Steps to Create an Azure Storage Account in Azure Portal

1. Login to Azure Portal:
    a. Navigate to https://portal.azure.com/ and sign in with your Azure credentials.
2. Navigate to Storage Accounts:
    a. In the left-hand navigation pane, click on "Create a resource".
    b. In the "New" window, select "Storage" and then select "Storage account".
3. Basics Tab:
    a. Subscription: Choose your Azure subscription.
    b. Resource Group: Create a new one or select an existing resource group.
    c. Storage account name: Provide a unique name. This will be part of the domain name for your storage endpoint.
    d. Location: Choose a region that's appropriate for you or your application.
    e. Performance: Choose between Standard (HDD) and Premium (SSD).
    f. Account kind: Choose between StorageV2 (general-purpose v2), Storage (general-purpose v1), and BlobStorage.
    g. Replication: Choose a redundancy option (e.g., LRS, GRS, etc.).
4. Network Tab:
    a. Decide on the connectivity method: Public endpoint (all networks), Public endpoint (selected networks), or Private endpoint.
    b. If selecting "Public endpoint (selected networks)", you'll need to configure firewall and virtual network settings to restrict access.
5. Data Protection Tab (Optional but Recommended):
    a. Enable or disable soft delete for blobs, files, or file shares.
    b. Configure point-in-time restore for containers if required.
6. Advanced Tab:
    a. Configure settings such as secure transfer (HTTPS), Azure AD integration, large file shares, blob public access, and more based on your needs.
7. Tags Tab (Optional):
    a. Add any tags if you want to categorize or label your storage account.
8. Review + Create:
    a. Review all the configurations you've set.
    b. Click the "Create" button once you're ready. Azure will validate your settings, and after passing the validation, your storage account will start deploying.
9. Deployment:
    a. Wait for Azure to deploy your storage account. You'll receive a notification once it's completed.

Steps to Create a Blob Container in Azure Portal

1. Login to Azure Portal:
    a. Navigate to https://portal.azure.com/ and sign in with your Azure credentials.
2. 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.
3. Blob Service:
    a. Under the "Blob service" section in the storage account blade, click on "Containers".
4. Add a New Container:
    a. On the top menu of the "Containers" pane, click the "+ Container" button.
5. 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.
    c. 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.
6. 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.
7. 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.

Azure Blob Storage File Upload using Python

1. 1. Initial Setup:
    a. Python Environment: Set up a new Python project.
    b. Tip: Consider using virtual environments (e.g., venv or virtualenv) to manage dependencies.
2. 2. Package Installation:
    a. Ensure the following packages are installed:
        i. azure-identity: For Azure SDK authentication.
            1. Install with: pip install azure-identity
        ii. azure-storage-blob: SDK for Azure Blob Storage operations.
            1. Install with: pip install azure-storage-blob
3. Code Setup:
    a. Refer to the Lab1/main directory and integrate the provided Python code into your project.
4. Authentication:
    a. Familiarize yourself with different authentication mechanisms available in the Azure SDK.
    b. Note: Always prioritize secure authentication methods and avoid hardcoding credentials.

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.
2. Delete a specific blob.
3. Subsequently, demonstrate the recovery of the deleted blob, leveraging the soft delete feature.