. Updating or deleting of existing blocks is not supported. Unlike a block blob, an append blob does not expose its block IDs.

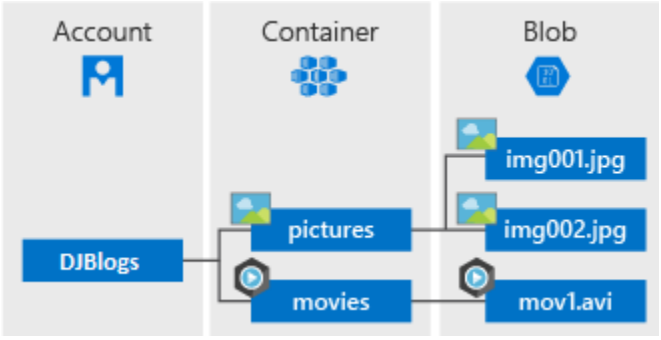
Page Blobs

Page blobs are made up of 512-byte pages up to 8 TB in total size and are designed for frequent random read/write operations. Page blobs are the foundation of Azure IaaS Disks. Page blobs are a collection of 512-byte pages, which provide the ability to read/write arbitrary ranges of bytes. Hence, page blobs are ideal for storing index-based and sparse data structures like OS and data disks for Virtual Machines and Databases.

Structure of Azure Blob storage

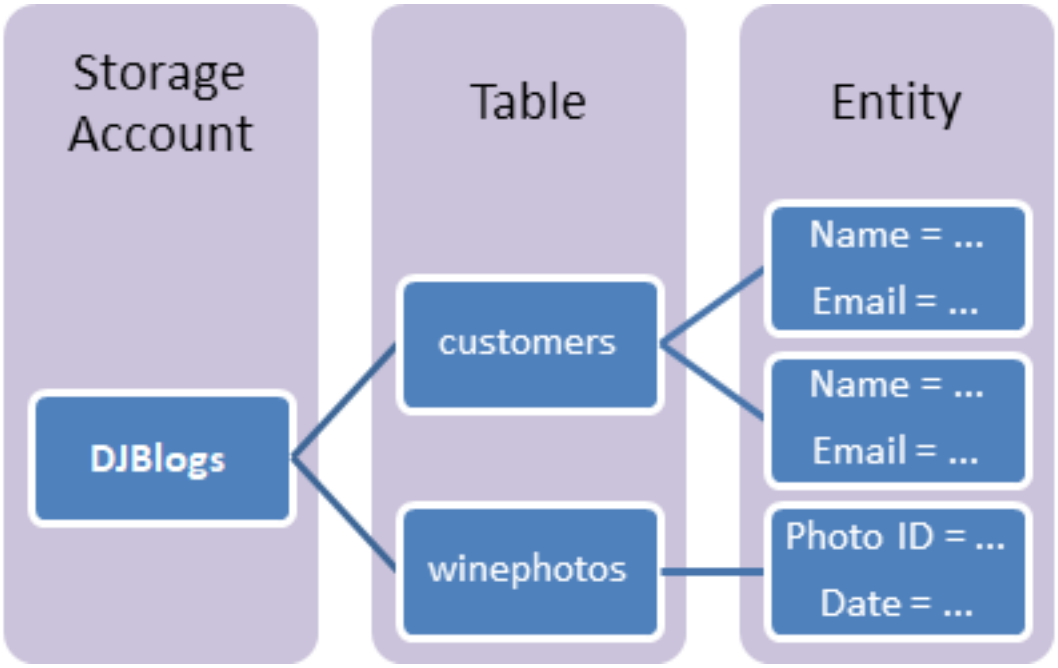
It has three types of resources

1. The storage Account
2. A container in the storage account
3. A blob



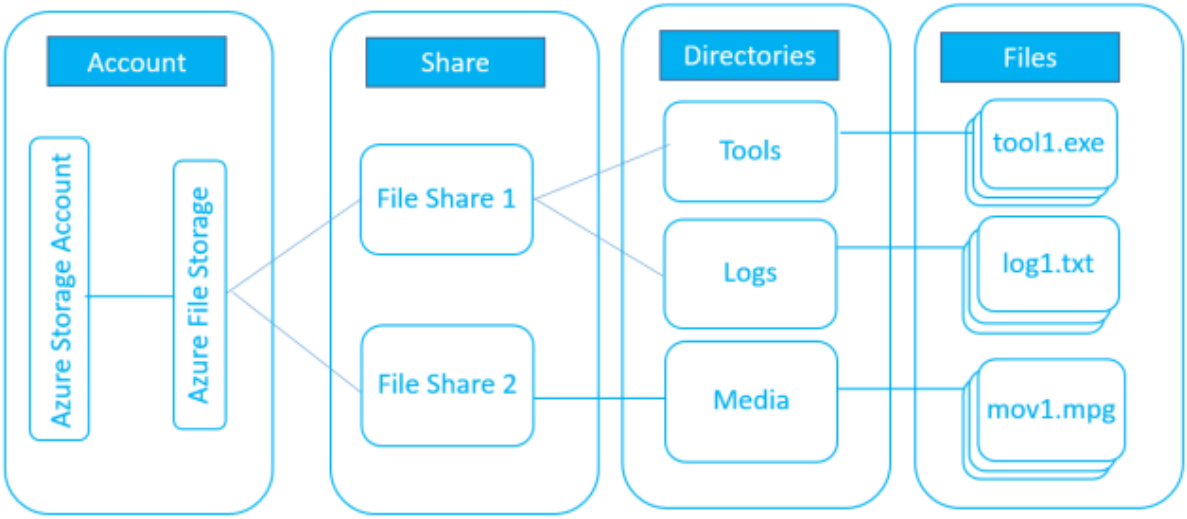
Tables Storage

Azure Table storage is a service that stores 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. Access to Table storage data is fast and cost-effective for many types of applications and is typically lower in cost than traditional SQL for similar volumes of data.
Table storage contains the following components



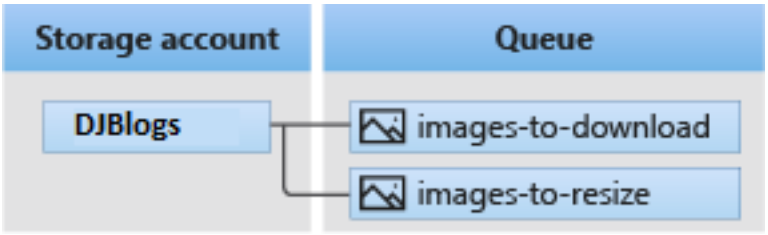
Files Storage

Azure Files is an alternative to Azure Blob Storage (Block Blob) but can reside within the same storage account. Azure File Storage exposes file shares using the Server Message Block (SMB) protocol, the predominantly used file share protocol for existing on-premises applications, it simplifies moving your existing applications to the cloud, and because Azure File Storage allows applications to mount file shares from anywhere in the world, your on-premises applications can take advantage of cloud storage without change. Azure File Storage also implements REST API protocol, which enables you to develop modern applications that integrate with existing applications.



