

## AzCopy Tool

AzCopy is a command-line utility provided by Microsoft to enable high-performance data transfer to and from Azure storage. It supports various scenarios such as copying data between Azure Blob, File, and Table storage, as well as copying data from on-premises to Azure or between Azure storage accounts.

With AzCopy, you can perform tasks like copying files, listing storage objects, syncing directories, and transferring data in parallel to achieve faster transfer speeds. It's particularly useful for handling large volumes of data efficiently.

Microsoft regularly updates AzCopy to introduce new features, improvements, and bug fixes, so it's a good idea to check for updates if you're using it frequently. Additionally, AzCopy provides options for logging, retry policies, and various other configurations to tailor it to your specific needs.

Overall, AzCopy is a powerful tool for managing data in Azure storage environments, offering flexibility and efficiency for data transfer operations.

### Use cases:

AzCopy serves a variety of use cases, primarily revolving around efficient data transfer to and from Azure storage. Here are some common scenarios where AzCopy proves invaluable:

1. **Migration to Azure:** AzCopy facilitates the seamless migration of data from on-premises servers, other cloud providers, or even other Azure regions to Azure storage. This includes migrating virtual machine disks, databases, media files, and other types of data.
2. **Backup and Restore:** It's often used for creating backups of critical data stored in Azure storage, allowing for disaster recovery and business continuity. AzCopy can also restore data from backups quickly when needed.
3. **Data Distribution:** AzCopy enables the distribution of large datasets across different Azure regions or storage accounts. This is useful for scenarios like content delivery networks (CDNs), where data needs to be replicated across multiple locations for improved performance and availability.
4. **Data Archiving:** Organizations often use AzCopy to archive older or infrequently accessed data to lower-cost storage tiers within Azure, such as Azure Blob Storage's Cool and Archive tiers.
5. **Content Migration for Applications:** For applications hosted in Azure, AzCopy can help migrate content such as static website assets, media files, logs, and user-generated content between development, staging, and production environments.
6. **Big Data and Analytics:** AzCopy supports the transfer of large datasets generated by big data and analytics workloads to Azure storage, facilitating data processing

and analysis using Azure services like Azure Data Lake Storage, Azure Synapse Analytics, and Azure Databricks.

7. **Hybrid Cloud Scenarios:** AzCopy is useful in hybrid cloud setups where data needs to be synchronized between on-premises infrastructure and Azure storage. It can efficiently handle periodic data syncs and updates.
8. **Content Syncing for Web Applications:** Web applications often require syncing of user-uploaded content (images, documents, etc.) between frontend servers and Azure storage. AzCopy simplifies this process by automating the transfer of new or updated content.
9. **Media and Entertainment:** In the media and entertainment industry, AzCopy assists in transferring large multimedia files, such as videos, images, and audio, to Azure storage for processing, streaming, and distribution.
10. **Continuous Integration/Continuous Deployment (CI/CD):** AzCopy can be integrated into CI/CD pipelines to automate the deployment of application artifacts (binaries, configuration files, etc.) to Azure storage for deployment to Azure services like Azure App Service, Azure Functions, and Azure Kubernetes Service.



## What are we doing in this lab?

1. **Download AzCopy:** Start by downloading the AzCopy tool from Microsoft's official website, choosing the appropriate version for your operating system.
2. **Prepare for Data Transfer:** Place the downloaded AzCopy executable in a temporary folder along with the data you want to transfer to Azure Storage.
3. **Create an Azure Storage Account:** If you don't already have an Azure Storage account, create one via the Azure Portal.
4. **Generate a Shared Access Signature (SAS):** In your storage account, create a SAS token with all necessary permissions to enable secure data transfer.
5. **Open Command Prompt:** Navigate to the temporary folder containing AzCopy and your data.
6. **Create a Container:** Use AzCopy to create a new container in your Azure Storage account, specifying the storage account name, container name, and the SAS token.
7. **Verify Container Creation:** Confirm that the container has been successfully created by checking your Azure Storage account.
8. **Copy Files to Azure:** Use AzCopy to transfer files from your local machine to the Azure Storage container, specifying the file name and SAS token.
9. **Verify File Transfer:** Confirm that the file has been successfully copied to the Azure Storage container.

**10. Download Files from Azure:** Test the download functionality by deleting the local file and using AzCopy to transfer it back from Azure Storage.

