

# CELEBAL SUMMER INTERNSHIP

## PROJECT REPORT



**Project Title:** Change Data Capture (CDC) ETL Pipeline using Azure Data Factory and Databricks

**Submitted by:** Sanchi Mishra

**Domain:** Data Engineering

**Student ID:** CT\_CSI\_DE\_5042

**Internship Duration:** 2 months

**College Name:** Manipal University Jaipur

**Project Description:** This project focuses on implementing Change Data Capture (CDC) using Databricks, a unified data analytics platform. The primary objective is to efficiently identify and process data changes (inserts, updates, deletes) in source systems and reflect them in the target data lake or data warehouse in near real-time. Leveraging Apache Spark, Delta Lake, and streaming capabilities within Databricks, the project aims to build a scalable and reliable CDC pipeline that supports incremental data ingestion, ensures data consistency, and minimizes processing overhead.

Github Repository: [Celebal Project](#)

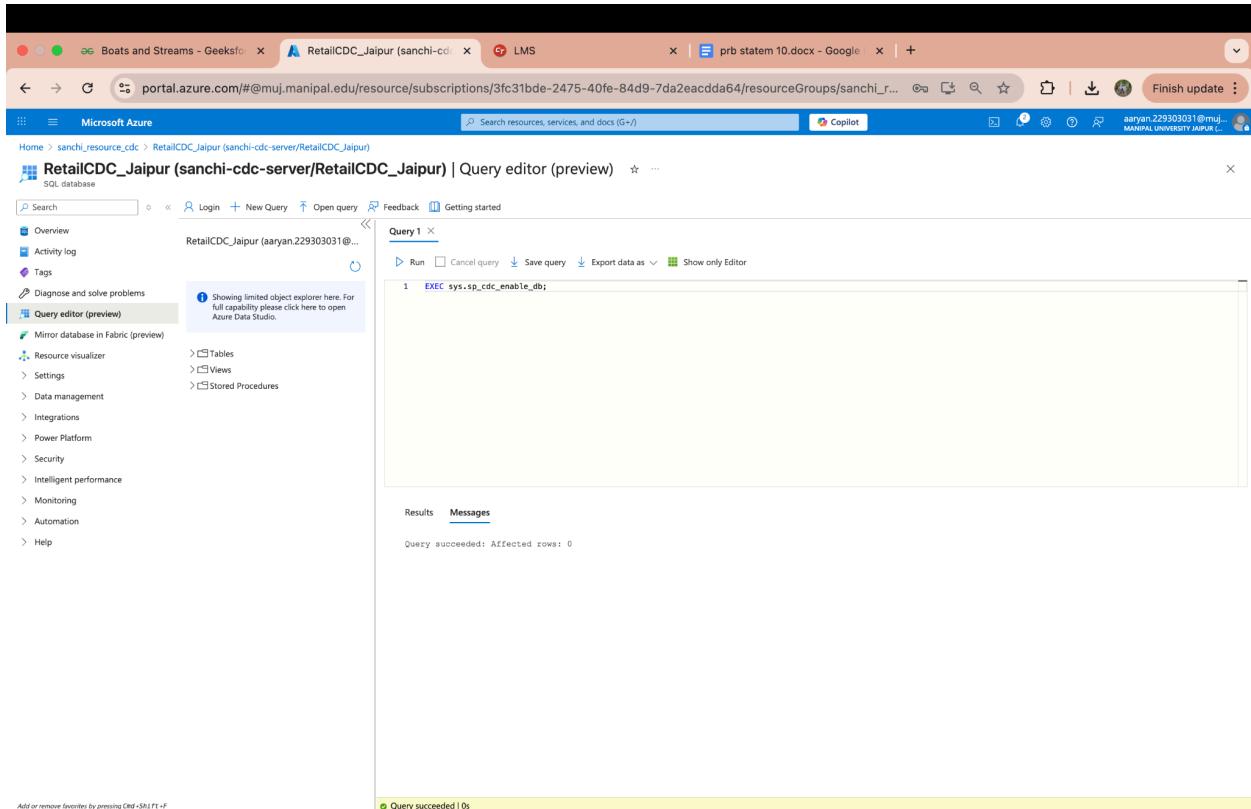
## 1. SQL Database Creation: RetailCDC\_Jaipur