Queues Storage

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.



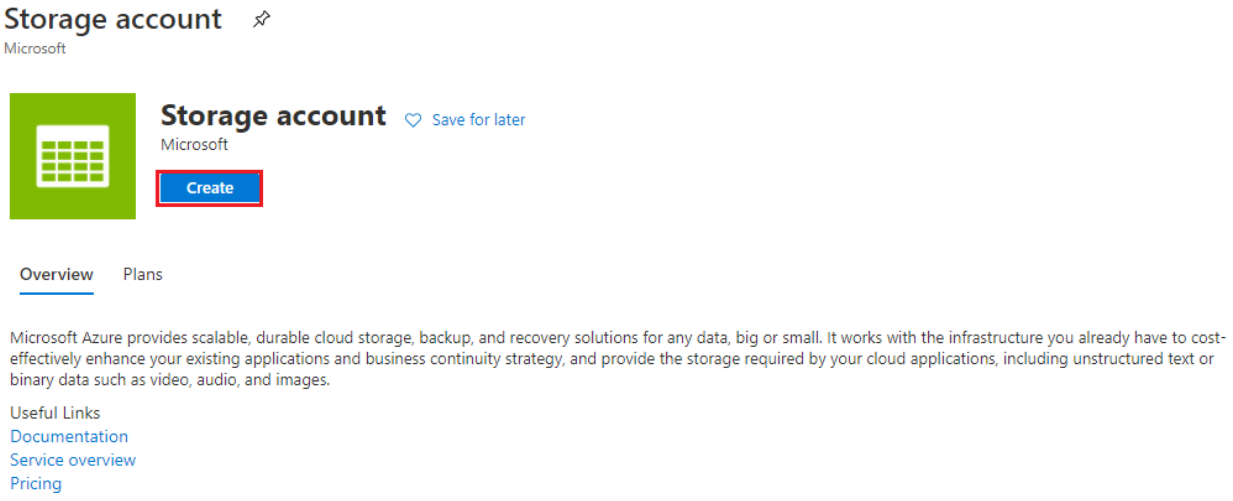
Create Azure Storage Account

We need to follow below steps to “Create storage account”

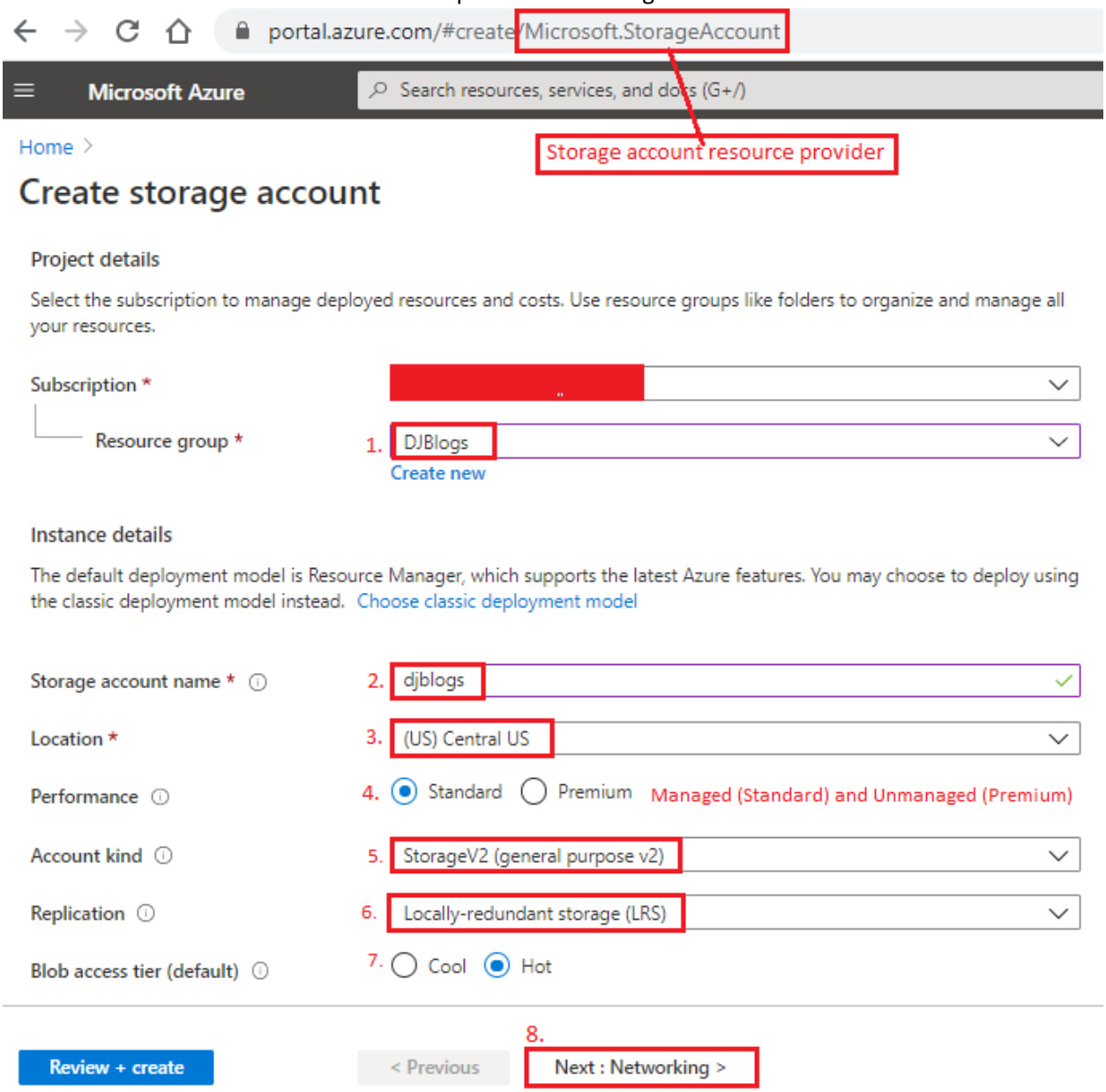
1. Go to azure portal https://portal.azure.com

2. Once you are login in portal click in (+) sign as below “Azure Marketplace” search page will open type “Storage Account”