### End Goal

The main objective of this process is to demonstrate how to use AzCopy for efficient data transfer to and from Azure Storage. This includes creating storage containers, uploading files to Azure, and downloading files back to your local system. The goal is to familiarize you with the tool's capabilities and ensure you can manage data in Azure storage environments effectively.

### To begin with the Lab:

1. First, we are going to download the AzCopy tool from Microsoft's official website. You can visit the site using the link mentioned below. As I am on a Windows laptop, I'll be downloading it for Windows 64-bit.

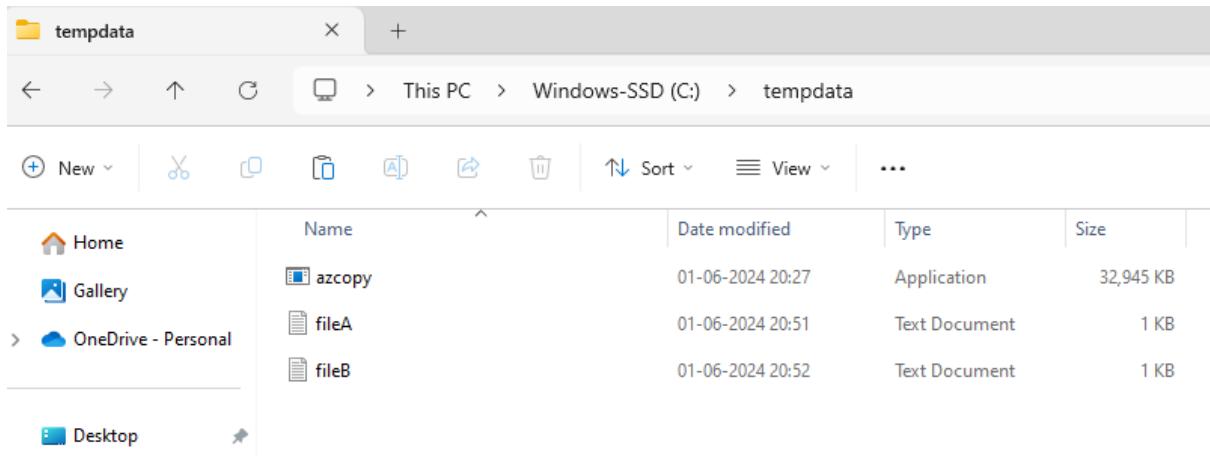
<https://learn.microsoft.com/en-us/azure/storage/common/storage-use-azcopy-v10>

## Download AzCopy

First, download the AzCopy V10 executable file to any directory on your computer. AzCopy V10 is just an executable file, so there's nothing to install.

- [Windows 64-bit](#) (zip)
- [Windows 32-bit](#) (zip)
- [Linux x86-64](#) (tar)
- [Linux ARM64](#) (tar)
- [macOS](#) (zip)
- [macOS ARM64 Preview](#) (zip)

2. Once it is downloaded take the exe file for azcopy and paste it into a temporary folder with the data you want to copy to your storage account.



3. After that you need to go to Azure Portal and create a storage account if you don't have any. After that, in your storage account, create a Shared Access Signature (SAS) so we can run the commands. Give it all the permission.

Allowed services ①

Blob  File  Queue  Table

Allowed resource types ①

Service  Container  Object

Allowed permissions ①

Read  Write  Delete  List  Add  Create  Update  Process  Immutable storage  Permanent delete

Blob versioning permissions ①

Enables deletion of versions

Allowed blob index permissions ①

Read/Write  Filter

4. Once your SAS is generated then you need to copy the SAS Token.

[Generate SAS and connection string](#)

Connection string

```
BlobEndpoint=https://stoage12030.blob.core.windows.net/QueueEndpoint=https://stoage12030.queue.core.windows.net/FileEndpoint=https://stoage12030.file.core.windows.net/TableEndpoint=https://st...
```

SAS token ①

```
sv=2022-11-02&ss=b&srt=sco&sp=rwdlacjytfx&se=2024-06-01T23:17:27Z&st=2024-06-01T15:17:27Z&spr=https&sig=bITy9gPfpN76mX%2BQuw09K3CkejUTW5R68%2B2L0oKcNuw%3D
```