The screenshot shows the Microsoft Azure portal interface for creating a new SQL database. The top navigation bar includes 'Microsoft Azure', a search bar, and user information. The main page title is 'RetailCDC\_Jaipur (sanchi-cdc-server/RetailCDC\_Jaipur)'. The left sidebar lists various management options like Overview, Activity log, Tags, and Resource visualizer. The central content area displays the database's properties, including its resource group ('sanchi\_resource\_cdc'), status ('Online'), location ('Southeast Asia'), and subscription details ('Azure for Students'). It also shows server name ('sanchi-cdc-server.database.windows.net'), connection strings, pricing tier ('Free - General Purpose - Serverless: Gen5, 2 vCores'), and other configuration details. Below the properties, there's a section titled 'Start working with your database' with four steps: 'Configure access', 'Connect to application', 'Start developing', and 'Mirror database in Fabric'. Each step has a 'Configure' button and a link to 'Learn more'. At the bottom, there's a note about adding or removing favorites.

## 2. Resource Creation: sanchi\_resource\_cdc

The screenshot shows the Microsoft Azure portal interface for creating a new resource group named 'sanchi\_resource\_cdc'. The top navigation bar includes 'Microsoft Azure', a search bar, and user information. The main page title is 'sanchi\_resource\_cdc'. The left sidebar lists various management options like Overview, Activity log, Access control (IAM), Tags, and Resource visualizer. The central content area displays the resource group's properties, including its subscription ('Azure for Students'), subscription ID ('3fc31bde-2475-40fe-84d9-7da2eacdd64'), and location ('Southeast Asia'). It also shows deployment status ('1 Succeeded') and deployment count ('1'). Below the properties, there's a table listing resources: 'RetailCDC\_Jaipur' (SQL database) and 'sanchi-cdc-server' (SQL server). A modal window on the right provides options to switch between a list view and a summary chart view of resource counts. At the bottom, there are navigation links for 'Previous', 'Page 1 of 1', 'Next >', and a 'Give feedback' link.

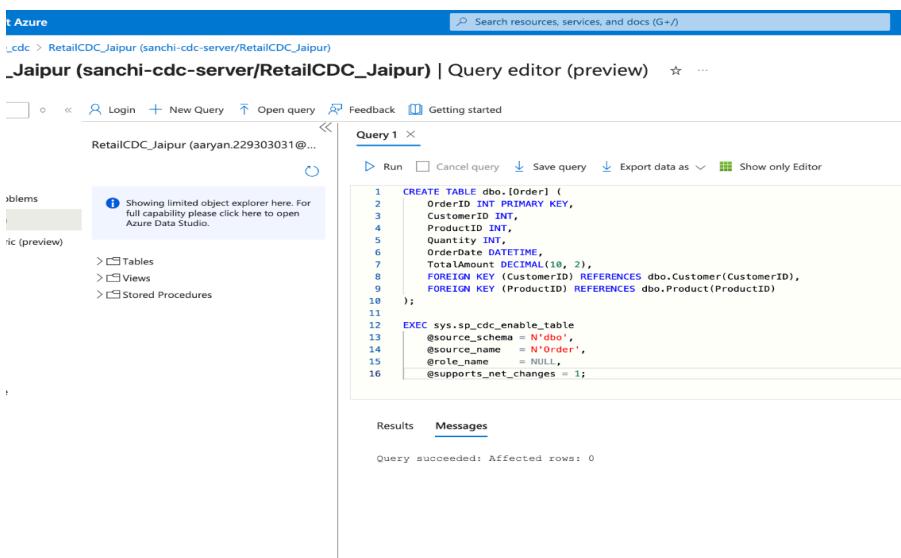
### 3. Enable CDC at Database Level



The screenshot shows the Microsoft Azure portal interface for a SQL database named 'RetailCDC\_Jaipur'. The left sidebar contains navigation links like Overview, Activity log, Tags, Diagnose and solve problems, and a highlighted 'Query editor (preview)'. The main area is a 'Query editor (preview)' titled 'RetailCDC\_Jaipur (sanchi-cdc-server/RetailCDC\_Jaipur) | Query editor (preview)'. It displays a single query line: 'EXEC sys.sp\_cdc\_enable\_db;'. Below the query, the status bar shows 'Query succeeded | 0s'.

```
EXEC sys.sp_cdc_enable_db;
```

### 4. Creating Tables (Customer, Order, Product, Inventory) and Enabling CDC (attaching just one eg)



The screenshot shows the Microsoft Azure portal interface for a SQL database named 'RetailCDC\_Jaipur'. The left sidebar contains navigation links like Overview, Activity log, Tags, Diagnose and solve problems, and a highlighted 'Query editor (preview)'. The main area is a 'Query editor (preview)' titled '\_Jaipur (sanchi-cdc-server/RetailCDC\_Jaipur) | Query editor (preview)'. It displays two queries. The first query creates the 'Order' table with columns: OrderID (INT PRIMARY KEY), CustomerID (INT), ProductID (INT), Quantity (INT), OrderDate (DATETIME), and TotalAmount (DECIMAL(10, 2)). It includes foreign key constraints linking CustomerID to 'Customer' and ProductID to 'Product'. The second query, starting at line 12, executes 'sys.sp\_cdc\_enable\_table' with parameters: @source\_schema = 'dbo', @source\_name = 'Order', @role\_name = NULL, and @supports\_net\_changes = 1. Below the queries, the status bar shows 'Query succeeded | 0s'.

```
CREATE TABLE dbo.[Order] (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    ProductID INT,
    Quantity INT,
    OrderDate DATETIME,
    TotalAmount DECIMAL(10, 2),
    FOREIGN KEY (CustomerID) REFERENCES dbo.Customer(CustomerID),
    FOREIGN KEY (ProductID) REFERENCES dbo.Product(ProductID)
);

EXEC sys.sp_cdc_enable_table
    @source_schema = 'dbo',
    @source_name = 'Order',
    @role_name = NULL,
    @supports_net_changes = 1;
```

## 5. Azure Data Factory Creation and Deployment

The screenshot shows the Microsoft Azure Deployment Overview page for a deployment named "Microsoft.DataFactory-20250716235406". The status is "Your deployment is complete". Deployment details include a name of "Microsoft.DataFactory-20250716235406", a subscription of "Azure for Students", and a resource group of "sanchi\_resource\_cdc". The start time was 7/16/2025, 11:56:22 PM, and the correlation ID is bfe59b9a-9de6-4426-a18b-acc20d06b750. The page also includes sections for "Deployment details" and "Next steps", and a "Go to resource" button.

## 6. Azure Linked Service

The screenshot shows the "Edit linked service" configuration for an "Azure SQL Database" type. The "Name" is set to "AzureSqlDb\_Linked". The "Type" is "Azure SQL". The "Connect via integration runtime" dropdown is set to "AutoResolveIntegrationRuntime". The "Version" is set to "2.0 (Recommended)". The "Account selection method" is "Enter manually". The "Fully qualified domain name" is "sanchi-cdc-server.database.windows.net", the "Database name" is "RetailCDC\_Jaipur", and the "Authentication type" is "SQL authentication". The "User name" is "sqladmin". The "Password" field is empty. There are "Apply" and "Cancel" buttons at the bottom, and a "Test connection" link.