3. Select “Storage account” option

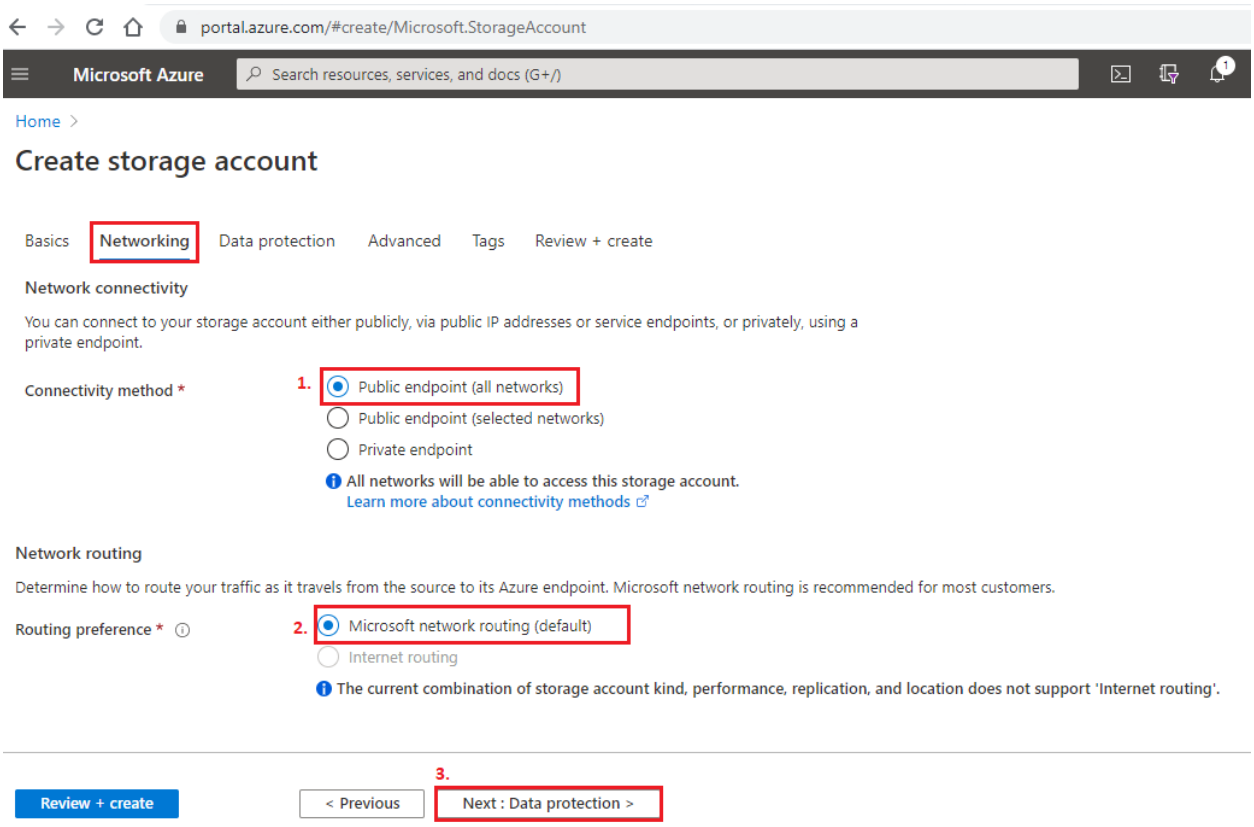


4. Click on above “Create” button and it will open “Create storage account” form like this

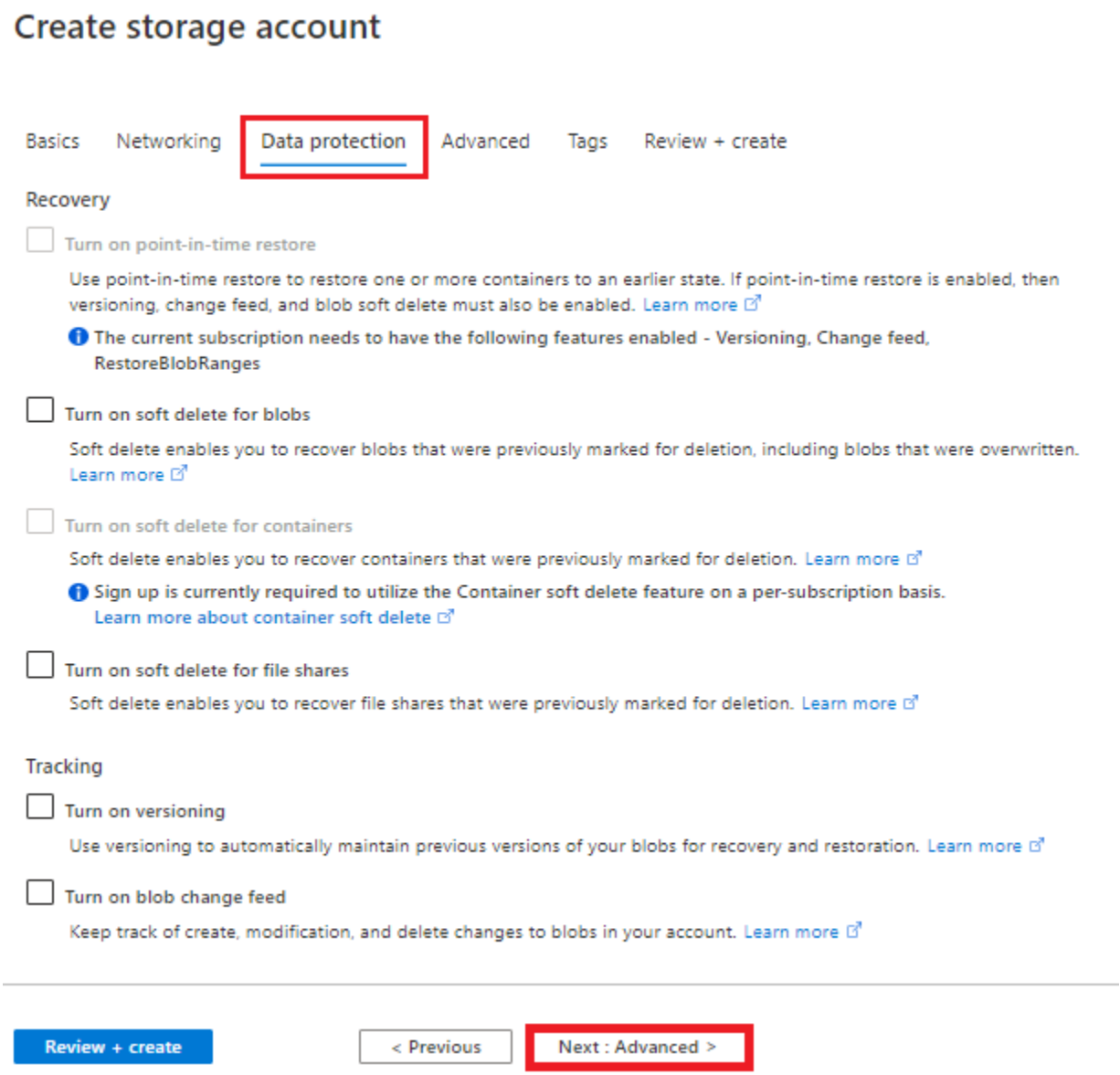


   a. **Performance** - A premium performance tier for storing unmanaged virtual machine disks. Microsoft recommends using managed disks with Azure virtual machines instead of unmanaged disks.
   b. **Replication** - Azure Storage always stores multiple copies of your data so that it is protected from planned and unplanned events, including transient hardware failures, network or power outages, and massive natural disasters.