Blob service SAS URL

```
https://stoage12030.blob.core.windows.net/?sv=2022-11-02&ss=b&srt=sco&sp=rwdlacjytfx&se=2024-06-01T23:17:27Z&st=2024-06-01T15:17:27Z&spr=https&sig=bITy9gPfpN76mX%2BQuw09K3CkejUT...
```

5. After that you need to open Command Prompt and go to that temporary folder.

```

C:\tempdata>dir
Volume in drive C is Windows-SSD
Volume Serial Number is 4ED4-19FB

Directory of C:\tempdata

01-06-2024  20:52    <DIR>          .
01-06-2024  20:27      33,735,192 azcopy.exe
01-06-2024  20:51           15 fileA.txt
01-06-2024  20:52           14 fileB.txt
              3 File(s)   33,735,221 bytes
              1 Dir(s)  47,303,806,976 bytes free

C:\tempdata>

```

6. Now we are going to create a container in our storage account using AzCopy tool. In the code below you need to give your storage account name then you need to specify the container name and then paste the SAS token. The two highlighted parts show the storage account name and the container name. After the container name “data” we have pasted the SAS Token.

```

azcopy make "https://stoage12030.blob.core.windows.net/data?sv=2022-11-02&ss=b&srt=sco&sp=rwdlaciytfx&se=2024-06-01T23:17:27Z&st=2024-06-01T15:17:27Z&spr=https&sig=bITy9gPfpN76mX%2BQuw09K3CkejUTW5R68%2B2L0oKcNuw%3D"

```

```

C:\tempdata>azcopy make "https://stoage12030.blob.core.windows.net/data?sv=2022-11-02&ss=b&srt=sco&sp=rwdlaciytfx&se=2024-06-01T23:17:27Z&st=2024-06-01T15:17:27Z&spr=https&sig=bITy9gPfpN76mX%2BQuw09K3CkejUTW5R68%2B2L0oKcNuw%3D"
Successfully created the resource.
C:\tempdata>

```

7. Now from the above snapshot you can see that it is saying that the resource is created successfully and if you go to your storage account then to container then you will see that the container is created successfully.

Name	Last modified	Anonymous access level	Lease state
Slogs	1/6/2024, 8:41:22 pm	Private	Available
data	1/6/2024, 8:55:42 pm	Private	Available

8. Then we used the command to copy the file from our laptop to our storage account. The command is the same as before this time we just added the file name.

```

azcopy copy "https://stoage12030.blob.core.windows.net/data/fileA.txt?sv=2022-11-02&ss=b&srt=sco&sp=rwdlaciytfx&se=2024-06-01T23:17:27Z&st=2024-06-01T15:17:27Z&spr=https&sig=bITy9gPfpN76mX%2BQuw09K3CkejUTW5R68%2B2L0oKcNuw%3D" fileA.txt

```

```

C:\tempdata>azcopy copy fileA.txt "https://stoage12030.blob.core.windows.net/data/fileA.txt?sv=2022-11-02&ss=b&srt=sco&sp=rwdlaciytfx&se=2024-06-01T23:17:27Z&st=2024-06-01T15:17:27Z&spr=https&sig=bITy9gPfpN76mX%2BQuw09K3CkejUTW5R68%2B2L0oKcNuw%3D"
INFO: Scanning ...
INFO: Any empty folders will not be processed, because source and/or destination doesn't have full folder support
Job 5fcfed16c-c82c-9649-7701-e4f0f19d949a has started
Log file is located at: C:\Users\JULIKIT\azcopy\5fcfed16c-c82c-9649-7701-e4f0f19d949a.log
100.0 %, 1 Done, 0 Failed, 0 Pending, 0 Skipped, 1 Total, 2-sec Throughput (Mb/s): 0.0001

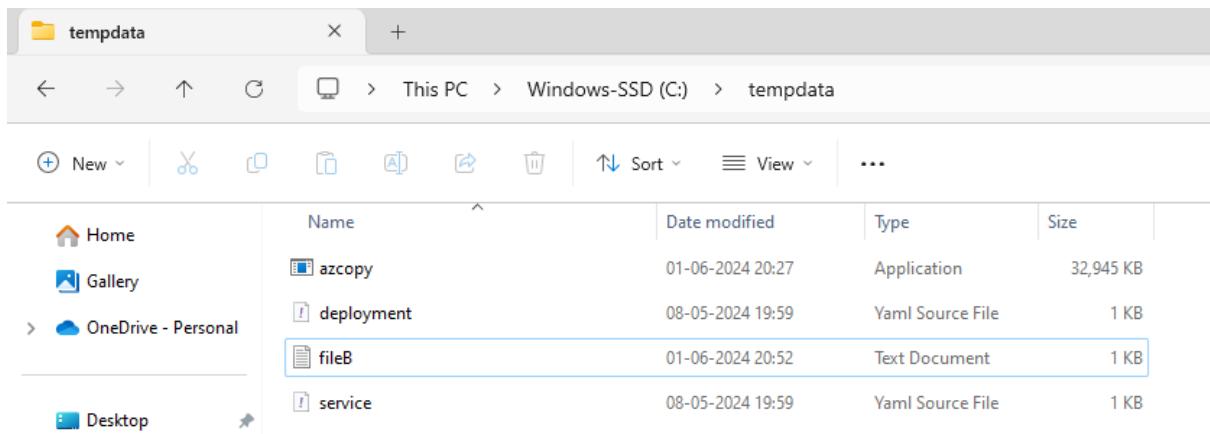
Job 5fcfed16c-c82c-9649-7701-e4f0f19d949a summary
Elapsed Time (Minutes): 0.0336
Number of File Transfers: 1
Number of Folder Transfers: 0
Number of Symlink Transfers: 0
Total Number of Transfers: 1
Number of File Transfers Completed: 1
Number of Folder Transfers Completed: 0
Number of File Transfers Failed: 0
Number of Folder Transfers Failed: 0
Number of File Transfers Skipped: 0
Number of Folder Transfers Skipped: 0
Total Number of Bytes Transferred: 15
Final Job Status: Completed

