**Copy CSV Data from Azure Blob Storage to Azure SQL Database using Azure Data Factory**

**Objective:**

Move a CSV file from Azure Blob Storage into an Azure SQL Database using a no-code pipeline in ADF.

**Pre-requisites:**

* Azure Subscription
* Azure Blob Storage account with a container and sample CSV file
* Azure SQL Database with a pre-created table
* Azure Data Factory created and accessible

**Step-by-Step Lab Instructions**

**Step 1: Prepare the Environment**

**1.1 Create Azure Blob Storage**

* Go to Azure Portal.
* Search for **Storage accounts** > Click **+ Create**.
* Fill the required fields:
  + Subscription, Resource Group
  + Name: adfdemostorage
  + Region: Your preferred region
  + Performance: Standard
  + Redundancy: Locally-redundant storage (LRS)
* Click **Review + Create** > then **Create**.

**1.2 Create a Container**

* Navigate to the created storage account > go to **Containers** > **+ Container**.
* Name it input-data > Set public access level to **Private**.
* Click **Create**.

**1.3 Upload CSV File**

* Inside input-data, click **Upload**.
* Upload a file like employees.csv:

ID,Name,Department

1,John Doe,Finance

2,Jane Smith,HR

3,Bob Johnson,IT

**Step 2: Create Azure SQL Database**

**2.1 Create SQL Server**

* Go to **SQL servers** > **+ Create**.
* Fill:
  + Server name: adfdemosqlserver
  + Admin login: sqladmin
  + Password: YourSecurePassword123
  + Location: Same as blob storage
* Click **Review + Create** > then **Create**.

**2.2 Create SQL Database**

* Go to **SQL databases** > **+ Create**.
* Fill:
  + Database name: EmployeeDB
  + Server: select the one you just created
  + Use SQL elastic pool: No
  + Compute + storage: Basic tier
* Click **Review + Create** > then **Create**.

**2.3 Create a Table**

* Connect to your SQL database using **Query Editor (Preview)** or **SQL Server Management Studio (SSMS)**.
* Run this SQL:

CREATE TABLE Employees (

ID INT,

Name NVARCHAR(100),

Department NVARCHAR(100)

);

**Step 3: Create Azure Data Factory**

**3.1 Create ADF**

* Go to **Azure Data Factory** > **+ Create**.
* Fill:
  + Name: adfdemo
  + Region: Same as storage and SQL
  + Version: V2
  + Git Configuration: Not required for now
* Click **Review + Create** > then **Create**.

**3.2 Open ADF Studio**

* After creation, click **Author & Monitor** to launch ADF Studio.

**Step 4: Create Linked Services**

**4.1 Linked Service to Azure Blob Storage**

* In ADF Studio, go to **Manage** > **Linked Services** > **+ New**.
* Search for **Azure Blob Storage**.
* Name: LS\_BlobStorage
* Connect using:
  + Authentication type: **Account key**
  + Storage account: Select your storage
* Test connection > Create.

**4.2 Linked Service to Azure SQL Database**

* Click **+ New** again > Search for **Azure SQL Database**.
* Name: LS\_AzureSQLDB
* Server name: Select your server
* Authentication type: SQL Authentication
* Username/Password: Use what you configured
* Test connection > Create.

**Step 5: Create Datasets**

**5.1 Dataset for Source (CSV in Blob)**

* Go to **Author** > **+ Add new Dataset**
* Choose **Azure Blob Storage** > Format: **DelimitedText** > Continue
* Name: DS\_Blob\_CSV
* Linked service: LS\_BlobStorage
* File path:
  + Container: input-data
  + File: employees.csv
* File format settings:
  + First row as header: Checked
  + Column delimiter: Comma
* Click **OK** > Publish.

**5.2 Dataset for Sink (SQL Table)**

* Add another Dataset > Choose **Azure SQL Database** > Table
* Name: DS\_SQL\_Employees
* Linked service: LS\_AzureSQLDB
* Select table: Employees
* Click **OK** > Publish.

**Step 6: Create the Pipeline**

**6.1 Create Pipeline**

* Go to **Author** > Pipelines > **+ New pipeline**
* Name it: CopyEmployeesPipeline

**6.2 Add Copy Data Activity**

* In the pipeline canvas, search and drag **Copy data** activity.
* Name: CopyFromBlobToSQL

**6.3 Configure Source**

* Click **Source** tab
* Select dataset: DS\_Blob\_CSV

**6.4 Configure Sink**

* Click **Sink** tab
* Select dataset: DS\_SQL\_Employees

**6.5 (Optional) Mapping**

* Click **Mapping** tab > click **Import schemas**
* Verify auto-mapping is correct:
  + ID → ID
  + Name → Name
  + Department → Department

**6.6 Validate & Publish**

* Click **Validate All** (top bar) > Fix if errors
* Click **Publish All**

**Step 7: Trigger the Pipeline**

**7.1 Manual Trigger**

* In the pipeline editor > Click **Add trigger** > **Trigger now**
* Confirm trigger

**7.2 Monitor Run**

* Go to **Monitor** tab > Click on pipeline run
* Verify if status = Succeeded

**Step 8: Verify Data in SQL**

* Go to **SQL Database** > **Query editor**
* Run:

SELECT \* FROM Employees;

You should see:

| **ID** | **Name** | **Department** |
| --- | --- | --- |
| 1 | John Doe | Finance |
| 2 | Jane Smith | HR |
| 3 | Bob Johnson | IT |