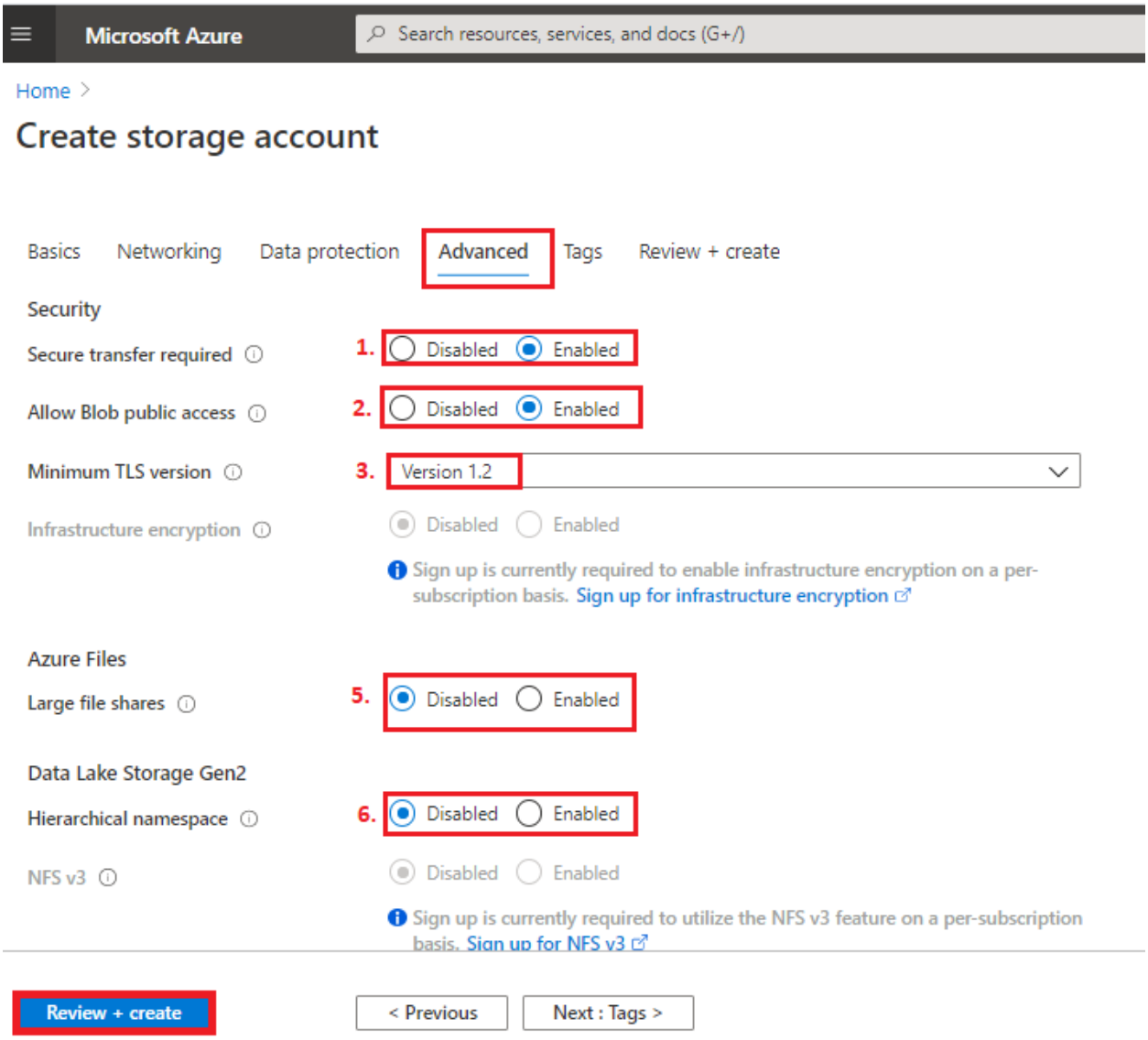
5. Once we fill “Create storage account” form as above it asks to choose “Network”



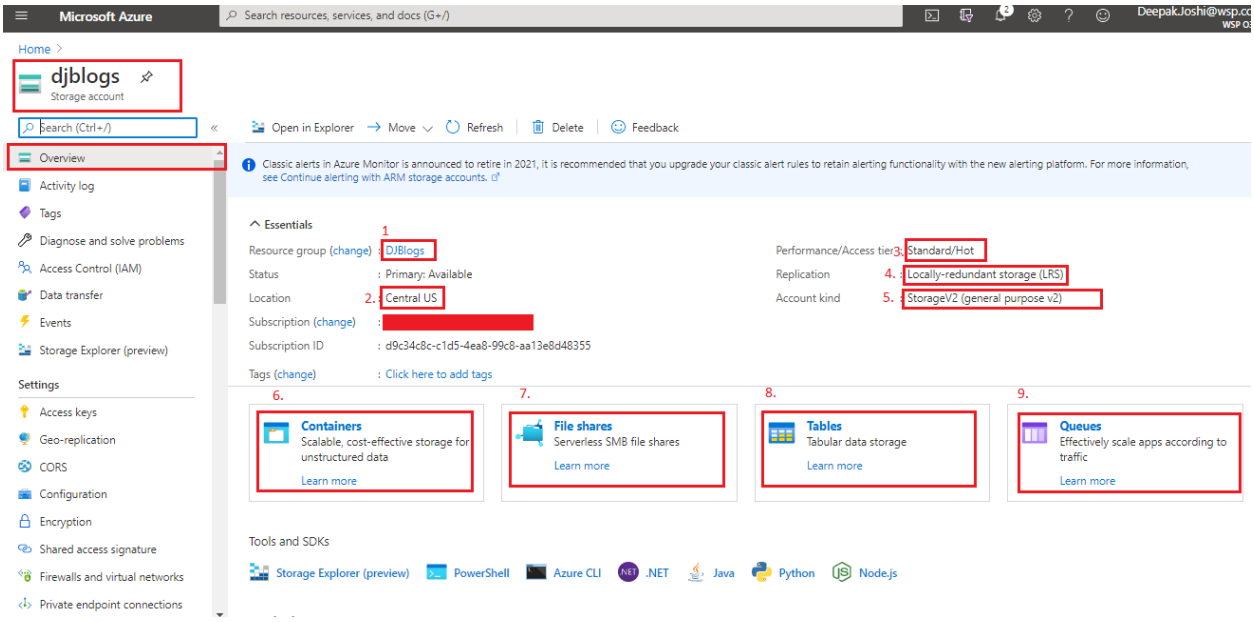
6. We have not made any change in “Network” default option just click in “Next: Data protection”



7. We have not made any change in this screen as well just click in “Next: Advanced”



8. On above screen we have not made any changed all these options selected by default. Just click in “Review + Create” button it will create “djblogs” as storage account.

