

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](#)

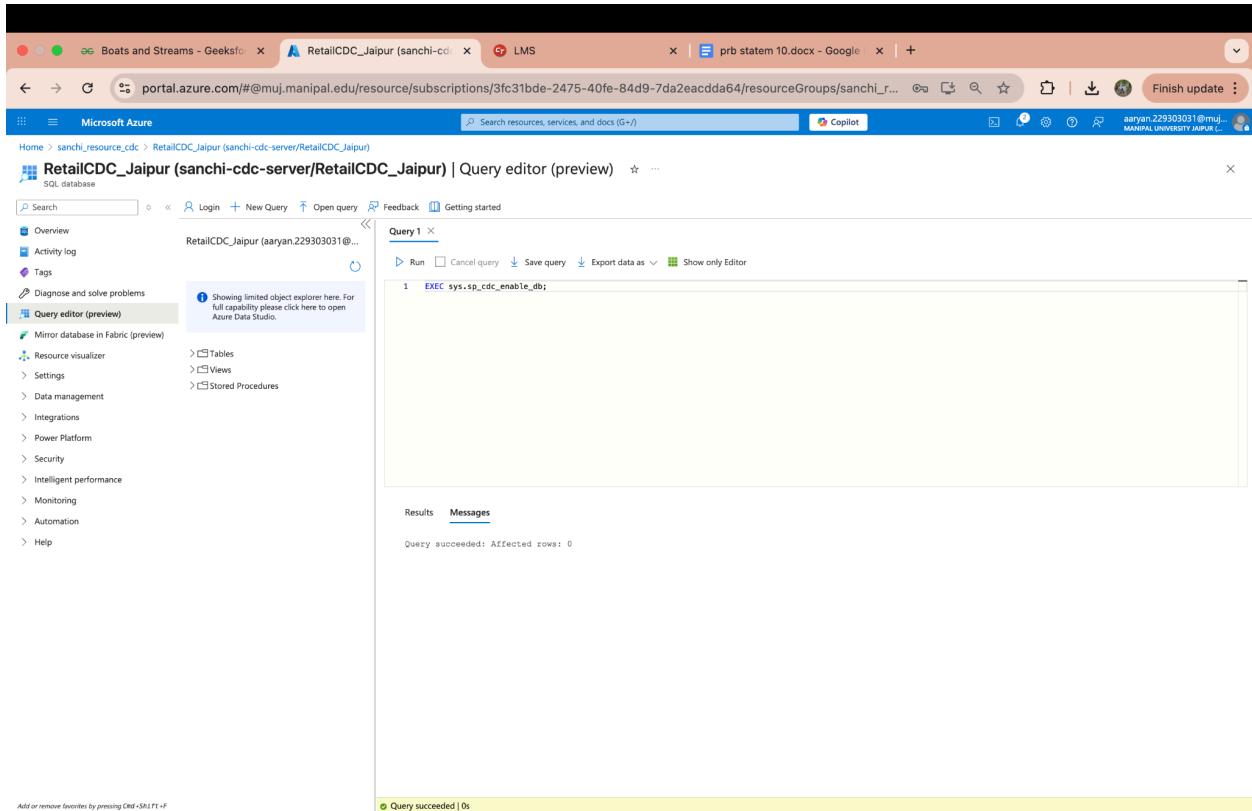
1. SQL Database Creation: RetailCDC_Jaipur

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'). 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 a Microsoft Azure portal window with the URL https://portal.azure.com/#@muj.manipal.edu/resource/subscriptions/3fc31bde-2475-40fe-84d9-7da2eacdda64/resourceGroups/sanchi_resource_cdc/providers/Microsoft.DBforSQL/databases/RetailCDC_Jaipur. The page title is "RetailCDC_Jaipur (sanchi-cdc-server/RetailCDC_Jaipur) | Query editor (preview)". The left sidebar shows navigation options like Overview, Activity log, Tags, Diagnose and solve problems, and Query editor (preview). The main area contains a query editor with the following content:

```
Query 1 < x
Run Cancel query Save query Export data as Show only Editor
1 EXEC sys.sp_cdc_enable_db;
```

The results section shows the message: "Query succeeded: Affected rows: 0".

