**Case Study 1: Retail Sales Data Processing (using Azure blob storage and Azure Data Factory)**

**Client:** XYZ Retail Inc.

**Objective:** To streamline the process of ingesting, transforming, and storing retail sales data for reporting and analysis.

**Background:** XYZ Retail Inc. collects daily sales data from multiple store locations in CSV format. This data needs to be processed, aggregated, and transformed before being loaded into a new storage container for analysis.

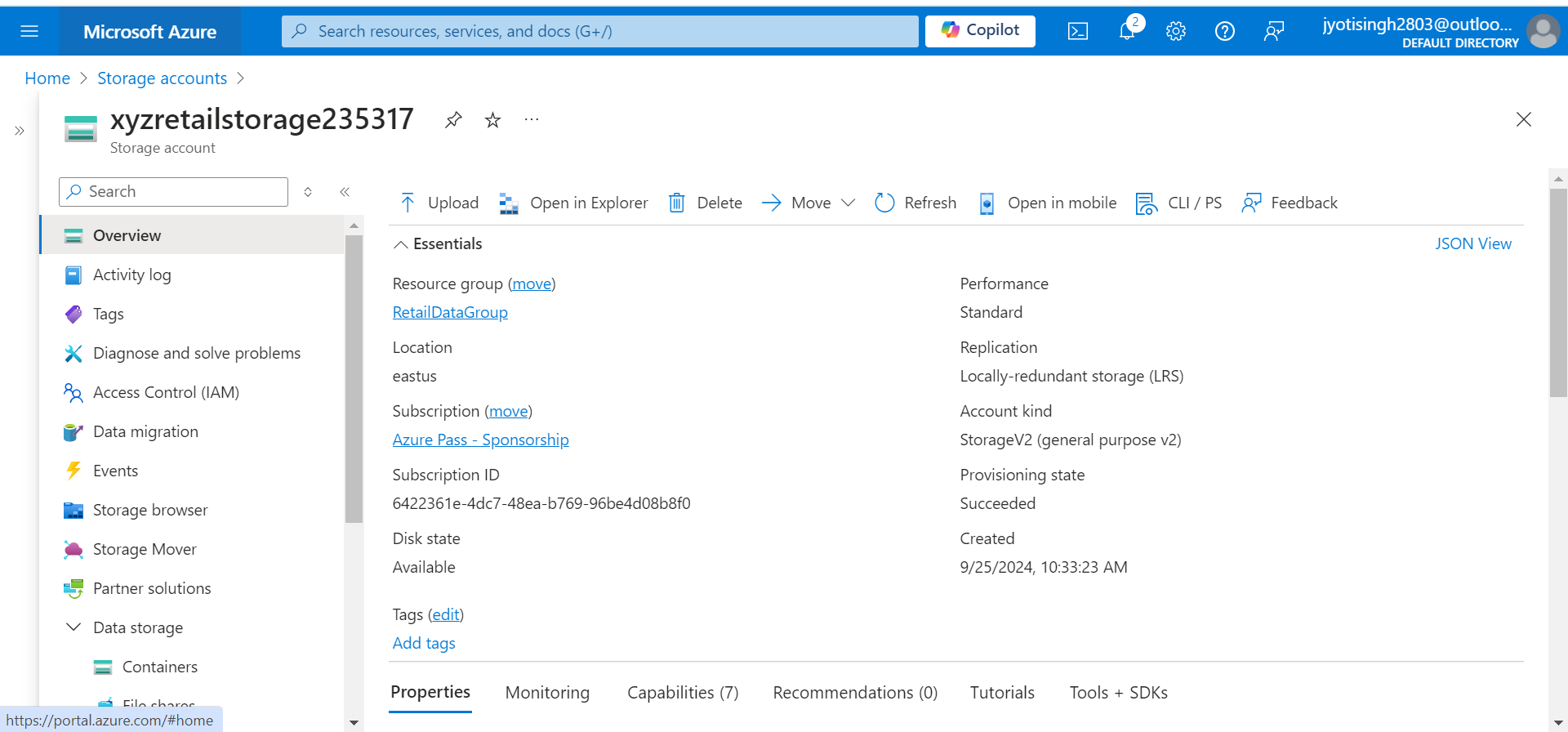
**Requirements:**

1. **Data Storage:**
   * **Source Container:** source-data in Azure Blob Storage where raw CSV files will be uploaded.
   * **Destination Container:** processed-data in Azure Blob Storage where transformed CSV files will be stored.
2. **Data Processing:**
   * **Data Transformation:** Perform basic transformations such as data cleaning (e.g., removing null values), aggregating sales data by date, and filtering irrelevant columns.
   * **Data Movement:** Copy the processed data from the source container to the destination container.
3. **Automation:**
   * **Pipeline Execution:** Automate the ingestion, transformation, and storage process using Azure Data Factory.
   * **Scheduling:** Run the pipeline daily to process new data files.

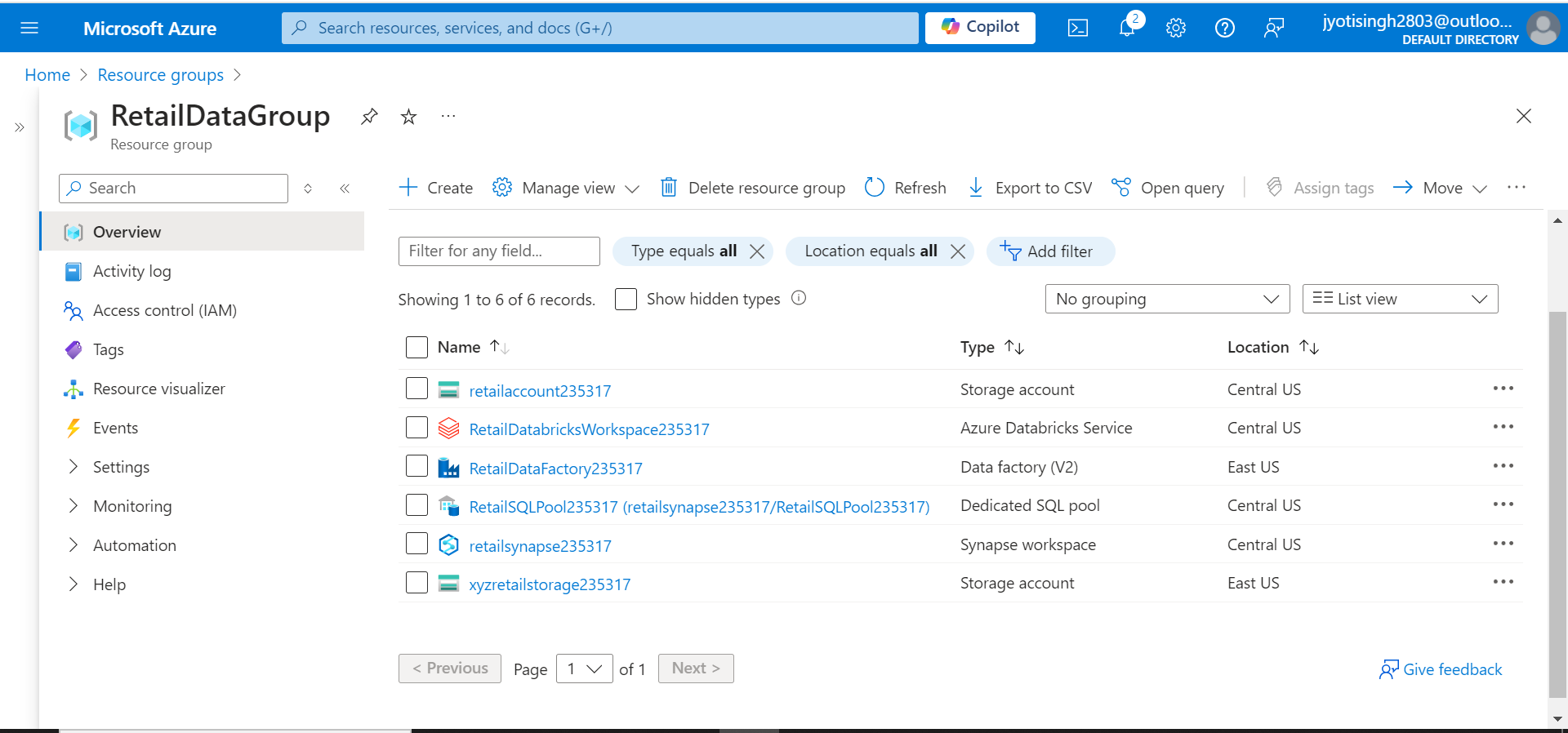
**High-Level Steps:**

**1. Set Up Azure Blob Storage**

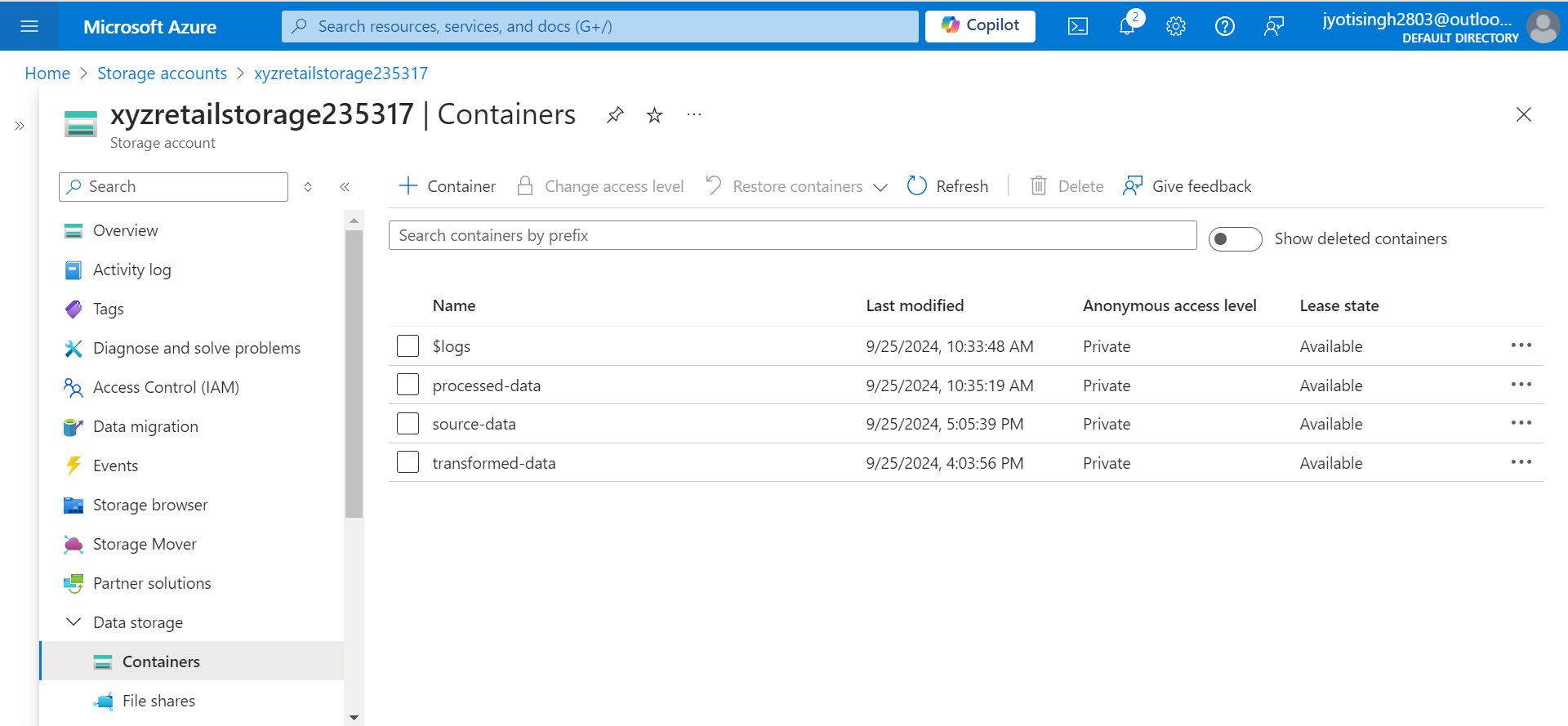
1. **Create Storage Accounts: done**
   * **Storage Account Name:** xyzretailstorage235317
   * **Resource Group:** RetailDataGroup
   * **Region:** Choose an appropriate region (e.g., East US)



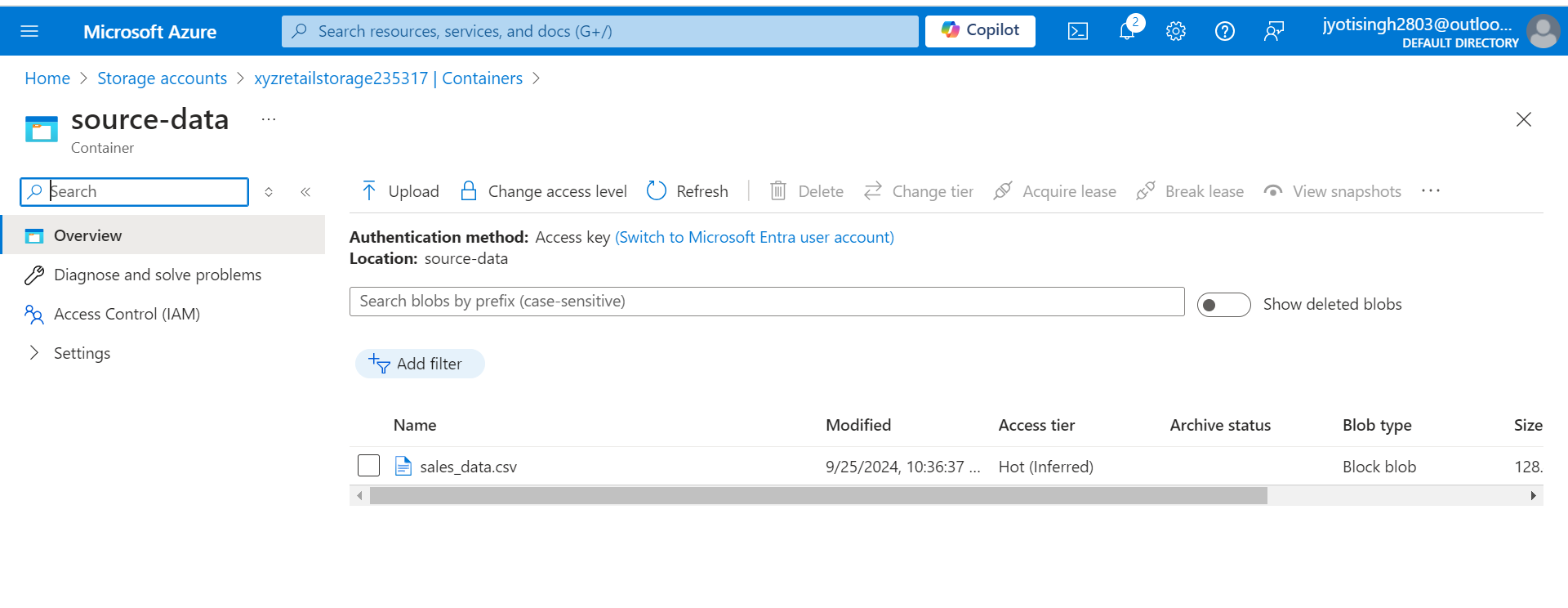
* + **Resource Group:** RetailDataGroup



1. **Create Containers: done**
   * **Source Container:** source-data
   * **Destination Container:** processed-data