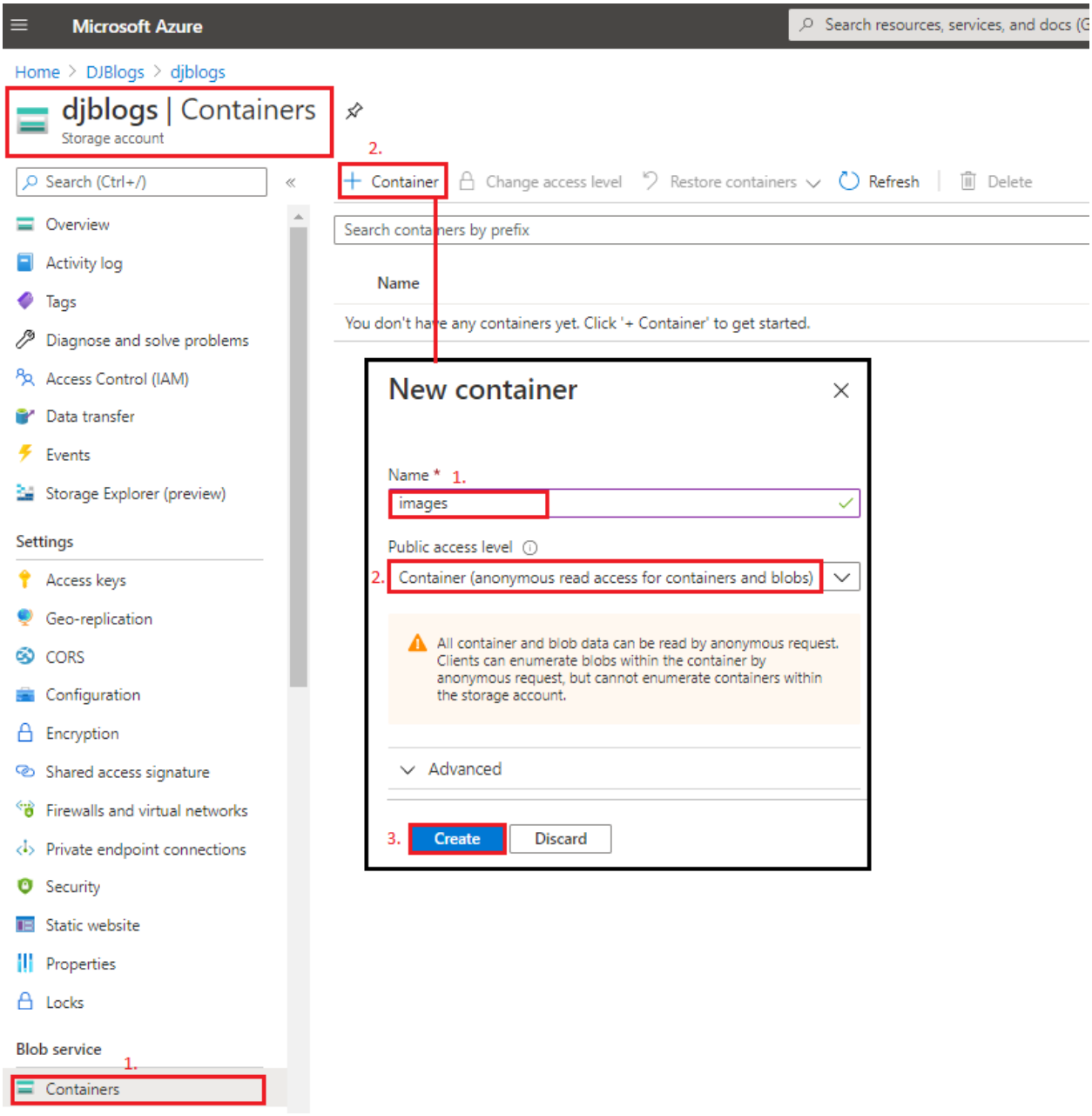


How to save image in blob container

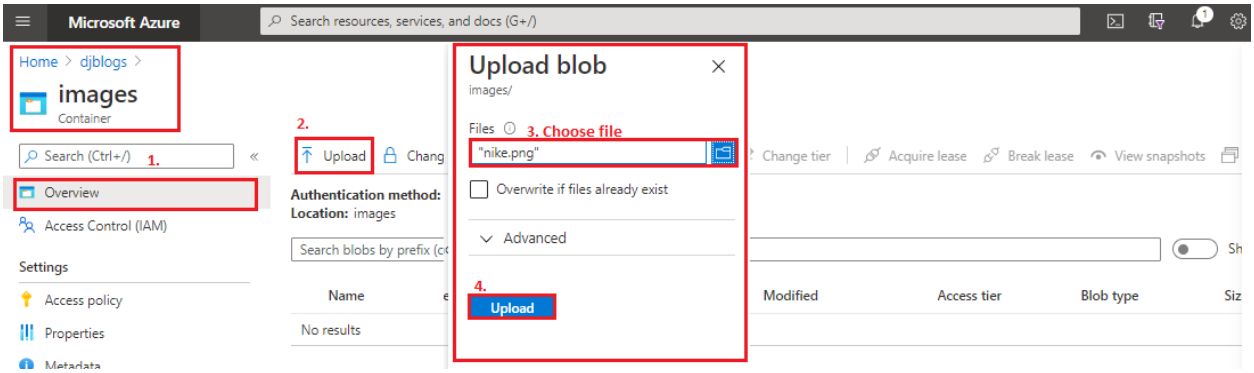
Now I will create demo application to save images in blob storage container. Need to follow below steps