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

The screenshot shows the Azure Data Studio interface with a query editor titled "Query 1". The code entered is:

```

1 CREATE TABLE dbo.[Order] (
2     OrderID INT PRIMARY KEY,
3     CustomerID INT,
4     ProductID INT,
5     Quantity INT,
6     OrderDate DATETIME,
7     TotalAmount DECIMAL(10, 2),
8     FOREIGN KEY (CustomerID) REFERENCES dbo.Customer(CustomerID),
9     FOREIGN KEY (ProductID) REFERENCES dbo.Product(ProductID)
10 );
11
12 EXEC sys.sp_cdc_enable_table
13     @source_schema = N'dbo',
14     @source_name   = N'Order',
15     @role_name     = NULL,
16     @supports_net_changes = 1;

```

The results tab shows the message: "Query succeeded! Affected rows: 0".

5. Azure Data Factory Creation and Deployment

The screenshot shows the Microsoft Azure portal with a deployment overview page for "Microsoft.DataFactory-20250716235406". The status is "Your deployment is complete". Deployment details include:

- Deployment name: Microsoft.DataFactory-20250716235406
- Subscription: Azure for Students
- Resource group: sanchi_resource_cdc
- Start time: 7/16/2025, 11:56:22 PM
- Correlation ID: bfe59b9a-9de6-4426-a18b-acc20d06b750

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

6. Azure Linked Service

Microsoft Azure | Data Factory > cdcadfclebal

Search factory and documentation

Validate all Publish all

General

Factory settings

Connector upgrade advisor

Connections

Linked services

Integration runtimes

Microsoft Purview

ADF in Microsoft Fabric

Source control

Git configuration

ARM template

Author

Triggers

Global parameters

Data flow libraries

Security

Credentials

Customer managed key

Outbound rules

Managed private endpoints

Workflow orchestration manager

Linked services

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

+ New

Filter by name Annotations : Any

Showing 1 - of 1 items

Name	Type
AzureSqlDb_Linked	Azure SQL

Edit linked service

Azure SQL Database [Learn more](#)

Name * AzureSqlDb_Linked

Description

Connect via integration runtime * [AutoResolveIntegrationRuntime](#)

Version 2.0 (Recommended) 1.0

Account selection method From Azure subscription Enter manually

Fully qualified domain name * sanchi-cdc-server.database.windows.net

Database name * RetailCDC_Jaipur

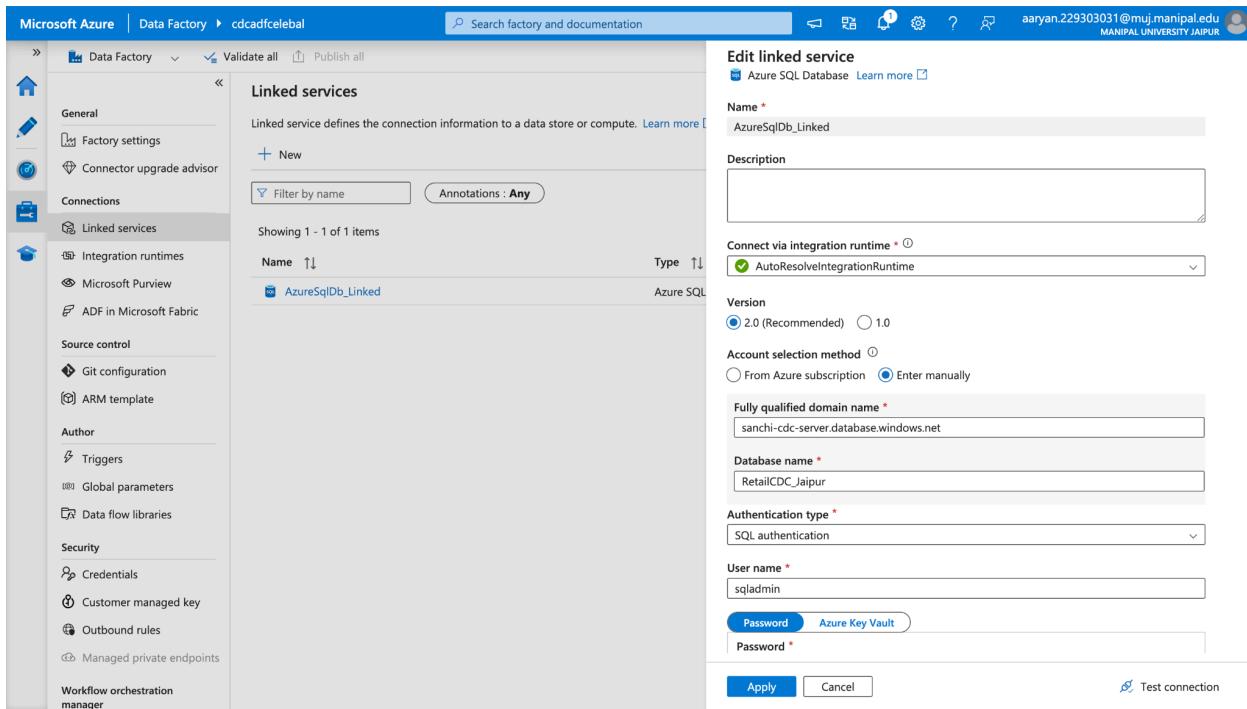
Authentication type * SQL authentication

User name * sqldadmin

Password [Azure Key Vault](#)

Test connection

Apply Cancel



7. Data Bricks WorkSpace Creation

Microsoft Azure

Search resources, services, and docs (G+)

Copilot

Home > sanchi_resource_cdc_cdc-databricks | Overview

Deployment

Search Delete Cancel Redeploy Download Refresh

Overview

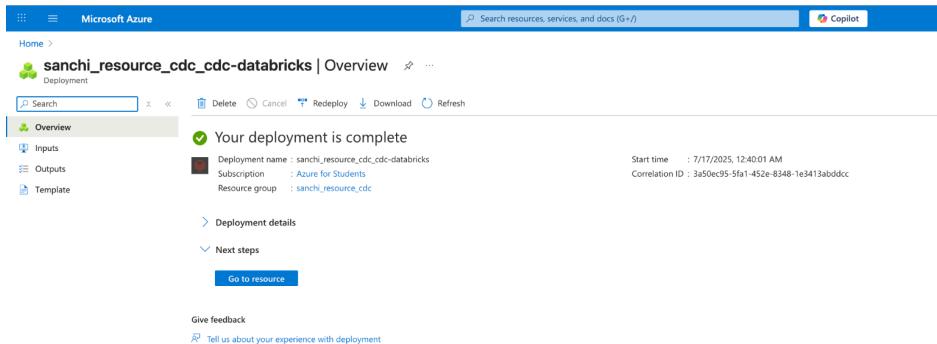
Your deployment is complete

Deployment name : sanchi_resource_cdc_cdc-databricks
Subscription : Azure for Students
Resource group : sanchi_resource_cdc

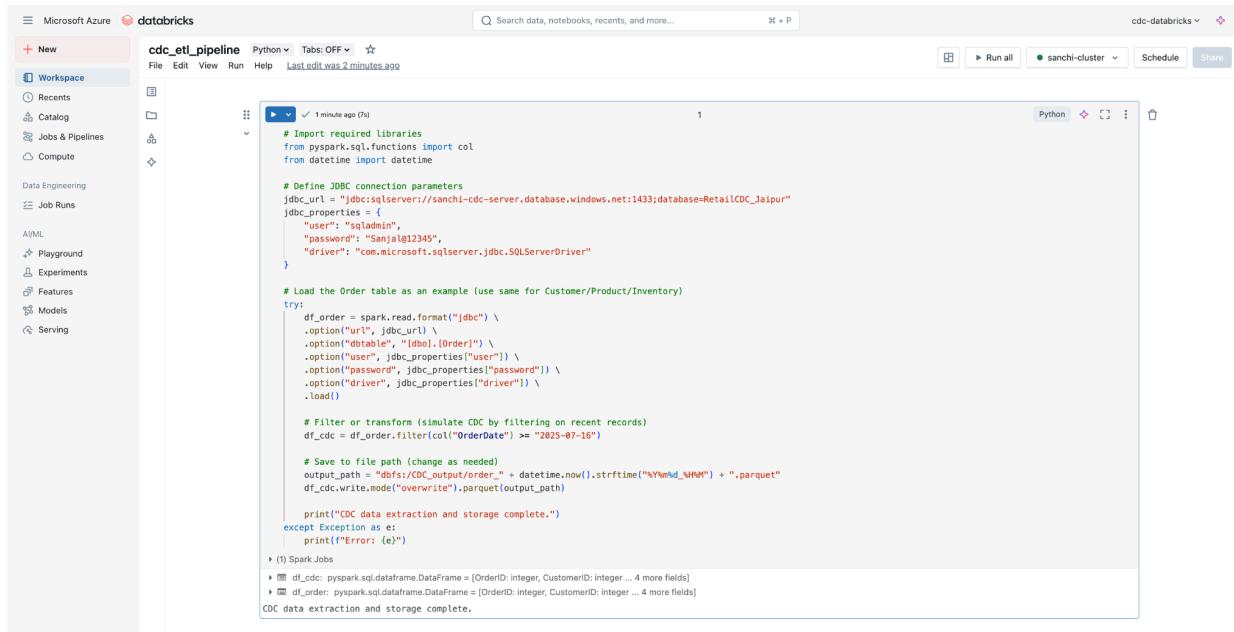
Start time : 7/17/2025, 12:40:01 AM
Correlation ID : 3a50ec95-5fa1-452e-8348-1e3413abddcc

