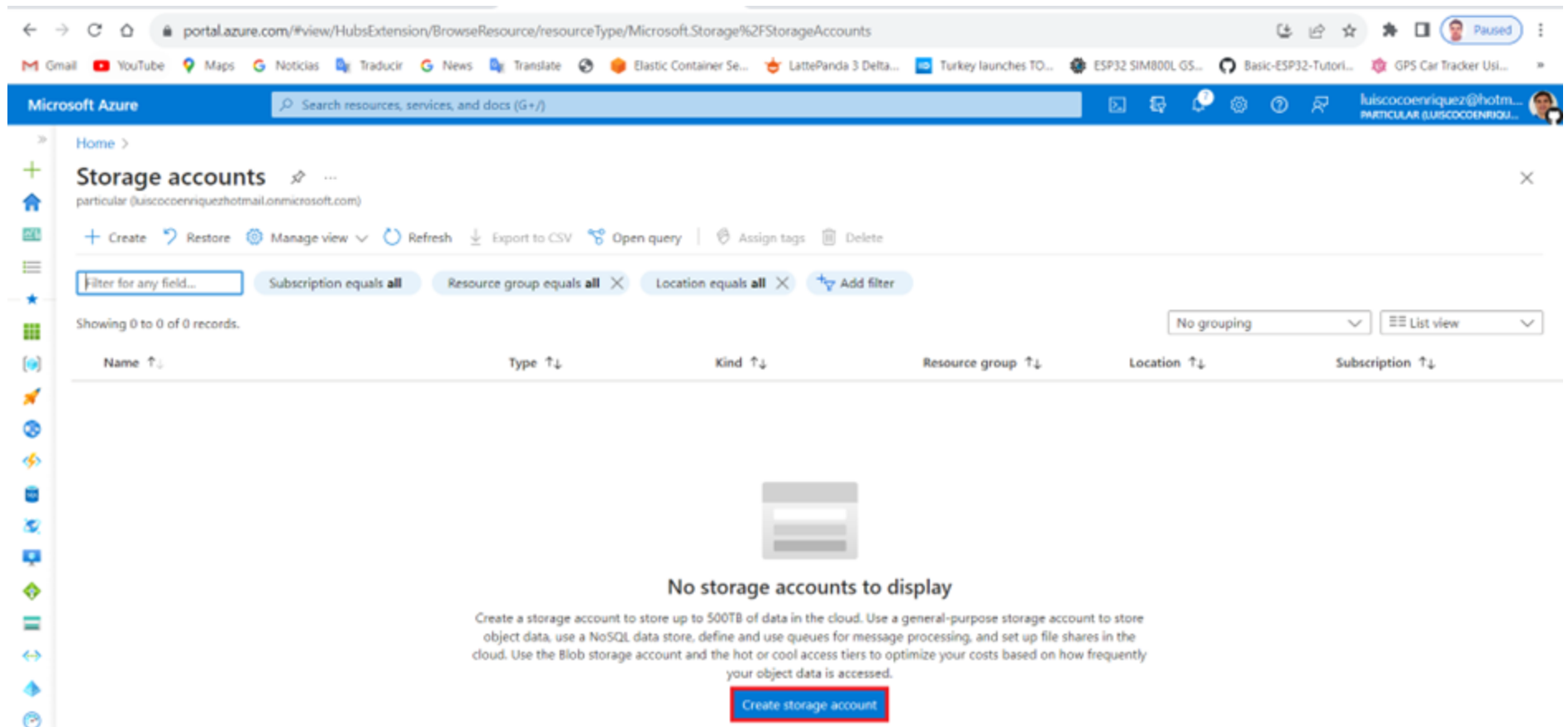


# Azure SDK Sample8: how to create an Azure Blob container and Upload/Download a file from/to your local hard drive disk

<https://learn.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-dotnet>

## 1. Create an Azure Storage account



[Home](#) > [Storage accounts](#) >

# Create a storage account ...

**Basics**

Advanced

Networking

Data protection

Encryption

Tags

Review

Subscription \*

Azure subscription 1

Resource group \*

(New) myNewRG

[Create new](#)

## Instance details

Storage account name ⓘ \*

mynewstorageaccount1974

Region ⓘ \*

(Europe) West Europe

[Deploy to an edge zone](#)

Performance ⓘ \*

☒ **Standard:** Recommended for most scenarios (general-purpose v2 account)☐ **Premium:** Recommended for scenarios that require low latency.

