**ACL(ACCESS CONTROL LISTS)**

**Access Control List (ACL) in Azure – Everything You Need to Know**

In Azure, **Access Control Lists (ACLs)** are used to provide fine-grained permission control over files and directories in **Azure Data Lake Storage Gen2** and **Azure Blob Storage**. Unlike **Role-Based Access Control (RBAC)**, which controls permissions at the **Azure resource level**, ACLs provide granular access control at the **file and folder level**.

**1. What is an ACL (Access Control List)?**

An **ACL (Access Control List)** is a set of rules that defines who can access a file or folder and what actions they can perform (read, write, execute). ACLs allow **multiple users, groups, or service principals** to have different permissions on the same file or directory.

**Key Features of ACLs in Azure**

✅ Granular control: Define permissions at the **file or folder level**  
✅ Supports **hierarchical permissions** (inheritance)  
✅ Works **alongside RBAC** (RBAC controls access to the storage account, while ACLs control access inside the storage)  
✅ Supports **POSIX-style ACLs** (used in Linux/Unix systems)

**2. Where Are ACLs Used in Azure?**

🔹 **Azure Data Lake Storage Gen2**  
🔹 **Azure Blob Storage (Hierarchical Namespace enabled)**  
🔹 **Azure Files (when integrated with Active Directory)**

ACLs are mainly used in **Azure Data Lake Storage Gen2**, which allows fine-grained access control to folders and files.

**3. How Do ACLs Work?**

Each **file or folder** has two types of ACLs:

1. **Access ACLs** – Define permissions for users, groups, or service principals
2. **Default ACLs** – Define inherited permissions for new files and folders created inside a directory

**ACL Permissions**

| **Permission** | **Symbol** | **Meaning** |
| --- | --- | --- |
| **Read** | r | View the file/folder contents |
| **Write** | w | Modify the file/folder contents |
| **Execute** | x | List folder contents or execute a file |

**4. ACL vs. RBAC in Azure**

| **Feature** | **ACL (Access Control List)** | **RBAC (Role-Based Access Control)** |
| --- | --- | --- |
| **Granularity** | File & Folder Level | Storage Account Level |
| **Permissions** | Read, Write, Execute | Reader, Contributor, Owner, etc. |
| **Inheritance** | Default ACLs can be inherited | No inheritance |
| **Scope** | Applies inside the storage account | Applies to entire Azure resources |
| **Use Case** | Fine-grained control over files/folders | Broad access management |

👉 **Best Practice**: Use **RBAC** to manage access at the storage account level and **ACLs** to control access at the file/folder level.

**5. How to Manage ACLs in Azure Data Lake Storage Gen2**

**Step 1: Check If Hierarchical Namespace is Enabled**

ACLs work **only when Hierarchical Namespace (HNS)** is enabled on **Azure Data Lake Storage Gen2**.

1. Go to **Azure Portal**
2. Open your **Storage Account**
3. Navigate to **Configuration**
4. Check if **Hierarchical Namespace** is enabled

**Step 2: Assign ACLs to a File/Folder**

You can assign ACLs using:

✅ **Azure Portal**  
✅ **Azure CLI**  
✅ **PowerShell**  
✅ **Python SDK**  
✅ **REST API**

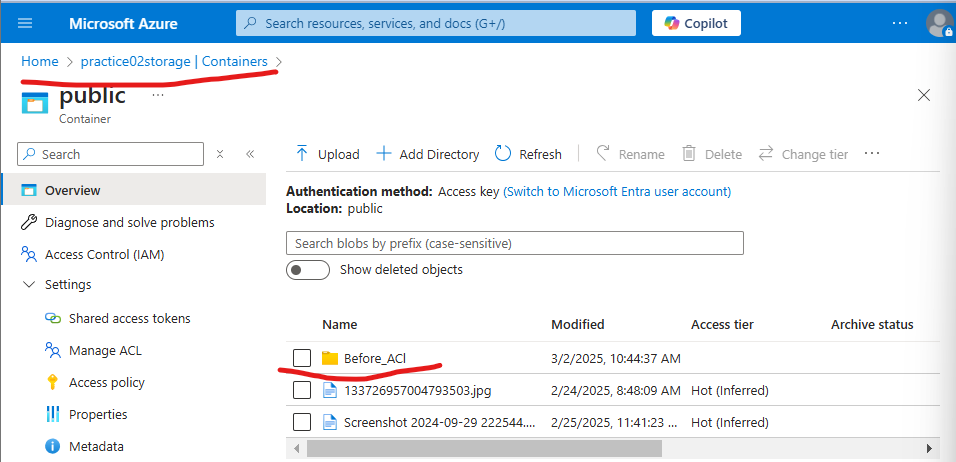
**6. Common Issues & Troubleshooting**

| **Issue** | **Possible Cause** | **Solution** |
| --- | --- | --- |
| Access denied | Missing **Execute (x)** permission on parent folder | Ensure execute (x) is set for the directory |
| Inherited permissions not applied | Default ACLs not set | Set **default ACLs** on the parent directory |
| User cannot modify files | Missing **Write (w)** permission | Assign w permission for the user |
| Cannot delete files | Lacking **Write (w)** and **Execute (x)** on parent directory | Grant wx on the parent directory |

