Steps to Create an Azure Storage Account in Azure Portal

Prerequisites

- An active Azure subscription (with permissions to create resources).
- Azure Portal access.
- Azure Storage Explorer (optional but recommended).
- Python 3.x installed + pip install azure-storage-blob.

Create an Azure Storage Account

1. Sign in
    a. Go to https://portal.azure.com and log in with your Azure credentials.

2. Create a Storage Account
    a. In the left-hand menu, click Create a resource → search for Storage account → click Create.

3. Basics TabSubscription:
    a. Select **Landing zone A1**
    b. Resource Group: Select **StroageDemoRg** (or create a new one).
    c. Storage account name: Enter a unique, lowercase name (e.g., lab1storage123).
    d. Region: (Europe) West Europe.
    e. Primary service: Azure Blob Storage or Azure Data Lake Storage Gen2.
    f. Performance: Standard (recommended for general-purpose v2).
    g. Redundancy: Locally-redundant storage (LRS).

4. Advanced Tab
    a. Require secure transfer for REST API operations: **Enabled** (recommended).
    b. Allow enabling anonymous access on individual containers: **Disable** (this will block you to later set a container to public access).
    c. Enable storage account key access: **Enabled** (default).
    d. Default to Microsoft Entra authorization in the Azure portal: **Enabled**.
    e. Minimum TLS version: 1.2.
    f. Hierarchical namespace: Leave **disabled** (only enable if you want Data Lake features).
    g. Access protocols: Leave defaults unless you specifically need **SFTP** or **NFS**.

5. Networking Tab

   a. Public network access: **Enable**.
   b. Public network access scope: **Enable** from all networks (for this lab, we allow broad access).
   c. Private endpoint: Leave unconfigured.

6. Data Protection Tab
   a. Enable point-in-time restore for containers: Leave **enabled** (requires extra features).
   b. Enable soft delete for blobs: **Enabled**, retention = 7 days.
   c. Enable soft delete for containers: **Enabled**, retention = 7 days.
   d. Enable soft delete for file shares: **Enabled**, retention = 7 days.

7. Encryption Tab
   a. Encryption type: Microsoft-managed keys (MMK) (default).
   b. Enable support for customer-managed keys: Leave at Blobs and files only.
   c. Enable infrastructure encryption: Leave unchecked.

8. Tags Tab (Optional)
   a. Add tags if needed (e.g., Environment = Lab).

9. Review + Create
   a. Review your configuration.
   b. Click Create.

10. Deployment
   a. Wait for the deployment to complete (usually 1–2 minutes).

Click Go to resource.
Checkpoint: You should see your new storage account’s Overview page showing it is Online.

Steps to Create a Blob Container in Azure Portal

1. Navigate to Your Storage Account:
   a. On the left-hand menu, select "Resource groups".
   b. Click on the resource group containing your storage account.
   c. In the resource group's resources list, click on your storage account.
2. Blob Service:
   a. Under the "Blob service" section in the storage account blade, click on "Containers".
3. Add a New Container:
   a. On the top menu of the "Containers" pane, click the "+ Container" button.
4. Configure the Container:

    a. Name: Provide a unique name for your blob container. Note that container names must be lowercase and can contain only letters, numbers, and hyphens.
    b. Public Access Level: Decide the level of public access:
        i. **Private** (no anonymous access): Only authorized users can access the blobs in the container.
        ii. **Blob** (anonymous read access for blobs only): Public read access for blobs, but container metadata and listings remain private.
        iii. **Container** (anonymous read access for containers and blobs): Public read access for both blobs and the container.

Note: It's recommended to keep containers private unless there's a specific need for public access. If public access is required, it's generally safer to use "Blob" rather than "Container" access level.

5. Create the Container:
    a. Click the "Create" button at the bottom to create your blob container.
    b. Verify the Container Creation:
    c. You'll be taken back to the "Containers" pane, where you should see your newly created container in the list.
6. Upload Blobs (Optional):
    a. If you want to immediately upload blobs into your new container:
    b. Click on the container name to open it.
    c. Click the "+ Upload" button at the top of the pane.
    d. Select and upload the desired blobs.

Remember, when creating and configuring blob containers, always consider the security implications of your choices, especially when setting the public access level. If public access is granted inadvertently, sensitive data might be exposed.

Generate a Shared Access Signature in Azure Portal

1. Login to Azure Portal:
    a. Navigate to https://portal.azure.com/ and sign in with your Azure credentials.
2. Access the Storage Account:
    a. In the left-hand navigation pane, search for "Storage accounts".
    b. Click on "Storage accounts" and then select the desired storage account from the list.
3. Shared Access Signature Section:
    a. In the storage account blade, under the "Security + networking" section, click on "Shared access signature".
4. Configure the SAS:
    a. Allowed services: Select the services (Blob, File, Queue, Table) for which you want the SAS to be applicable.

   b. Allowed resource types: Choose between **Service**, **Container**, and **Object**.
   c. Allowed permissions: Select the operations (Read, Write, Delete, etc.) you want to permit with the SAS.
   d. Start and expiry date/time: Define the time window during which the SAS will be valid.
   e. Allowed IP addresses (optional): To restrict the SAS to certain IP addresses, you can specify an IP range here.
   f. Allowed protocols: Choose between HTTPS and HTTP/HTTPS. For security reasons, it's recommended to use only HTTPS.
   g. Signing key: Choose which of the two storage account access keys (key1 or key2) to use for generating the SAS.
5. Generate SAS Token:
   a. After configuring the desired settings, click the "Generate SAS and connection string" button at the bottom.
6. Copy the SAS Token
   a. Once generated, you'll see the "Blob service SAS URL", "File service SAS URL", etc., based on the services you selected. You can copy the SAS token from these URLs.
   b. The "SAS token" field will also show the generated token, which you can copy directly.

