RETAIL X

*This document presents a comprehensive overview of the **RetailX project**, an end-to-end real-time retail analytics pipeline built using Microsoft Azure and Python. Developed as part of a data engineering learning initiative, this project demonstrates the integration of cloud-native tools to simulate, ingest, process, store, and visualize streaming retail transaction data.*

The pipeline leverages a combination of Azure services and open-source tools to simulate real-time transaction events, stream them through the Azure ecosystem, and generate actionable business insights through Power BI dashboards. The project emphasizes the practical application of cloud architecture and real-time data engineering principles in the retail domain.

Krishna Cha Muttevi

Project Description:

RetailX is a cloud-native, real-time retail data pipeline built using Microsoft Azure, Python, and Power BI. It simulates live transaction data, processes it on the fly using Azure Stream Analytics, stores it in Azure Data Lake Gen2, and visualizes the insights using Power BI dashboards. This project demonstrates scalable event-driven architecture and live business intelligence.

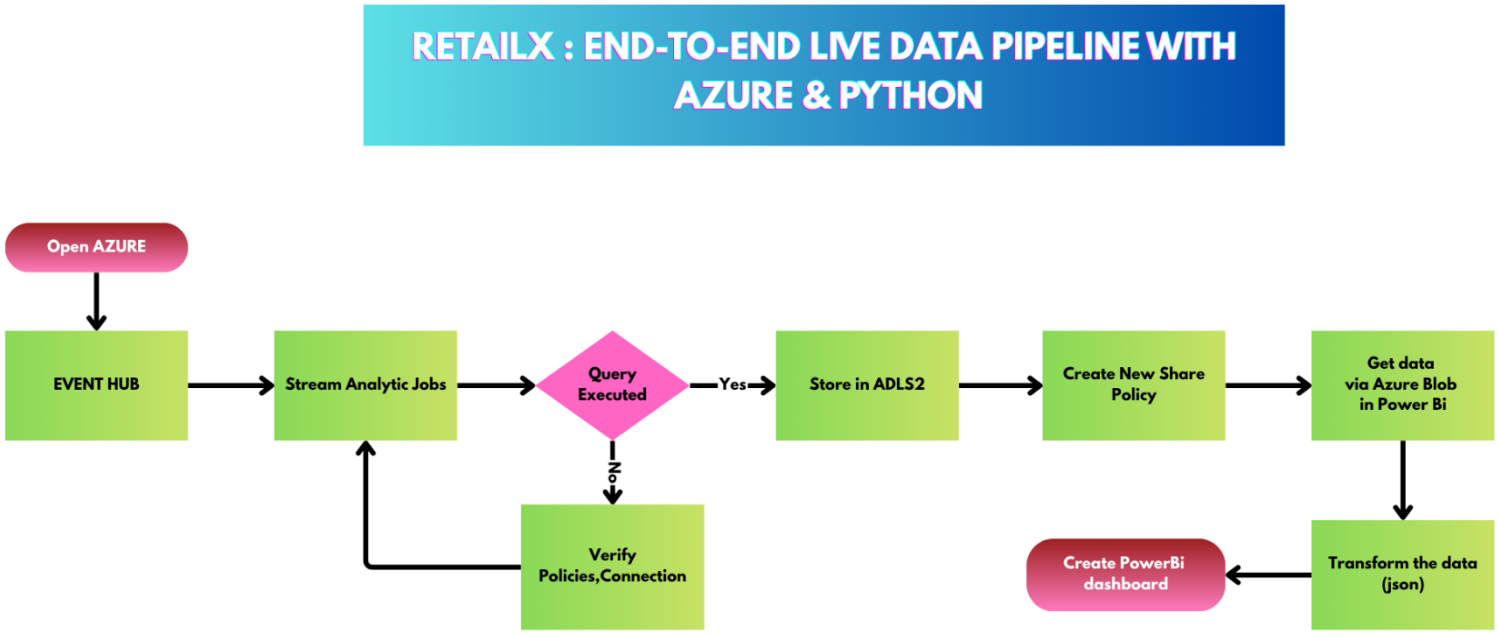
Tech Stack:

Tool	Purpose
Azure Event Hub	Real-time event ingestion
Azure Stream Analytics	SQL-based stream processing
Azure Data Lake Gen2	Storage for raw and transformed data
Python (VS Code)	Real-time data simulator
Power BI	Live data visualization
JSON	Stream data format

Key Features:

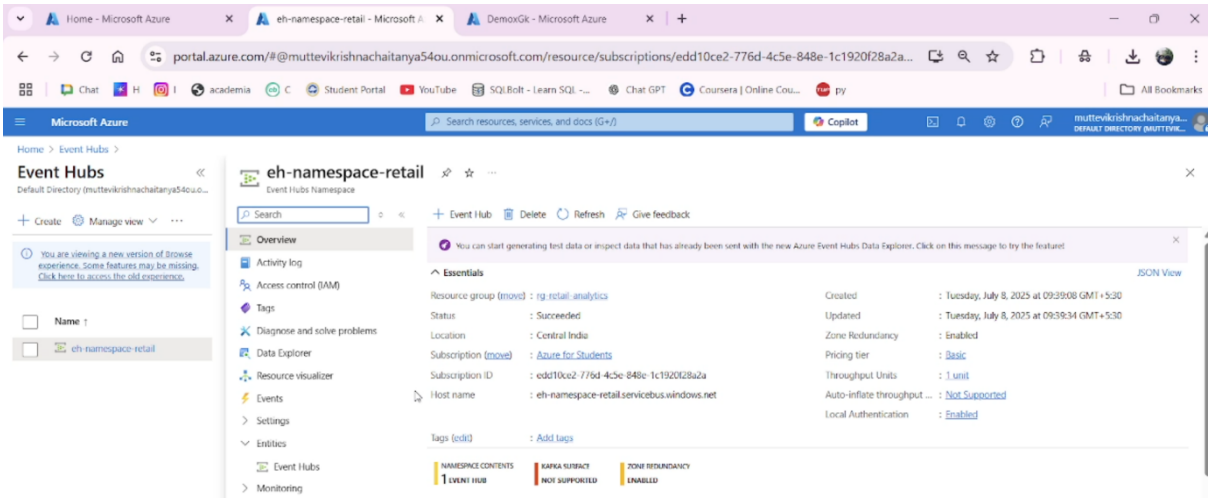
- Real-time data ingestion using **Azure Event Hub**
- SQL-based stream processing with **Azure Stream Analytics**
- Scalable cloud storage via **ADLS Gen2**
- Integration with **Power BI** for live dashboards
- Python-based data simulator for real-time publishing

Architecture Diagram:

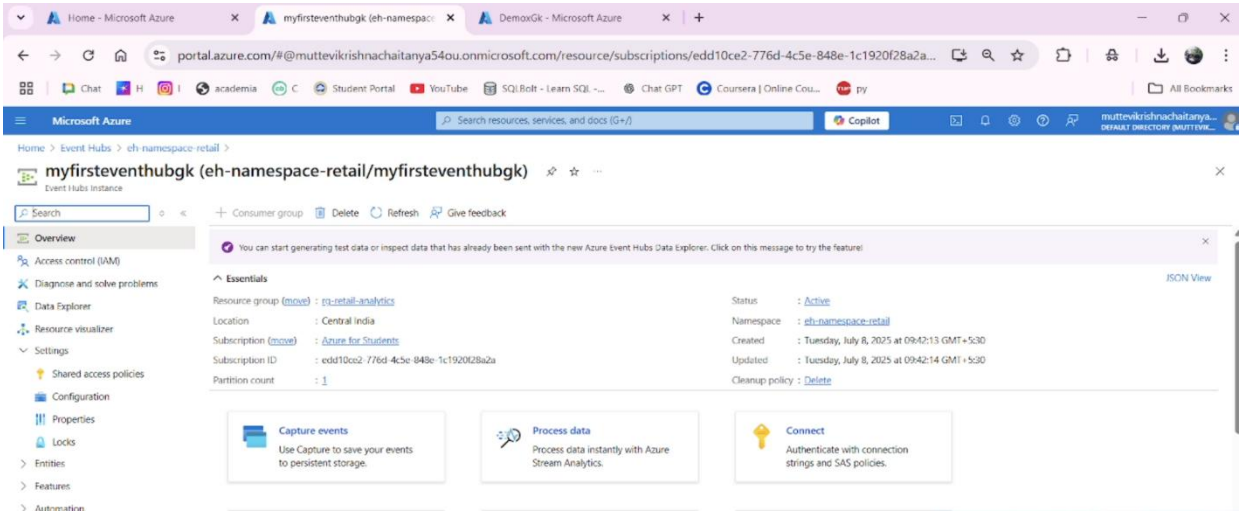


EVENT-HUB:

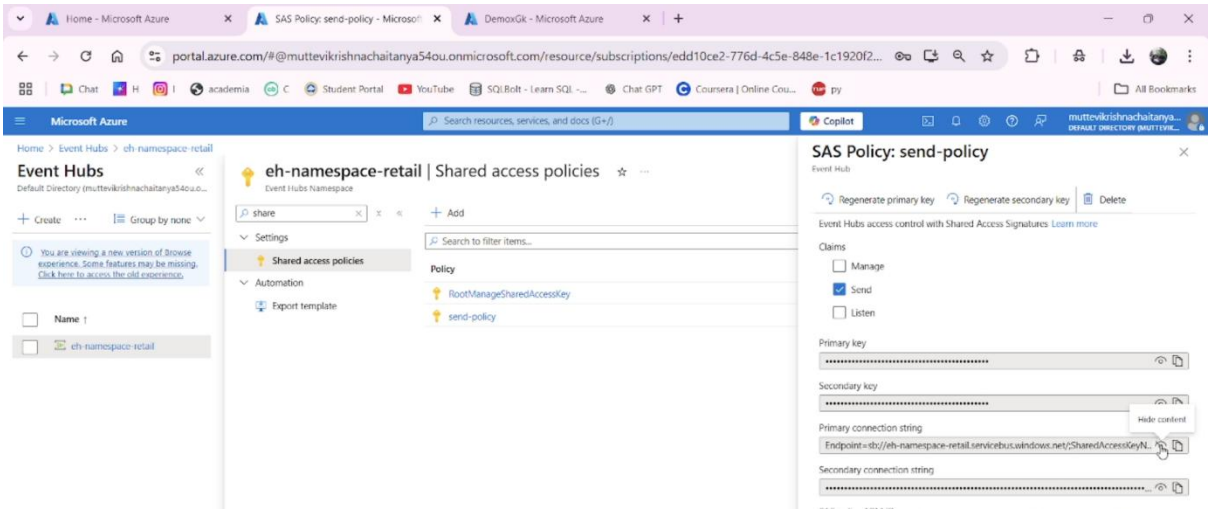
1.Namespace:



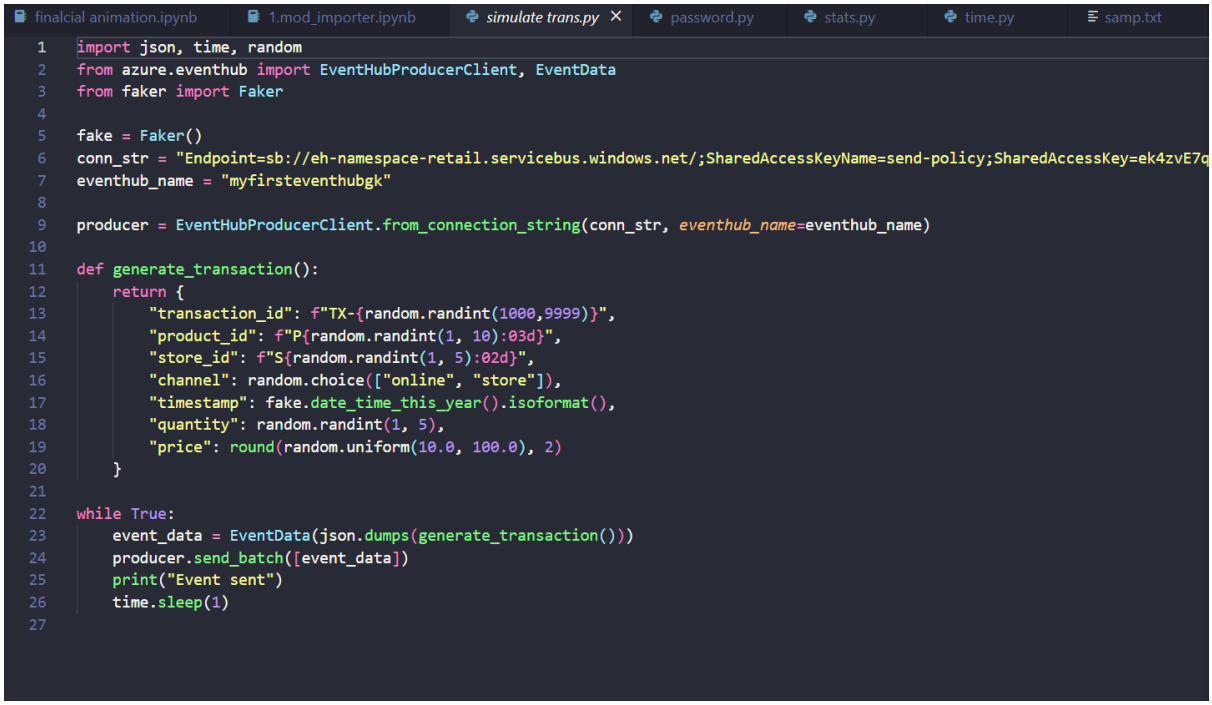
2. Event Hub:



3. Creation of New Shared Access Policy:



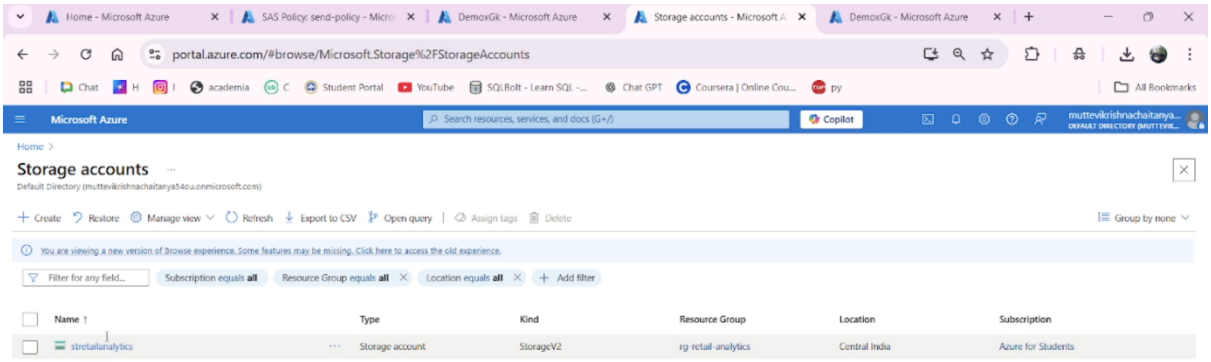
4.Python file:



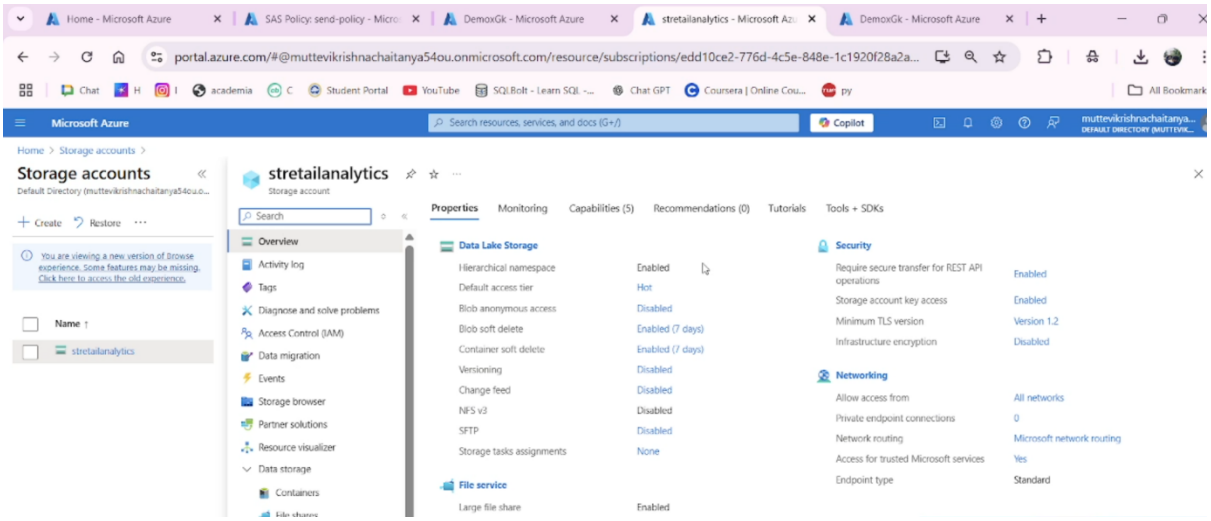
```python
import json, time, random
from azure.eventhub import EventHubProducerClient, EventData
from faker import Faker

fake = Faker()
conn_str = "Endpoint=sb://eh-namespace-retail.servicebus.windows.net/;SharedAccessKeyName=send-policy;SharedAccessKey=ek4zvE7q
eventhub_name = "myfirsteventhubgk"

producer = EventHubProducerClient.from_connection_string(conn_str, eventhub_name=eventhub_name)

def generate_transaction():
    return {
        "transaction_id": f"TX-{random.randint(1000,9999)}",
        "product_id": f"P{random.randint(1, 10):03d}",
        "store_id": f"S{random.randint(1, 5):02d}",
        "channel": random.choice(["online", "store"]),
        "timestamp": fake.date_time_this_year().isoformat(),
        "quantity": random.randint(1, 5),
        "price": round(random.uniform(10.0, 100.0), 2)
    }

while True:
    event_data = EventData(json.dumps(generate_transaction()))
    producer.send_batch([event_data])
    print("Event sent")
    time.sleep(1)
```

ADLS 2:

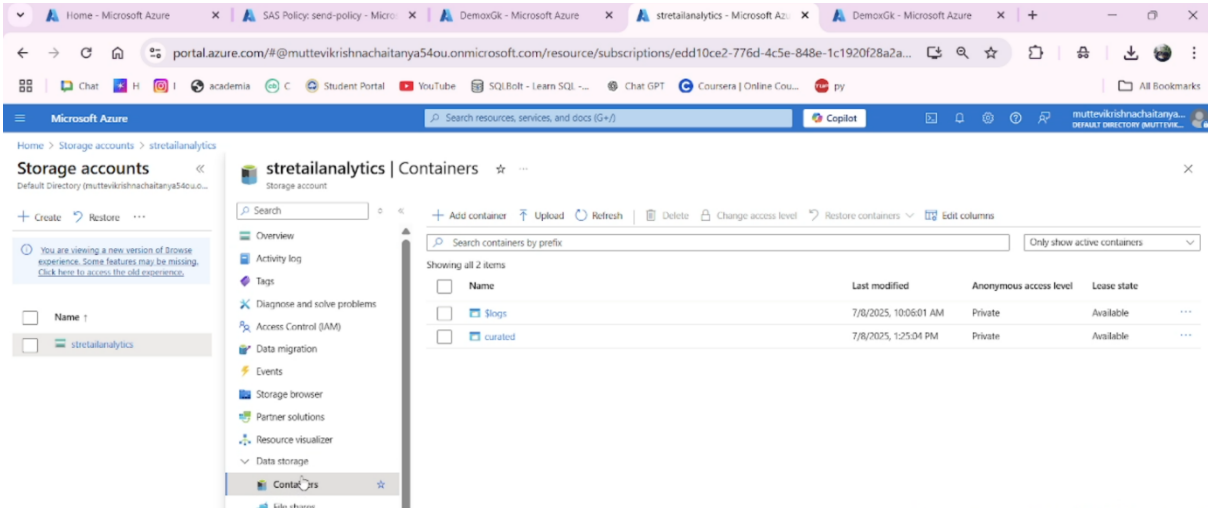
1. New Storage Account:



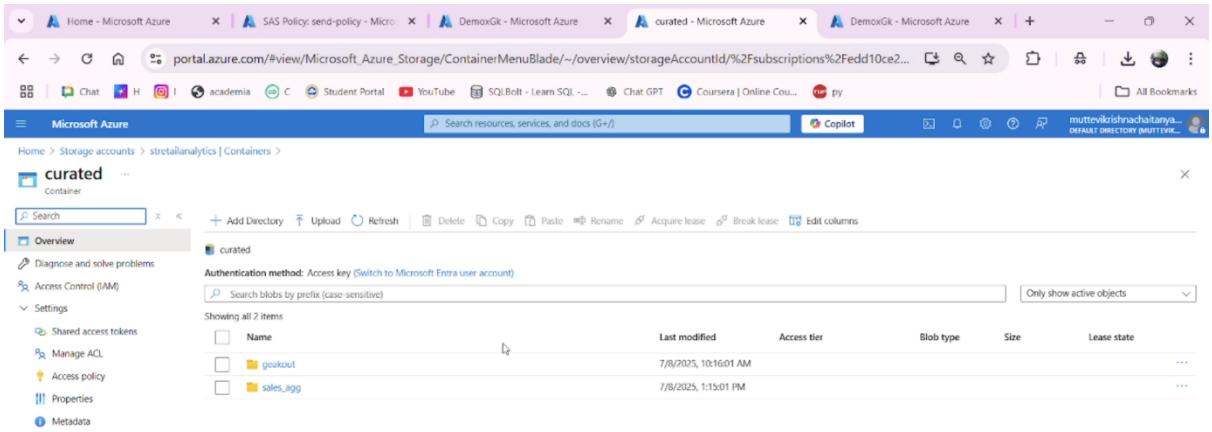
2. Enable Heirarchial Namespace:



3. Create a Container:

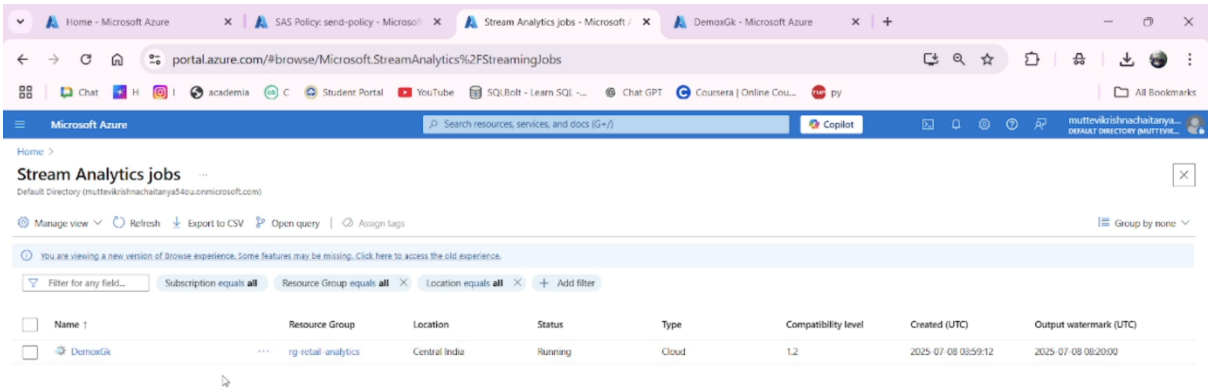


4. Inside Container Create a new Directory:

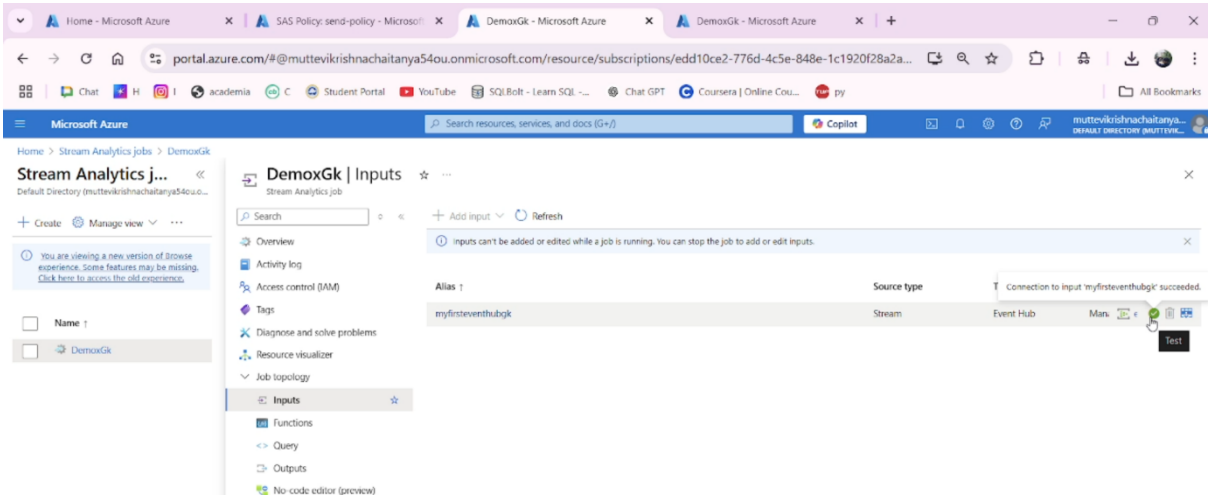


Stream Analytics:

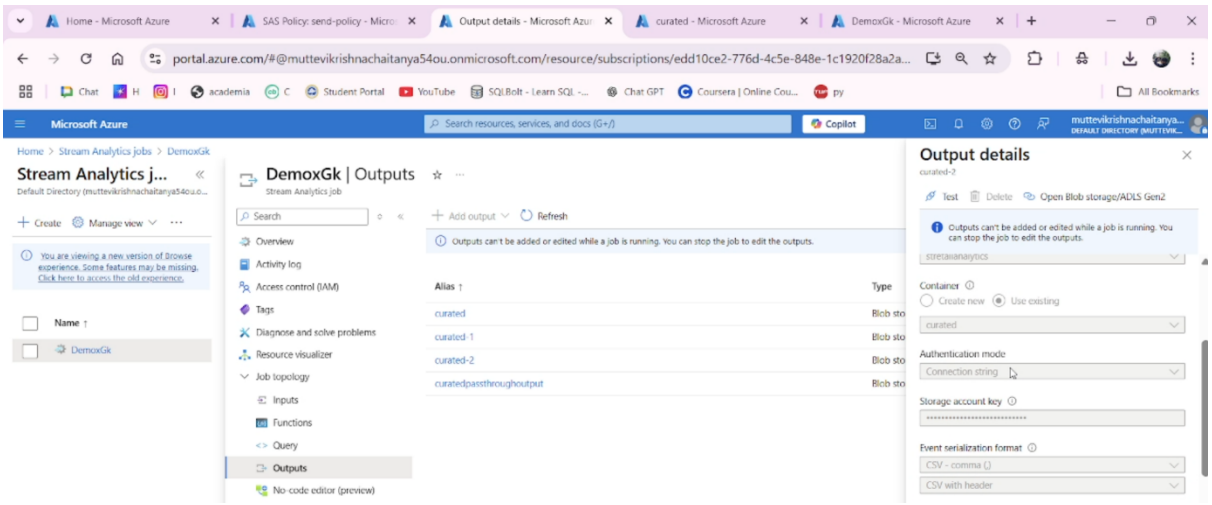
1. Job Creation:



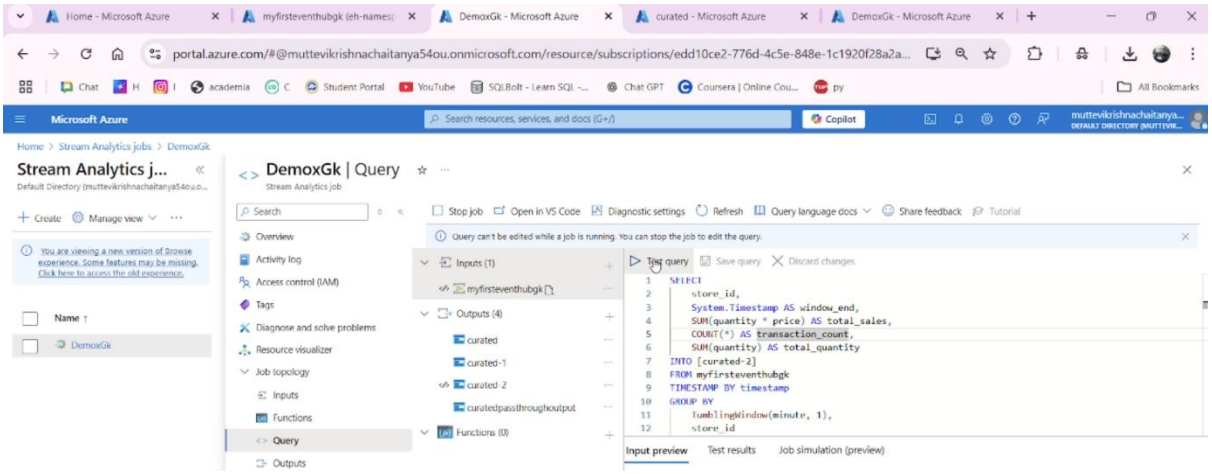
2. Input Definition:



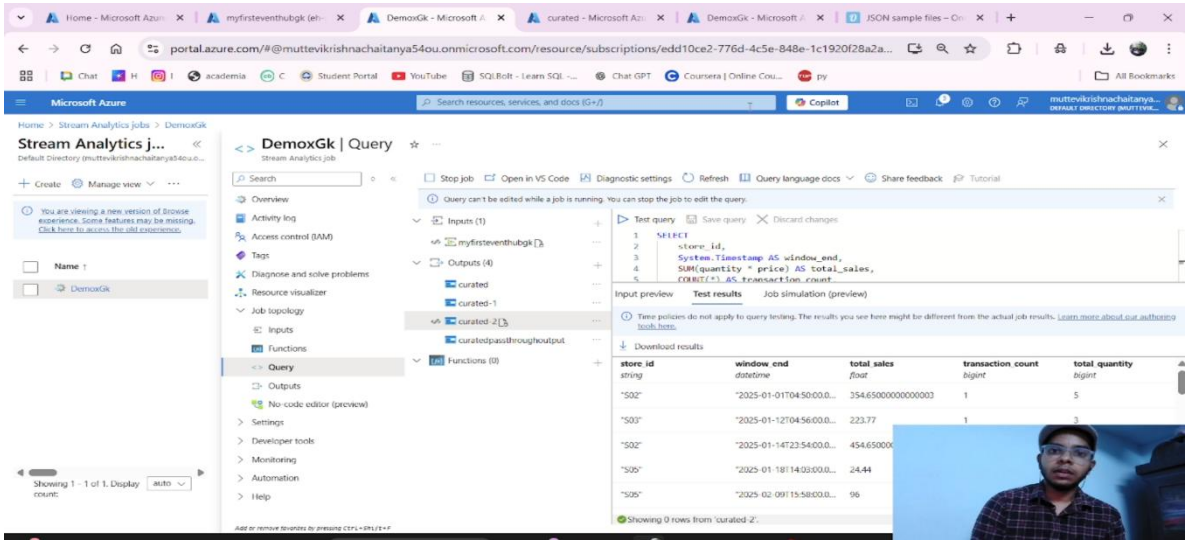
3. Output Definition:



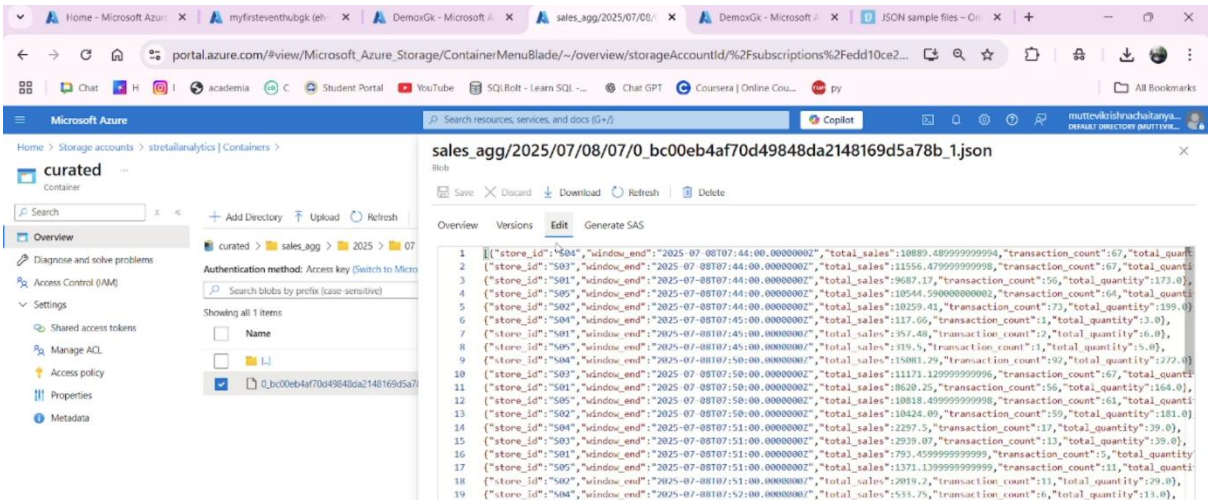
4. Start Job + Run Query:



5. Verify Query Output:

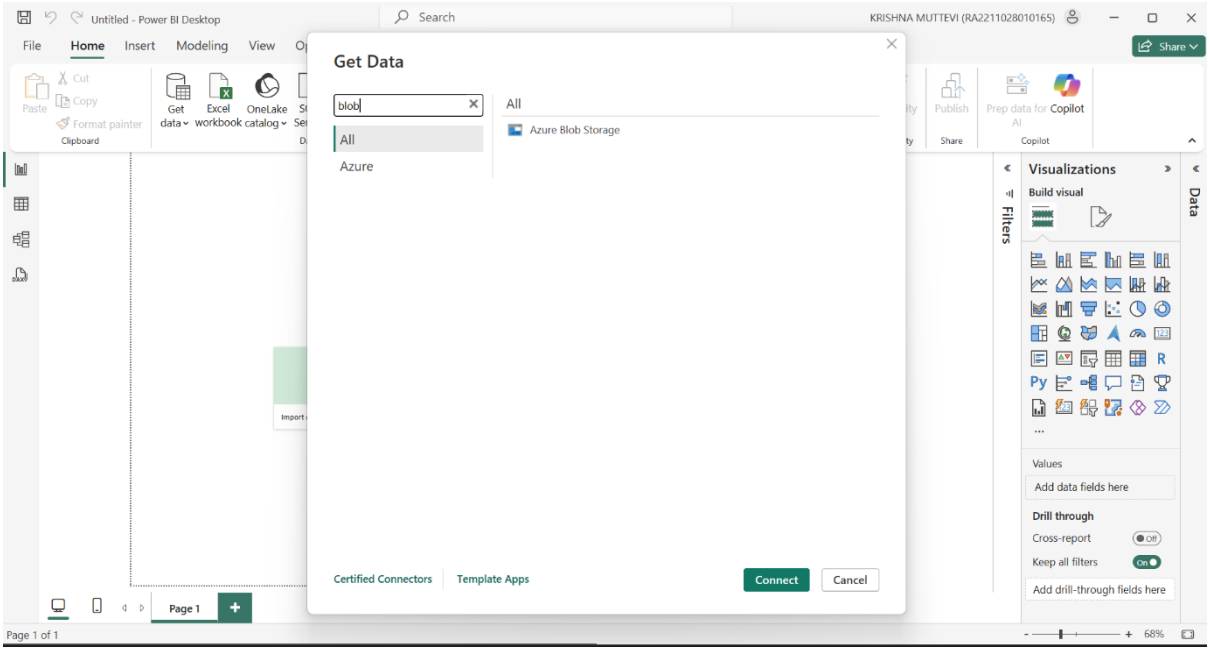


6. Check Output in ADLS2 :

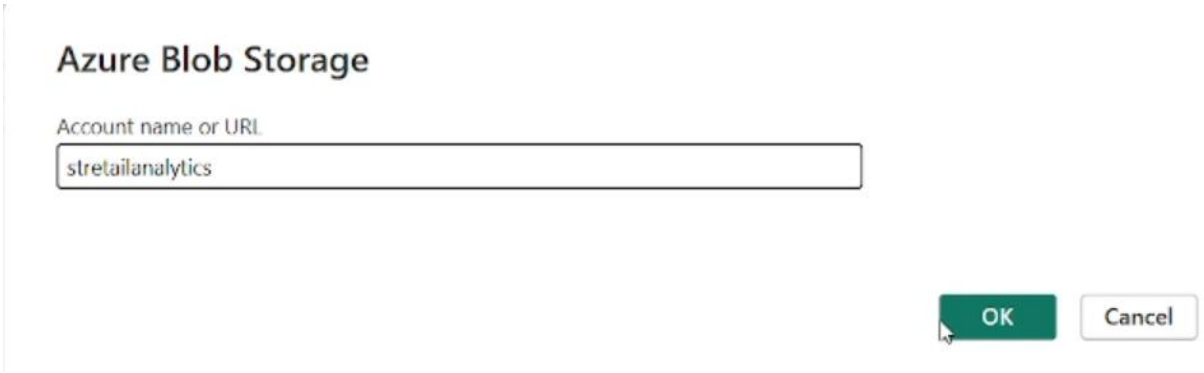


Power Bi Connection:

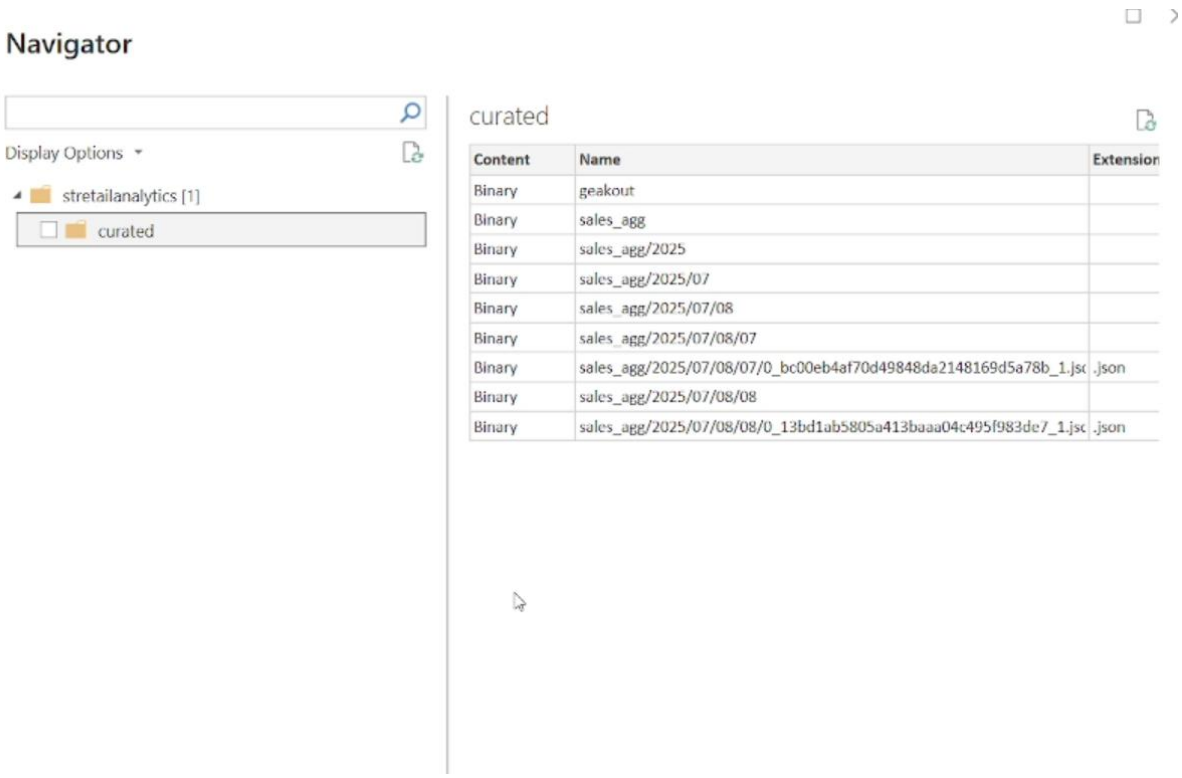
1. Get Blob Storage link and Use in Get Data of Power Bi:



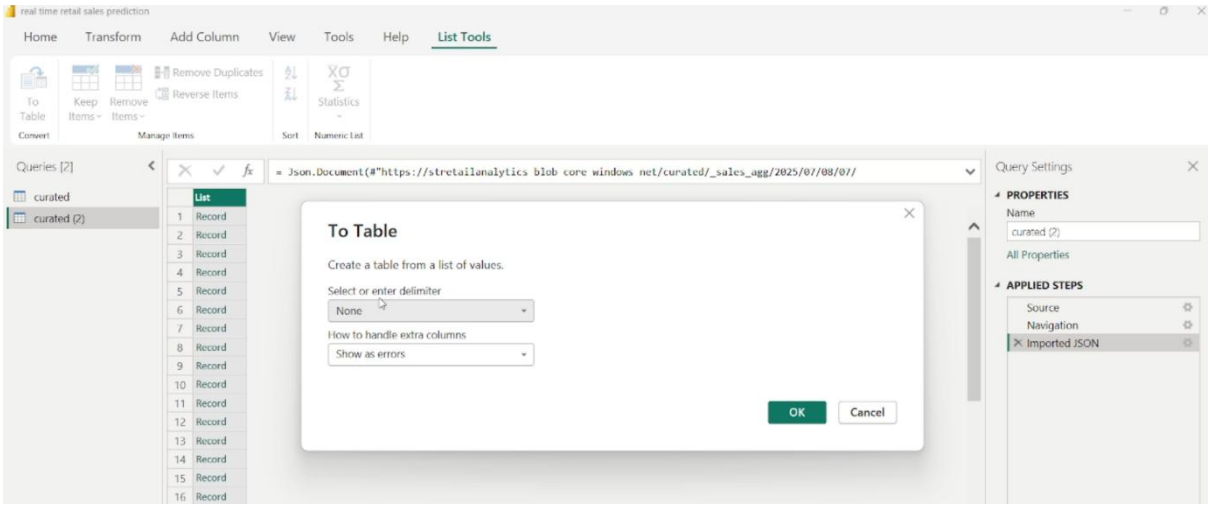
2. Enter Storage Account Details:



3. Enter Secure Blob Storage Url + Transform Data:



4. Select Json and Convert to Table:



5. Create a Dashboard: