**Step-by-Step Guide: Azure Storage Account Creation and Management**

**1. Create an Azure Storage Account**

**Using Azure Portal:**

* Go to **Azure Portal** (<https://portal.azure.com>)
* Click **Create a resource** → **Storage** → **Storage account**
* Fill in:
  + Subscription
  + Resource group (create new or existing)
  + Storage account name (unique globally)
  + Region
  + Performance: Standard or Premium (Standard for most cases)
  + Account kind: BlobStorage / StorageV2 (general purpose v2 recommended)
  + Replication: LRS (locally redundant), GRS (geo-redundant), ZRS, etc.
  + Access tier (Hot, Cool, Archive) - can be changed later for Blob data
* Click **Review + create** and then **Create**

**2. Explore Storage Account Options**

**Inside the storage account overview:**

* **Access keys** (two keys for redundancy) under **Security + networking**
* **Shared access signature (SAS)**
* **Firewalls and virtual networks**
* **Data protection** (soft delete, versioning, etc.)
* **Configuration** (performance, access tier)
* **Blob service** (containers, lifecycle management, replication)
* **File shares** (Azure Files)
* **Queues and Tables** (if enabled)
* **Networking** (private endpoints, firewall rules)

**3. Create Blob Container and Upload Blob**

* Go to **Containers** under Blob service
* Click **+ Container**, give a name and set public access level (Private/Blob/Container)
* Click **Upload**, select a file, upload

**4. Blob Access and Authentication Methods**

**Authentication methods to access blobs:**

* **Access Keys** (Full control keys of the storage account)
* **Shared Access Signature (SAS) tokens**
* **Azure Active Directory (Azure AD) Integration** (for RBAC permissions)
* **Anonymous public access** (if container set to public)

**5. Test Authentication Methods**

**Using Azure Storage Explorer:**

* Download and install **Azure Storage Explorer** (<https://azure.microsoft.com/en-us/features/storage-explorer/>)
* Connect to your Storage Account:
  + Using **Access Keys** (account name + key)
  + Using **SAS Token**
  + Using **Azure AD** (login with your Azure credentials)

Try upload, download, delete operations with different authentication modes.

**6. Provision Access Keys and Use Them**

* Go to **Access keys** in Azure Portal under your storage account
* Copy **key1** or **key2** and use in connection string like:

DefaultEndpointsProtocol=https;AccountName=<your\_account\_name>;AccountKey=<your\_access\_key>;EndpointSuffix=core.windows.net

* Use this in SDKs, Azure Storage Explorer, Azure CLI commands.

**7. Create Shared Access Signature (SAS)**

* Go to **Shared access signature** blade
* Select permissions (read, write, delete, list, add, create)
* Select services (Blob, File, Queue, Table)
* Select resource types (Service, Container, Object)
* Set start and expiry date/time
* Generate SAS token or URL

**Test:**

* Use SAS URL in browser or tools to access blobs with limited permissions and time scope.

**8. Create Stored Access Policy for SAS**

* In Blob Container → **Access policy** tab → Add policy
* Define permissions, start, expiry
* Generate SAS token linked to this policy

**Why use stored access policy?**

* You can revoke or modify SAS permissions centrally by updating policy

**9. Access Tiers and Use Cases**

* **Hot tier**: Frequently accessed data, lowest access latency, higher storage cost
* **Cool tier**: Infrequently accessed, lower cost but higher read/write cost and latency
* **Archive tier**: Long-term backup, rarely accessed, lowest cost, high latency for retrieval

**10. Apply Lifecycle Management Policy**

* Go to **Lifecycle management** blade
* Create rules like:
  + Move blobs older than X days to Cool or Archive
  + Delete blobs older than Y days
* Save and wait for policy to run (usually runs daily)

**11. Test Object Replication in Blob**

* Enable **Blob replication** like **Read-access geo-redundant storage (RA-GRS)** or **Object replication** for specific containers
* Test by uploading blob and verifying replication to secondary region
* Note: Object replication requires specific rules between source and destination storage accounts

**12. Create a File Share**

* Go to **File shares** blade
* Click **+ File share**, name it, specify quota
* Access file share via:
  + SMB protocol (map network drive in Windows)
  + Azure Storage Explorer
  + REST APIs