1. Go to azure portal https://portal.azure.com

2. Select **djblogs** storage account and add new container **images** like below



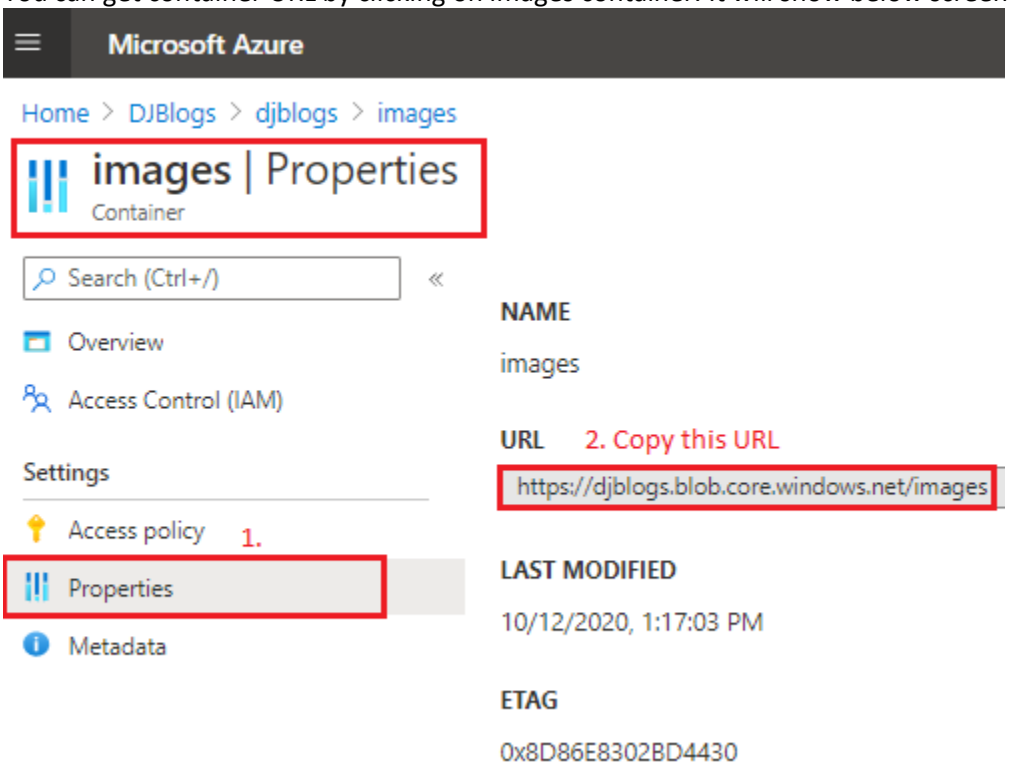
3. User can upload new image in **images** container directly from azure portal as below



4. Once **images** container created we need to save images inside it. We can access this container with help of
    a. Container URL
    b. Storage account access keys

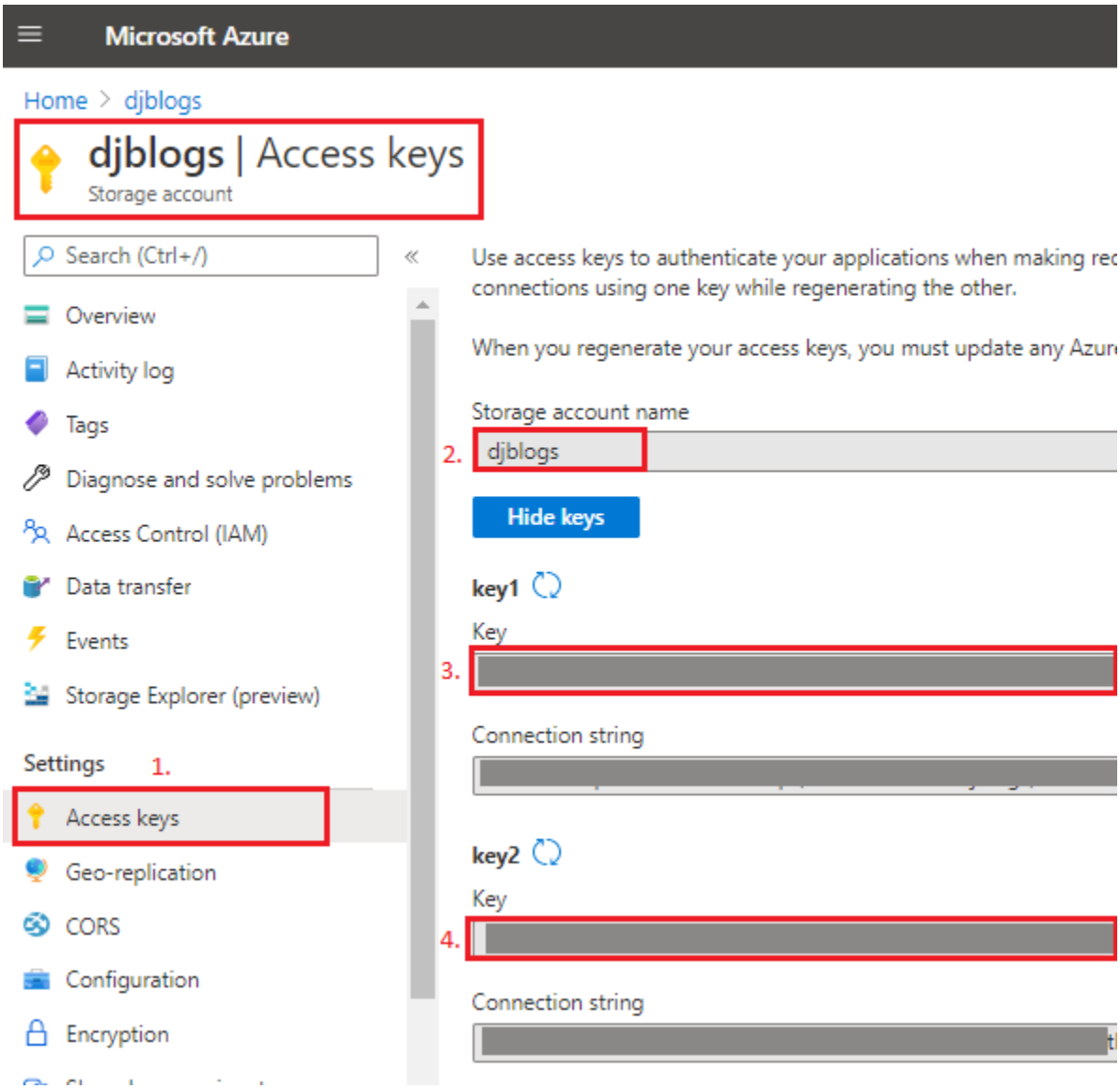
Container URL

You can get container URL by clicking on images container. It will show below screen



Storage account access keys

We can get storage account access key by clicking on storage account **Access keys** section. It provides two access keys. You can use any one of them



5. Now open visual studio and create new MVC project in .NET core. First we will install NuGet package “Microsoft.WindowsAzure.Storage” to for blob storage library.

6. You can get the images from blob storage with below piece of code.

```
// Get a reference to a container that's available for anonymous access.
CloudBlobContainer container = new CloudBlobContainer(new Uri(AppConstant.BlobContainerURL));
//New for core
OperationContext context = new OperationContext();
BlobRequestOptions options = new BlobRequestOptions();
// List blobs in the container.
var list = container.ListBlobsSegmentedAsync(null, true, BlobListingDetails.All, null, null, options, context);
var _list = new List<FileModel>();
foreach (IListBlobItem blobItem in list.Result.Results)
{
    _list.Add(new FileModel()
    {
        URL = blobItem.Uri.ToString(),
        Name = Path.GetFileName(blobItem.Uri.ToString())
    });
    //Console.WriteLine(blobItem.Uri);
}
```

https://djblogs.blob.core.windows.net/images

GitHub Code:
https://github.com/deepakjoshiinfo/DJBlog.StorageAccount/blob/main/DJBlogs.StorageAccount/Controllers/HomeController.cs