Redundancy ⓘ \*

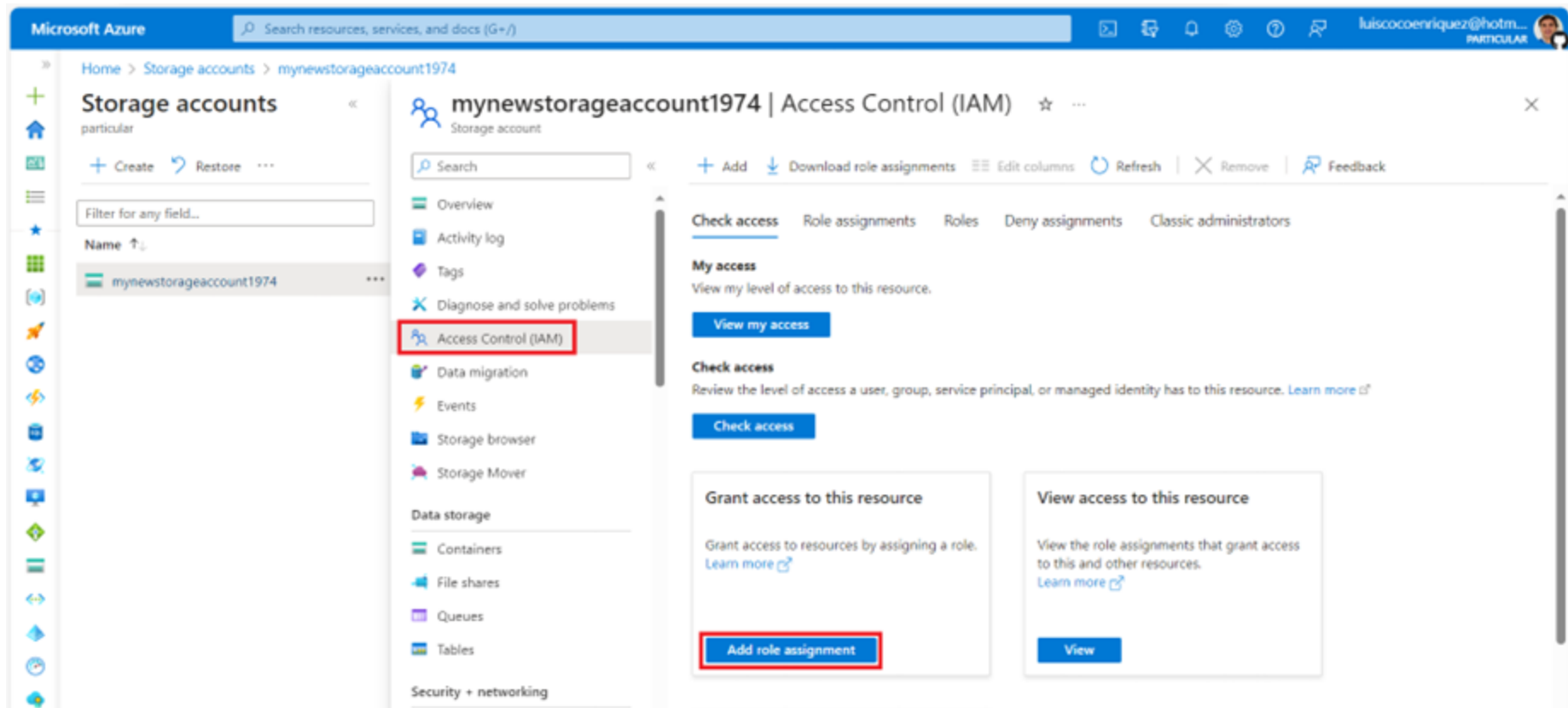
Locally-redundant storage (LRS)