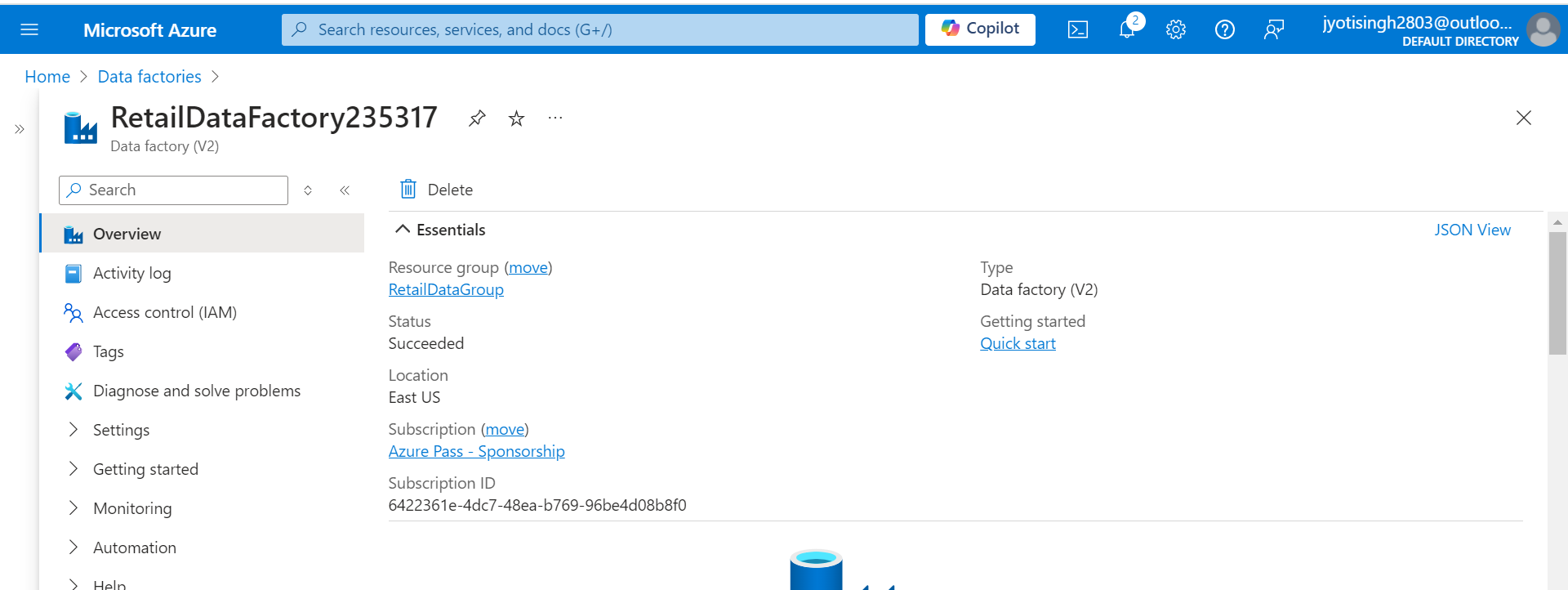


1. **Upload Sample Data: done**
   * Upload sample CSV files with sales data to the source-data container for initial testing.

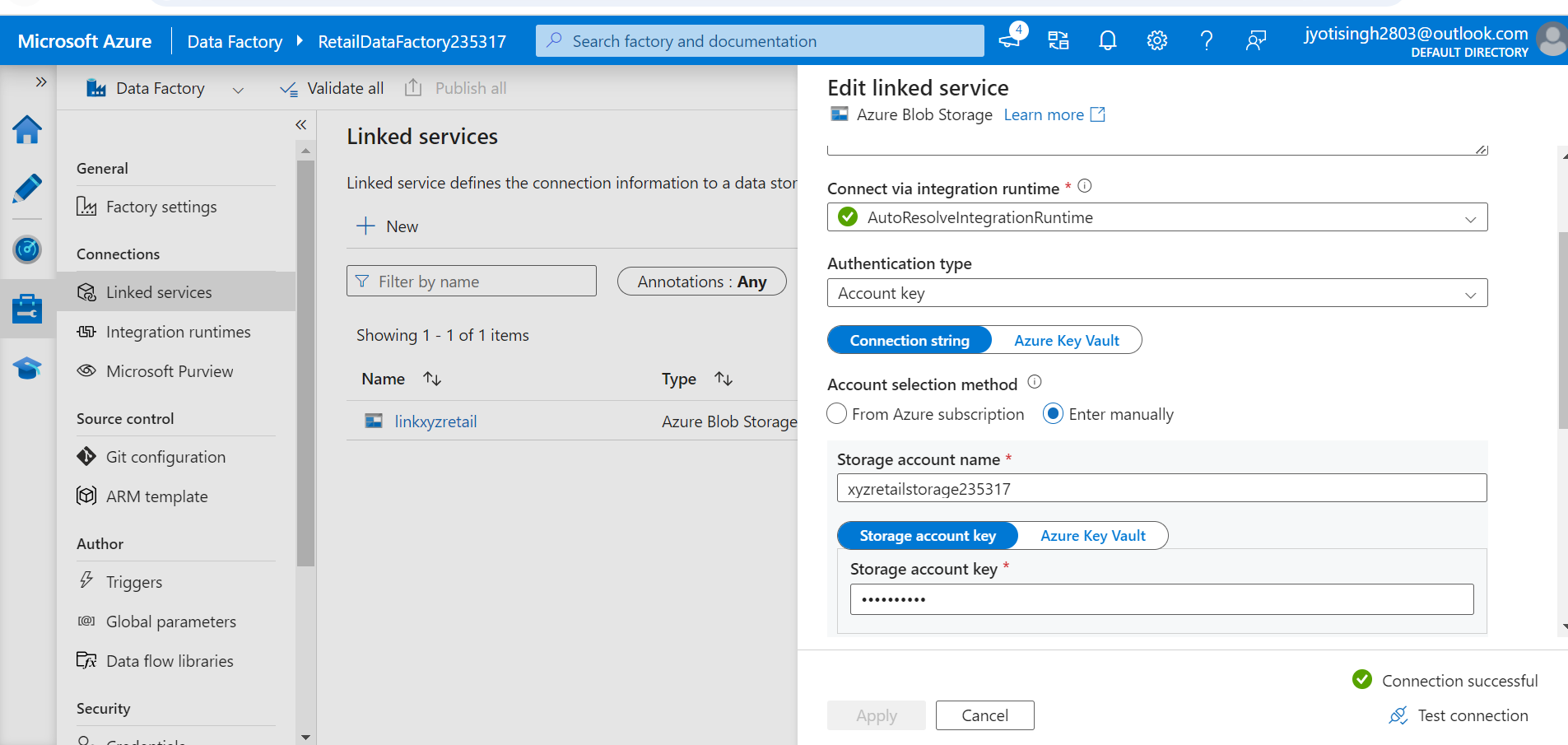


**2. Configure Azure Data Factory**

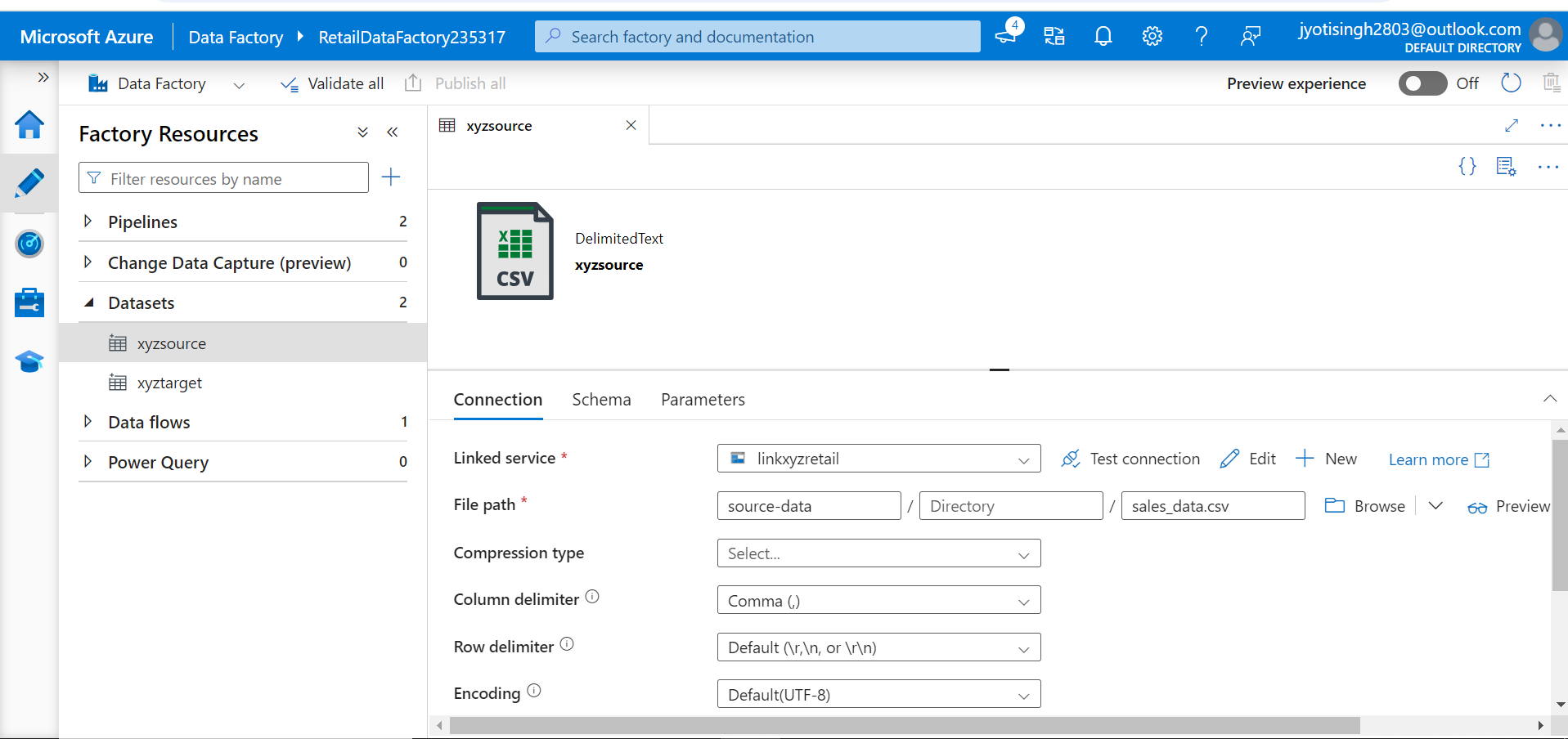
1. **Create Data Factory: done**
   * **Data Factory Name:** RetailDataFactory235317
   * **Resource Group:** RetailDataGroup
   * **Region:** Same as storage account (e.g., East US)



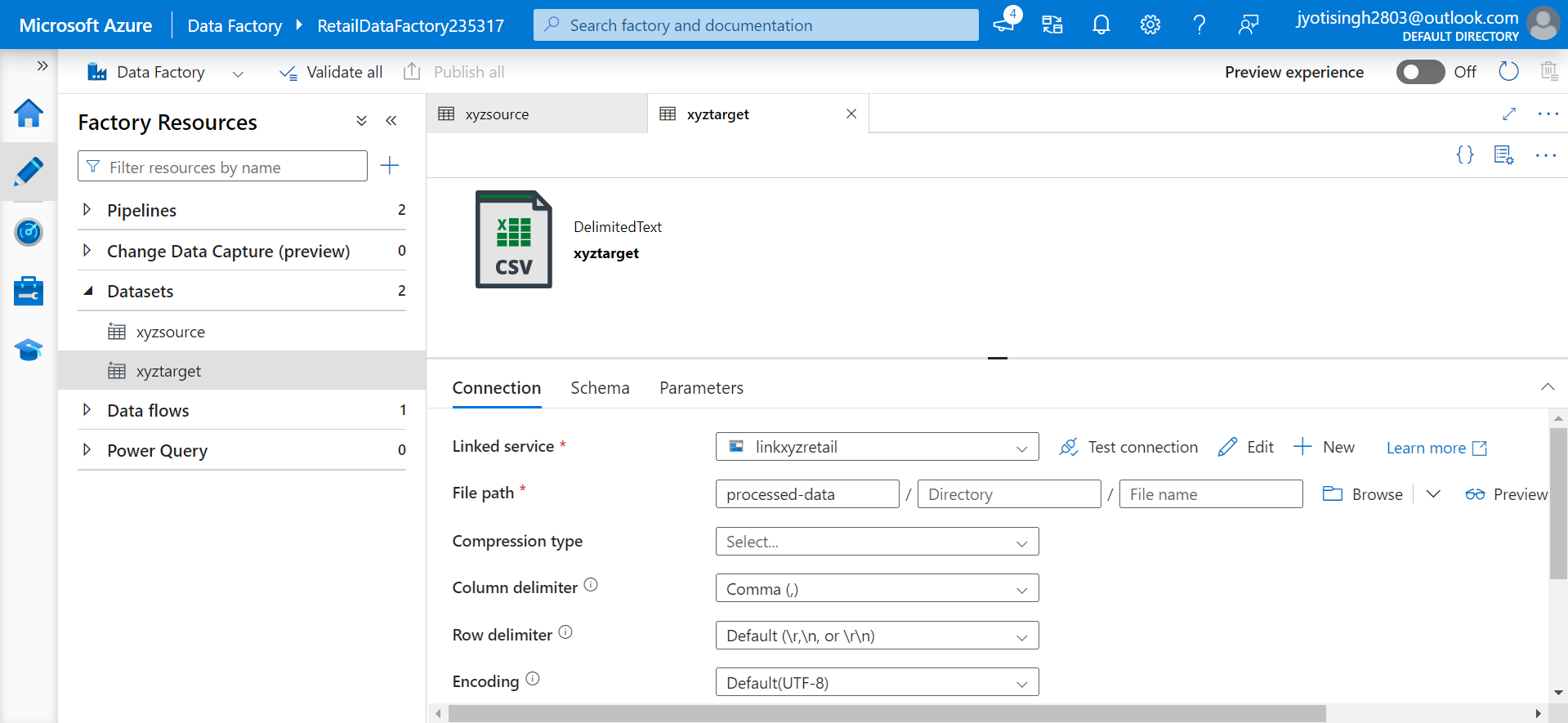
1. **Create Linked Services: done**
   * **Azure Blob Storage Linked Service: linkxyzretail**
     + Configure connection to xyzretailstorage.
     + Use the storage account connection string.