## 7. Data Bricks WorkSpace Creation

The screenshot shows the Microsoft Azure Deployment Overview page for a deployment named "sanchi\_resource\_cdc\_cdc-databricks". The status is "Your deployment is complete". Key details include:

- Deployment name: sanchi\_resource\_cdc\_cdc-databricks
- Subscription: Azure for Students
- Resource group: sanchi\_resource\_cdc
- Start time: 7/17/2025, 12:40:01 AM
- Correlation ID: 3a50ec95-5fa1-452e-8348-1e3413abddcc

Navigation links include "Overview", "Inputs", "Outputs", "Template", "Deployment details", and "Next steps". A "Go to resource" button is present at the bottom.

## 8. Data Bricks Notebook Glimpse (Full Notebook attached in Github Repository)

The screenshot shows a Databricks Notebook titled "cdc\_etl\_pipeline". The notebook interface includes a sidebar with options like New, Workspace, Recents, Catalog, Jobs & Pipelines, Compute, Data Engineering, Job Runs, AI/ML, Playground, Experiments, Features, Models, and Serving. The main area displays the following Python code:

```
# Import required libraries
from pyspark.sql.functions import col
from datetime import datetime

# Define JDBC connection parameters
jdbc_url = "jdbc:sqlserver://sanchi-cdc-server.database.windows.net:1433;database=RetailCDC_Jaipur"
jdbc_properties = {
    "user": "sqladmin",
    "password": "Sanjal@12345",
    "driver": "com.microsoft.sqlserver.jdbc.SQLServerDriver"
}

# Load the Order table as an example (use same for Customer/Product/Inventory)
try:
    df_order = spark.read.format("jdbc") \
        .option("url", jdbc_url) \
        .option("dbtable", "dbo.Order") \
        .option("user", jdbc_properties["user"]) \
        .option("password", jdbc_properties["password"]) \
        .option("driver", jdbc_properties["driver"]) \
        .load()

    # Filter or transform (simulate CDC by filtering on recent records)
    df_cdc = df_order.filter(col("OrderDate") >= "2025-07-16")

    # Save to file path (change as needed)
    output_path = "dbfs:/CDC_output/order_" + datetime.now().strftime("%Y%m%d_%H%M") + ".parquet"
    df_cdc.write.mode("overwrite").parquet(output_path)

    print("CDC data extraction and storage complete.")
except Exception as e:
    print(f"Error: {e}")

# (1) Spark Jobs
# df_cdc: pyspark.sql.dataframe.DataFrame = [OrderID: integer, CustomerID: integer ... 4 more fields]
# df_order: pyspark.sql.dataframe.DataFrame = [OrderID: integer, CustomerID: integer ... 4 more fields]
CDC data extraction and storage complete.
```

## 9. Linking Database to Azure

The screenshot shows the Microsoft Azure Data Factory pipeline editor. On the left, the 'Factory Resources' sidebar lists 'Pipelines' (1), 'Datasets' (5), and other components. The main area displays 'pipeline1' with an 'Activities' list containing 'Move and transform', 'Synapse', 'Azure Data Explorer', 'Azure Function', 'Batch Service', and 'Databricks'. Under 'Databricks', there are options for 'Notebook', 'Jar', 'Python', and 'Job (Preview)'. A 'New linked service' dialog is open on the right, titled 'New linked service (Azure Databricks)'. It includes fields for 'Name' (SanchiDatabricks\_Linked), 'Description', 'Connect via integration runtime' (AutoResolveIntegrationRuntime selected), 'Account selection method' (From Azure subscription selected), 'Azure subscription' (Azure for Students selected), 'Databricks workspace' (cdc-databricks selected), 'Select cluster' (Existing interactive cluster selected), 'Databrick Workspace URL' (https://adb-3243773246735880.0.azuredatabricks.net), 'Authentication type' (Access Token selected), and 'Access token' (selected). A 'Create' button is at the bottom, with a 'Connection successful' message and a 'Test connection' link.