7. User can save images and files with below code

```
if (files == null || files.Count == 0)
    return Content("files not selected");
foreach (var file in files)
{
    var stream = file.OpenReadStream();
    // Create storagecredentials object by reading the values from the configuration (appsettings.json)
    //deeppawncloudcustomer - Access keys
    StorageCredentials storageCredentials = new StorageCredentials("djblogs", AppConstant.Accesskey);

    // Create cloudstorage account by passing the storagecredentials
    CloudStorageAccount storageAccount = new CloudStorageAccount(storageCredentials, true);

    // Create the blob client.
    CloudBlobClient blobClient = storageAccount.CreateCloudBlobClient();

    // Get reference to the blob container by passing the name by reading the value from the configuration (appsettings.json)
    CloudBlobContainer container = blobClient.GetContainerReference("images");

    // Get the reference to the block blob from the container
    CloudBlockBlob blockBlob = container.GetBlockBlobReference(file.FileName);

    // Upload the file
    await blockBlob.UploadFromStreamAsync(stream);
}
```

Storage account

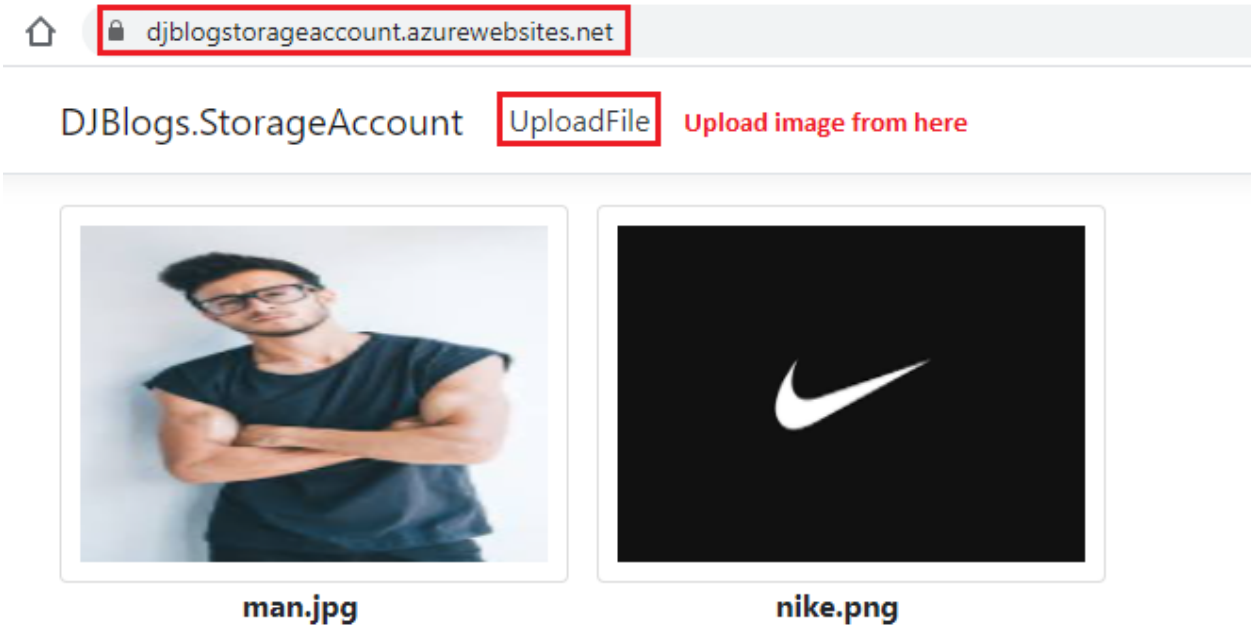
Access keys

Container name inside blob

GitHub Code:
https://github.com/deepakjoshiinfo/DJBlog.StorageAccount/blob/main/DJBlogs.StorageAccount/Controllers/HomeController.cs

8. I have added all this code in my “DJBlogs.StorageAccount” project. You can download this project code from GitHub.
   GitHub URL: https://github.com/deepakjoshiinfo/DJBlog.StorageAccount

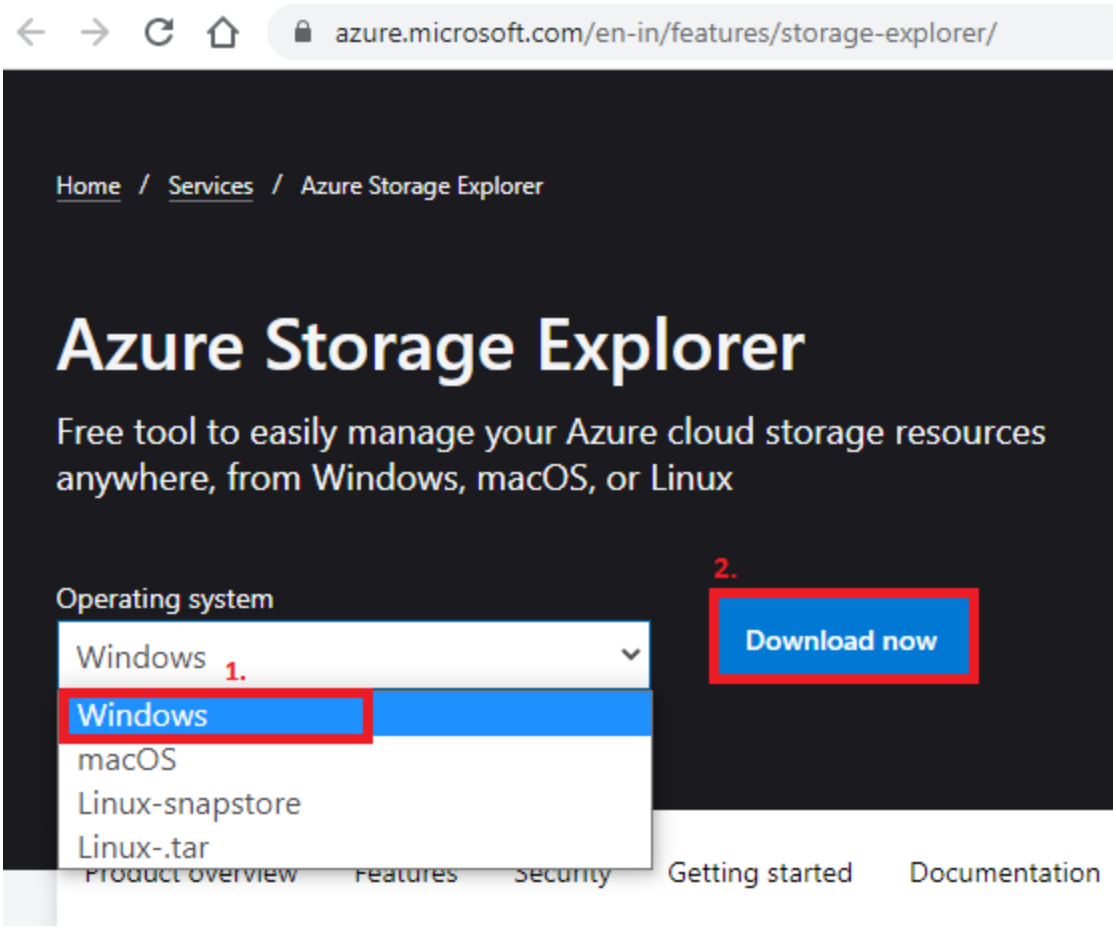
9. Also deployed this code in Azure WebApp as well
   URL: https://djblogstorageaccount.azurewebsites.net/
   I have uploaded 2 images it will look like as below