```

9. Also in your container you will be able to see the file.

The screenshot shows the Azure Storage Explorer interface. On the left, there's a sidebar with navigation links like 'Overview', 'Diagnose and solve problems', 'Access Control (IAM)', 'Settings', 'Shared access tokens', 'Access policy', 'Properties', and 'Metadata'. The main area displays a table of blobs. The table has columns: Name, Modified, Access tier, Archive status, Blob type, Size, and Lease state. There is one entry: 'fileA.txt' (Modified: 1/6/2024, 9:04:06 pm, Access tier: Hot (Inferred), Blob type: Block blob, Size: 15 B, Lease state: Available).

10. Now our last command is to download the blob data for that first we will delete the file from our laptop.



11. Here you can see that the transfer is complete.

```

azcopy copy "https://stoage12030.blob.core.windows.net/data/fileA.txt?sv=2022-11-02&ss=b&srt=sco&sp=rwdlaciytfx&se=2024-06-01T23:17:27Z&st=2024-06-01T15:17:27Z&spr=https&sig=bITy9gPfpN76mX%2BQuw09K3CkejUTW5R68%2B2L0oKcNuw%3D" "fileA.txt"

```

```
C:\tempdata>azcopy copy "https://storage12038.blob.core.windows.net/data/fileA.txt?sv=2022-11-02&ss=b&st=sco&sp=rwdlaciytfx&se=2024-06-01T23:17:27Z&st=2024-06-01T15:17:27Z&spr=https&sig=bITY9gPfpN76mX%2BQuw09K
3CkejUTW5R68%2B2L8oKcNuw%3D" "fileA.txt"
INFO: Scanning...
INFO: Any empty folders will not be processed, because source and/or destination doesn't have full folder support
Job d910fa20-6da4-594c-5cf4-f242a43e154d has started
Log file is located at: C:\Users\PULKIT\azcopy\d910fa20-6da4-594c-5cf4-f242a43e154d.log
100.0 %, 1 Done, 0 Failed, 0 Pending, 0 Skipped, 1 Total, 2-sec Throughput (Mb/s): 0.0001

Job d910fa20-6da4-594c-5cf4-f242a43e154d summary
Elapsed Time (Minutes): 0.0334
Number of File Transfers: 1
Number of Folder Property Transfers: 0
Number of Symlink Transfers: 0
Total Number of Transfers: 1
Number of Transfers Completed: 1
Number of Folder Transfers Completed: 0
Number of File Transfers Failed: 0
Number of Folder Transfers Failed: 0
Number of File Transfers Skipped: 0
Number of Folder Transfers Skipped: 0
Total Number of Bytes Transferred: 15
Final Job Status: Completed
```

12. Below you can see that we have file in place.