## 10. Notebook in Azure Published

The screenshot shows the Microsoft Azure Data Factory pipeline editor. The 'Factory Resources' sidebar is identical to the previous screenshot. The main area displays 'pipeline1' with the same 'Activities' list. A 'Notebook' activity named 'Sanchi\_Notebook\_DB' is now present in the pipeline. A 'General' tab for this activity is open, showing its configuration: 'Name' (Sanchi\_Notebook\_DB), 'Activity state' (Activated selected), 'Timeout' (0.12:00:00), 'Retry' (0), and 'Retry interval (sec)' (30). A 'Publish all' button is visible at the top, and a 'Publishing completed' message is displayed in the top right corner.

## 11. Data Bricks Pipeline Ran Successfully

The screenshot shows the Microsoft Azure Databricks interface under the 'Jobs & Pipelines' section. On the left, there's a sidebar with options like Workspace, Recents, Catalog, Jobs & Pipelines, Compute, Data Engineering, and Job Runs. The 'Job Runs' tab is selected. In the main area, there are three cards: 'Ingestion pipeline' (Ingest data from popular apps, databases and file sources), 'ETL pipeline' (Build ETL pipelines using SQL and Python), and 'Job' (Orchestrate notebooks, pipelines, queries and more). Below these cards, there are two tabs: 'Jobs & pipelines' (selected) and 'Job runs'. Under 'Job runs', there are two rows of data. The first row has a status of 'Failed' (red dot) and the second has a status of 'Succeeded' (green dot). Both rows show the start time as 'Jul 17, 2025, 03:25 PM', the job name as 'ADF\_cdcadfccelebal\_pipeline1\_S...', the run as 'By runs submit API', duration '32s', status 'Succeeded', error code '0', and run parameters. The timeline at the bottom shows the run starting at 16 Jul, 12 AM and ending at 17 Jul, 12 AM.

## 12. Notebook in Azure Linked and Published

The screenshot shows the Microsoft Azure Data Factory pipeline editor. The left sidebar lists 'Factory Resources' including Pipelines, Datasets, Data flows, and Power Query. A pipeline named 'pipeline1' is selected. The main workspace shows a pipeline activity named 'Notebook'. The 'Activities' pane on the right lists various activity types: Move and transform, Synapse, Azure Data Explorer, Azure Function, Batch Service, Databricks, Notebook (selected), Jar, Python, Job (Preview), Data Lake Analytics, General, HDInsight, Iteration & conditionals, and Machine Learning. The 'Notebook' activity is highlighted with a blue border. The 'General' tab of the configuration pane shows the 'Name' field set to 'Sanchi\_Notebook\_DB', 'Activity state' set to 'Activated' (radio button selected), and 'Timeout' set to '0:12:00:00'. A message in the top right corner says 'Publishing completed' and 'Successfully published'.