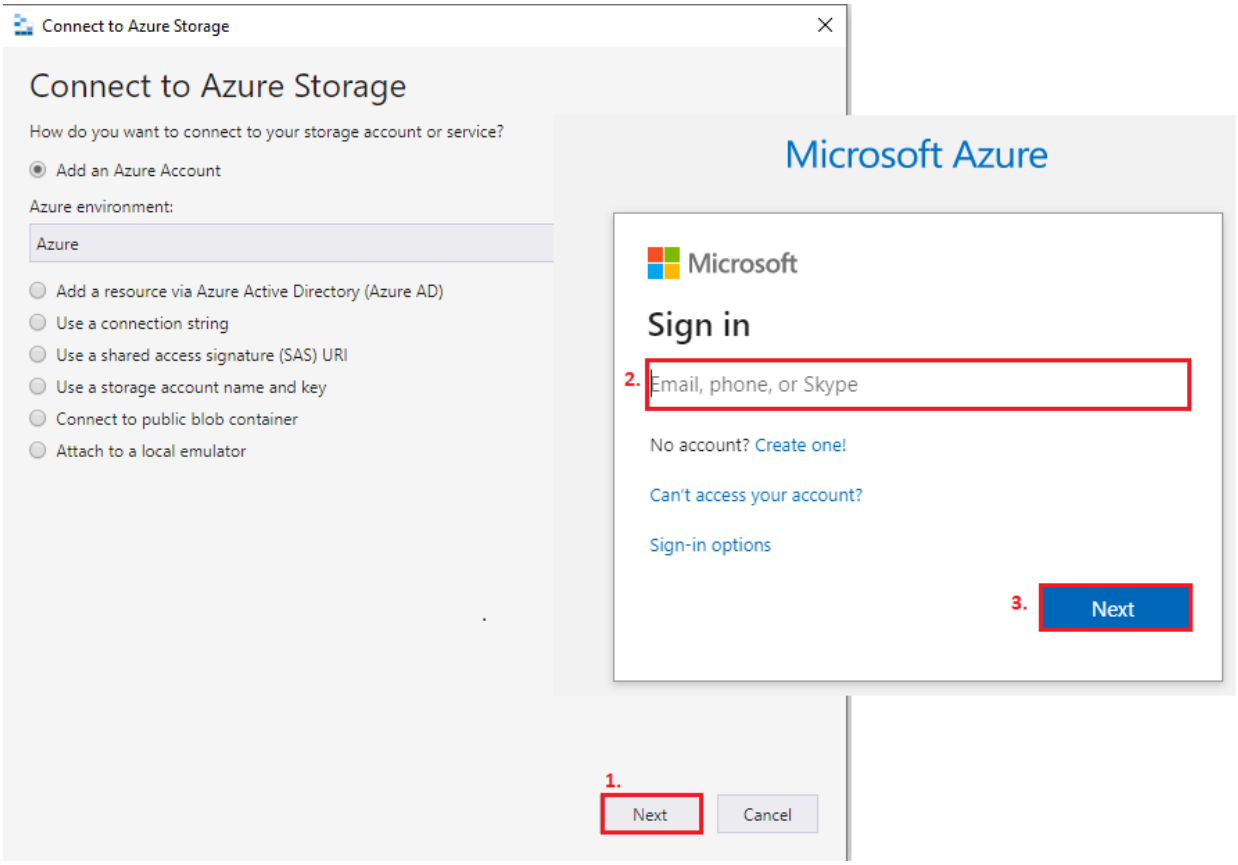
What is Azure Storage Explorer

Azure Storage Explorer is an application which helps you to easily access the Azure storage account through any device on any platform, be it Windows, MacOS, or Linux. You can easily connect to your subscription and manipulate your tables, blobs, queues, and files.
We need to follow below steps to see all storage account details in “Azure Storage Explorer”

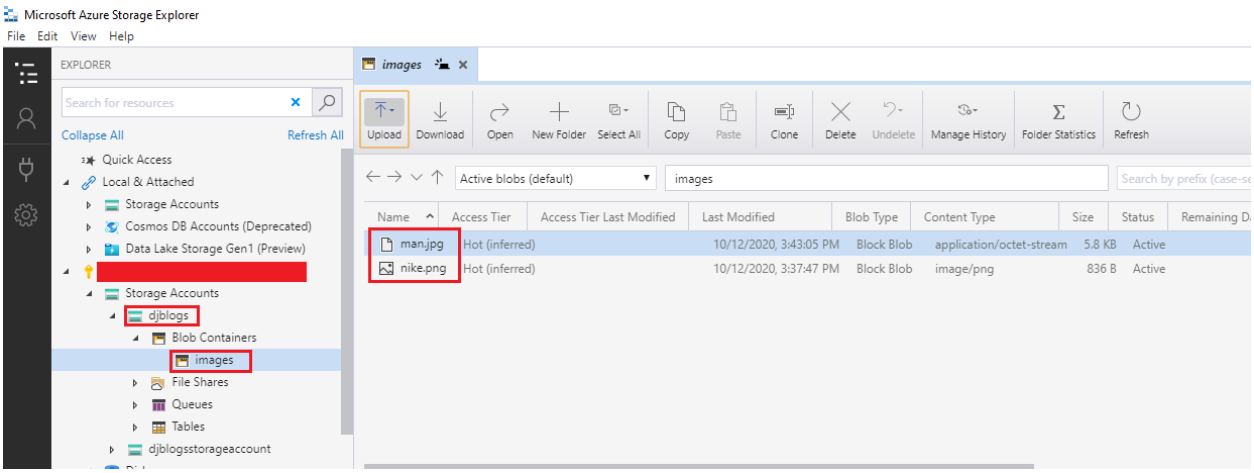
1. You can download “Azure Storage Explorer” from below link and install your machine
   Download Link: https://azure.microsoft.com/en-in/features/storage-explorer



2. Once you “Azure Storage Explorer” installed in your machine then need to be added your azure to see all your storage accounts.



3. Once your azure account added in "Azure Storage Explorer" then you can see all storage account added in azure subscription. As we have created "**djblogs**" storage account and added **images** container to store images. It will look like below



Once you installed it in your machine need to login with
Hope it will help you to understand azure storage account and where we can use them
Keep sharing keep learning

Cheers