**Review**

&lt; Previous

Next : Advanced &gt;

## 2. Grant permission to the Azure Storage account for creating a Blog storage container



Microsoft Azure Search resources, services, and docs (G+)

Home > Storage accounts > mynewstorageaccount1974 | Access Control (IAM) >

## Add role assignment

**Role** Members Conditions (optional) Review + assign

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. [Learn more](#)

Assignment type

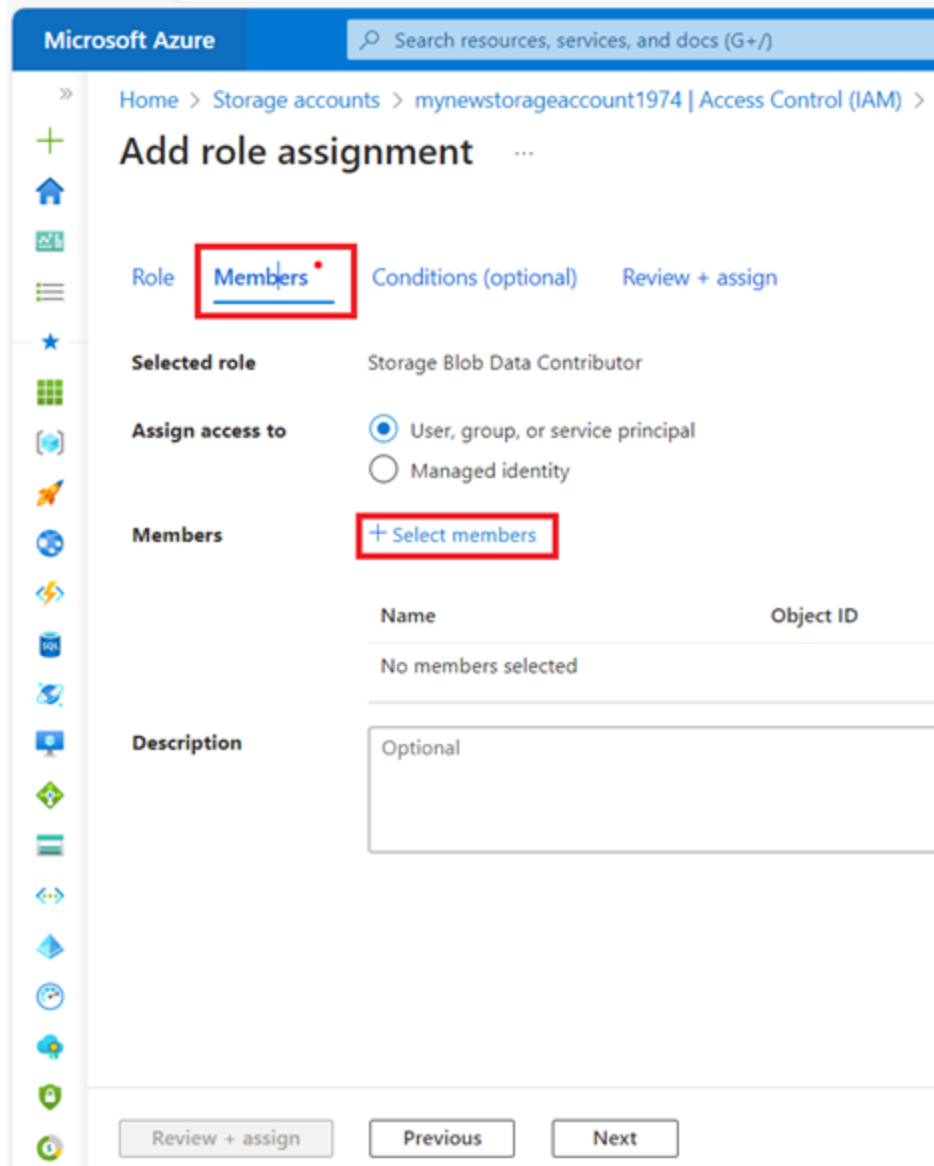
**Job function roles** Privileged administrator roles

