1. **Design Document**

* **Objective:**

The objective of this project is to develop a scalable, efficient, and robust end-to-end data pipeline for XYZ Retail Inc. This pipeline will enable the ingestion, integration, and processing of sales data from diverse sources, including online platforms, physical stores, and mobile applications. The solution aims to facilitate data-driven decision-making by delivering advanced analytics capabilities, improving operational efficiency, and enhancing the overall business strategy.

* **Scope**

The scope of this project encompasses the following:

* Data Ingestion: Capture sales data from multiple channels (online, physical stores, mobile apps).
* Data Integration and Cleaning: Consolidate diverse data formats and ensure data accuracy and consistency.
* Data Storage: Implement scalable and secure storage solutions to accommodate growing data volumes.
* Data Transformation: Design ETL (Extract, Transform, Load) workflows to prepare data for analysis.
* Data Governance: Ensure compliance, security, and quality throughout the data lifecycle.
* Data Visualization: Provide actionable insights through user-friendly dashboards and reports for advanced analytics.
* **Problem Statement**

XYZ Retail Inc. operates in a highly competitive retail landscape, requiring effective data management and analytics to remain agile and customer-focused. The company's operations span various sales channels, including online platforms, physical stores, and mobile applications, resulting in complex and fragmented data silos. These disparate data sources create challenges in deriving cohesive insights and making informed decisions in real-time.

* **Challenges**

XYZ Retail faces the following key challenges:

* Data Fragmentation: Sales data is scattered across multiple channels in varying formats, making it difficult to consolidate and analyse.
* Scalability Issues: Existing systems are unable to handle the increasing volume and velocity of sales data.
* Data Quality and Consistency: Inaccurate, incomplete, and inconsistent data hinders reliable analysis and reporting.
* Integration Complexity: Seamlessly integrating data from diverse platforms and systems is a significant hurdle.
* Real-Time Processing: Limited capability to process and analyse data in real-time affects decision-making agility.
* Governance and Compliance: Ensuring data security, privacy, and adherence to regulatory standards adds complexity.
* Visualization Limitations: Current reporting tools lack the depth and interactivity needed for actionable insights.
* **Proposed Solution:**

The proposed solution involves setting up a robust database system in Azure to manage, transform, and analyze data effectively. The workflow includes creating essential cloud resources such as storage accounts, SQL databases, and Azure Data Factory. These components work together to automate data processing and ensure secure and efficient data handling. The process also incorporates connecting to Azure Data Studio for advanced transformations and query execution.

* **Key Components**

Resource Creation: Initial setup of cloud resources to support data operations, including storage and database infrastructure.

Storage Account: Provides scalable and secure storage for hosting raw and processed data.

SQL Database: Serves as the structured data repository for efficient data management and retrieval.

Azure Data Factory: Facilitates data integration and workflow orchestration through pipelines.

Pipelines: Automates data movement and transformation processes, connecting various stages of the workflow.

Pipeline Linking: Ensures seamless integration between pipelines to maintain data flow consistency.

Azure Data Studio Connection: Enables detailed data transformation and analysis capabilities for the stored data.

Firewall and Networking Configuration: Ensures secure access to resources by implementing appropriate networking rules.

* **Data Storage Choice**

**Options Considered:**

Azure Data Lake Storage (ADLS)

Azure Data Lake Storage is a scalable data storage solution designed for big data analytics. It provides a single, centralized repository for storing all types of data, both structured and unstructured1. Key features include:

* Hadoop-compatible access: Works seamlessly with Hadoop and other frameworks using the Apache Hadoop Distributed File System (HDFS).
* Hierarchical directory structure: Organizes data in a hierarchical manner for easier management.
* Optimized cost and performance: Offers low-cost storage with high throughput.
* Finer grain security model: Provides advanced security features for data access and encryption.
* Massive scalability: Can handle petabytes of data and sustain high throughput.

Azure SQL Database

Azure SQL Database is a fully managed relational database service provided by Microsoft. It offers several advantages, especially when compared to ADLS:

* High availability: Guarantees 99.99% availability with built-in disaster recovery options.
* Automatic backups and data protection: Manages backups and data integrity checks automatically.
* Automatic tuning: Optimizes database performance without manual intervention.
* Scalability: Easily scales up or down based on usage, providing flexibility during peak times.
* Cost-effective: Uses a pay-as-you-go model, reducing costs.
* Integration with other Microsoft services: Seamlessly integrates with other Azure services and Microsoft tools.

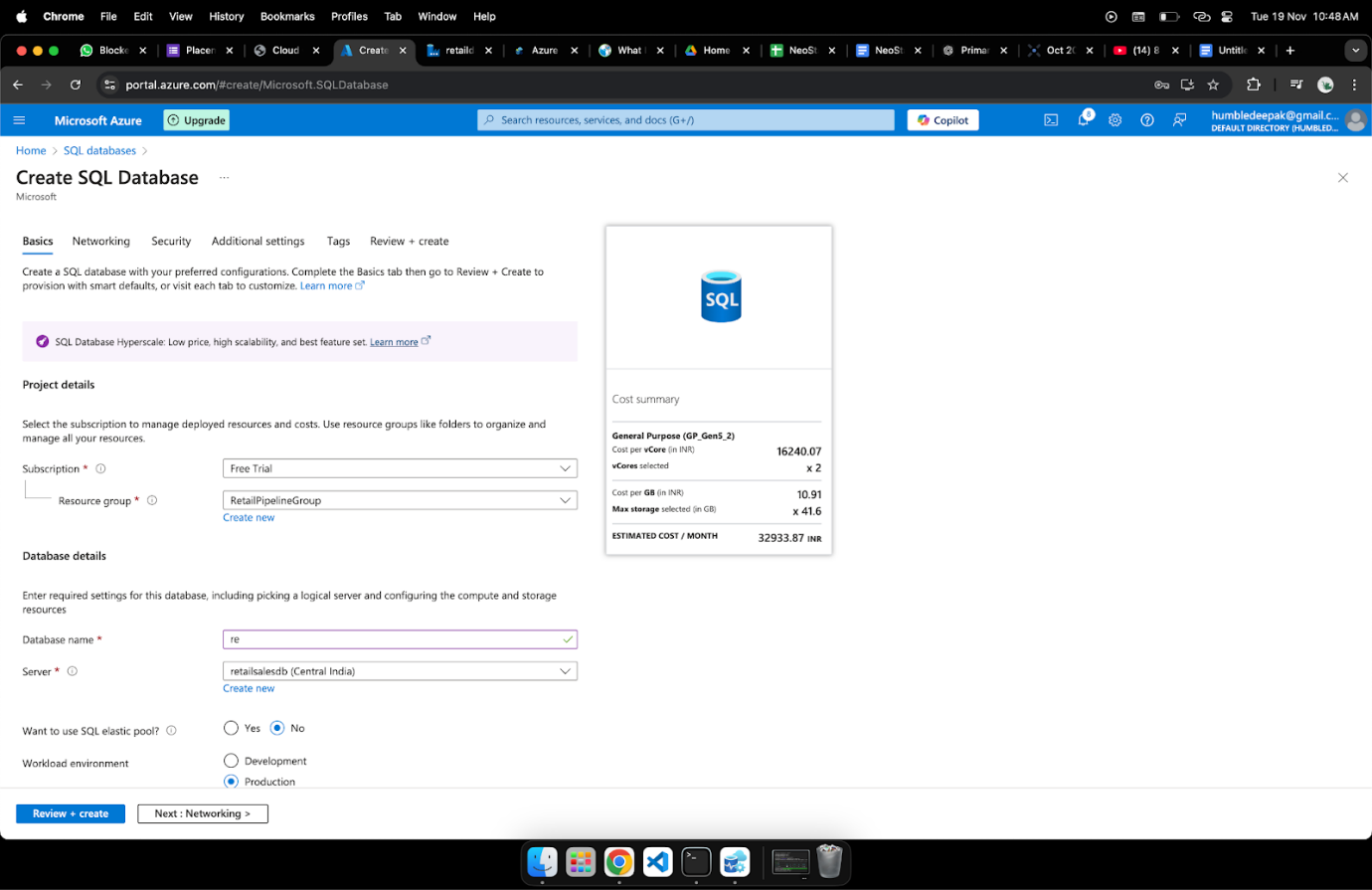
**Chosen Solution:**

I have chosen Azure SQL database, because of its

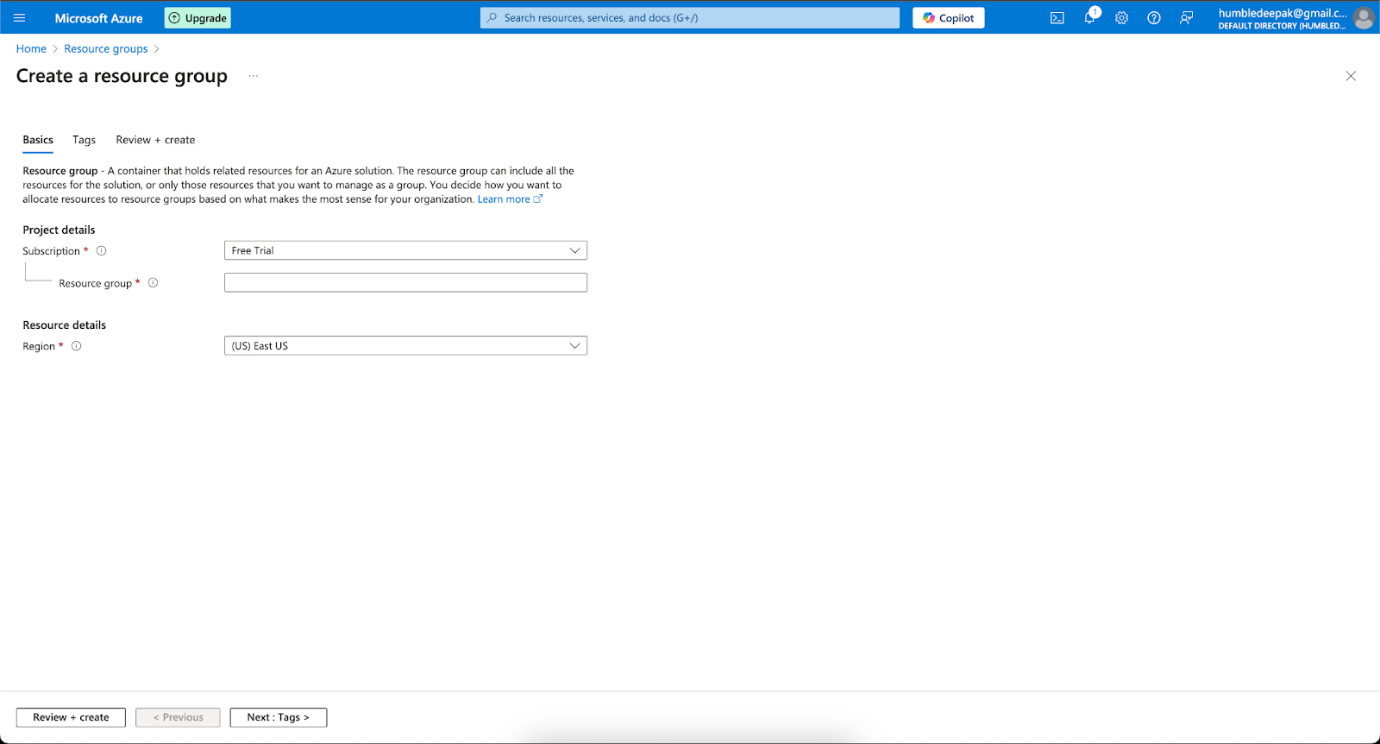
* Structured Data Management: Azure SQL Database is ideal for managing structured data, whereas ADLS is better suited for unstructured data2.
* Database Management Features: Azure SQL Database offers advanced database management features like automatic tuning, backups, and high availability, which are not available in ADLS.
* Ease of Use: Azure SQL Database provides a more user-friendly interface for SQL developers and database administrators.
* Performance Optimization: Automatic tuning and performance optimization features in Azure SQL Database help maintain optimal performance without manual intervention.

Here is a step-by-step process on what approach I used and how I did it:

**Step 1 – Database Creation:**

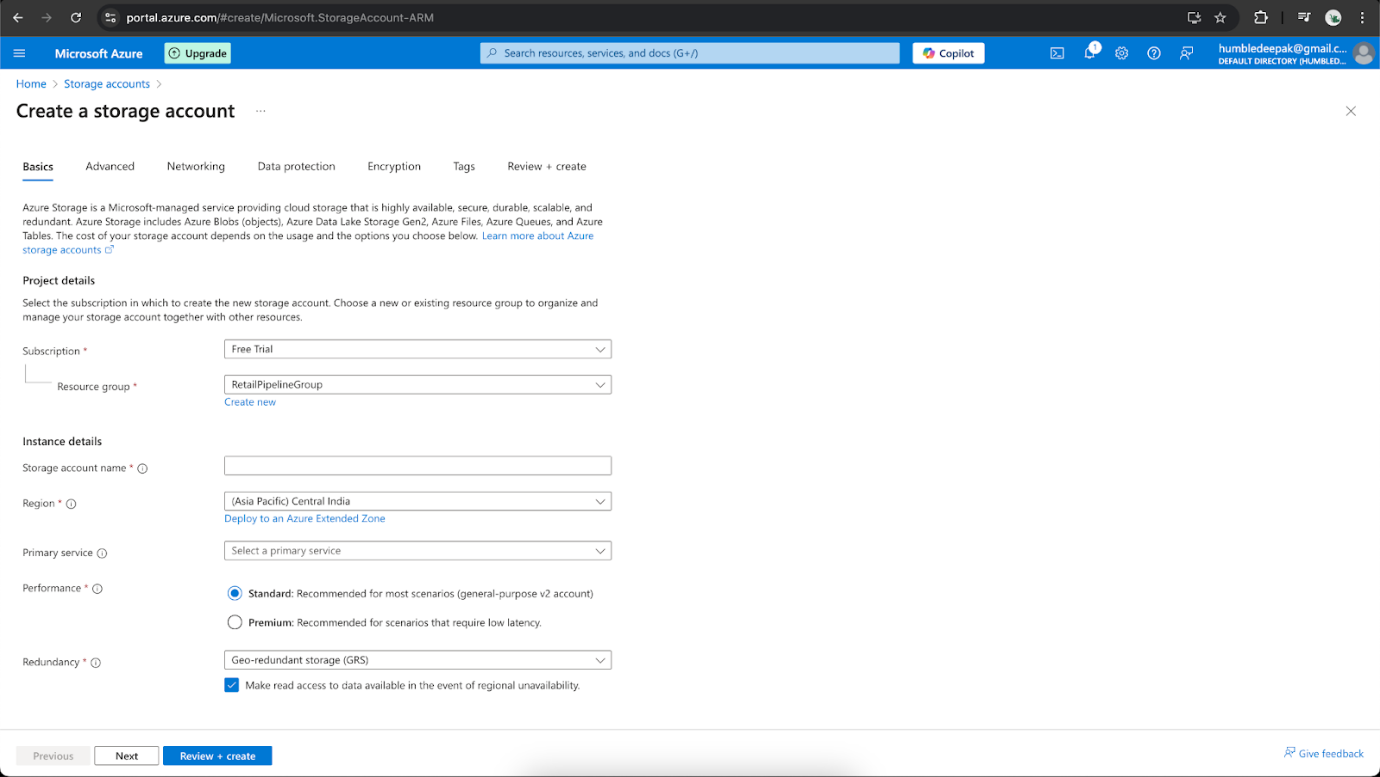


**Step 2 – Resource Creation**



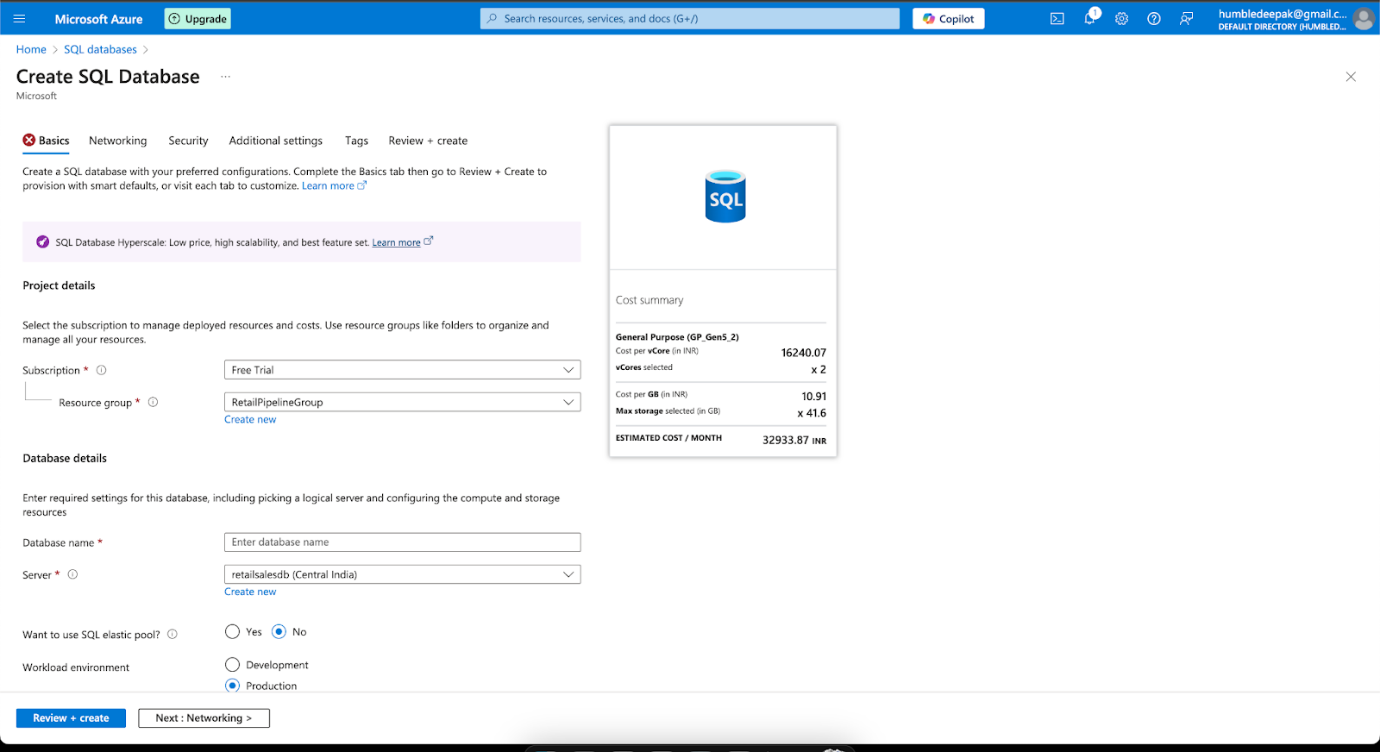
The initial step demonstrates setting up the necessary resources for the project, laying the foundation for database and related infrastructure.

**Step 3- Storage Account:**



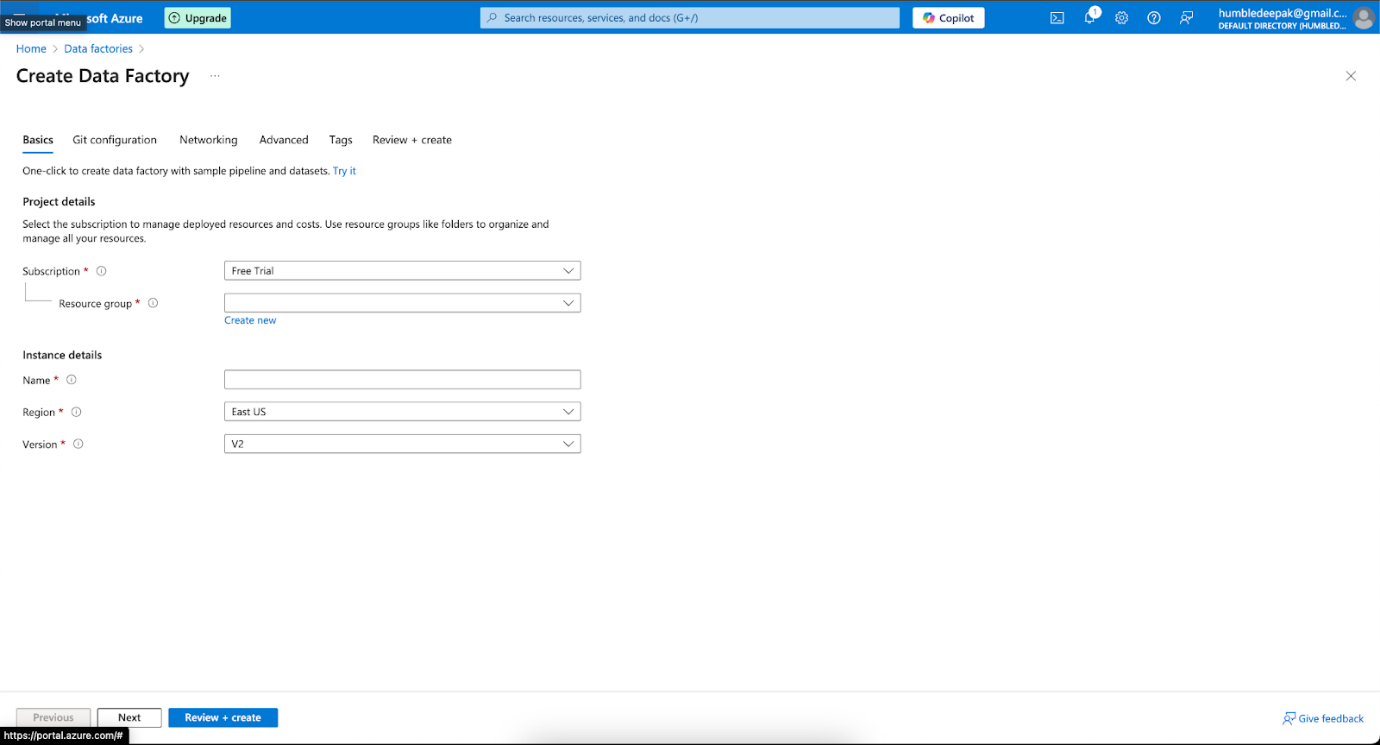
Highlights the creation of a storage account in Azure, essential for hosting and managing data in the cloud.

Step 4- **SQL Database**:



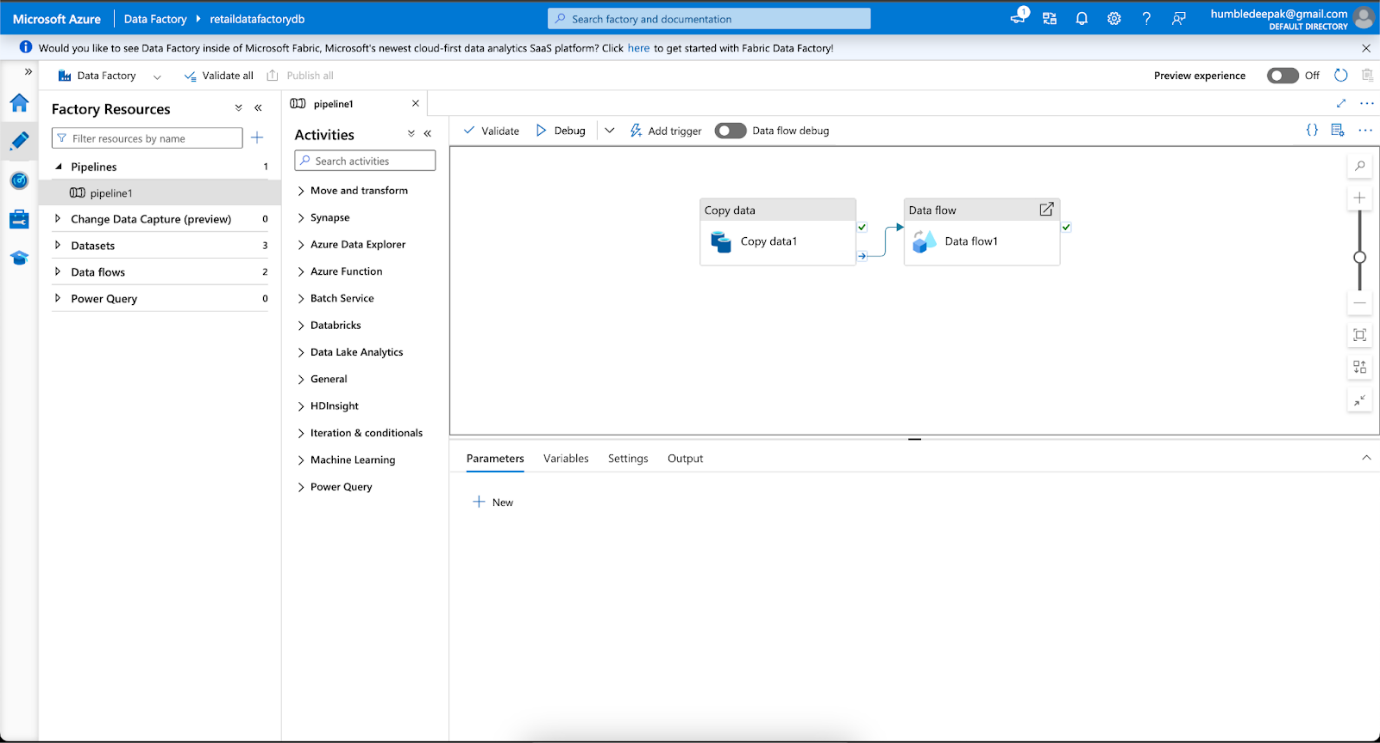
Showcases the configuration and deployment of an SQL database to store and retrieve structured data efficiently.

Step 5- **Create Data Factory**:



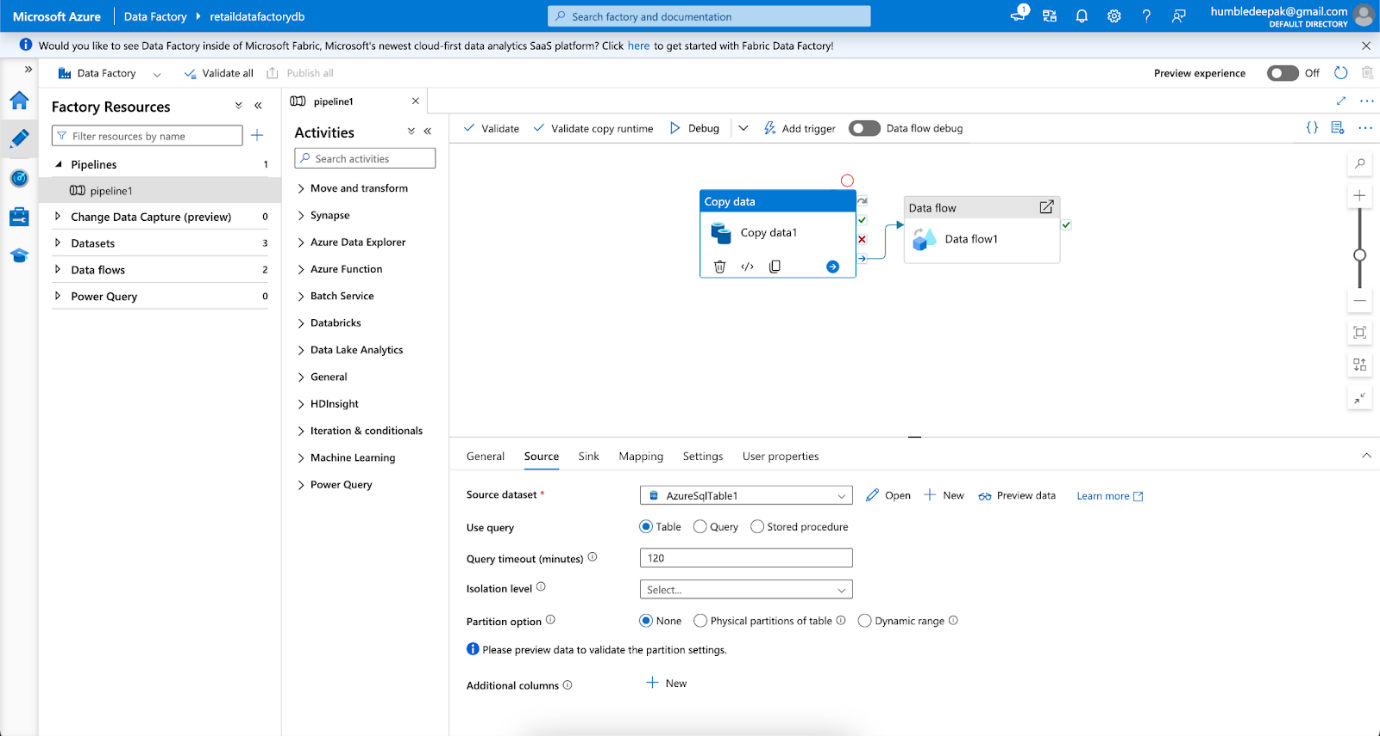
Illustrates the setup of an Azure Data Factory, a cloud service for integrating and transforming data from various sources.

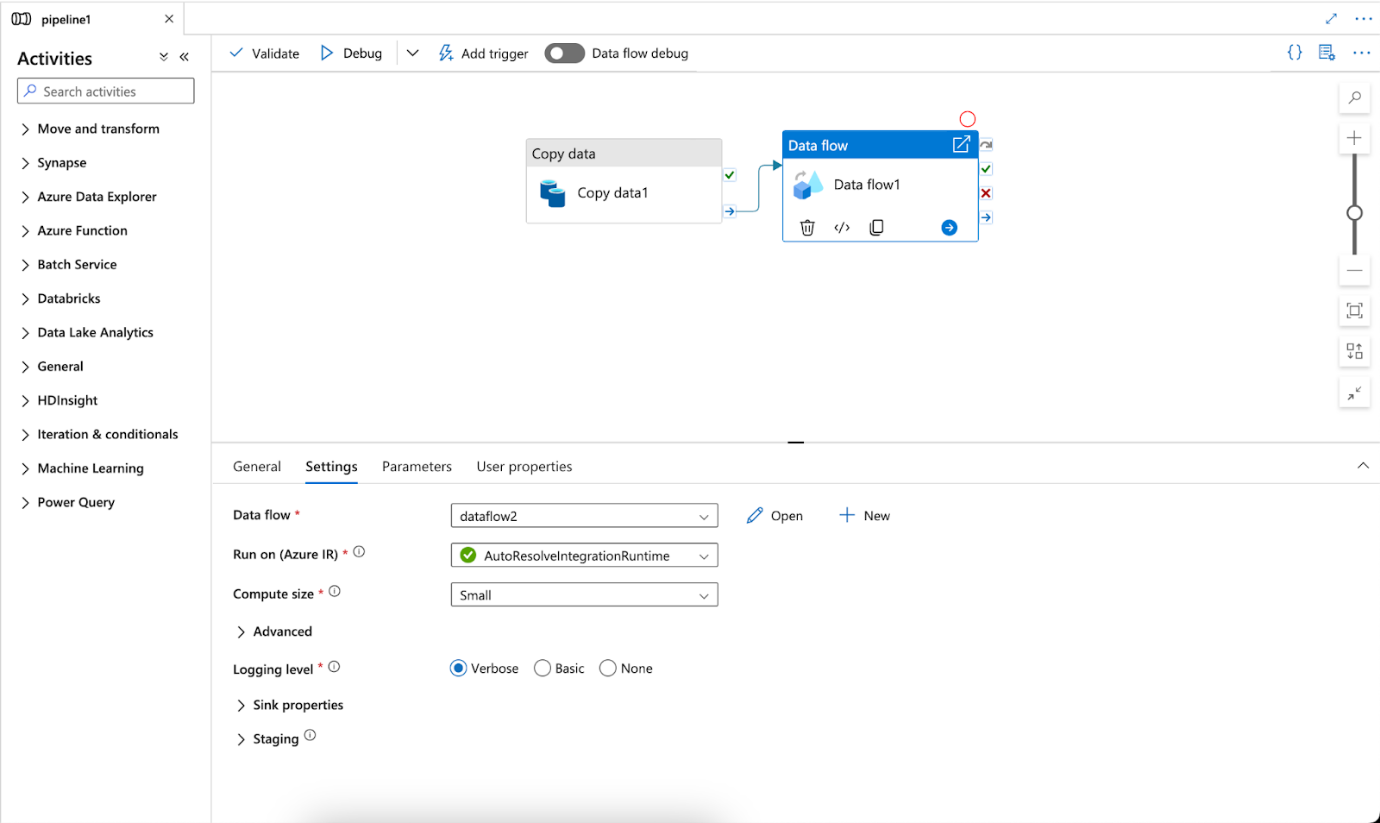
**Step 6- Pipelines**:



Represents the creation of data pipelines within the Data Factory to orchestrate workflows and automate data processes.

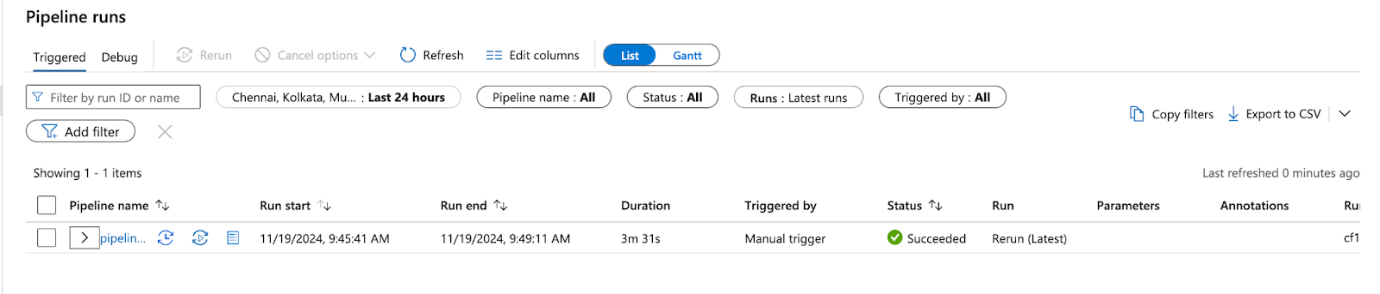
**Step 7- Linking Pipelines**:



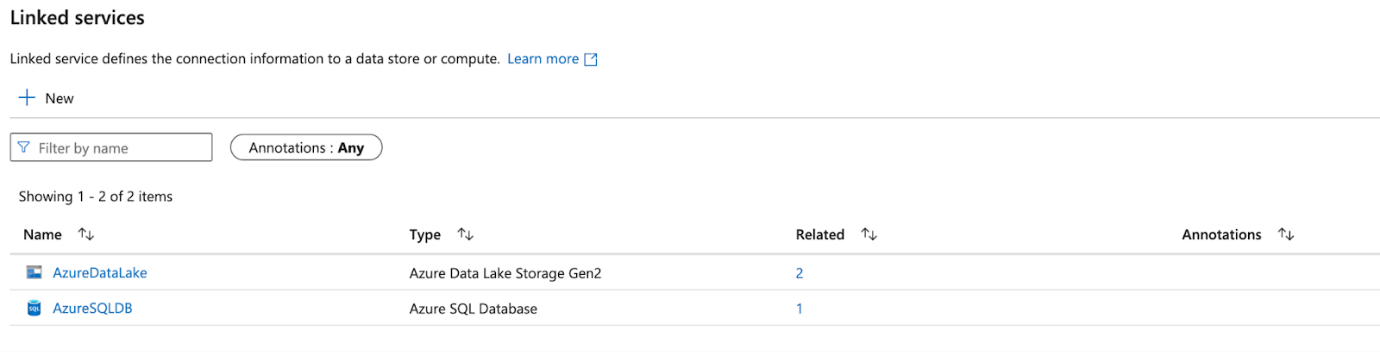


Depicts the connection between multiple pipelines, ensuring cohesive data flow and process integration.

**Step 8- Trigger the Pipelines (Ensure Success)**:

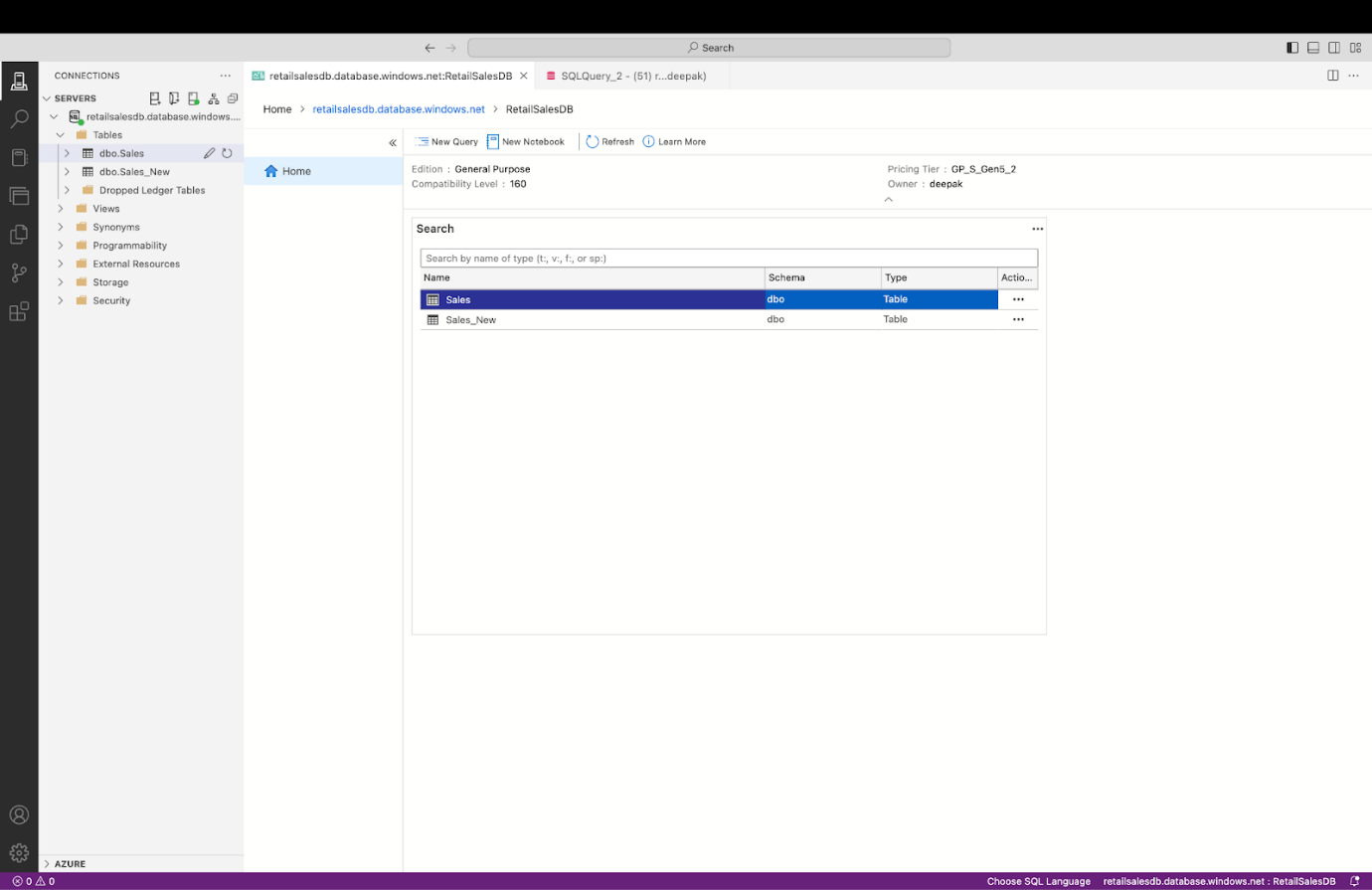


Link the pipelines



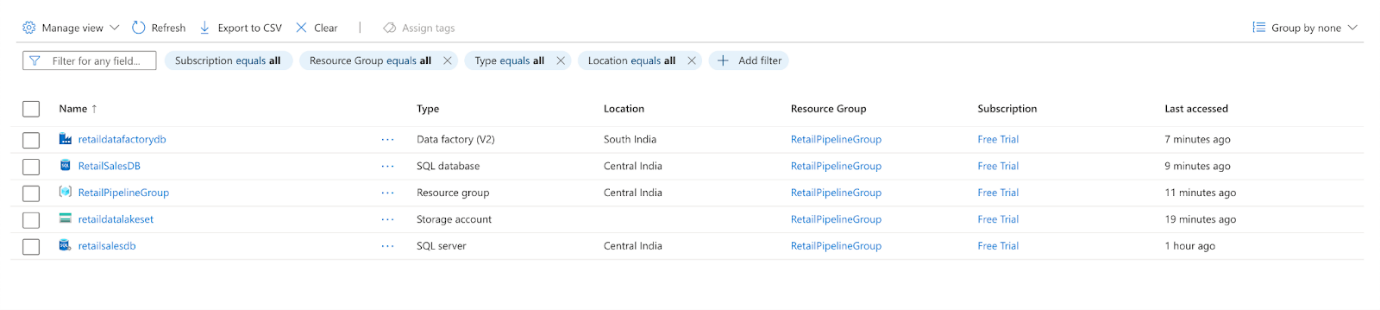
Demonstrates the execution of the pipelines and monitoring for successful operation to validate workflow functionality.

**Step 9- Connect to Azure Data Studio**:



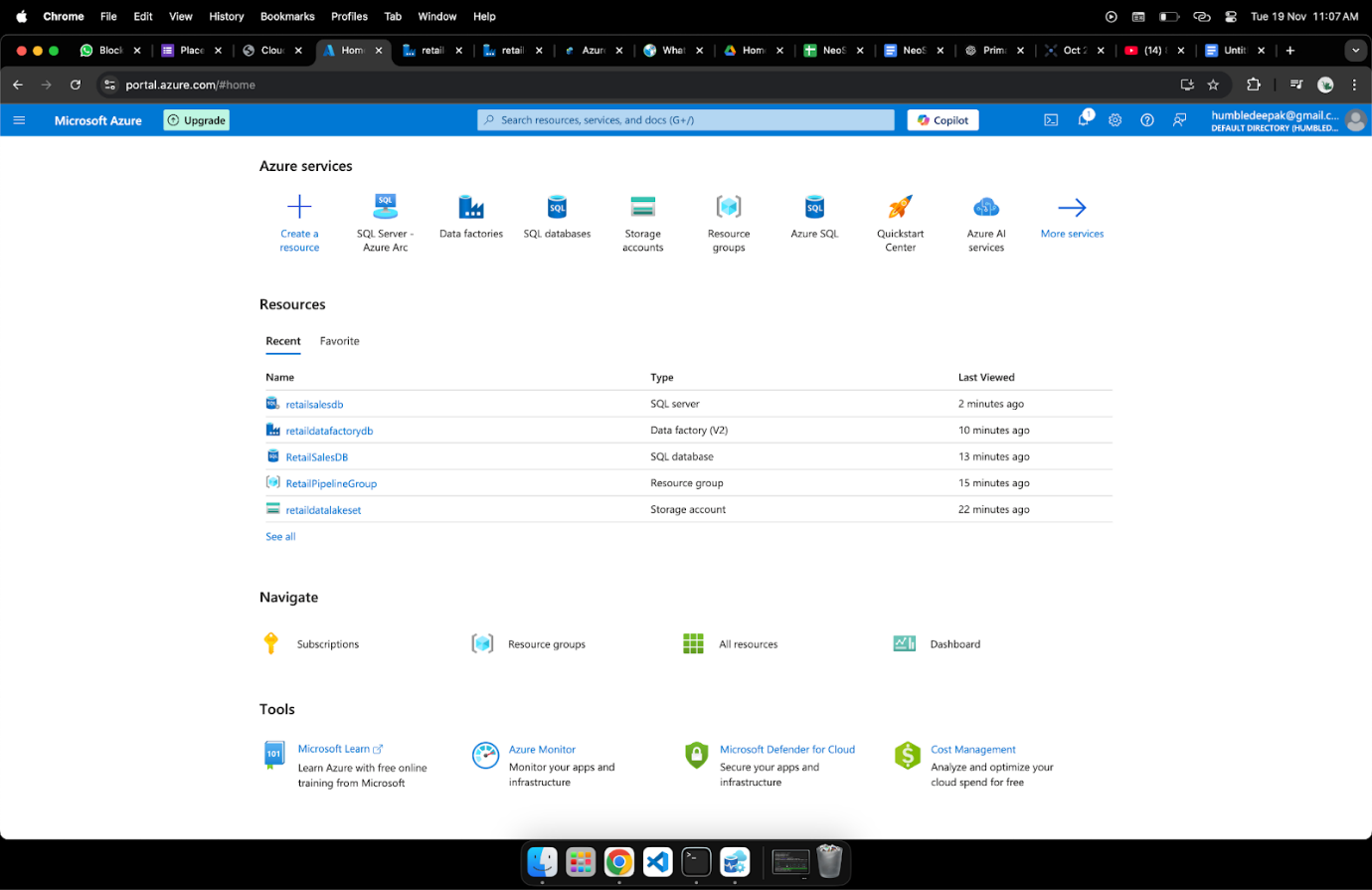
Highlights the connection to Azure Data Studio for performing advanced data transformations and queries.

**Step 10- All Resources in One Place**:

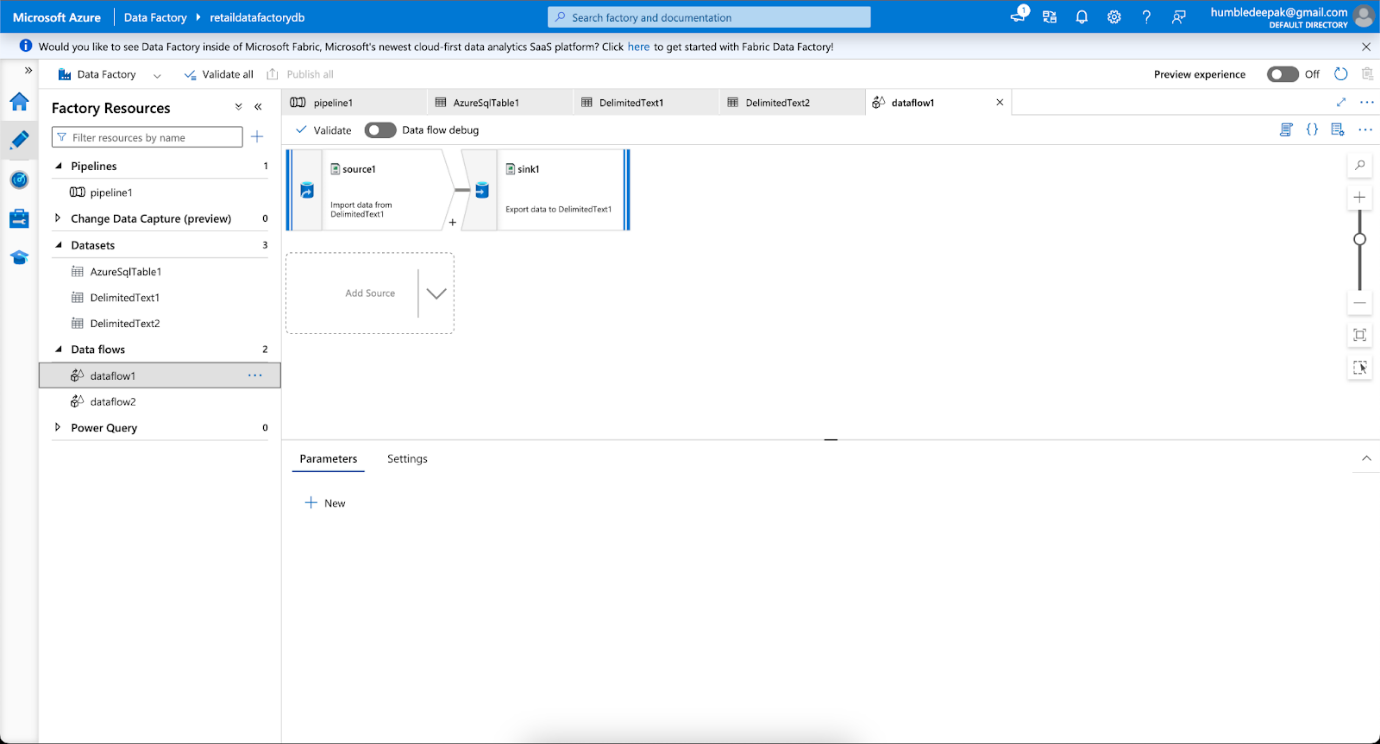


Summarizes the consolidated view of all created resources within the Azure portal for easy management.

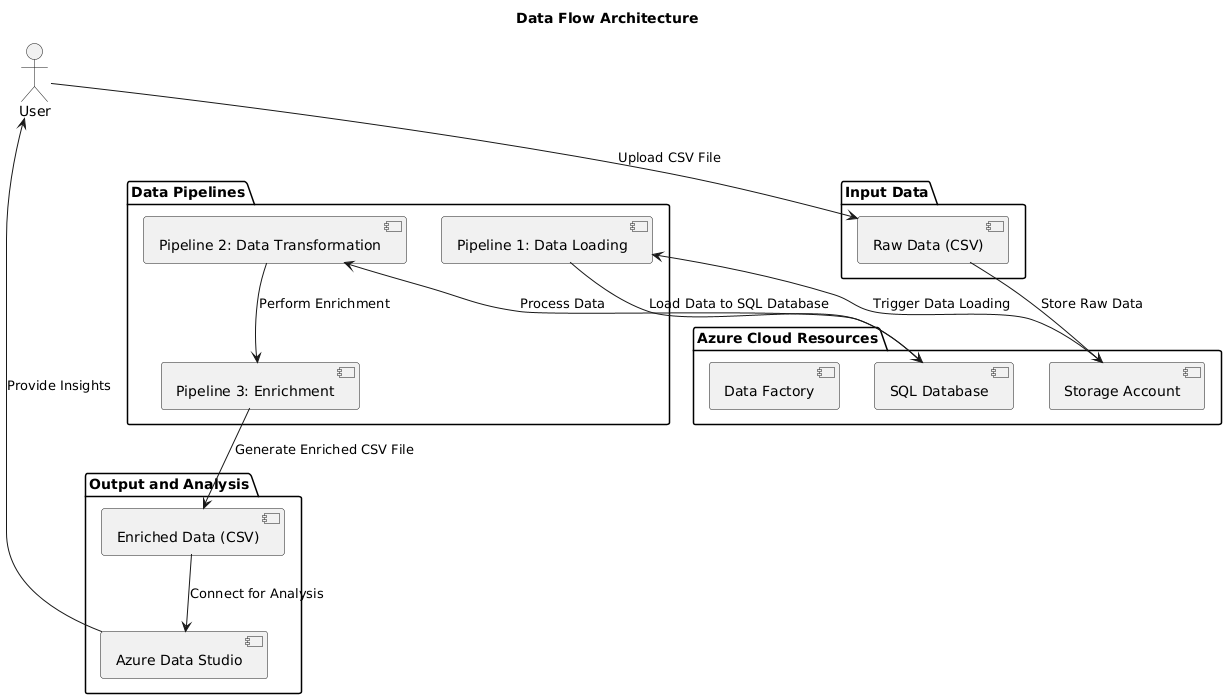
Overall Dashboard:



Source Linking:



* **Architecture Diagram:**

****

**Explanation:**

**1. User Interaction**

User: The starting point of the workflow. The user uploads a raw data file (CSV) to initiate the process.

**2. Input Data**

Raw Data (CSV): Represents the input dataset uploaded by the user. This serves as the raw data source for all subsequent transformations.

**3. Azure Cloud Resources**

**Storage Account:**

Stores the uploaded raw data securely in the cloud.

Acts as the central repository for the input data.

**SQL Database:**

Receives the raw data from the storage account via the pipeline.

Used to store, organize, and process structured data for further operations.

Data Factory:

Orchestrates the data flow and triggers the execution of pipelines.

Serves as the backbone for managing the data pipeline processes.

**4. Data Pipelines**

Pipeline 1: Data Loading:

Transfers raw data from the Storage Account into the SQL Database for structured processing.

Ensures data is loaded accurately for downstream operations.

Pipeline 2: Data Transformation:

Processes the data within the SQL database to perform transformations, such as computing metrics (e.g., total sales or average order value).

Prepares the data for enrichment and final output.

Pipeline 3: Enrichment:

Adds additional fields or metrics (e.g., customer details, calculated totals) to the data.

Ensures the final dataset is enriched and ready for analysis.

**5. Output and Analysis**

Enriched Data (CSV):

Represents the final output of the data pipelines, containing all processed and enriched data.

This is stored as an enriched CSV file for further use.

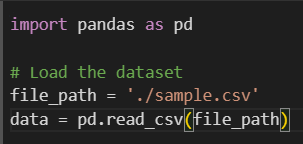
Azure Data Studio:

Connects to the enriched dataset for advanced analysis and visualization.

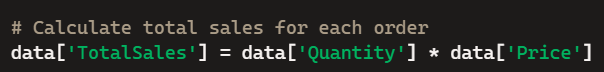
Provides the user with actionable insights derived from the processed data.

1. **Code Implementation:**

**Transformation steps:**

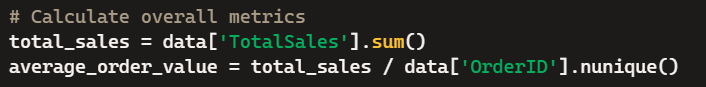
****

Load the Dataset: Read the sales data from a CSV file using pandas.

****

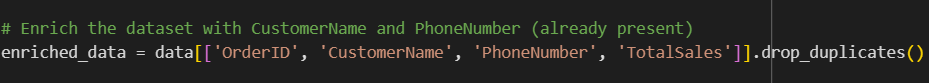
Calculate Total Sales: Compute the total sales for each order by multiplying the quantity of items sold by their price.

**Metrics Calculation:**

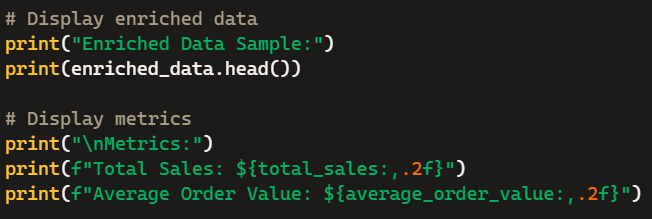
****

* Total Sales: Sum up the total sales across all orders.
* Average Order Value: Calculate the average value of an order by dividing the total sales by the number of unique orders.

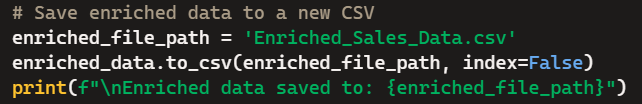
**Enrichment:**

****

Enrich the Dataset: Add additional fields like CustomerName and PhoneNumber to the dataset. Ensure there are no duplicate records.

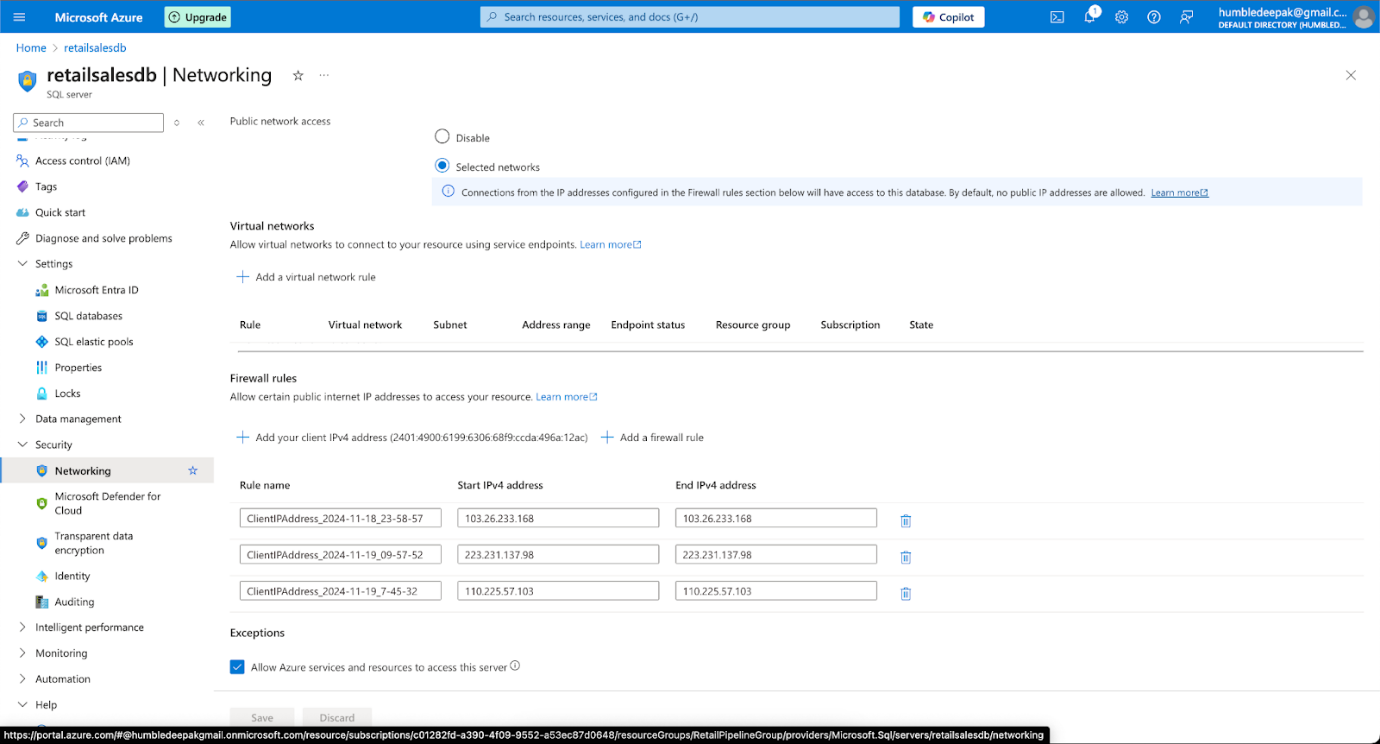
****

Display Data and Metrics: Show a sample of the enriched data and the calculated metrics for verification.

****

Save Enriched Data: Store the enriched dataset in a new CSV file for future use.

**Handle PII data securely:**



Indicates the configuration of network firewalls, ensuring secure access to the deployed resources.

1. **Dataset Documentation**

**Structure of the Sales Dataset**

**The sales dataset contains 99 records with the following columns and their data types:**

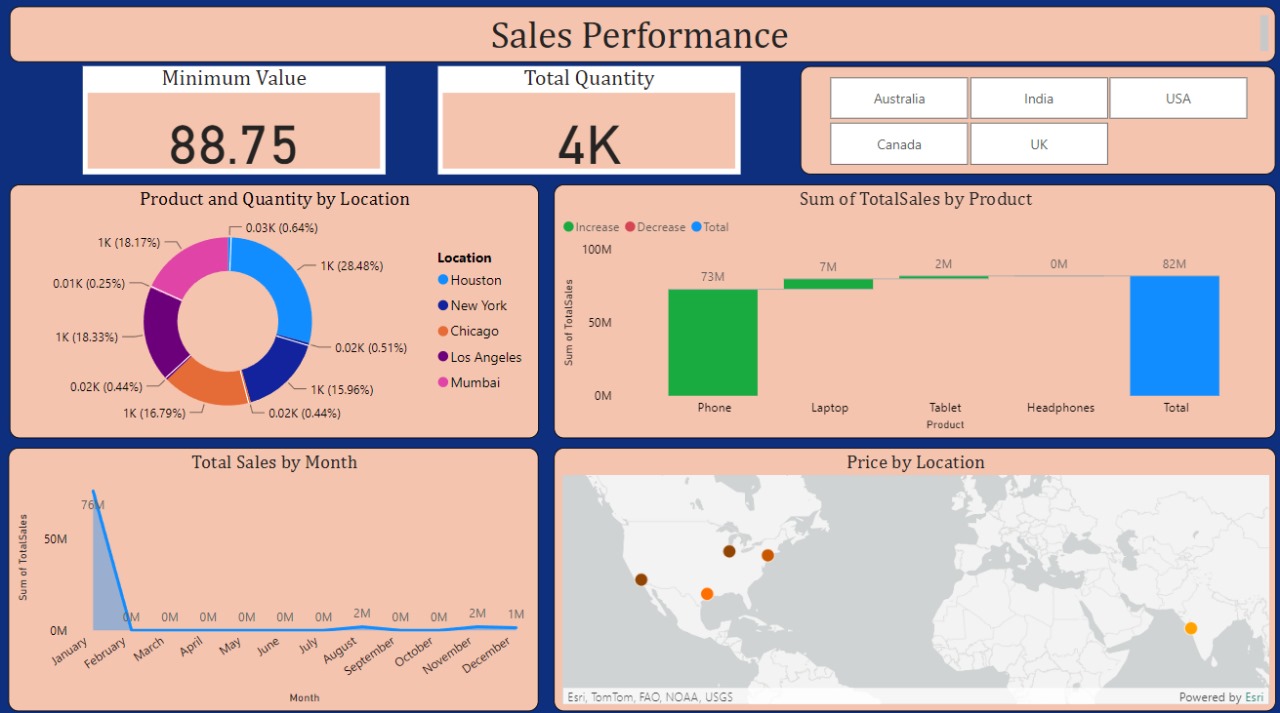
* OrderID (object): Unique identifier for each order.
* CustomerName (object): Name of the customer placing the order.
* PhoneNumber (object): Contact number of the customer.
* Location (object): City where the order was placed.
* Country (object): Country of the order placement.
* StoreCode (object): Code representing the store fulfilling the order.
* Product (object): Type of product purchased.
* Quantity (int64): Number of items purchased in the order.
* Price (float64): Unit price of the product. Some values are missing.
* Date (object): Date of the order in DD-MM-YYYY format.
* CreditCardNumber (object): Customer's credit card number (masked for security).
* ExpiryDate (object): Credit card expiry date in MMM-YY format.

**Data Preprocessing Steps**

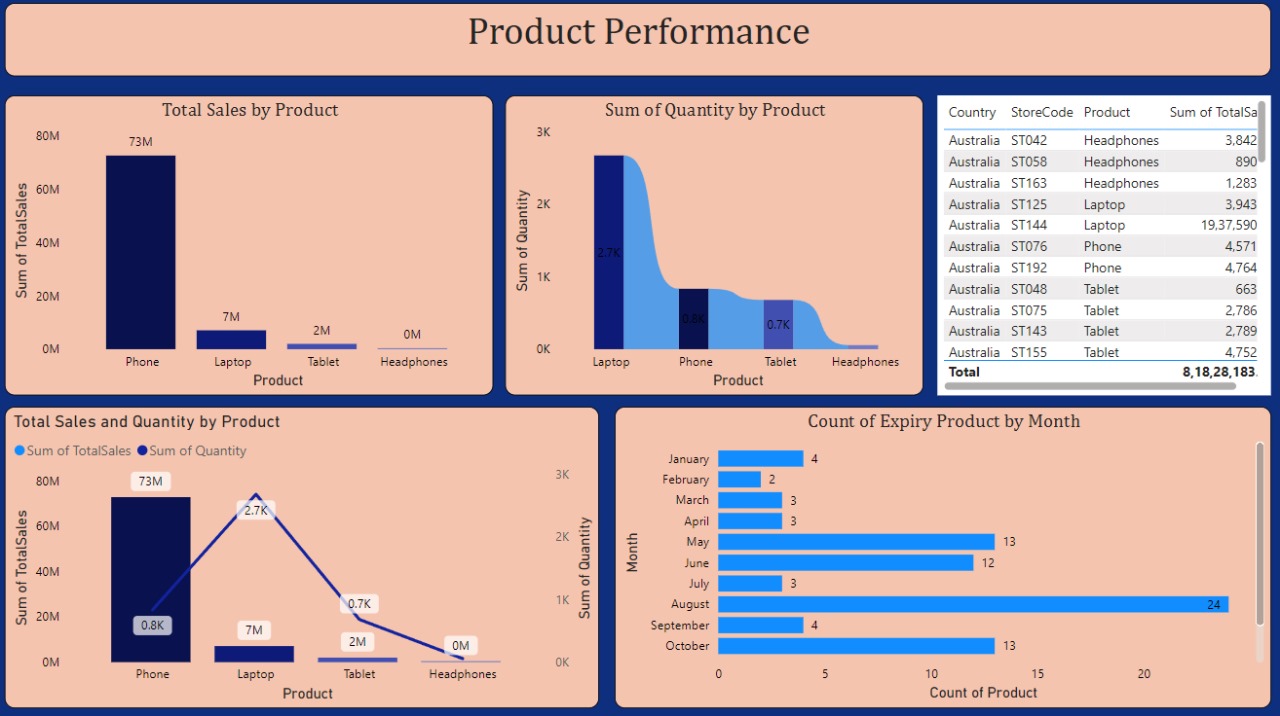
* Data Loading: The dataset was loaded using the pandas library for analysis.
* Handling Missing Data: Observed missing values in the Price column. These need to be handled through imputation or filtering.
* New Column Creation:
* TotalSales: Calculated as Quantity \* Price to represent total revenue per order.
* Data Validation: Ensured numeric columns (e.g., Quantity, Price) were correctly interpreted and verified non-null values for critical fields like OrderID and Product.
* Data Saving: Enriched data (with new columns) was saved into a new CSV file for further processing.

1. **Power BI Dashboard**

**Power BI report:**



The dashboard provides a comprehensive analysis of sales performance across locations and products. Key highlights include total sales distribution by product category (Phones dominating with 73M) and a breakdown of product quantity by location, showing Houston and New York as major contributors. Additional insights include monthly sales trends, location-wise pricing details on a map, and overall metrics like minimum value (88.75) and total quantity (4K).



The Product Performance Dashboard provides insights into sales and quantities of products:

* Total Sales by Product: Phones dominate sales with 73M, while laptops and tablets contribute 7M and 2M, respectively. Headphones have negligible sales.
* Sum of Quantity by Product: Laptops lead in quantity sold (2.7K units), followed by phones (0.8K units). Tablets and headphones contribute much less to the total quantity.
* Expiry Product Count by Month: August sees the highest count of expired products (24), followed by May (13) and June (12), indicating potential seasonal patterns.
* Detailed Data Table: The table highlights specific store codes, products, and their corresponding total sales in Australia, summing up to a total of 8,18,28,183.
* Overall Summary: The dashboard effectively identifies top-performing products, low contributors, and temporal patterns in product expiry.