Remember, a SAS token provides delegated access to the Azure Storage resources, so you should be cautious about the permissions you grant, and always set the shortest possible time span for which the SAS is valid. Also, never share your SAS token recklessly, as whoever has the token can access your storage resources as per the permissions given in the SAS.

Blob Operations

1. Soft Delete:
   a. Ensure "soft delete" is enabled for blobs. This provides a safety net for accidental blob deletions by retaining them for a specified period.
   b. This can be enabled in Data Management -> Data Protection -> Enable soft delete for blobs
2. Delete a specific blob.
3. Subsequently, demonstrate the recovery of the deleted blob, leveraging the soft delete feature.

Allow Public Blob Access and Test

1. Enable public access at the storage account level
   a. In the Azure Portal → go to your Storage Account.

    b. Under Settings, select Configuration.
    c. Set Allow Blob anonymous access = **Enabled**.
    d. Save changes.

2. Set the container to public
    a. Go to Data storage → Containers → select lab1-container.
    b. Click Change access level.
    c. Choose Blob (anonymous read access for blobs only).
    d. Save.

3. Verify public blob access
    a. Open the container → click on the file
    b. Copy the Blob URL (ends with the name of the file and extension).
    c. Open a private/incognito browser window and paste the URL.
    d. You should see the file displayed directly in the browser without login.

Check: If you can read the file in incognito, the configuration worked. If not, double-check both storage account Configuration and container Access Level.

Create a Lifecycle Management Policy

1. Go to your storage account → Data management → Lifecycle management.
    a. Click + Add rule:
    b. Rule name: moveToCool
    c. Scope: limit to lab1-container
    d. Action: Move blobs to Cool tier if last modified > 7 days.
    e. Save the policy.

Check: Policy appears under Lifecycle rules.

Reminder: Storage lifecycle management policy is an automated set of rules that help manage data over time by moving blobs between access tiers (Hot, Cool, Archive) or deleting them after a defined period; it's especially useful for reducing costs by archiving infrequently accessed files, meeting compliance requirements for data retention, and simplifying cleanup of old data such as logs, backups, or temporary files.

Connect to the storage account with Access Key

Open Storage Explorer

- Click Manage Accounts (left sidebar) and ensure you're signed out (to prove key-based auth works independently). Close the account pane.
- Click Add account (plug icon) → Storage account or service → Account name and key → Next.

- Enter:
  - Account name: youraccountname
  - Account key: paste Key1 (or Key2) from the portal.
  - Display name: anything helpful (e.g., youraccountname (key)).

Success check:

- In the Explorer tree, expand your account. You should see Blob Containers, File Shares, Queues, Tables.
- Open Blob Containers → create a test container, upload a small file.

Cleanup (optional)

- Right-click the attached account → Detach.

Connect with SAS (Shared Access Signature)

1. Generate a SAS at:
   a. Account level (grants access to multiple services), or
   b. Resource level (e.g., a single container, object).
2. From Azure Portal, generate SAS:
   a. Storage account → Security + networking → Shared access signature (account SAS), or
   b. Specific container → Shared access tokens (service SAS).
   c. Select the desired permissions

3. Copy the SAS URL **Connection string**

4. Connect via the Connection string
   a. In Storage Explorer, click Add account → Storage account or service → Shared access signature (SAS) → Next.
   b. Paste the connecton string
   c. Give a Display name and click Next → Connect.

Success check

- Browse only the services/containers permitted by the SAS.
- Try upload/list/delete operations that your SAS allows; blocked actions should fail (e.g., if write not granted).

Connect with Azure AD (RBAC)

Prereqs

- Appropriate Azure RBAC role assignments on the storage account or resource:
  - For Blob data: e.g., Storage Blob Data Reader, Storage Blob Data Contributor, or Owner (data plane).
  - For File shares: e.g., Storage File Data SMB Share Reader/Contributor and (optionally) NTFS permissions for AD DS if using Azure Files with AD.
  - Role assignment propagation may take a few minutes.

Steps

1. In Storage Explorer, click Manage Accounts → Add an account → sign in with your Azure AD user (select the right tenant/subscription).
2. Close the account pane. In the left tree, expand Subscriptions → your subscription → locate your Storage account.
3. Expand Blob Containers (or other services) and test access (list/upload/download according to your role).

Success check

- If you have Storage Blob Data Reader, you can list and download blobs but cannot upload/delete.
- With Storage Blob Data Contributor, you should be able to create containers, upload, and delete blobs.

Quick Comparison & Tips

- **Access Key**: Full control; simplest; highest risk if leaked. Rotate keys regularly.
- **SAS**: Principle of least privilege; time-bound; easiest to share safely. Prefer resource-level SAS when feasible.
- **Azure AD (RBAC)**: No secrets; auditable; best for enterprises. Ensure you grant data plane roles.