Grant access to Azure resources based on job function, such as the ability to create virtual machines.

storage contrib × Type: All Category: All

Name ↑↓	Description ↑↓
Log Analytics Contributor	Log Analytics Contributor can read all monitoring data and edit monitoring settings. Edit
Managed Application Contributor Role	Allows for creating managed application resources.
Monitoring Contributor	Can read all monitoring data and update monitoring settings.
Resource Policy Contributor	Users with rights to create/modify resource policy, create support ticket and read resour
Storage Account Backup Contributor	Lets you perform backup and restore operations using Azure Backup on the storage acc
Storage Account Contributor	Lets you manage storage accounts, including accessing storage account keys which prov
<b>Storage Blob Data Contributor</b>	Allows for read, write and delete access to Azure Storage blob containers and data

Review + assign Previous Next



### 3. Install .NET 8 SDK

Download from <https://dotnet.microsoft.com/en-us/download>) and then we can access the .NET CLI tools

### 4. Create a new .Net 8 C# console application

Open in VSCode a Terminal window and run the following command for :

```
dotnet new console --framework net8.0
```

## 5. Load the libraries for Azure SDK for .NET: <https://www.nuget.org/packages>

For loading the libraries in the application we run the commands:

```
dotnet add package Azure.Identity --version 1.10.4
dotnet add package Azure.Storage.Blobs --version 12.19.1
```

After loading the libraries we run the command:

```
dotnet restore
```

And we check in the **csproj** file . the libraries were loaded in the application

## 6. Input the source code in the program.cs file

```
using Azure.Storage.Blobs;
using Azure.Storage.Blobs.Models;
using Azure.Identity;

// TODO: Replace <storage-account-name> with your actual storage account name
string storageAccountName = "mynewstorageaccount1974";
var blobServiceClient = new BlobServiceClient(
    new Uri("https://" + storageAccountName + ".blob.core.windows.net"),
    new DefaultAzureCredential());

//Create a unique name for the container
string containerName = "quickstartblobs" + Guid.NewGuid().ToString();
```

```
// Create the container and return a container client object
BlobContainerClient containerClient = await blobServiceClient.CreateBlobContainerAsync(containerName);

// Create a local file in the ./data/ directory for uploading and downloading
string localPath = "data";
Directory.CreateDirectory(localPath);
string fileName = "quickstart" + Guid.NewGuid().ToString() + ".txt";
string localFilePath = Path.Combine(localPath, fileName);

// Write text to the file
await File.WriteAllTextAsync(localFilePath, "Hello, World!");

// Get a reference to a blob
BlobClient blobClient = containerClient.GetBlobClient(fileName);

Console.WriteLine("Uploading to Blob storage as blob:\n\t {0}\n", blobClient.Uri);

// Upload data from the local file
await blobClient.UploadAsync(localFilePath, true);

Console.WriteLine("Listing blobs...");

// List all blobs in the container
await foreach (BlobItem blobItem in containerClient.GetBlobsAsync())
{
    Console.WriteLine("\t" + blobItem.Name);
}

// Download the blob to a local file
// Append the string "DOWNLOADED" before the .txt extension
// so you can compare the files in the data directory
string downloadFilePath = localFilePath.Replace(".txt", "DOWNLOADED.txt");

Console.WriteLine("\nDownloading blob to\n\t{0}\n", downloadFilePath);

// Download the blob's contents and save it to a file
```

```
await blobClient.DownloadToAsync(downloadFilePath);

// Clean up
Console.WriteLine("Press any key to begin clean up");
Console.ReadLine();

Console.WriteLine("Deleting blob container...");
await containerClient.DeleteAsync();

Console.WriteLine("Deleting the local source and downloaded files...");
File.Delete(localFilePath);
File.Delete(downloadFilePath);

Console.WriteLine("Done");
```