**13. Test File Share Functionality**

* Upload files via portal or explorer
* Map file share on Windows using SMB connection string:

php-template:-

\\<storageaccountname>.file.core.windows.net\<sharename>

* Test read/write/delete on mapped drive

**14. Create Azure File Sync**

**Steps:**

1. **Create a Storage Sync Service**
   * In Azure portal, search for **Storage Sync Services**
   * Create a new service
2. **Create a Sync Group**
   * Inside your Storage Sync Service, create a Sync Group
   * Add your Azure File Share as cloud endpoint
3. **Install Azure File Sync Agent on Windows Server**
   * Download the agent from Microsoft site
   * Install on your on-premises or Azure VM Windows Server
4. **Register Windows Server with Storage Sync Service**
   * Use Server Registration UI during installation
   * Register server to your Storage Sync Service
5. **Add Server Endpoint**
   * On registered server, add a folder as Server Endpoint to sync with the cloud endpoint (file share)
6. **Test Sync**
   * Add/change/delete files in local folder and verify sync with Azure File Share

**1. Create Storage Account (Azure CLI)**

az group create --name myResourceGroup --location eastus

az storage account create \

--name mystorageacct1234 \

--resource-group myResourceGroup \

--location eastus \

--sku Standard\_LRS \

--kind StorageV2 \

--access-tier Hot

**2. Create Blob Container**

az storage container create \

--account-name mystorageacct1234 \

--name mycontainer \

--public-access blob

**3. Upload Blob**

az storage blob upload \

--account-name mystorageacct1234 \

--container-name mycontainer \

--file ./localfile.txt \

--name remotefile.txt

**4. Get Access Keys (Azure CLI)**

az storage account keys list \

--resource-group myResourceGroup \

--account-name mystorageacct1234

Copy the key value for further use.

**5. Connect Using Access Key (Example with Azure Storage Explorer)**

* Open **Azure Storage Explorer**
* Click **Add Account** → Select **Use a storage account name and key**
* Enter:
  + Storage Account Name: mystorageacct1234
  + Storage Account Key: <copied key>

**6. Generate SAS Token (Azure CLI)**

az storage container generate-sas \

--account-name mystorageacct1234 \

--name mycontainer \

--permissions rwl \

--expiry 2030-01-01T00:00:00Z \

--https-only \

--output tsv

This gives you a SAS token string you can append to URLs.

**7. Create Stored Access Policy (Azure CLI)**

az storage container policy create \

--account-name mystorageacct1234 \

--container-name mycontainer \

--name readpolicy \

--permissions r \

--start 2025-05-01T00:00:00Z \

--expiry 2030-01-01T00:00:00Z

**8. Generate SAS Using Stored Access Policy**

az storage container generate-sas \

--account-name mystorageacct1234 \

--name mycontainer \

--policy-name readpolicy \

--https-only \

--output tsv

**9. Set Lifecycle Management Policy (JSON + CLI)**

Create a JSON file lifecycle.json with content:

json

{

"rules": [

{

"enabled": true,

"name": "move-to-cool-after-30-days",

"type": "Lifecycle",

"definition": {

"filters": {

"blobTypes": ["blockBlob"],

"prefixMatch": [""]

},

"actions": {

"baseBlob": {

"tierToCool": {

"daysAfterModificationGreaterThan": 30

}

}

}

}

}

]

}

Apply policy:

az storage account management-policy create \

--account-name mystorageacct1234 \

--resource-group myResourceGroup \

--policy @lifecycle.json

**10. Enable Geo-Replication (Example with GRS)**

This is done at storage account creation by choosing replication SKU:

az storage account create \

--name mystorageacctgrs \

--resource-group myResourceGroup \

--sku Standard\_GRS \

--location eastus

**11. Create Azure File Share**

az storage share create \

--account-name mystorageacct1234 \

--name myfileshare \

--quota 5120

**12. Map Azure File Share on Windows**

1. Get the connection string from Azure Portal or CLI:

az storage account show-connection-string \

--name mystorageacct1234 \

--resource-group myResourceGroup

1. Use this command in PowerShell to map:

net use Z: "\\mystorageacct1234.file.core.windows.net\myfileshare" /user:Azure\<storage-account-name> <storage-account-key>

**13. Azure File Sync Setup Overview**

* Create Storage Sync Service:

az storagesync service create --resource-group myResourceGroup --name mystoragesyncsvc

* Create Sync Group:

az storagesync sync-group create --resource-group myResourceGroup --storage-sync-service mystoragesyncsvc --name mysyncgroup

* Add Cloud Endpoint:

az storagesync cloud-endpoint create --resource-group myResourceGroup --storage-sync-service mystoragesyncsvc --sync-group mysyncgroup --storage-account mystorageacct1234 --azure-file-share myfileshare

* Install Azure File Sync agent on your Windows Server VM or on-premises server (manually from Microsoft site).
* Register Server using the agent GUI.
* Add Server Endpoint (local folder path to sync).