18. End-to-End Streaming Data Pipeline with Azure Data Factory and Databricks. Data Engineering Project 2.

-Sarthak Niranjan Kulkarni (Maverick)

sarthakkul2311@gmail.com (+91) 93256 02791

Title: *Set up an end-to-end pipeline with Azure Data Factory for ingesting streaming data and Azure Databricks for real-time processing and analysis of the streaming data.*

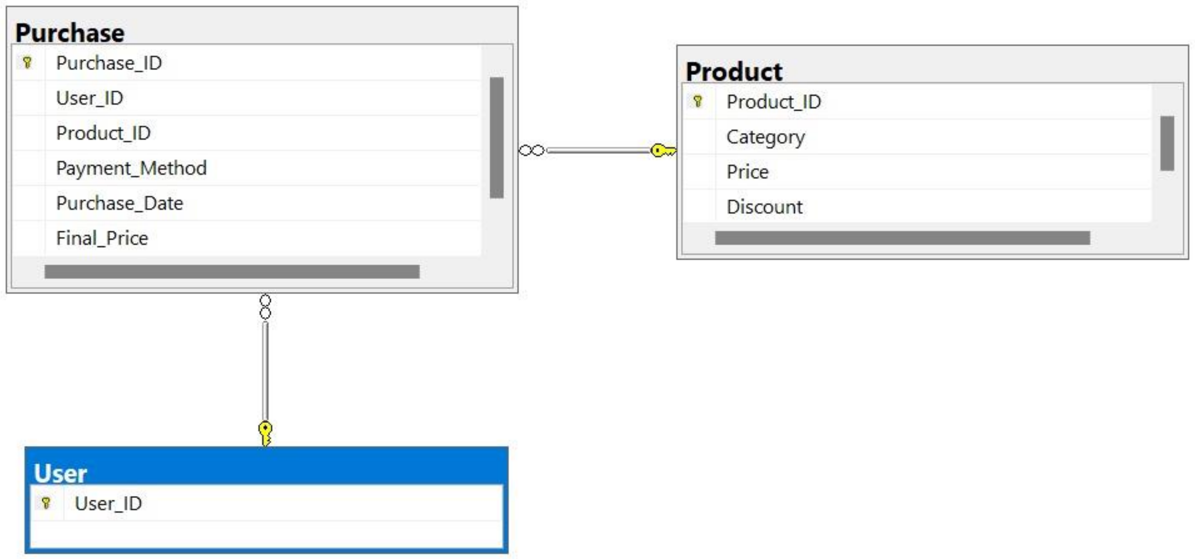
1. Project Overview:

I built an end-to-end data pipeline using Azure tools to automate the process of handling streaming data. The raw data was first stored in Azure Blob Storage and then transferred to Azure Data Lake Storage using the Copy Data feature in Azure Data Factory. After the data was moved, a Databricks notebook was connected to the pipeline to perform data transformations on the stored data. The transformed data was then saved back to Data Lake Storage for further use. This entire workflow: data ingestion, transformation, and storage was automated within the Azure Data Factory pipeline, ensuring a seamless and efficient process.

2. Data Overview:

The e-commerce dataset contains information about 3,660 online purchases, with each purchase uniquely identified by a User_ID and Product_ID. It provides details about the products, such as their category (like Sports, Clothing, Toys, and Beauty), their original price, and the discount percentage applied. The final price customers paid after the discount is also recorded. Additionally, the dataset includes information about the payment methods used, such as Net Banking, Credit Card, and UPI, along with the purchase date for each transaction. Importantly, there are no missing values in the dataset, making it reliable for analysis. Overall, this dataset gives a clear picture of customer buying habits, product popularity, and pricing patterns on the e-commerce platform.

3. Architecture diagram –ER diagram:



---

4. Source Data File:

The source data for this project was obtained from Kaggle, where the e-commerce dataset is available. The dataset is provided in CSV (Comma Separated Values) format, which is commonly used for structured data storage and analysis.