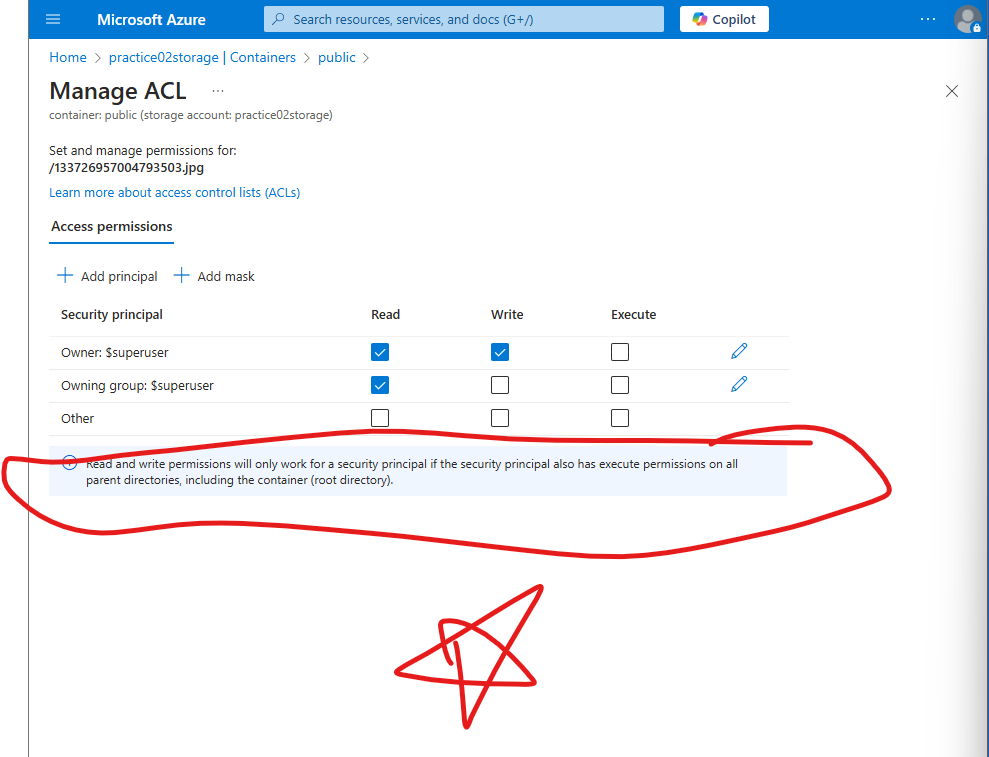
**7. Key Takeaways**

✅ ACLs provide **fine-grained file and folder-level access control**  
✅ They work **inside Azure Data Lake Storage Gen2 (HNS-enabled)**  
✅ ACLs **complement** RBAC but do not replace it  
✅ **Access ACLs** define permissions for a specific file/folder  
✅ **Default ACLs** ensure new files inherit correct permissions  
✅ Use **Azure CLI, PowerShell, Portal, or SDK** to manage ACLs

Practice:

So, I have a practice 02 stoarge account and here I am trying to figure out how ACL actually works: 

Here I created a directory and I had files and a driectory named Before\_ACL that are created before I manage permissions using ACL so I am trying to created an ACL for a user (I MEAN FOR A FILE) and here I observed this :

  
Read and write permissions will only work for a security principal if the security principal also has execute permissions on all parent directories, including the container (root directory).

Which means the

If you **give read (r) or write (w) permissions on a file**, but **the user doesn’t have execute (x) permission on the parent directories**, they **will not** be able to access the file.

**📌 Example:**

Imagine you have the following structure inside your **"public"** container:

pgsql

CopyEdit

/public (container)

│

├── /reports (folder) [NO execute permission]

│ ├── sales.xlsx (file) [User has read permission]

│

├── /logs (folder)

│ ├── error.log (file)

If **you give a user read (r) access to /reports/sales.xlsx** but **do not give execute (x) access on /public and /reports**, the user **will not** be able to open the file.