## 7. Build and run the application

Type the command:

```
dotnet run
```



The screenshot displays the Visual Studio Code interface with a C# program named `Program.cs` open. The Explorer sidebar on the left shows the project structure with folders `.vs`, `bin`, `data`, and `obj`, and files `Program.cs`, `sampleCreateBlobandUplo...`, and `sampleCreateBlobandUplo...`. The main editor area shows the following C# code:

```
1 using Azure.Storage.Blobs;
2 using Azure.Storage.Blobs.Models;
3 using Azure.Identity;
4
5 // TODO: Replace <storage-account-name> with your actual storage account name
6 string storageAccountName = "mynewstorageaccount1974";
7 var blobServiceClient = new BlobServiceClient(
8     new Uri("https://" + storageAccountName + ".blob.core.windows.net"),
9     new DefaultAzureCredential());
10
11 //Create a unique name for the container
12 string containerName = "quickstartblobs" + Guid.NewGuid().ToString();
13
14 // Create the container and return a container client object
15 BlobContainerClient containerClient = await blobServiceClient.CreateBlobContainerAsync(containerName);
16
```

Below the code editor, the TERMINAL tab is active, showing the output of the `dotnet run` command:

```
PS C:\Azure SDK for .NET\sampleCreateBlobandUploadFile> dotnet run
Uploading to Blob storage as blob:
https://mynewstorageaccount1974.blob.core.windows.net/quickstartblobs591e5549-546d-42a7-8bf4-a539c6a7ee63/quickstart914b4d76-7d8c-493f-848b-e20518f46e0f.txt

Listing blobs...
quickstart914b4d76-7d8c-493f-848b-e20518f46e0f.txt

Downloading blob to
data\quickstart914b4d76-7d8c-493f-848b-e20518f46e0fDOWNLOADED.txt

Press any key to begin clean up
```

The bottom status bar shows the current project is `main*` with 71 lines of code, 0 errors, and 0 warnings. It also indicates the active AWS profile is `default` and shows various extension icons like CodeWhisperer, Cloud Code, Go Live, Quokka, Prettier, and Formatting.

Microsoft Azure Search resources, services, and docs (G+/)

Home > Storage accounts > mynewstorageaccount1974

## Storage accounts

particular

+ Create Restore ...

Filter for any field...

Name ↑↓

mynewstorageaccount1974

### mynewstorageaccount1974 | Containers

Storage account

Search

+ Container Change access level Restore containers Refresh Delete Give feedback

Search containers by prefix ☐ Show deleted containers

Name	Last modified	Anonymous access l...	Lease state
<input type="checkbox"/> \$logs	11/28/2023, 10:21:28...	Private	Available
<input checked="" type="checkbox"/> quickstartblobs591e5549-546d-42a7-8bf4-a539c6a7ee63	11/28/2023, 11:16:14...	Private	Available

Microsoft Azure Search resources, services, and docs (G+/)

Home > Storage accounts > mynewstorageaccount1974 | Containers >

## quickstartblobs591e5549-546d-42a7-8bf4-a539c6a7ee63

Container

Search

Upload Change access level Refresh Delete Change tier Acquire lease Break lease View snapshots Create snapshot Give feedback

**Authentication method:** Access key ([Switch to Microsoft Entra user account](#))  
**Location:** quickstartblobs591e5549-546d-42a7-8bf4-a539c6a7ee63

Search blobs by prefix (case-sensitive) ☐ Show deleted blobs

Add filter

Name	Modified	Access tier	Archive status	Blob type	Size
<input type="checkbox"/> quickstart914b4d76-7d8c-493f-848b-e20518f46e0f.txt	11/28/2023, 11:16:14 AM	Hot (Inferred)		Block blob	13 B