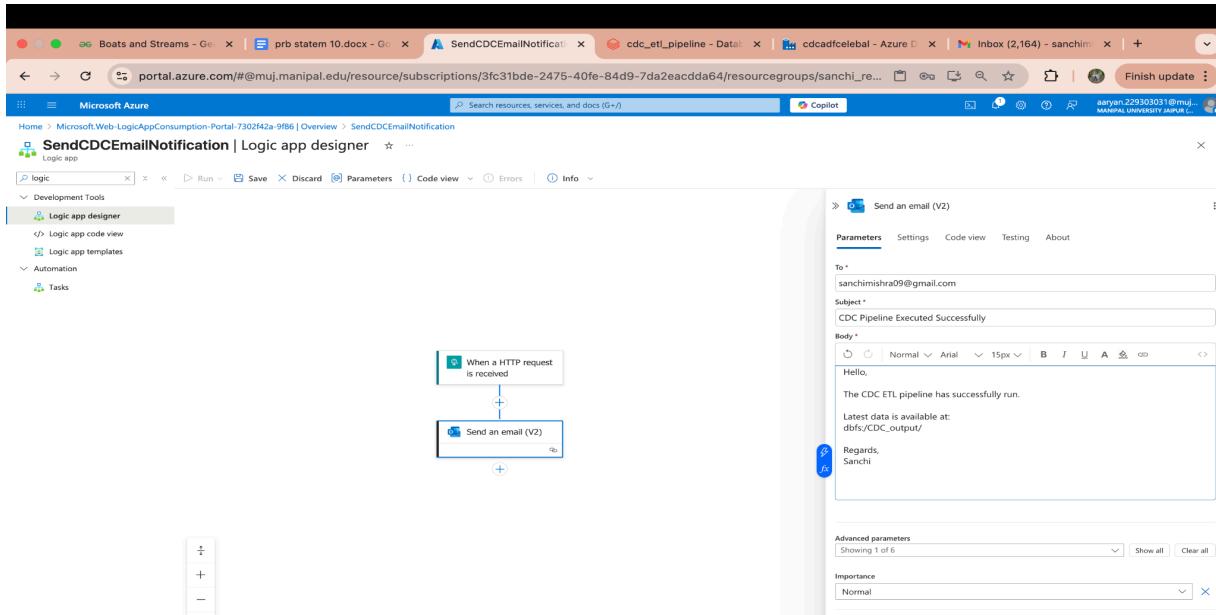
### 13. Pipeline Ran Successfully

The screenshot shows the Microsoft Azure Data Factory interface. The left sidebar is titled 'Runs' and includes options like 'Pipeline runs', 'Trigger runs', 'Change Data Capture', 'Runtimes & sessions', 'Integration runtimes', 'Data flow debug', 'Notifications', and 'Alerts & metrics'. The main area is titled 'All pipeline runs > pipeline1 - Activity runs'. A modal window titled 'Run Succeeded' is open, stating 'Successfully ran pipeline1 (Pipeline). View pipeline run'. Below the modal, the pipeline run details are shown: Pipeline run ID: 3a1c2325-37fe-4c26-b5a2-9965a3c74061, Status: Succeeded, Activity name: Sanchi\_Notebook\_DB, Duration: 2m 35s, User properties: AutoResolveIntegrationRuntime (Southeast Asia), and Activity ID: 461. The pipeline run Gantt chart shows two tasks: 'Notebook' and 'Sanchi\_Notebook\_DB', both marked with green checkmarks.

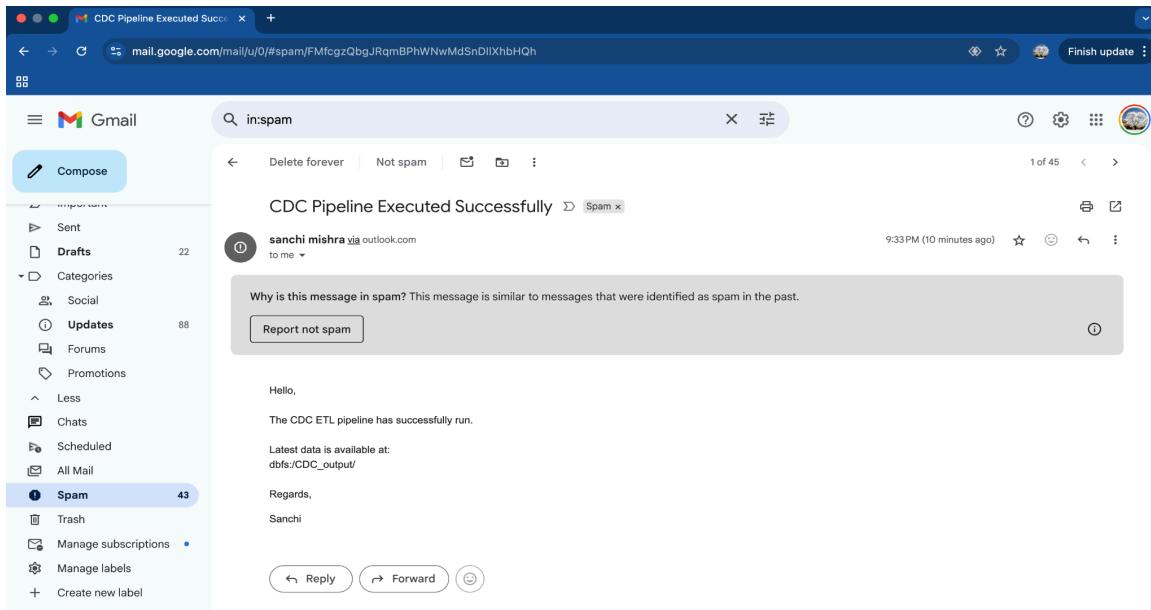
### 14. Email and Notebook Connected and Ran

