Hands-On Lab 9: ADF → Databricks Integration (Simple & Beginner-Friendly)

Lab Created by : Dr. Sandeep Kumar Sharma

🔷What You Will Learn

Why integrate ADF with Databricks?

Azure Data Factory is an orchestration and ETL/ELT service. Databricks provides a fast, scalable Apache Spark environment ideal for heavy transformations, ML training, and complex analytics. Integrating ADF and Databricks lets you orchestrate data movement with ADF and run scalable compute in Databricks where heavy lifting belongs.

Key concepts covered in this lab:

1. **Linked Service to Databricks** — how ADF connects securely to Databricks using personal access token (PAT) or managed identity. You'll learn what information is required (workspace URL, token) and how authentication differs from storage linked services.

2. **Databricks Linked Service vs Cluster configuration** — ADF can submit jobs to existing interactive clusters, new job clusters, or use all-purpose clusters. We'll show a simple job-cluster approach (ADF triggers Databricks job which creates a short-lived cluster).

3. **Notebook activity** — Using ADF's Databricks Notebook activity to run a notebook stored in DBFS or workspace. You'll learn how to pass parameters from ADF into notebooks (for dynamic behavior) and how to retrieve results via notebook exit values or by writing outputs to storage.

4. **Data flow of the lab** — ADF will move sample CSV to ADLS/Blob, then call a Databricks notebook which reads the file, performs a simple transformation (e.g., convert CSV → Parquet + add a column), writes output back to storage. Finally ADF validates the output.

5. **Simple error handling** — Basic pattern: capture activity status, use If Condition activity to route on success/failure and write logs.

6. **Best practices (brief)** — use short-lived job clusters for cost control, parameterize paths, write results to storage as Parquet/Delta, and keep idempotent notebooks.

✔️Concepts You Will Understand

1. **How ADF connects to Databricks** using a Linked Service.

2. **How to run a Databricks Notebook from ADF** using the Databricks Notebook Activity.
3. **How to pass parameters** from ADF → Notebook.
4. **How Databricks processes a file** and writes output back to storage.
5. **A complete flow**: ADF → Databricks → Storage.

Step 1 — Prepare Storage and Upload File

Create folder: `lab9/input/`

Upload file **sample.csv**:

```
id,name,salary
1,Amit,50000
2,Rekha,60000
3,Rahul,70000
```

Step 2 — Create Linked Services

1. Storage Linked Service

- Go to **Manage → Linked Services → New**
- Choose **ADLS Gen2 / Blob Storage**
- Authenticate using *account key* or *managed identity*

2. Databricks Linked Service

- Choose **Azure Databricks**
- Enter **workspace URL**
- Enter **Personal Access Token (PAT)**

  This allows ADF to submit jobs to your Databricks workspace.

Step 3 — Create Databricks Notebook

Open **Azure Databricks → Workspace → Create Notebook**

Create notebook named: `lab9_notebook`

Use Python and paste this simple code:

```
# Input parameters from ADF
input_path = dbutils.widgets.get("input_path")
output_path = dbutils.widgets.get("output_path")

# Read CSV
input_df = spark.read.option("header", "true").csv(input_path)

# Add new column
output_df = input_df.withColumn("processed_by", lit("ADF_Databricks_Lab9"))

# Write result
output_df.write.mode("overwrite").parquet(output_path)

print("Processing completed!")
```

Add widgets in top cell:

```
dbutils.widgets.text("input_path", "")
dbutils.widgets.text("output_path", "")
```

Step 4 — Create Pipeline in ADF

Create pipeline: **pl_adf_dbx_lab9**

Add Activity → Databricks Notebook

- Choose notebook path: `/Users/your.email/lab9_notebook`
- Select your Databricks Linked Service

Add Parameters

Add 2 parameters:

- `input_path` → `adls://lab9/input/sample.csv`
- `output_path` → `adls://lab9/output/processed/`

ADF will pass these into the notebook.

Step 5 — Debug & Run

1. Click **Debug**

2. Open **Monitor** to watch progress
3. Check output folder: `lab9/output/processed/`

You should see Parquet files.

Expected Output (Parquet → CSV view)

```
id,name,salary,processed_by
1,Amit,50000,ADF_Databricks_Lab9
2,Rekha,60000,ADF_Databricks_Lab9
3,Rahul,70000,ADF_Databricks_Lab9
```

Lab Completed

This lab covered ONLY the basics of ADF → Databricks integration in the simplest possible way.

✔️ADF calls Databricks notebook ✔️Notebook transforms data ✔️Output stored back to storage