Download Ecommerce Dataset

---

5. Execution Overview:

This project utilizes Apache Spark in combination with Azure Data Lake Storage Gen2 to process real-time e-commerce transaction data. The process begins by configuring Spark to connect to the Azure Data Lake Storage, where the raw e-commerce data is stored. The dataset includes key fields such as Product ID, User ID, Category, Price, Discount, Final Price, Payment Method, and Purchase Date.

The data is ingested, with several transformations applied:

- **Total Sales Calculation:** The total sales for each transaction is calculated by applying the discount to the original price.

- **Data Filtering:** Transactions with missing product categories are filtered out to ensure the dataset is clean and complete.
- **Date Formatting:** The Purchase Date is standardized to the yyyy-MM-dd format for consistency in future analysis.

The transformed data is then written back to Azure Data Lake in CSV format, ensuring that new data is appended as it arrives. A checkpointing mechanism is implemented to track the progress of the streaming job, enabling the system to recover from any failures. The job continues to run, processing and updating the data in real-time, making it available for further analysis or reporting.

---

6. Azure Resources Used for this Project:

- Azure Data Lake Storage Gen2
- Azure Databricks
- Azure Storage Account
- Azure Spark Service
- Azure Blob Storage

---

7. Project Requirements:

- **Azure Subscription:** An active Azure subscription to create and manage Azure resources such as Data Lake Storage, Databricks, and other necessary services.
- **Azure Data Lake Storage Gen2:** A provisioned Azure Data Lake Storage Gen2 account for storing raw and processed data.
- **Azure Databricks Workspace:** A Databricks workspace to set up and execute Spark-based transformations on the streaming data.
- **Apache Spark:** A Spark environment within Databricks for processing large-scale data in real-time.
- **Data Source:** A CSV-formatted e-commerce dataset containing transaction details like User_ID, Product_ID, Category, Price, Discount, Payment_Method, and Purchase_Date.
- **Python & PySpark Libraries:** Python environment with PySpark libraries installed for accessing and processing data from Azure Data Lake Storage and Databricks.
- **Azure Storage Accounts:** Storage accounts for raw data input and processed output, with appropriate keys and permissions to access the data.
- **Checkpointing:** Checkpoint locations for streaming jobs to ensure recovery and fault tolerance during data processing.

---

8. Tasks Performed:

- **Data Ingestion from Azure Data Lake**
  1. Configured the Spark environment to access Azure Data Lake Storage Gen2 using the storage account key.
  2. Set up a streaming read process to ingest the e-commerce transaction data stored in CSV format.
- **Data Transformation**
  1. Defined the schema for the e-commerce dataset to properly interpret the columns during ingestion.
  2. Applied transformations to calculate total sales by considering the final price and discount.

3. Filtered data to exclude rows with missing or null values in specific columns (e.g., Product Category).

4. Reformatted the Purchase_Date field from the original format into yyyy-MM-dd format for consistency.

- **Real-Time Streaming Processing**

  1. Set up a real-time streaming process using Spark Structured Streaming to continuously process and transform incoming data from the Data Lake.

  2. Added custom transformations like calculating total sales and formatting the date to create a clean, processed dataset.