Deployment details
Next steps
Go to resource

Give feedback
Tell us about your experience with deployment



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



```
# 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}")

# 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 interface. On the left, the 'Factory Resources' sidebar lists Pipelines, Datasets, Data flows, and Power Query. In the center, the 'Activities' section shows a pipeline named 'pipeline1'. A modal window titled 'New linked service' is open, specifically for 'Azure Databricks'. The 'Name' field is set to 'SanchiDatabricks_Linked'. Under 'Connect via integration runtime', 'AutoResolveIntegrationRuntime' is selected. The 'Account selection method' is set to 'From Azure subscription'. The 'Azure subscription' dropdown shows 'Azure for Students (3fc31bde-2475-40fe-84d9-7da2eacdda64)'. The 'Databricks workspace' dropdown shows 'cdc-databricks'. Under 'Select cluster', 'Existing interactive cluster' is selected. The 'Databrick Workspace URL' is set to 'https://adb-3243773246735880.azuredatabricks.net'. The 'Authentication type' is set to 'Access Token'. At the bottom right of the modal, there are 'Create' and 'Cancel' buttons, along with a 'Connection successful' message and a 'Test connection' link.

10. Notebook in Azure Published

The screenshot shows the Microsoft Azure Data Factory interface. The 'Factory Resources' sidebar and 'Activities' section are similar to the previous screenshot. A modal window titled 'Publishing completed' is displayed, stating 'Successfully published'. The main pipeline configuration area shows the 'General' tab for a notebook activity named 'Sanchi_Notebook_DB'. The 'Name' field is set to 'Sanchi_Notebook_DB'. The 'Activity state' is set to 'Activated'. Other settings include 'Timeout' (0:12:00:00), 'Retry' (0), and 'Retry interval (sec)' (30). The pipeline is shown in the background with the notebook activity highlighted.

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 'New', 'Workspace', 'Recents', 'Catalog', 'Jobs & Pipelines', 'Compute', 'Data Engineering', 'Job Runs', 'AI/ML', 'Playground', 'Experiments', 'Features', 'Models', and 'Serving'. The 'Job Runs' tab is selected. At the top, 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, the 'Job runs' section displays a timeline from Jul 17, 2025, 03:25 PM to Jul 17, 2025, 03:30 PM. It shows two runs: one succeeded (32s) and one succeeded (36s). The table headers are 'Start time', 'Job', 'Run as', 'Launched', 'Duration', 'Status', 'Error code', and 'Run parameters'. The status column shows green checkmarks for both runs.

12. Notebook in Azure Linked and Published

The screenshot shows the Microsoft Azure Data Factory pipeline editor. On the left, the 'Factory Resources' sidebar lists 'Pipelines' (1), 'Datasets' (5), 'Data flows', and 'Power Query'. The main area shows a pipeline named 'pipeline1' with an 'Activities' list containing 'Move and transform', 'Synapse', 'Azure Data Explorer', 'Azure Function', 'Batch Service', 'Databricks', 'Notebook' (selected), 'Jar', and 'Python'. A preview message 'Publishing completed' and 'Successfully published' is visible. On the right, the 'General' tab for the 'Notebook' activity is shown, with fields for 'Name' (set to 'Sanchi_Notebook_DB'), 'Description', 'Activity state' (set to 'Activated'), and 'Timeout' (set to '0:12:00:00').

13. Pipeline Ran Successfully

The screenshot shows the Microsoft Azure Data Factory interface. On the left, the navigation menu is open, with 'Pipeline runs' selected. The main area displays '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 'Activity runs' section shows one item: 'Sanchi_Notebook_DB' with status 'Succeeded'. The activity details table is as follows:

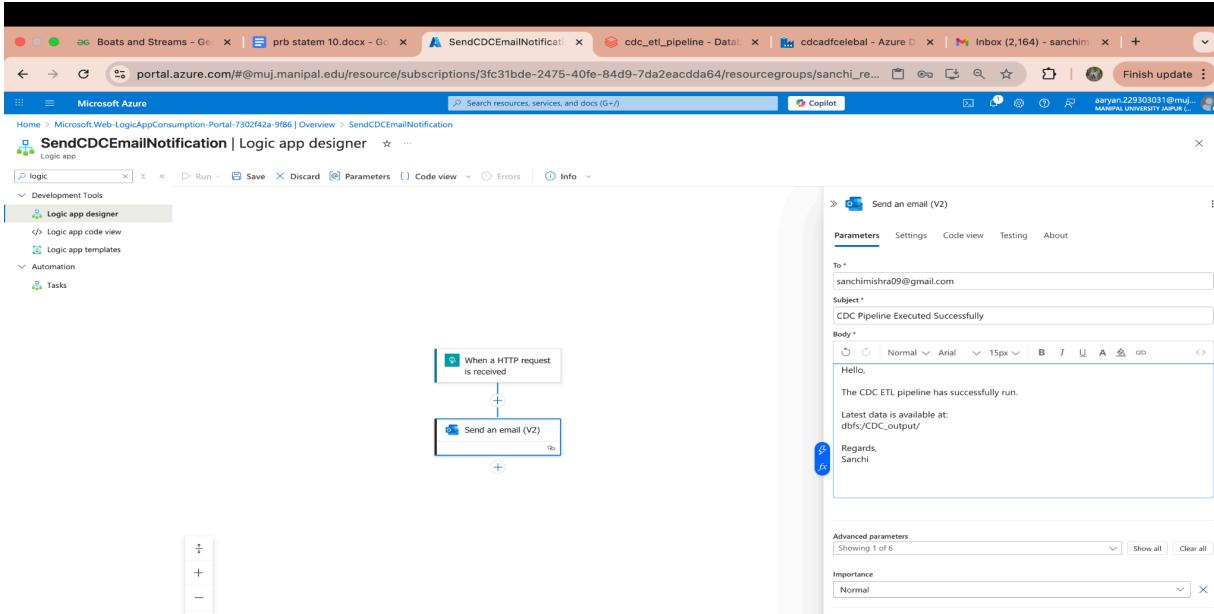
Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime	User prop...	Acti...
Sanchi_Notebook_DB	Succeeded	Notebook	7/17/2025, 3:23:47 PM	2m 35s	AutoResolveIntegrationRuntime (Southeast Asia)	461	

14. Email and Notebook Connected and Ran

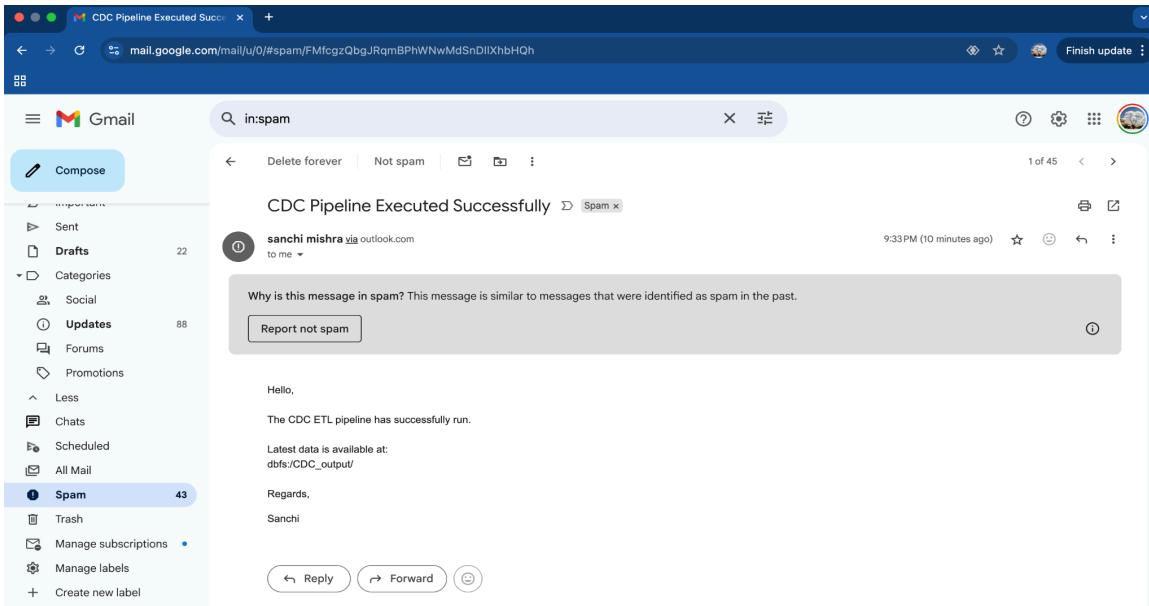
This screenshot shows the same Microsoft Azure Data Factory interface as the previous one, but with two activities listed in the 'Activity runs' section. The 'Notebook' activity has already been completed successfully. The second activity, 'SendEmailNotification', is also shown with a green checkmark indicating success. The activity details table is as follows:

Activity name	Activity st...	Activit...	Run start	Duration	Integration runtime	User prop...	Acti...
Sanchi_Notebook_DB	Succeeded	Notebook	7/17/2025, 6:26:32 PM	34s	AutoResolveIntegrationRuntime (Southeast Asia)	0b1	
SendEmailNotification	Succeeded	Web					

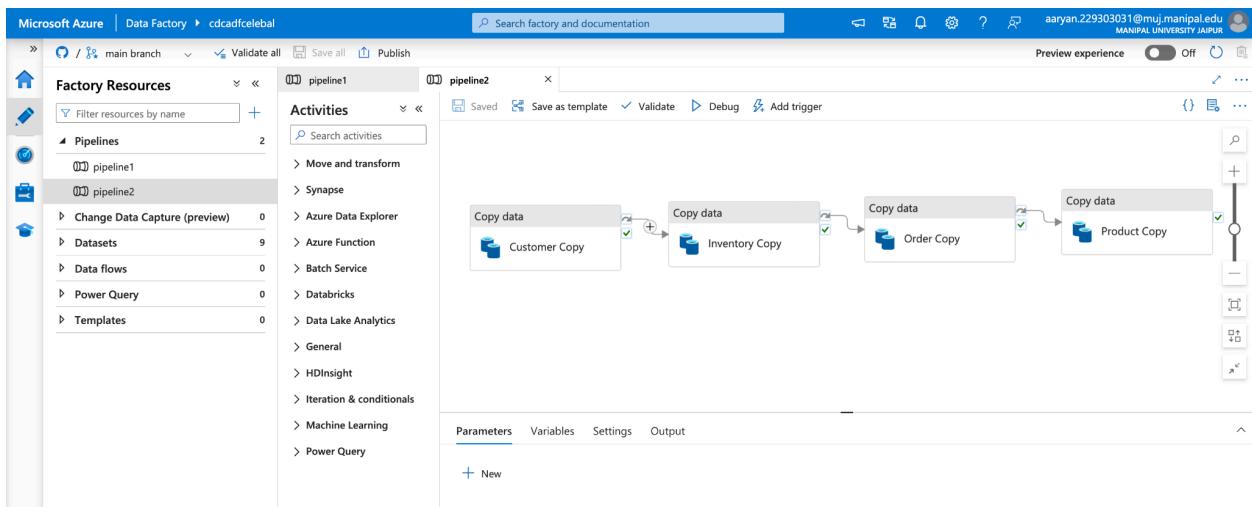
15. Email Setup



16. Email Received



17. A few more screenshots from pipeline where dataset and outputs were loaded



◀ 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](#)

+ New

Showing 1 - 5 of 5 items

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'. The 'Type' column shows 'Schedule'. On the right, the 'Edit trigger' dialog is open for 'HourlyTrigger'. The 'Name' field is set to 'HourlyTrigger'. The 'Type' dropdown is set to 'ScheduleTrigger'. The 'Start date' is set to '7/16/2025, 5:15:00 PM'. The 'Time zone' dropdown is set to 'Chennai, Kolkata, Mumbai, New Delhi (UTC+5:30)'. The 'Recurrence' section shows 'Every 60 Minute(s)' and has a checked checkbox for 'Specify an end date', with the value '7/19/2025, 7:00:00 PM'. The 'Annotations' section has a '+ New' button. The 'Status' section shows 'Started' (radio button unselected) and 'Stopped' (radio button selected).

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!