**Azure Storage**

**Lab 1: Create an Azure Storage Account**

**Objective**

Learn how to provision a storage account and understand the types (GPv1, GPv2, Blob).

**Steps (Azure Portal)**

1. Sign in to Azure Portal.
2. Navigate to **Storage accounts** → **+ Create**.
3. Fill in the details:
   * **Subscription**: Choose your subscription.
   * **Resource group**: Create new or select existing.
   * **Storage account name**: Must be globally unique, lowercase only.
   * **Region**: Choose nearest to you.
   * **Performance**: Standard (default) or Premium.
   * **Redundancy**: Choose **LRS** (for now).
   * **Account kind**: Select **General-purpose v2 (GPv2)**.
4. Review + Create → Deploy.

**Steps (Azure CLI)**

az storage account create \

--name mystorageaccount123 \

--resource-group my-rg \

--location eastus \

--sku Standard\_LRS \

--kind StorageV2

**Steps (PowerShell)**

New-AzStorageAccount -ResourceGroupName "my-rg" -Name "mystorageaccount123" `

-Location "EastUS" -SkuName "Standard\_LRS" -Kind StorageV2

**Lab 2: Blob Storage – Containers & Blobs**

**Objective**

Create containers and upload blobs (block, append, page).

**Steps (Portal)**

1. Go to your **storage account** → **Containers** → **+ Container**.
2. Enter name: sample-data.
3. Set **public access level** = Private.
4. Upload a file (e.g., demo.txt).

**Steps (CLI)**

# Create container

az storage container create --name sample-data --account-name mystorageaccount123

# Upload a blob (block blob default)

az storage blob upload --account-name mystorageaccount123 \

--container-name sample-data --name demo.txt --file demo.txt

**Steps (PowerShell)**

$ctx = (Get-AzStorageAccount -ResourceGroupName "my-rg" -Name "mystorageaccount123").Context

New-AzStorageContainer -Name "sample-data" -Context $ctx

Set-AzStorageBlobContent -File "demo.txt" -Container "sample-data" -Context $ctx

**Lab 3: ADLS Gen2 – Hierarchical Namespace**

**Objective**

Enable ADLS Gen2 features and create folder structures.

**Steps**

1. While creating the storage account, enable **Hierarchical namespace**.
2. Go to **Storage Explorer** (inside portal).
3. Create container: raw-data.
4. Inside it, create folder structure:
   * year=2025/month=08/day=20/
   * Upload files into respective folders.

**CLI Example**

az storage fs create -n raw-data --account-name mystorageaccount123

az storage fs directory create -n "year=2025/month=08/day=20" -f raw-data --account-name mystorageaccount123

**Lab 4: Table & Queue Storage Basics**

**Table Storage**

# Create table

az storage table create --name CustomerTable --account-name mystorageaccount123

**Queue Storage**

# Create queue

az storage queue create --name order-queue --account-name mystorageaccount123

Send a test message:

az storage message put --queue-name order-queue --account-name mystorageaccount123 --content "OrderID:12345"

**Lab 5: Azure Files (SMB & NFS)**

1. In storage account → **File shares** → **+ File share**.
2. Name: projectshare, set quota = 100GB.
3. Use **Connect** button to mount on Windows/Linux VM.

Windows Example:

net use Z: \\mystorageaccount123.file.core.windows.net\projectshare /u:mystorageaccount123 <storage-key>

**Lab 6: Authentication & Authorization**

**Shared Key**

* Get keys: **Storage account → Access keys**.

**SAS Token**

az storage container generate-sas --account-name mystorageaccount123 \

--name sample-data --permissions rwdl --expiry 2025-08-21T23:59Z --output tsv

**Azure AD**

* Assign role: **Storage Blob Data Contributor** to a user in **IAM**.

**Lab 7: Lifecycle Management**

1. Go to **Lifecycle management** in storage account.
2. Add rule:
   * If blob is older than 30 days → Move to Cool tier.
   * If older than 90 days → Delete.

**Lab 8: Data Redundancy**

When creating the account, test different redundancy options:

* **LRS, ZRS, GRS, RA-GRS**.  
  Observe replication in settings.

**Lab 9: Monitoring**

1. In **Monitoring** → Enable metrics & diagnostic settings.
2. Send logs to **Log Analytics** workspace.
3. Run sample queries:
4. AzureDiagnostics
5. | where ResourceType == "STORAGEACCOUNTS"

**Lab 10: Integration with Analytics**

**With ADF**

1. Open ADF → **Linked services** → New → Choose **Azure Data Lake Storage Gen2**.
2. Connect with Managed Identity.
3. Create pipeline to copy data from ADLS to SQL DB.

**With Databricks**

spark.conf.set("fs.azure.account.auth.type.mystorageaccount123.dfs.core.windows.net", "OAuth")