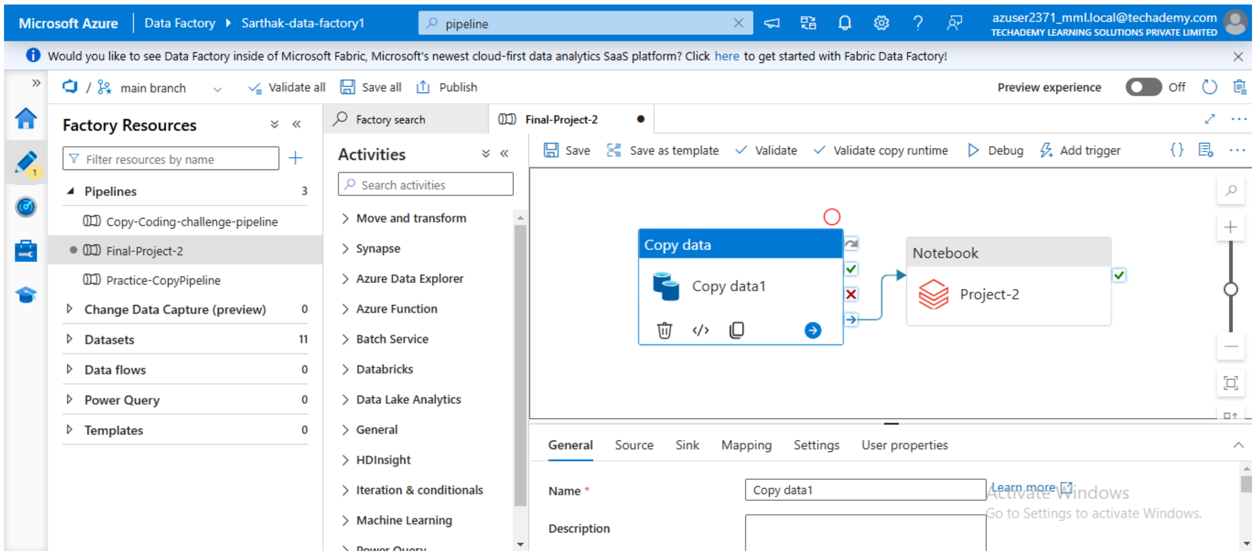
- **Data Storage in Azure Data Lake**

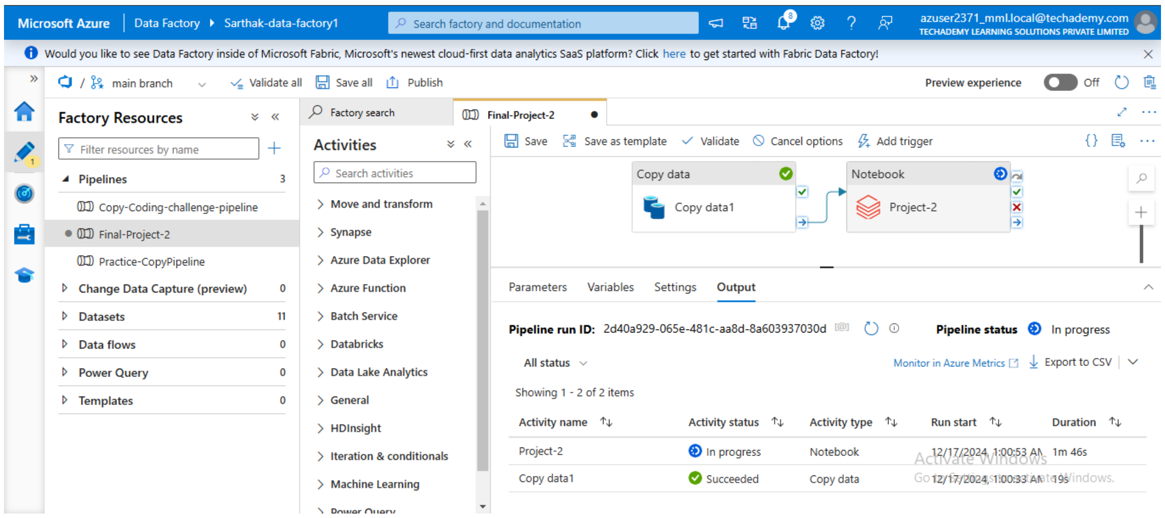
  1. Wrote the processed data to a separate container in Azure Data Lake Storage Gen2 in CSV format, ensuring that the data is accessible for future analysis or reporting.

  2. Implemented checkpointing to ensure that the stream can resume processing in case of failures or restarts.

---

9. Analysis Result:




Microsoft Azure | Data Factory ▸ Sarthak-data-factory1

Would you like to see Data Factory inside of Microsoft Fabric, Microsoft's newest cloud-first data analytics SaaS platform? Click here to get started with Fabric Data Factory!