This screenshot is similar to the previous one, showing the Microsoft Azure Data Factory interface. The left sidebar and main area are identical. The pipeline run details are: Pipeline run ID: 4aa5e7b9-e94c-42a2-b48a-f0f79ef0a0f8, Status: Succeeded, Activity name: Sanchi\_Notebook\_DB, Duration: 34s, User properties: AutoResolveIntegrationRuntime (Southeast Asia), and Activity ID: 0b1. The pipeline run Gantt chart now includes a second task, 'Web SendEmailNotification', which is also marked with a green checkmark, indicating both activities in the pipeline have completed successfully.

## 15. Email Setup



## 16. Email Received



## 17. A few more screenshots from the pipelines: Datasets, Triggers, Emails, Deployments and Outputs

Screenshot of Microsoft Azure Data Factory pipeline editor:

```

graph LR
    A[Customer Copy] --> B[Inventory Copy]
    B --> C[Order Copy]
    C --> D[Product Copy]
  
```

The pipeline consists of four sequential "Copy data" activities:

- Customer Copy (CustomerDataset)
- Inventory Copy (InventoryDataset)
- Order Copy (OrderDataset)
- Product Copy (ProductDataset)

**Datasets**

- CustomerDataset
- CustomerOutputDataset
- CustomerOutputDS
- InventoryDataset
- InventoryOutputDS
- OrderDataset
- OrderOutputDS
- ProductDataset
- ProductOutputDS

**Linked services**

Linked service defines the connection information to a data store or compute. [Learn more](#)

Name ↑	Type ↑	Related ↑	Annotations ↑
AzureBlobStorage1	Azure Blob Storage	4	
AzureDatabricksDeltaLake1	Azure Databricks Delta Lake	0	
AzureSqlDb_Linked	Azure SQL Database	4	
AzureStorageCDC_Linked	Azure Data Lake Storage Gen2	1	
SanchiDatabricks_Linked	Azure Databricks	1	

The screenshot shows the Azure Data Factory interface. On the left, the 'Triggers' blade is open, displaying a list with one item: 'HourlyTrigger' (Type: Schedule). On the right, a detailed view of 'HourlyTrigger' is shown in the 'Edit trigger' dialog. The 'Name' field is set to 'HourlyTrigger'. The 'Type' is selected as 'ScheduleTrigger'. The 'Start date' is set to '7/16/2025, 5:15:00 PM'. The 'Time zone' is set to 'Chennai, Kolkata, Mumbai, New Delhi (UTC+5:30)'. Under 'Recurrence', it is set to 'Every 60 Minute(s)' with the 'Specify an end date' checkbox checked, and the end date is '7/19/2025, 7:00:00 PM'. The 'Annotations' section has a '+ New' button. The 'Status' section shows 'Started' as the current status.

I am pleased to share that I have successfully completed the **Change Data Capture (CDC) ETL Pipeline Project** using Azure Data Factory and Azure Databricks

. The solution includes:

- CDC-enabled SQL Server tables
- Automated ETL using Databricks notebooks
- ADF pipelines for scheduling and integration
- Email notifications on successful runs
- Export of updated data to Azure Blob Storage

I have also attached a few screenshots as proof of successful pipeline runs, output verification, and final email notifications and a github repository.

I am truly grateful for the opportunity to work on this project and enhance my practical knowledge of modern data engineering pipelines. Looking forward to receiving your feedback and contributing further.

Thank you!

Sanchi Mishra