1. **Create Datasets:**
   * **Source Dataset: xyzsource**
     + Data source: source-data container
     + Format: CSV
     + Schema: Configure based on the CSV file

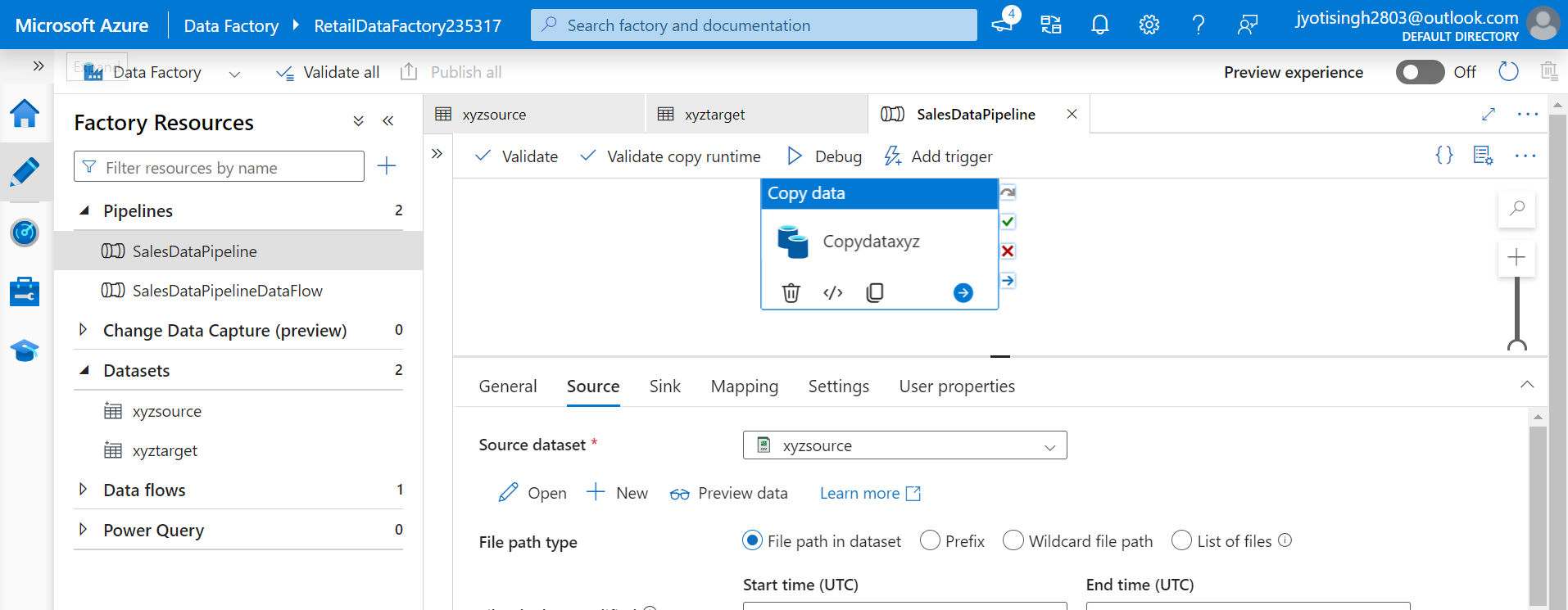


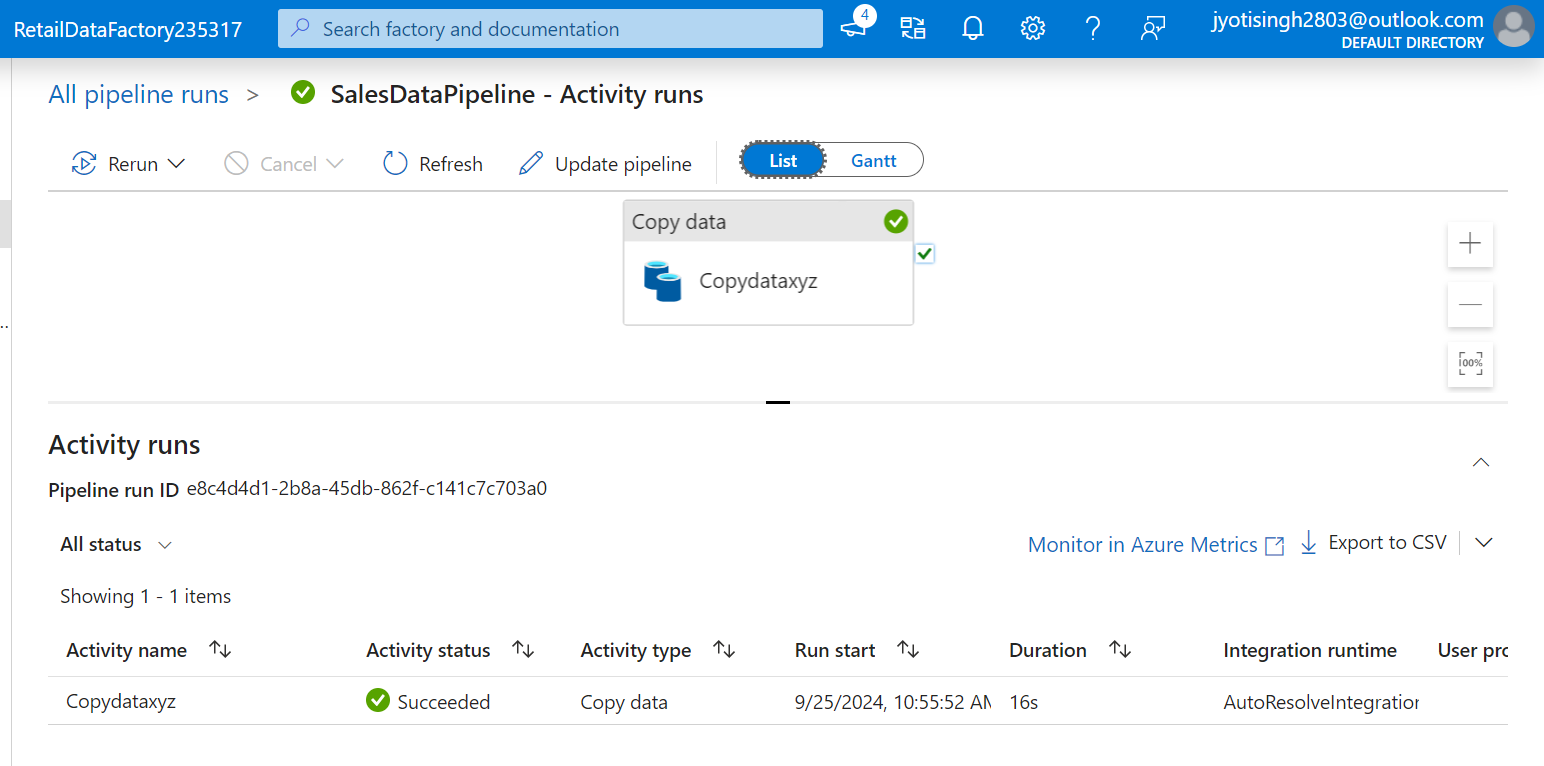
* + **Destination Dataset: xyztarget**
    - Data source: processed-data container
    - Format: CSV
    - Schema: Define or infer schema based on transformation.



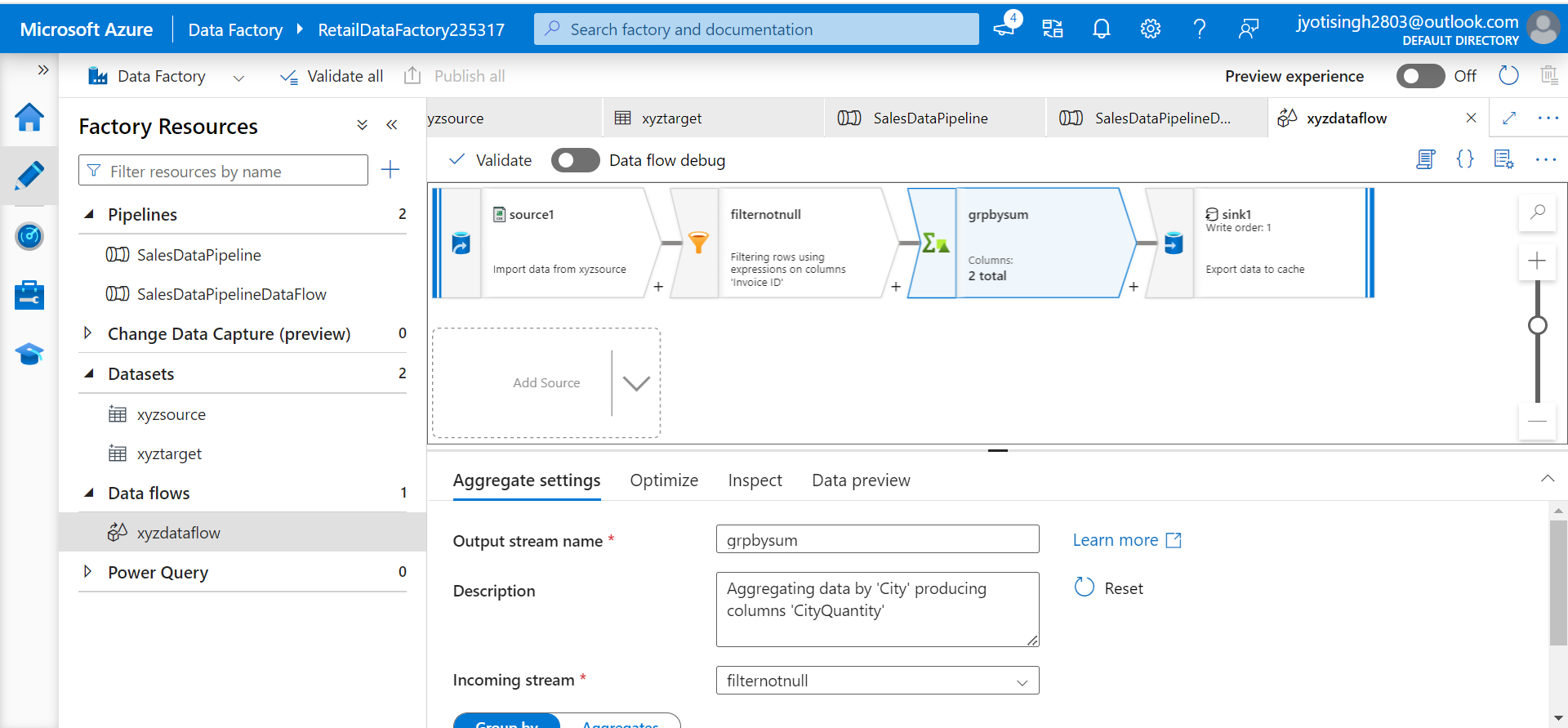
**3. Build and Configure Pipeline**

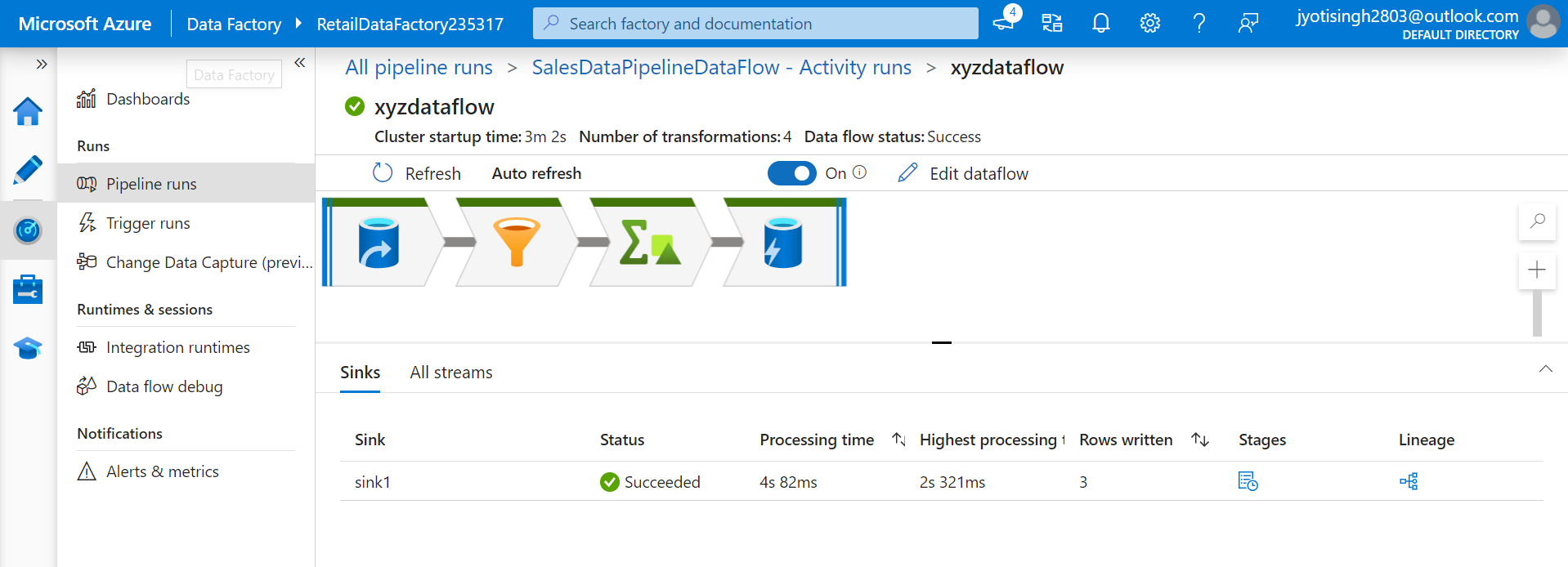
1. **Create Pipeline: done**
   * **Pipeline Name:** SalesDataPipeline
   * **Activities:**
     + **Copy Data Activity:**
       - **Source:** Link to source-data dataset.
       - **Sink:** Link to processed-data dataset.
       - **Transformation:** Optionally, configure transformations like column mapping and data filtering.





1. **Optional Data Flow: done** SalesDataPipelineDataFlow
   * Create a Data Flow if more complex transformations are needed. **xyzdataflow**
   * Add steps for data cleaning (e.g., removing null values) and aggregation (e.g., summing sales by date).



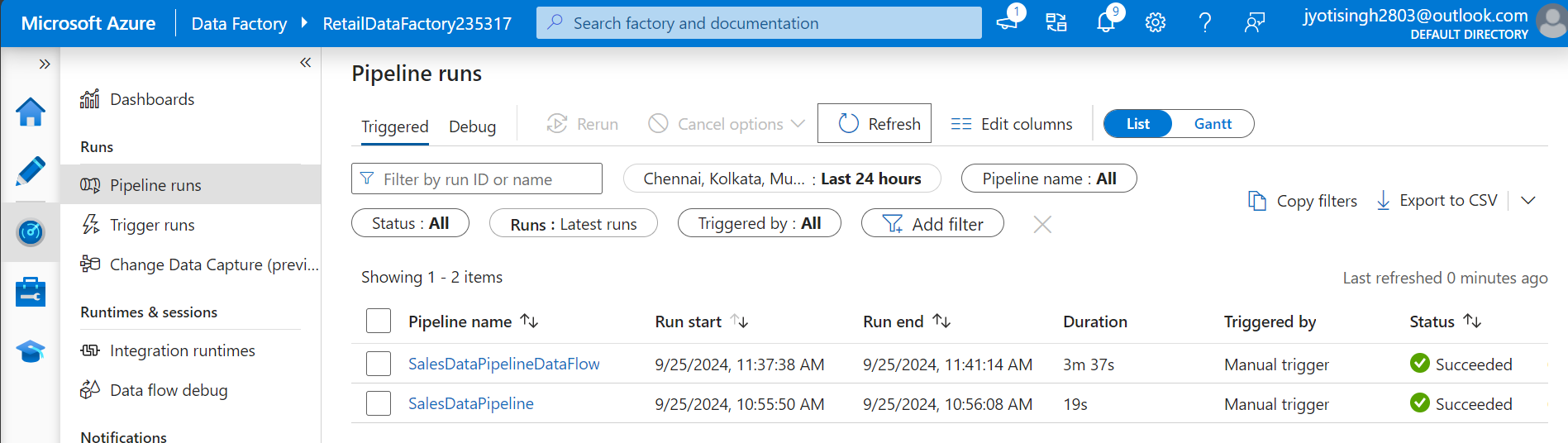


**4. Automate and Monitor**

1. **Create Trigger:**
   * **Trigger Type:** Schedule
   * **Frequency:** Daily
   * **Start Time:** Set the start time according to your business requirements (e.g., after daily data uploads)

**NOTE: Aware about steps, but implementation will cost.**

1. **Monitor Pipeline:**
   * Check the **"Monitor"** section in Azure Data Factory to track the execution and ensure successful data processing.



**Case Study 2: Retail Sales Data Processing (using Azure DataBricks)**

**Client:** XYZ Retail Inc.

**Objective:** To enhance the data processing pipeline by incorporating Azure Databricks for advanced data transformations and using Azure Data Factory for orchestration.

**Background:** XYZ Retail Inc. wants to process retail sales data by performing complex transformations with Azure Databricks . The goal is to load raw data from Blob Storage, transform it using Databricks, and store the processed data back in Blob Storage.

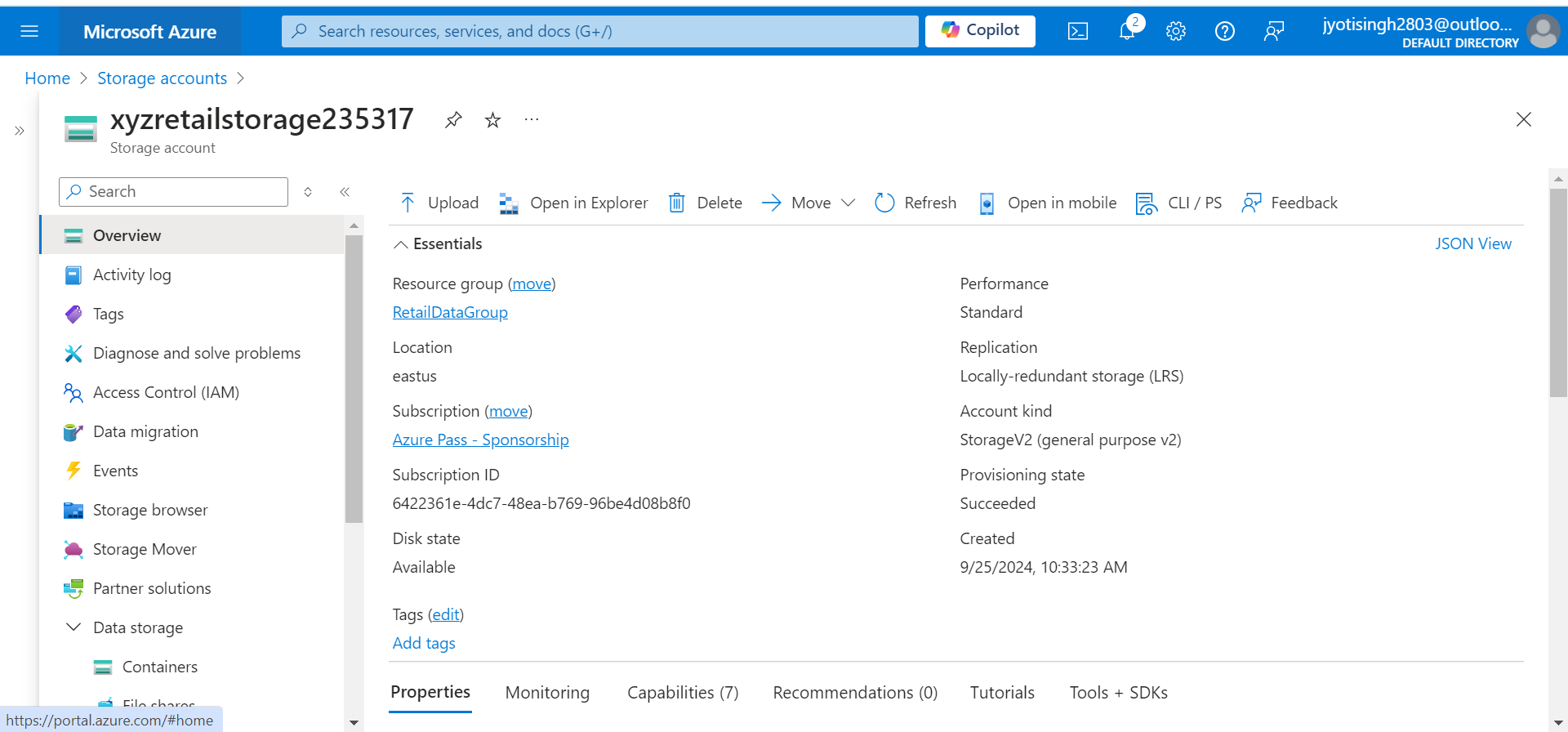
**Requirements:**

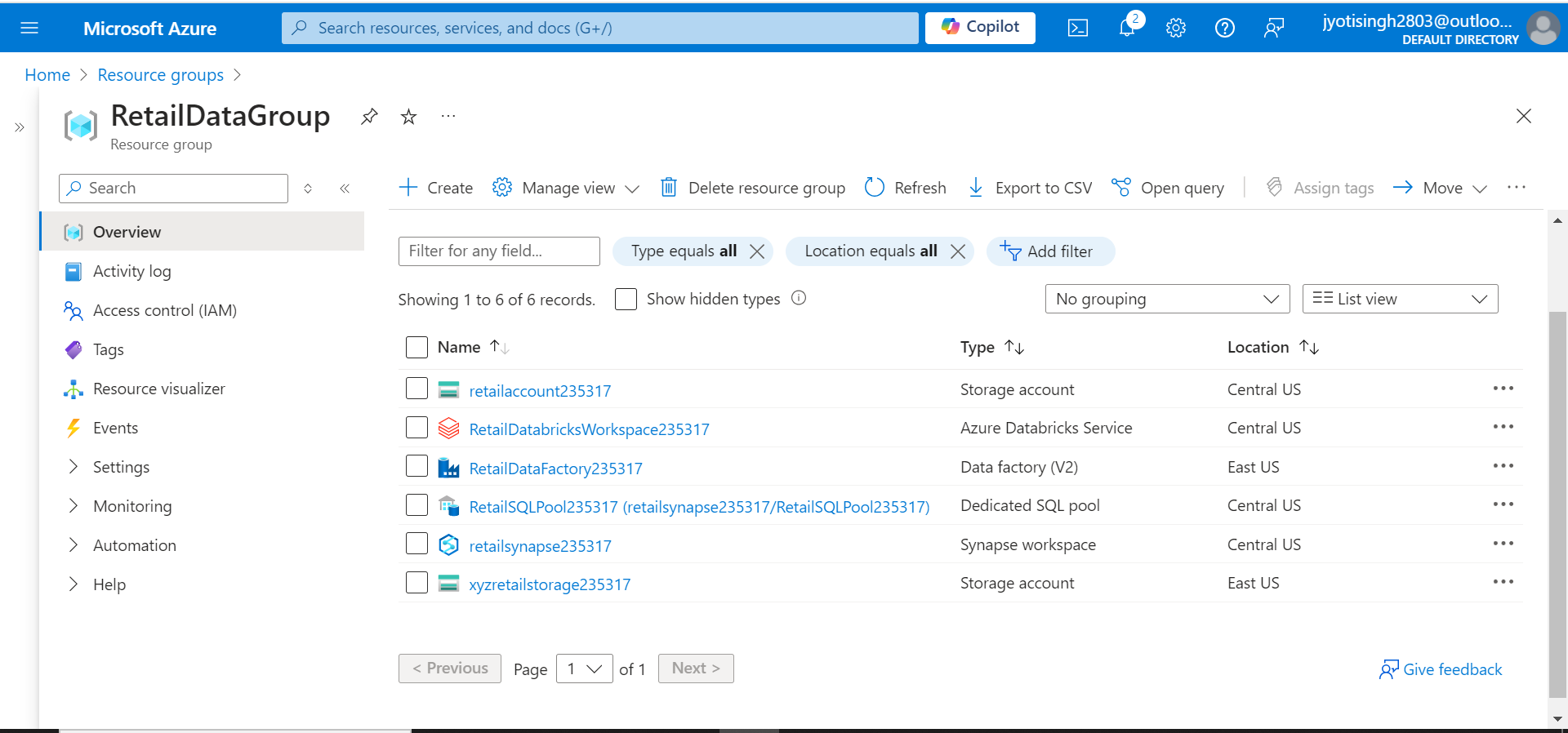
1. **Data Storage:**
   * **Source Container:** source-data in Azure Blob Storage for raw CSV files.
   * **Target Container:** transformed-data in Azure Blob Storage for data processed by Azure Databricks.
2. **Data Processing:**
   * **Transformation with Databricks:** Use PySpark and SQL in Azure Databricks to perform complex data transformations and aggregations.
   * **Data Movement:** Move data between Blob Storage and Databricks and store the processed data back in Blob Storage.

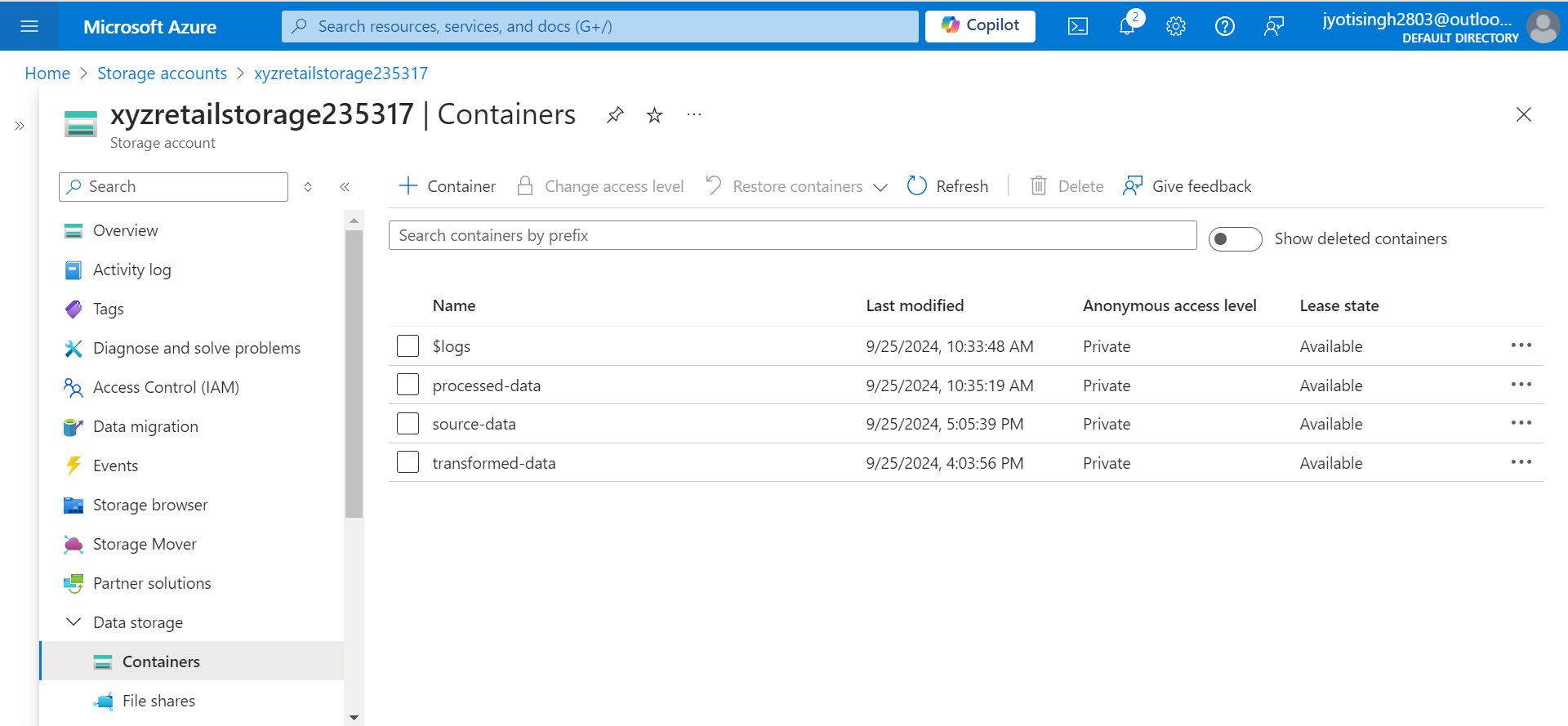
**High-Level Steps:**

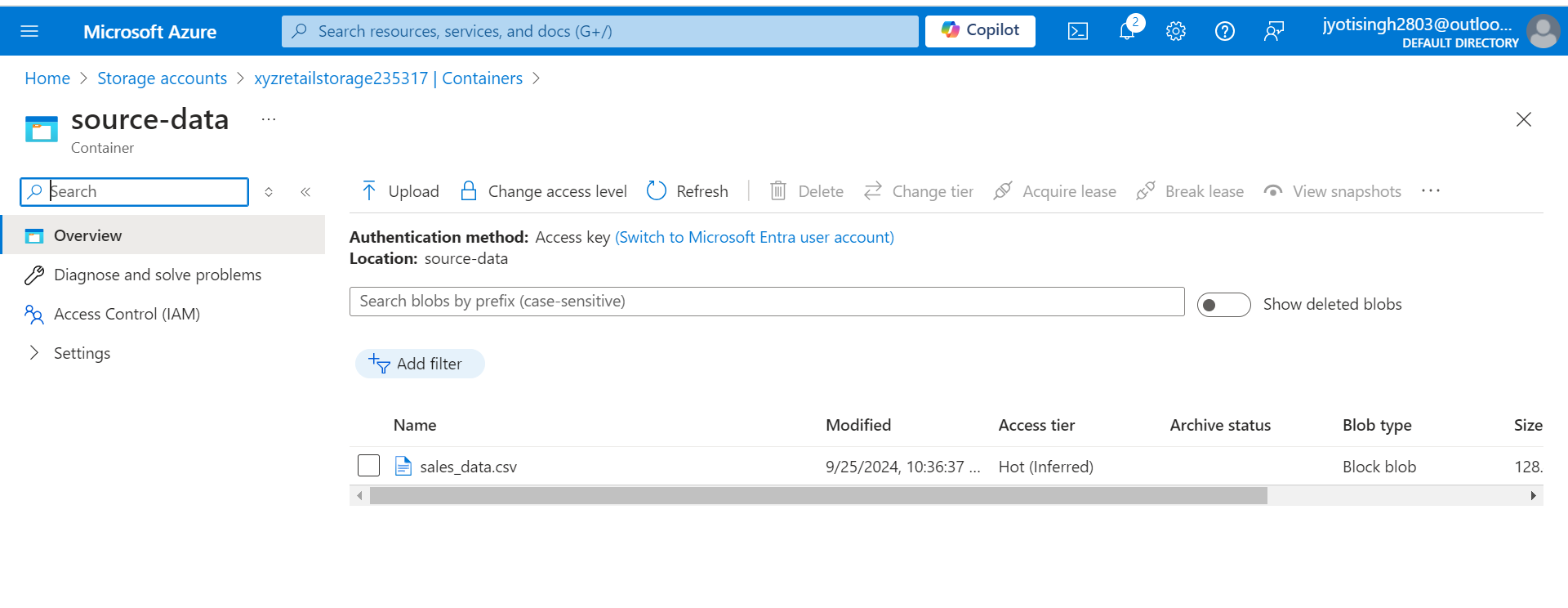
**1. Set Up Azure Blob Storage**

1. **Create Storage Accounts:**
   * **Storage Account Name:** xyzretailstorage
   * **Resource Group:** RetailDataGroup
   * **Region:** Choose an appropriate region (e.g., East US)
2. **Create Containers:**
   * **Source Container:** source-data
   * **Target Container:** transformed-data
3. **Upload Sample Data:**
   * Upload sales\_data.csv to the source-data container.



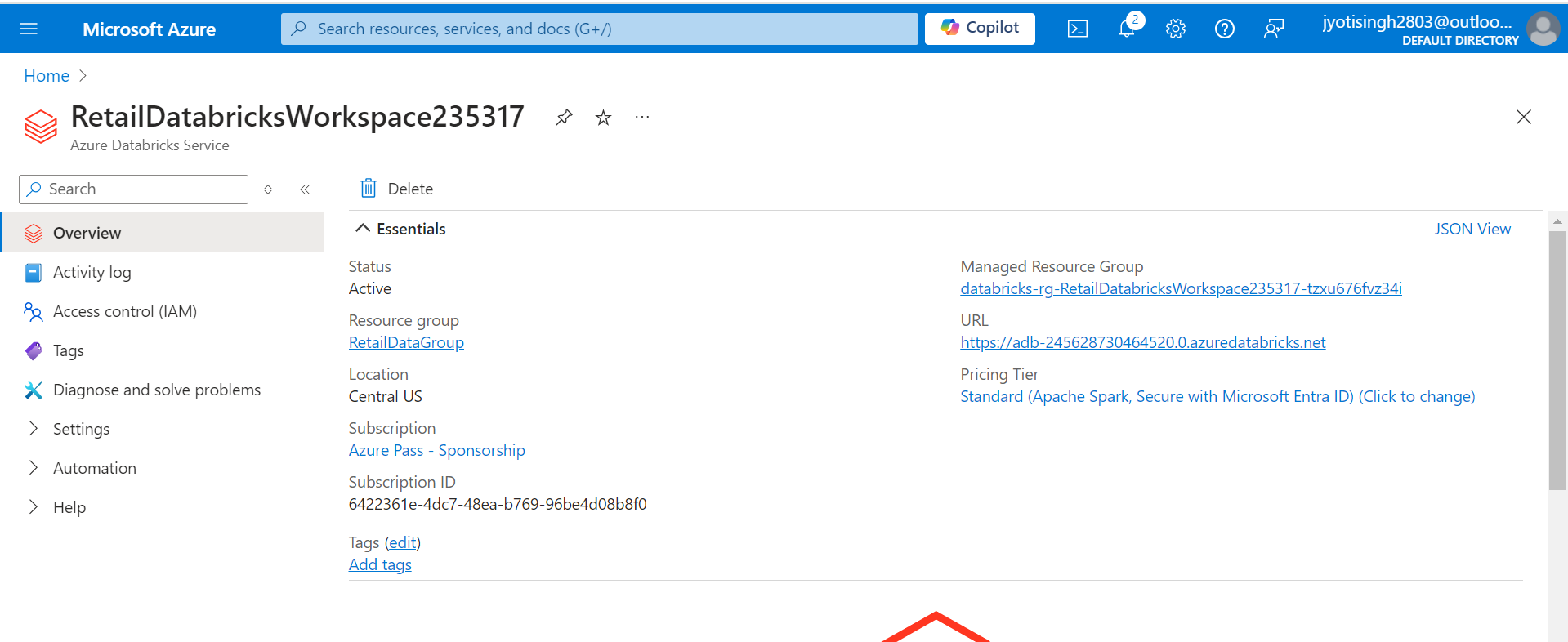




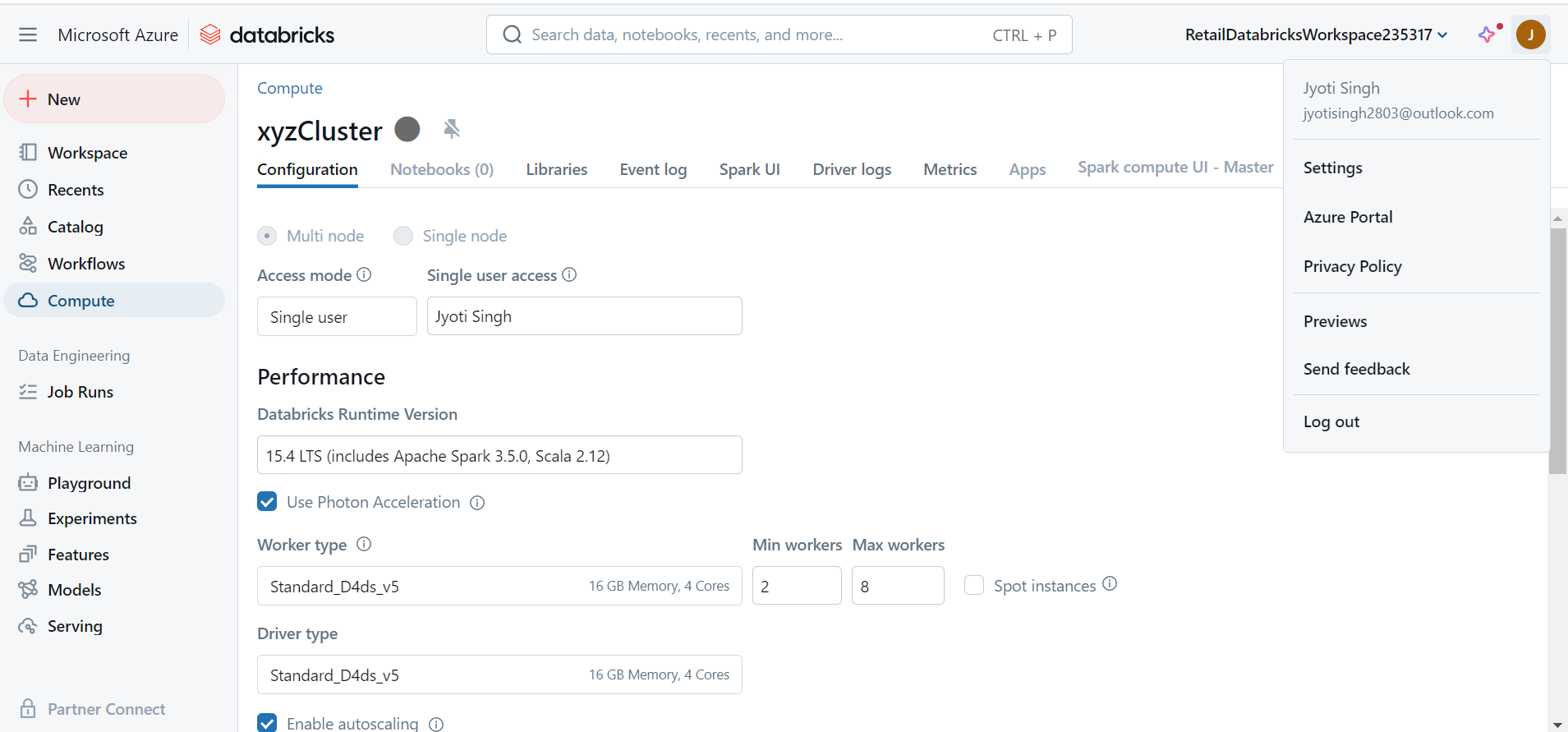


**2. Set Up Azure Databricks**

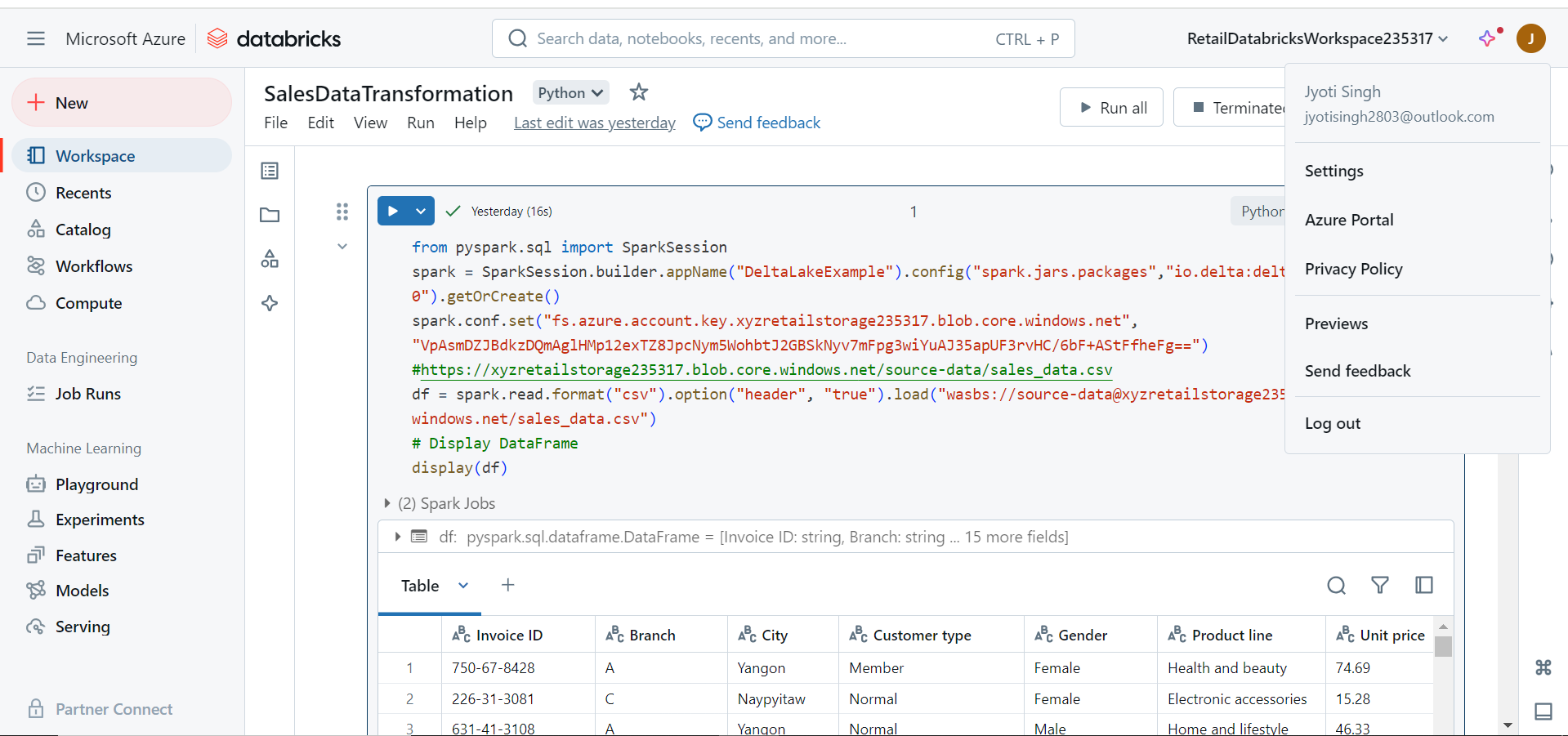
1. **Create an Azure Databricks Workspace: done**
   * **Workspace Name:** RetailDatabricksWorkspace235317
   * **Resource Group:** RetailDataGroup
   * **Region:** Same as storage account (e.g., East US)

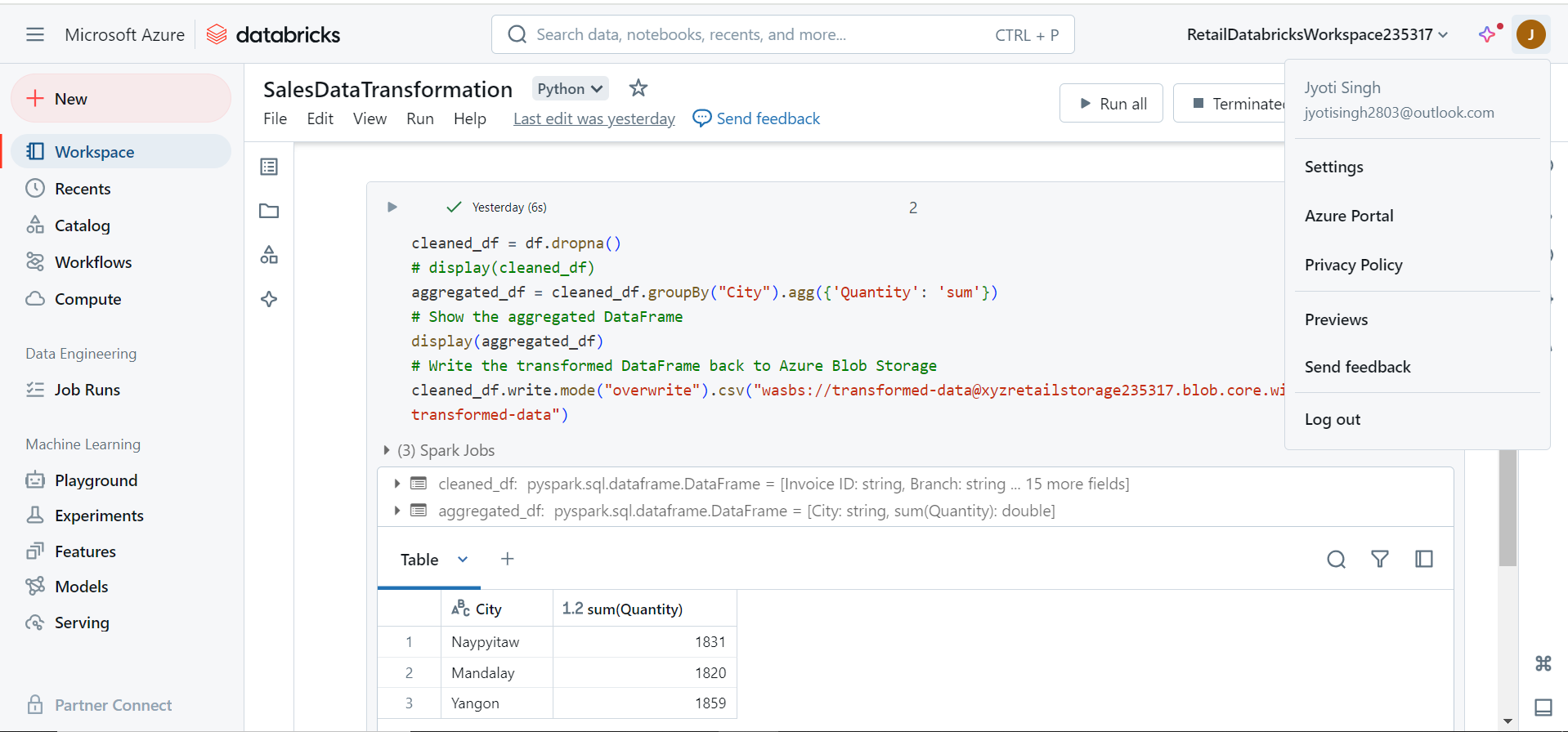


1. **Create a Cluster: xyzCluster done**
   * Set up a cluster in Azure Databricks with appropriate configurations for processing the data.



1. **Create a Databricks Notebook: done**
   * **Notebook Name:** SalesDataTransformation
   * Use PySpark or SQL to read the CSV file from Azure Blob Storage, perform transformations (e.g., cleaning, aggregating), and write the transformed data back to the transformed-data container.





from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("DeltaLakeExample").config("spark.jars.packages","io.delta:delta-core\_2.12:1.0.0").getOrCreate()

spark.conf.set("fs.azure.account.key.xyzretailstorage235317.blob.core.windows.net", "VpAsmDZJBdkzDQmAglHMp12exTZ8JpcNym5WohbtJ2GBSkNyv7mFpg3wiYuAJ35apUF3rvHC/6bF+AStFfheFg==")

#https://xyzretailstorage235317.blob.core.windows.net/source-data/sales\_data.csv

df = spark.read.format("csv").option("header", "true").load("wasbs://source-data@xyzretailstorage235317.blob.core.windows.net/sales\_data.csv")

# Display DataFrame

display(df)

cleaned\_df = df.dropna()

# display(cleaned\_df)

aggregated\_df = cleaned\_df.groupBy("City").agg({'Quantity': 'sum'})

# Show the aggregated DataFrame

display(aggregated\_df)

# Write the transformed DataFrame back to Azure Blob Storage

cleaned\_df.write.mode("overwrite").csv("wasbs://transformed-data@xyzretailstorage235317.blob.core.windows.net/transformed-data")

**Case Study 3 : Analyzing Retail Sales Data with Azure Synapse Analytics**

**Client:** XYZ Retail Inc.

**Objective:** To leverage Azure Synapse Analytics for data integration, transformation, and analysis to gain insights from retail sales data.

**Background:** XYZ Retail Inc. needs to analyze retail sales data stored in Azure Blob Storage using Azure Synapse Analytics. The goal is to perform data transformation and analysis to generate actionable insights, such as total sales per store and sales trends over time.

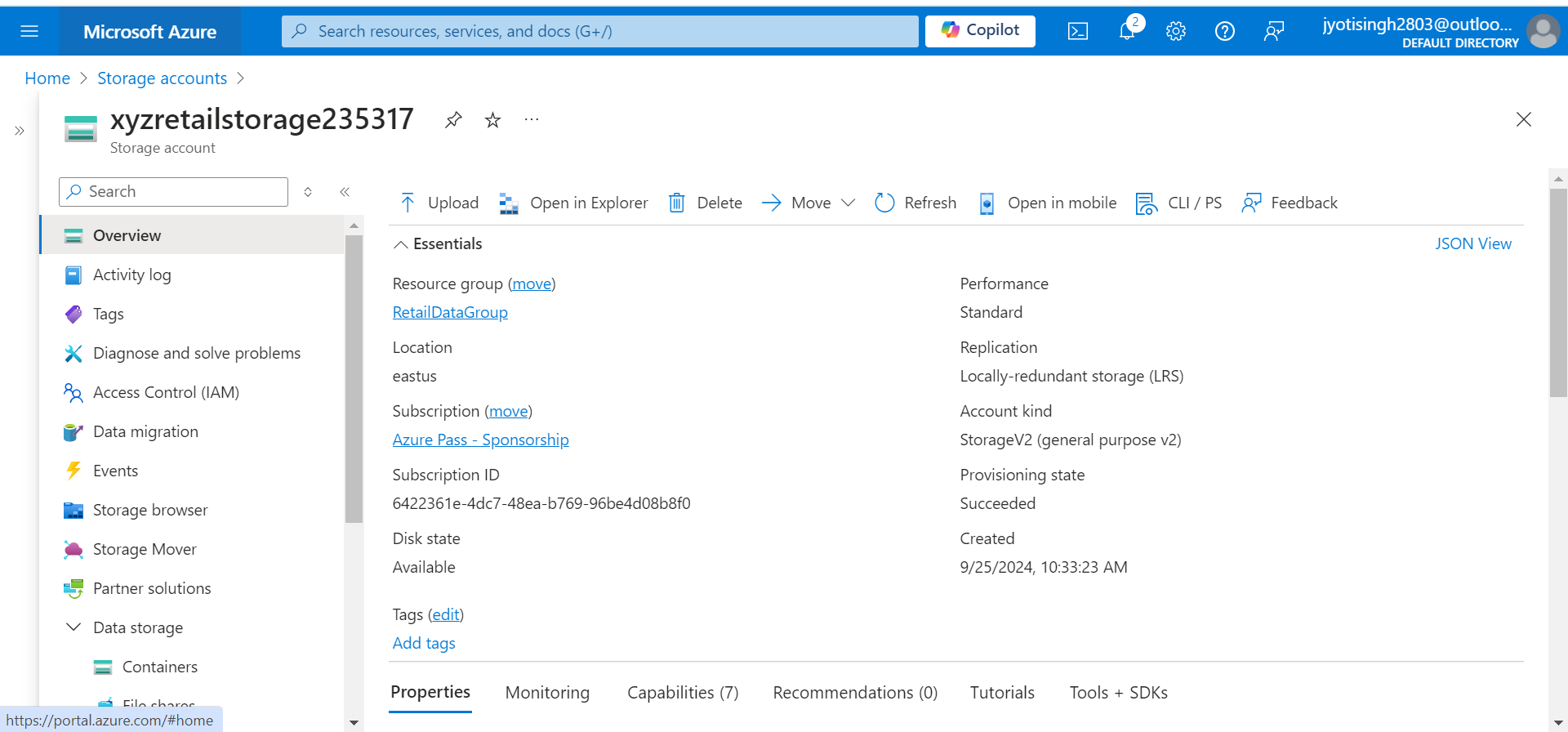
**Requirements:**

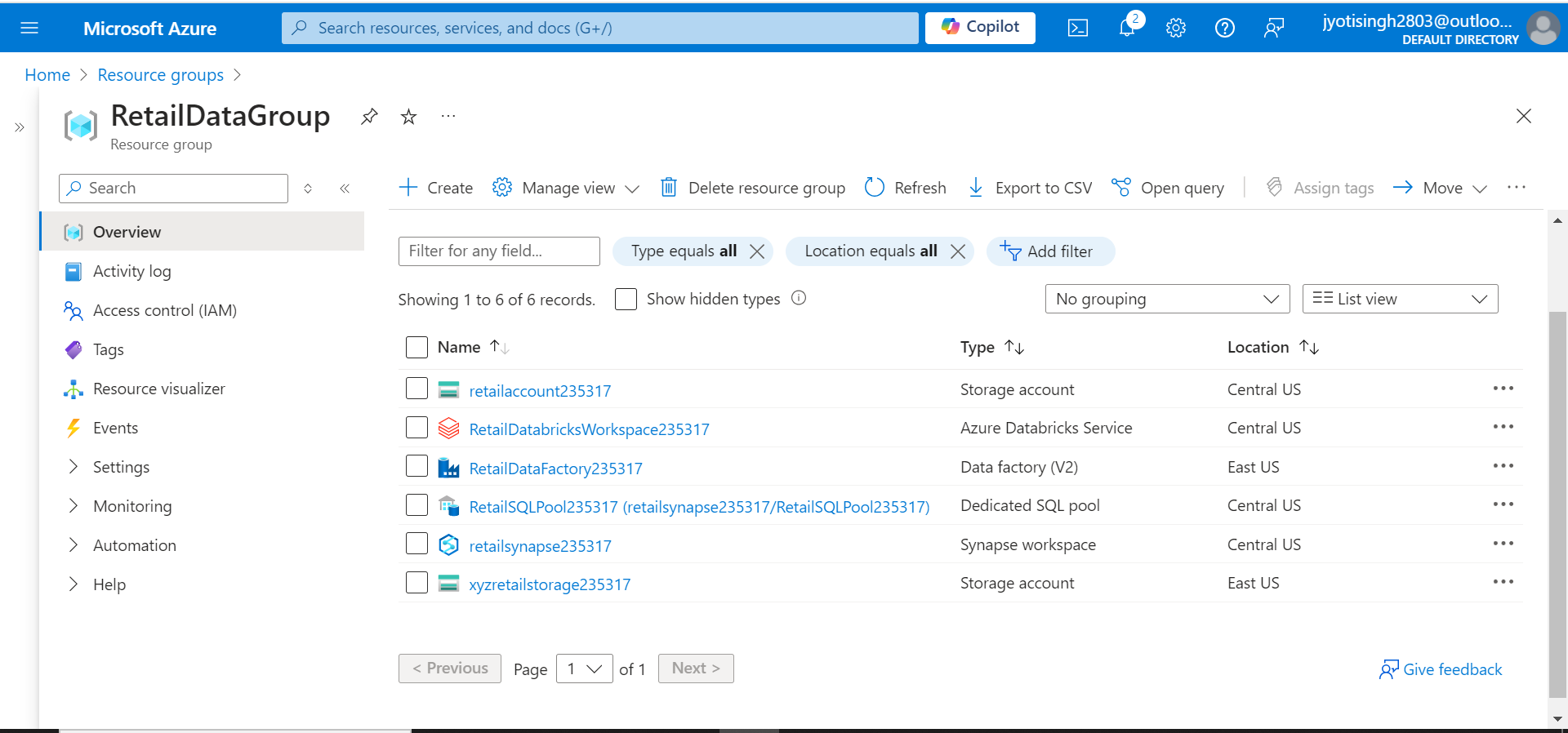
1. **Data Storage:**
   * **Source Container:** source-data in Azure Blob Storage for raw CSV files.
2. **Data Processing and Analysis:**
   * **Data Ingestion:** Load data from Azure Blob Storage into Azure Synapse Analytics.
   * **Data Transformation:** Perform basic transformations such as cleaning and aggregating data.
   * **Data Analysis:** Execute SQL queries to analyze the data and generate insights.

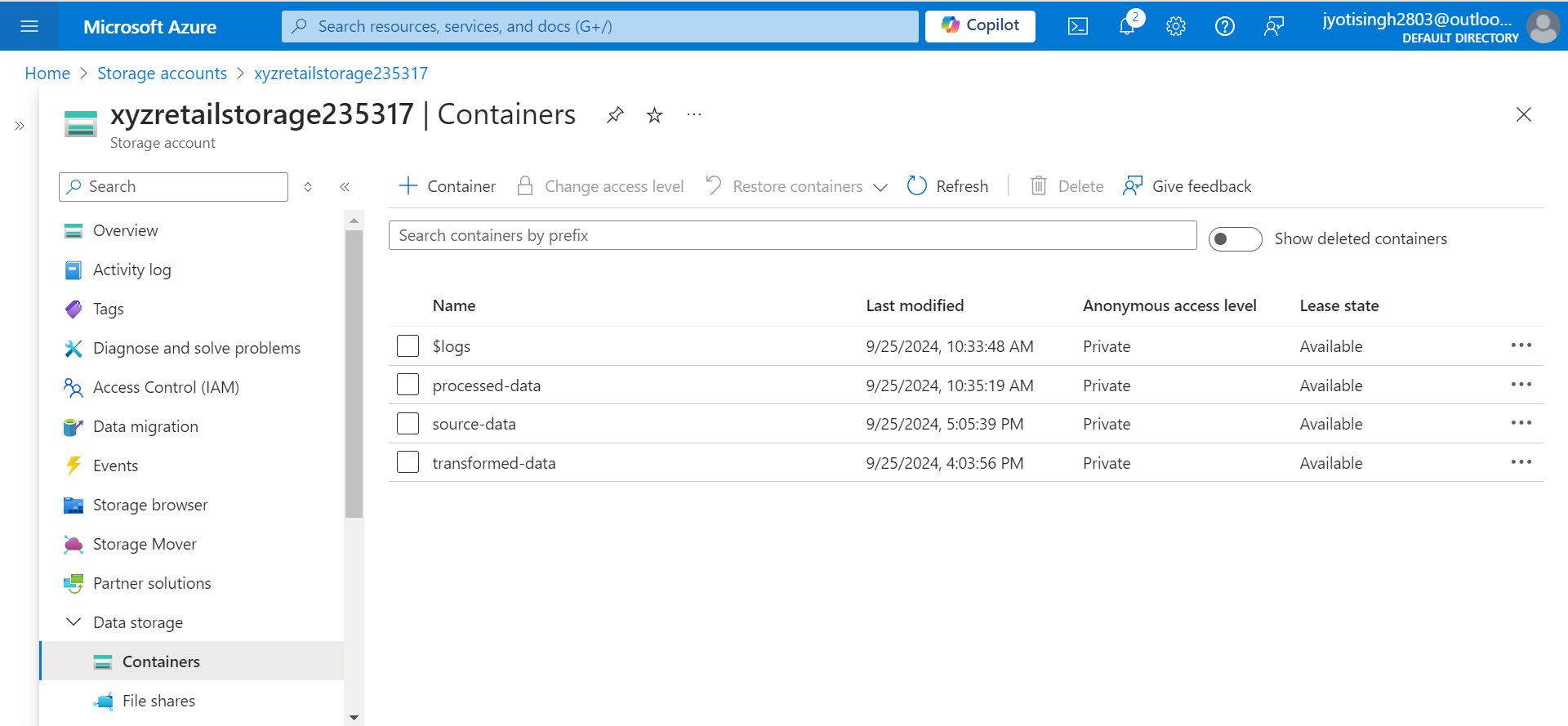
**High-Level Steps:**

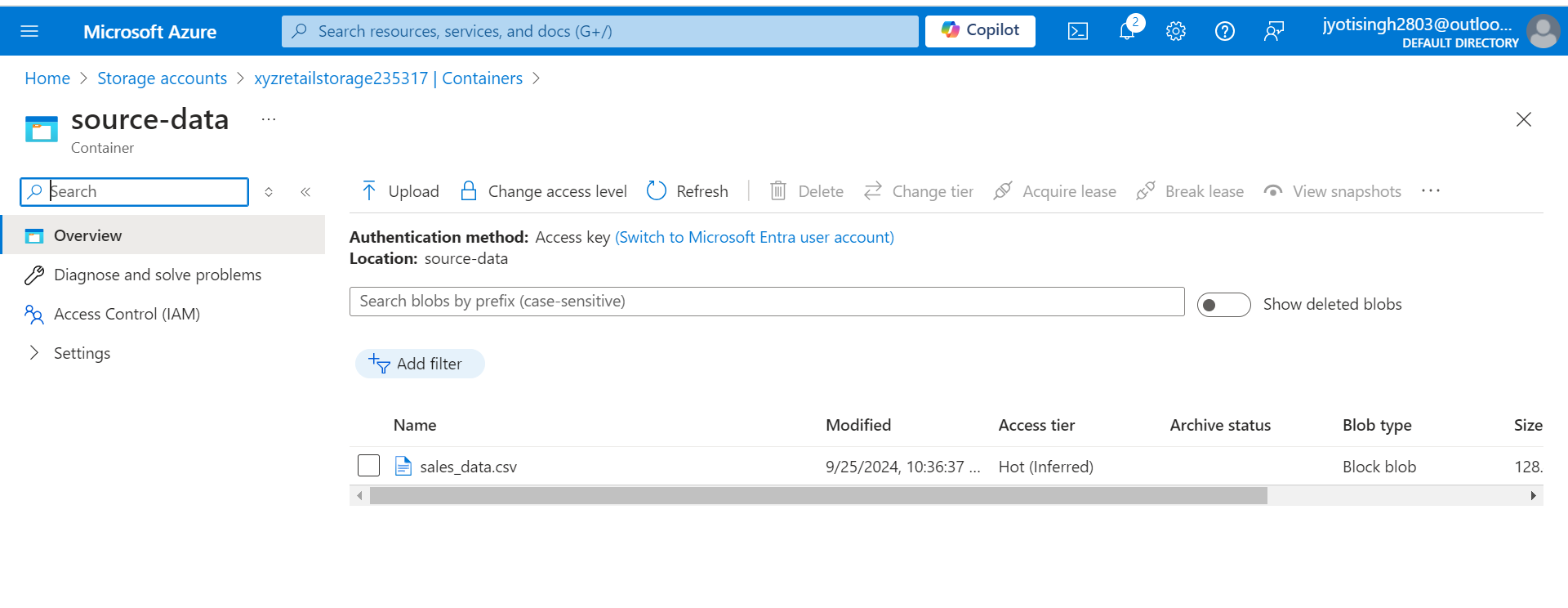
**1. Set Up Azure Blob Storage done**

1. **Create Storage Account:**
   * **Storage Account Name:** xyzretailstorage
   * **Resource Group:** RetailDataGroup
   * **Region:** Choose an appropriate region (e.g., East US)
2. **Create a Container:**
   * **Container Name:** source-data
3. **Upload Sample Data:**
   * Upload sales\_data.csv to the source-data container.



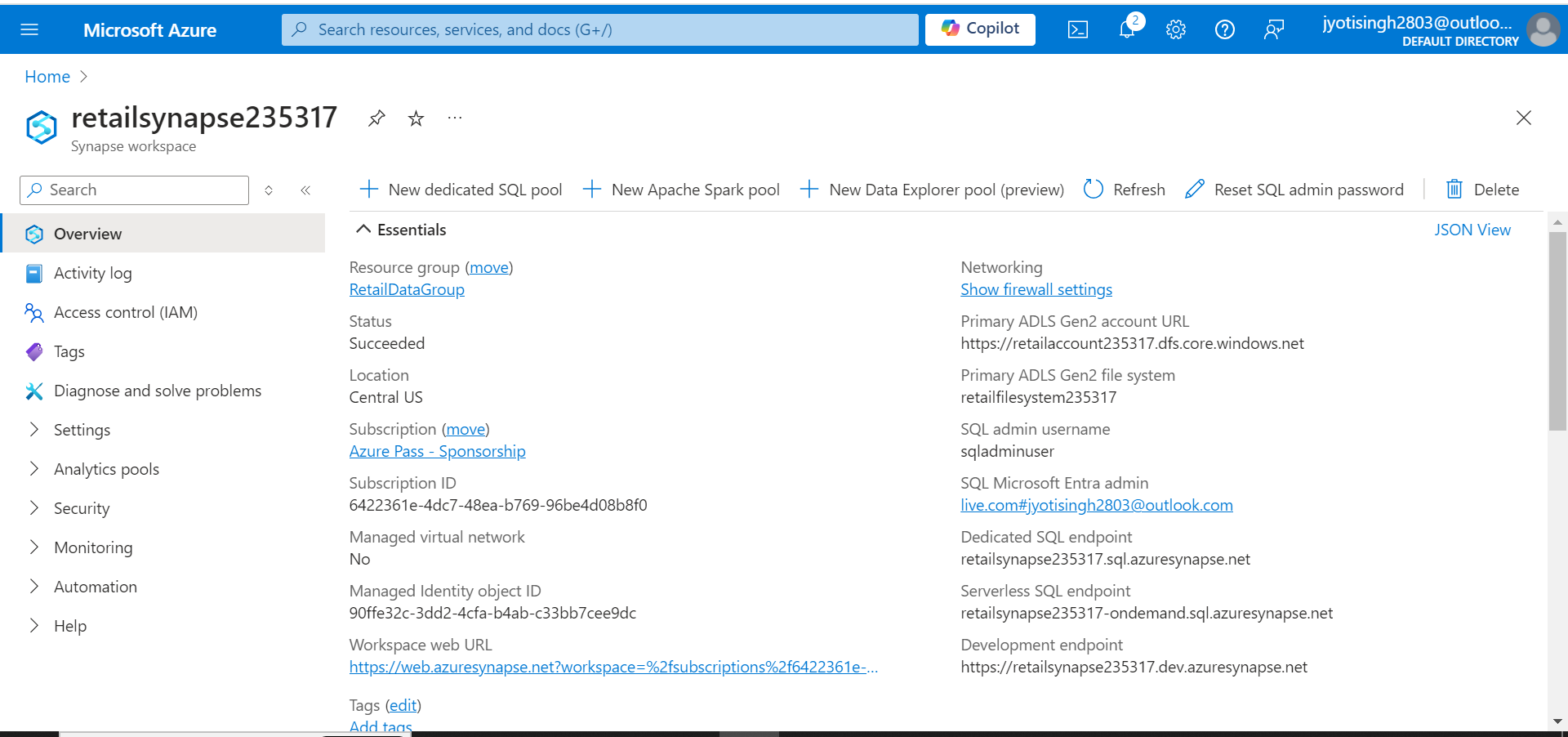




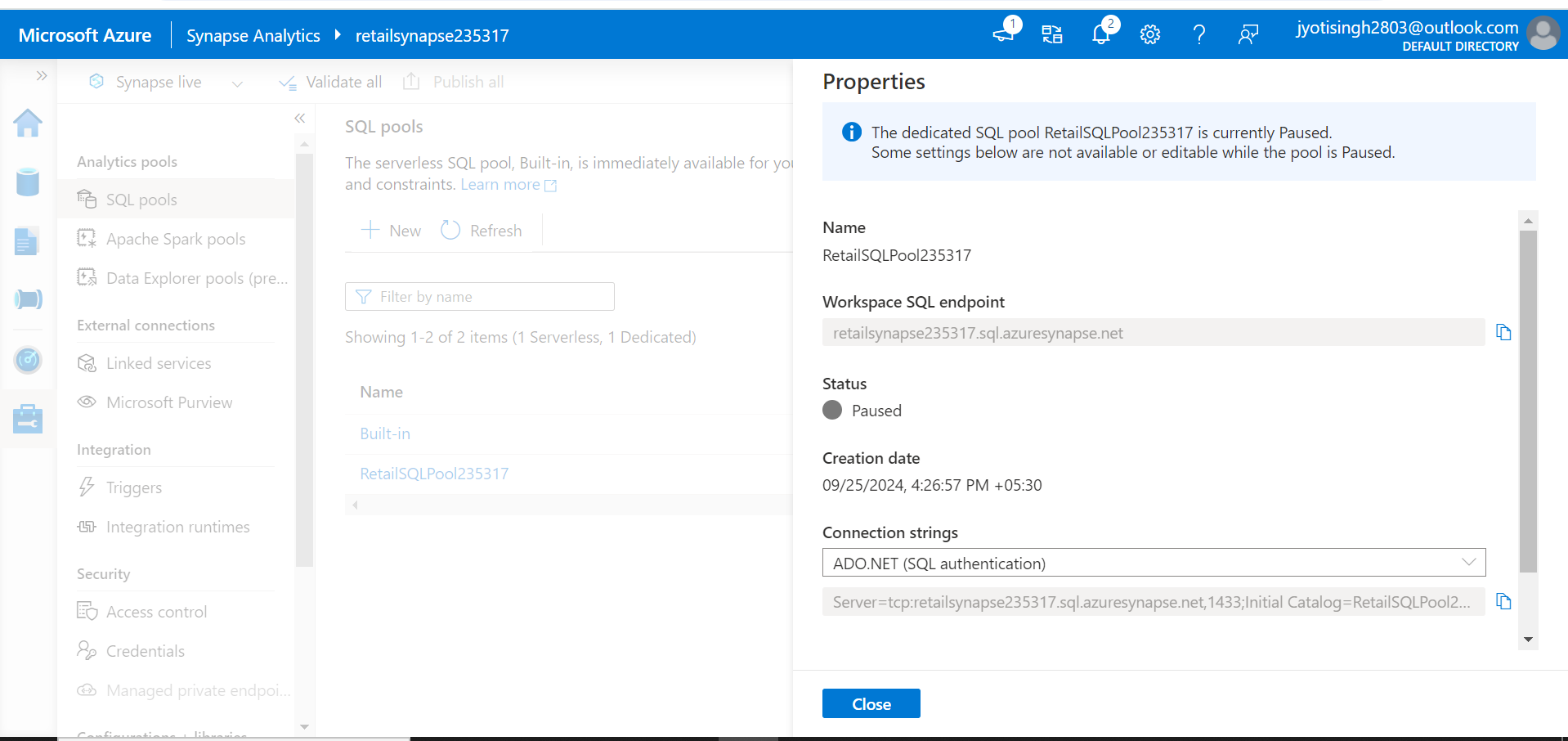


**2. Set Up Azure Synapse Analytics**

1. **Create an Azure Synapse Workspace: done**
   * **Workspace Name:** retailsynapse235317
   * **Resource Group:** RetailDataGroup
   * **Region:** Same as storage account (e.g., East US)

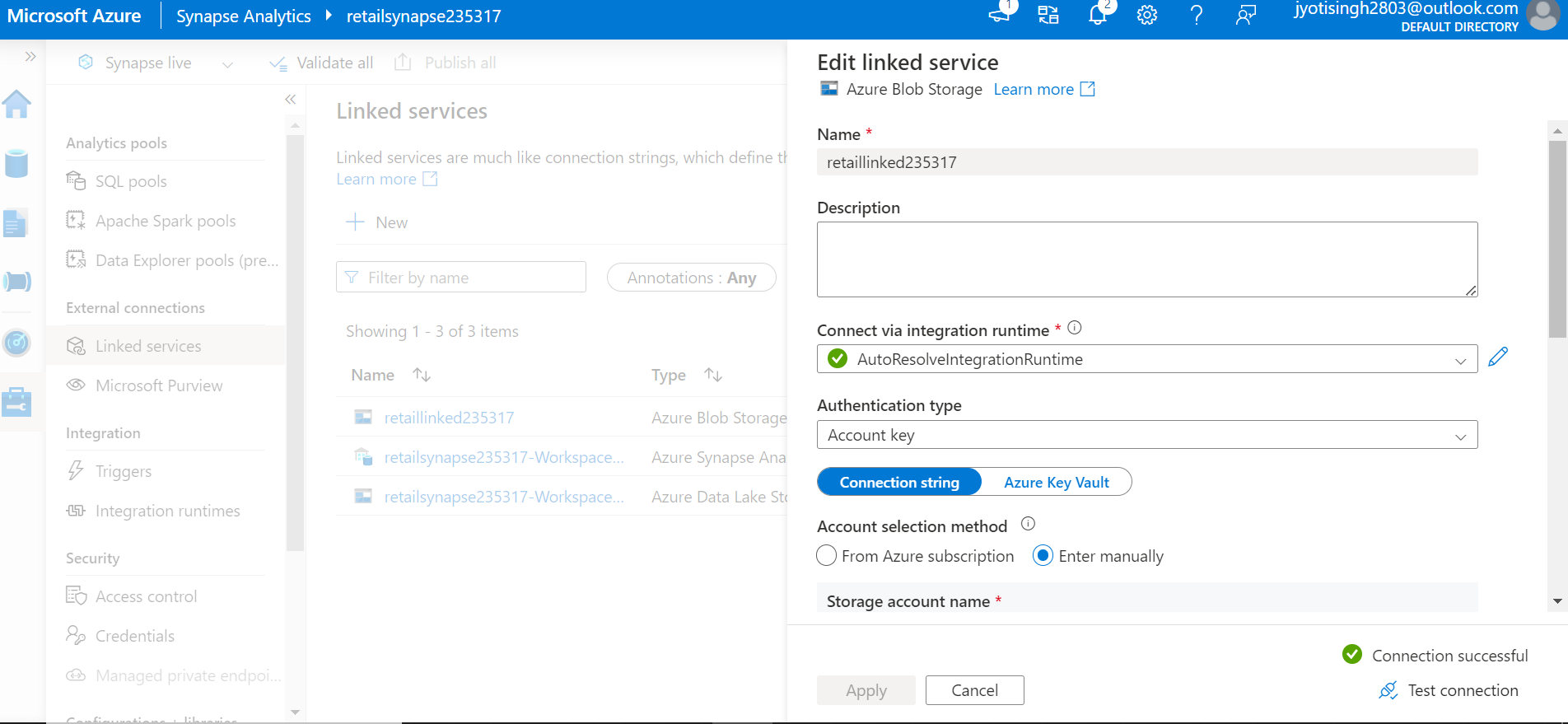


1. **Create a Dedicated SQL Pool: done**
   * **SQL Pool Name:** RetailSQLPool235317
   * Configure a dedicated SQL pool with appropriate performance levels.

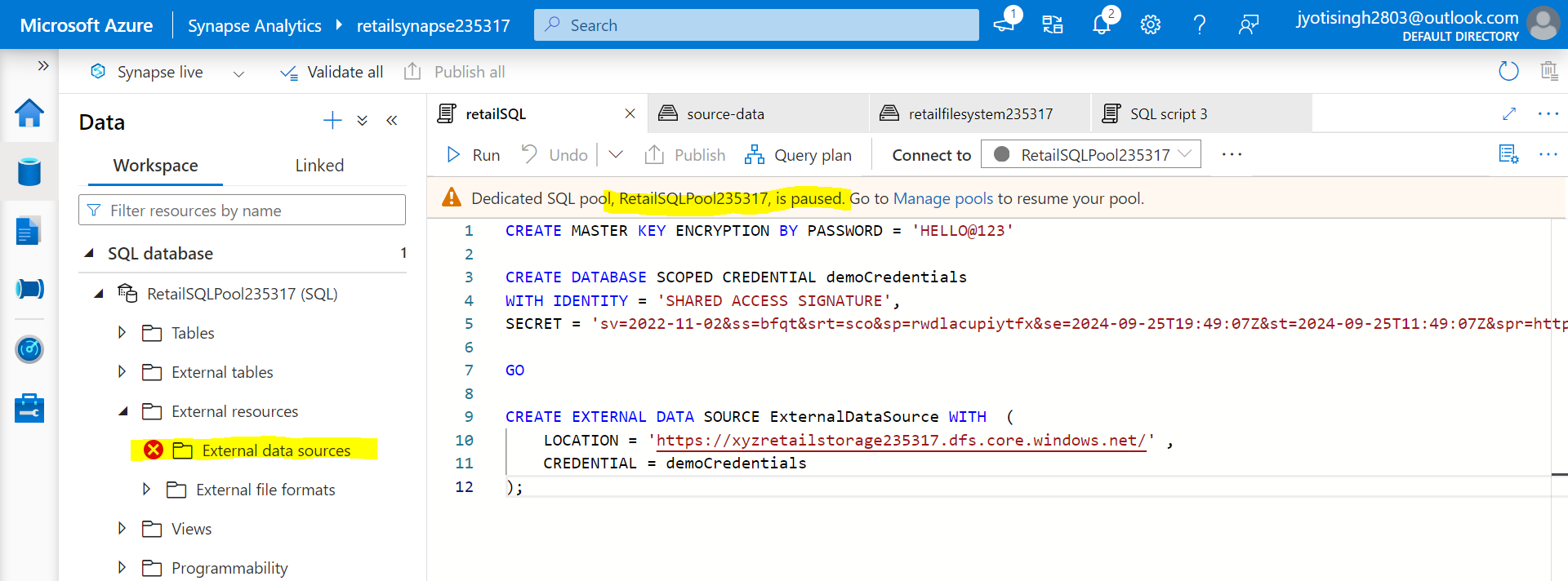


**3. Ingest Data into Azure Synapse Analytics**

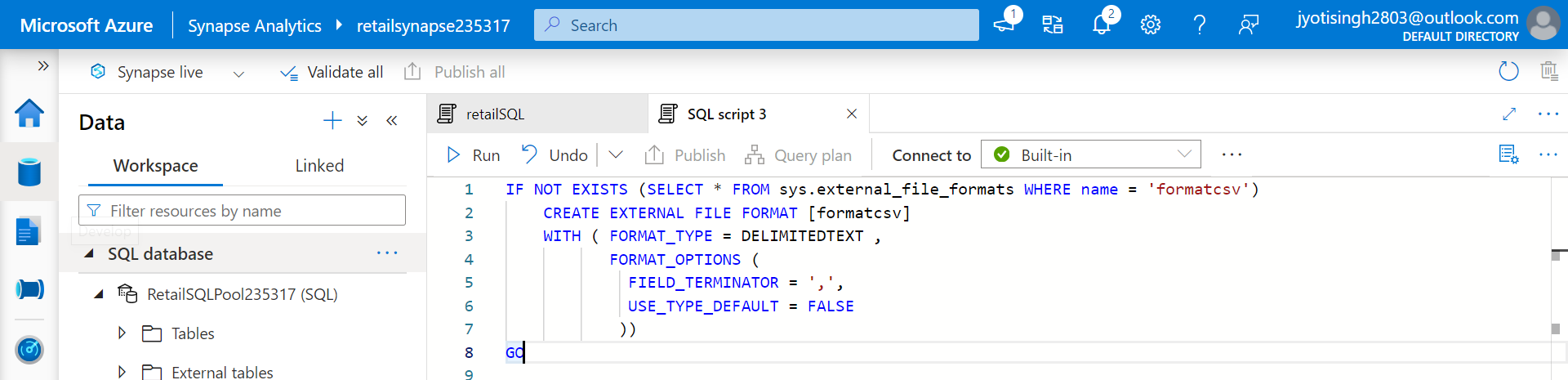
1. **Create Linked Service for Azure Blob Storage: retaillinked235317 done**
   * Go to your Synapse workspace.
   * Navigate to **Manage** > **Linked services**.
   * Click **+ New** and select **Azure Blob Storage**.
   * Configure the connection to the xyzretailstorage account using the storage account connection string.



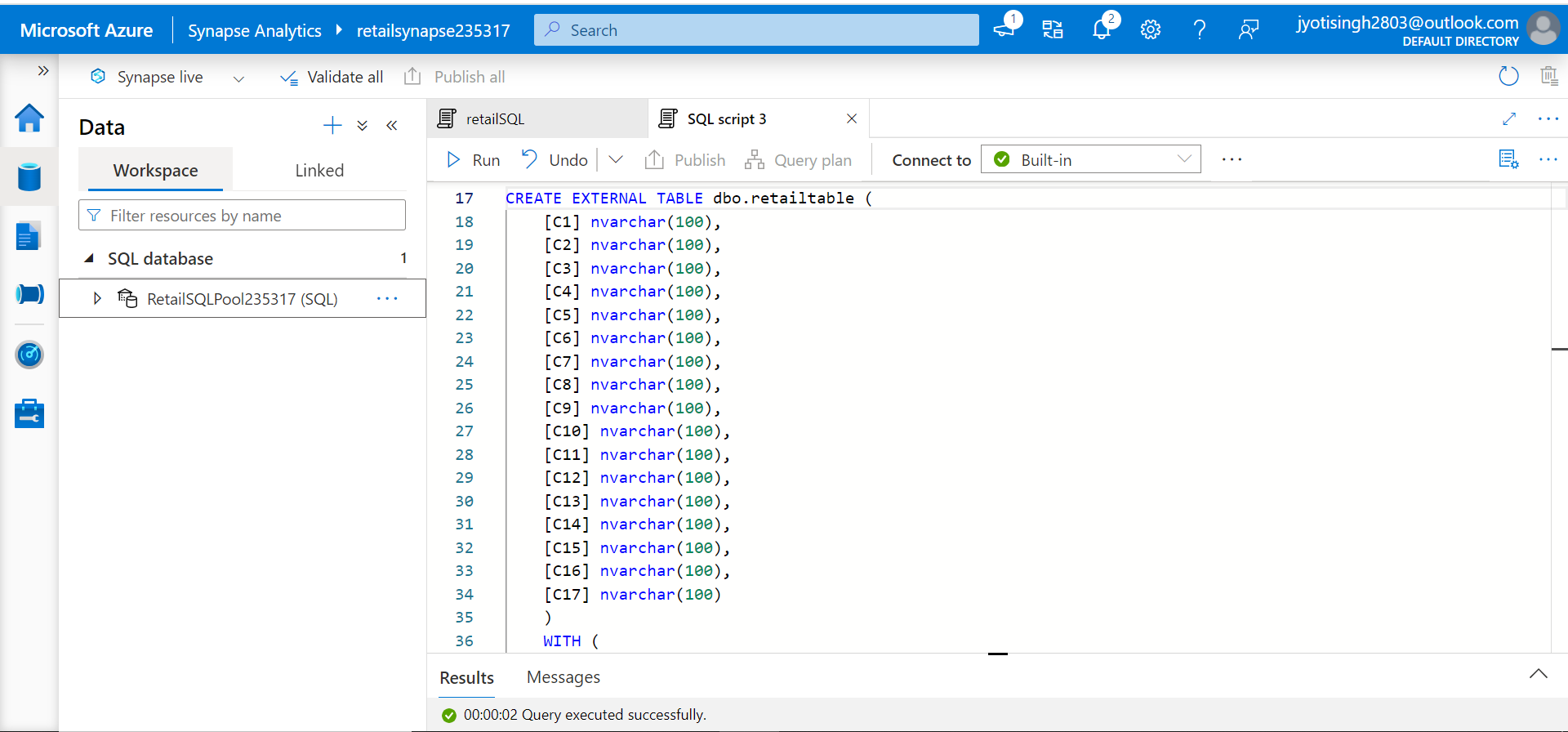
1. **Create External Data Source: DONE**
   * In Synapse Studio, open **Develop** and create a new SQL script.
   * Create an external data source to access data from Azure Blob Storage.



1. **Create External File Format: done**
   * Define the file format for reading CSV files.



1. **Create External Table: done**
   * Define the schema for the external table to query the CSV data.



**4. Transform and Analyze Data**

1. **Create SQL Scripts for Transformation:**
   * Write SQL queries to transform the data, such as aggregating sales by date and store.