Factory Resources

Pipelines	3
Copy-Coding-challenge-pipeline	
Final-Project-2	
Practice-CopyPipeline	
Change Data Capture (preview)	0
Datasets	11
Data flows	0
Power Query	0
Templates	0

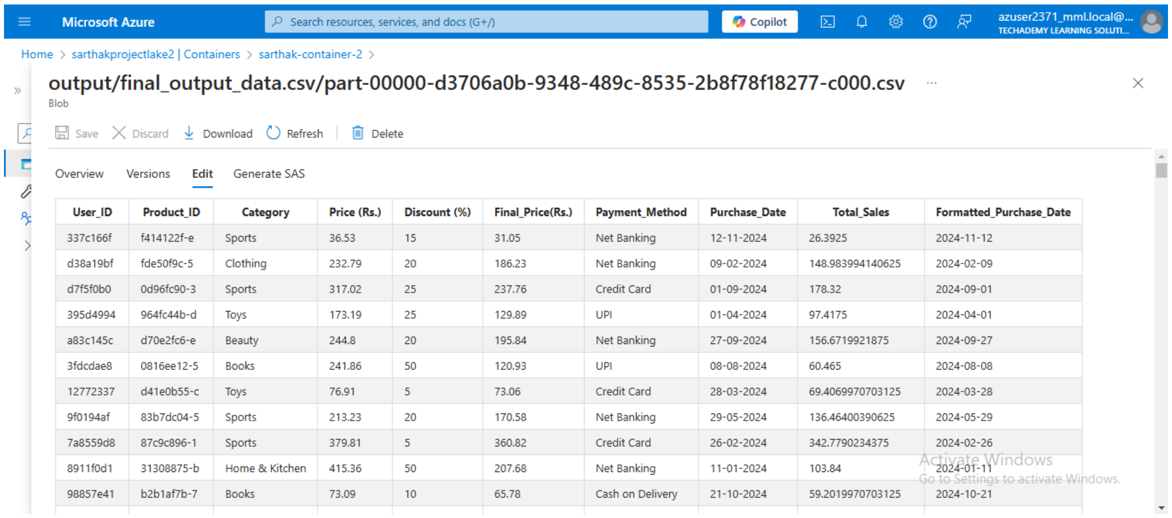
Pipeline run ID: 2d40a929-065e-481c-aa8d-8a603937030d

Pipeline status: In progress

Showing 1 - 2 of 2 items

Activity name	Activity status	Activity type	Run start	Duration
Project-2	In progress	Notebook	12/17/2024, 1:00:53 AM	1m 46s
Copy data1	Succeeded	Copy data	12/17/2024, 1:00:33 AM	19s

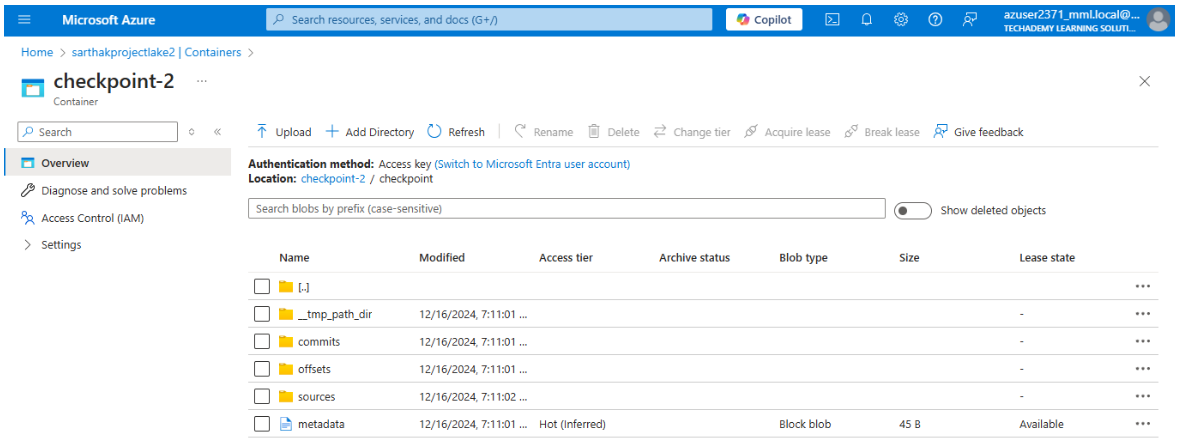
Note: - In the picture above, since the code is set up for streaming live data, the status of the notebook will remain as "In Progress" for up to 10 minutes, as the timeout for the streaming job is configured to 10 minutes.


Home > sarthakprojectlake2 | Containers > sarthak-container-2 >

output/final_output_data.csv/part-00000-d3706a0b-9348-489c-8535-2b8f78f18277-c000.csv

Blob

User_ID	Product_ID	Category	Price (Rs.)	Discount (%)	Final_Price(Rs.)	Payment_Method	Purchase_Date	Total_Sales	Formatted_Purchase_Date
337c166f	f414122f-e	Sports	36.53	15	31.05	Net Banking	12-11-2024	26.3925	2024-11-12
d38a19bf	fde50f9c-5	Clothing	232.79	20	186.23	Net Banking	09-02-2024	148.983994140625	2024-02-09
d7f5f0b0	0d96fc90-3	Sports	317.02	25	237.76	Credit Card	01-09-2024	178.32	2024-09-01
395d4994	964fc44b-d	Toys	173.19	25	129.89	UPI	01-04-2024	97.4175	2024-04-01
a83c145c	d70e2fc6-e	Beauty	244.8	20	195.84	Net Banking	27-09-2024	156.6719921875	2024-09-27
3fdcdae8	0816ee12-5	Books	241.86	50	120.93	UPI	08-08-2024	60.465	2024-08-08
12772337	d41e0b55-c	Toys	76.91	5	73.06	Credit Card	28-03-2024	69.4069970703125	2024-03-28
9f0194af	83b7dc04-5	Sports	213.23	20	170.58	Net Banking	29-05-2024	136.46400390625	2024-05-29
7a8559d8	87c9c896-1	Sports	379.81	5	360.82	Credit Card	26-02-2024	342.7790234375	2024-02-26
8911f0d1	31308875-b	Home & Kitchen	415.36	50	207.68	Net Banking	11-01-2024	103.84	2024-01-11
98857e41	b2b1af7b-7	Books	73.09	10	65.78	Cash on Delivery	21-10-2024	59.2019970703125	2024-10-21


Home > sarthakprojectlake2 | Containers >

checkpoint-2

Container

Authentication method: Access key (Switch to Microsoft Entra user account)
Location: checkpoint-2 / checkpoint

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
[..]						
_tmp_path_dir	12/16/2024, 7:11:01 ...					-
commits	12/16/2024, 7:11:01 ...					-
offsets	12/16/2024, 7:11:01 ...					-
sources	12/16/2024, 7:11:02 ...					-
metadata	12/16/2024, 7:11:01 ...	Hot (Inferred)		Block blob	45 B	Available

Note: - The image above shows the checkpoint directory created for the streaming job. In this directory, Spark stores the metadata and progress information of the streaming job, ensuring that the job can resume from the last processed state in case of a failure or restart.

---

10. Project Code and Output:

The output CSV file showcases the successful implementation of a real-time streaming pipeline, where data ingestion, transformation, and storage were seamlessly achieved using Azure Data Factory (ADF) and Azure Databricks (ADB). Initially, the dataset was uploaded to Azure Blob Storage, simulating streaming data. Using ADF, a pipeline was created with activities to copy and process the data in chunks, triggering a Databricks Notebook for advanced transformations. The Databricks Structured Streaming logic calculated the Final_Price (Rs.) by applying discounts to the original Price (Rs.), standardized purchase dates into Formatted_Purchase_Date, and aggregated the Total_Sales for each record. The pipeline ensured smooth orchestration between data ingestion and processing, while the output file reflects key insights such as product categories, payment methods, and sales metrics. This end-to-end process validates the pipeline's ability to process and transform large-scale datasets efficiently, delivering clean and structured data ready for analysis and reporting.

Final-Project-2.zip

Link to whole project: https://drive.google.com/drive/folders/1u5mBoxdRFqZg_OOj7iqBm-xkiepcdH5r?usp=drive_link

---