**Hexaware Data Engineering Project**

**Student Name:** Mahesh Ratneshwar Godale

Abhilash Gowda S

**Trainer:** Harinya Ma'am

**Title:** Global Health Data Analysis using Azure

**Problem Statement:**

Implement a serverless data processing pipeline where Azure Data Factory orchestrates data workflows, and Azure Databricks is used as a serverless processing engine for on-demand analytics and transformations.

**Project Overview**

The Serverless Data Processing Pipeline project focuses on implementing an automated, scalable solution to process and analyze global health data. The dataset contains statistics on various diseases, their prevalence, mortality rates, healthcare access, and treatment outcomes across multiple countries and years. This solution utilizes Azure Data Factory for orchestrating data workflows and Azure Databricks as a serverless engine for processing and transforming data. The primary aim is to extract valuable insights and trends from health data to support global health research and decision-making.

The pipeline's core functionalities include the automation of data ingestion, transformation, and analysis, enabling stakeholders to gain a better understanding of healthcare dynamics and disease patterns worldwide. The solution also provides an extensible framework for integrating with other data visualization tools like Power BI for further analysis and reporting.

**Objective**

The objective of this project is to implement a **Serverless Data Processing Pipeline** using **Azure Data Factory (ADF)** and **Azure Databricks**, with data stored in **Azure Data Lake**. The pipeline is designed to:

1. **Ingest Data**: Securely store and organize the input dataset (**Global Health Statistics.csv**) in **Azure Data Lake Storage (ADLS)** for easy access and processing.
2. **Process and Transform Data**: Use **Azure Databricks** to perform data cleaning, transformation, and enrichment, ensuring the data is ready for analysis.
3. **Analyze Data**: Extract meaningful insights from the processed data by performing analytical tasks such as identifying global health trends, comparing healthcare access, and evaluating disease management.
4. **Visualize Data**: Generate insightful visualizations to highlight key trends and metrics, such as mortality rates, disease prevalence, and treatment effectiveness.
5. **Orchestrate with Azure Data Factory**: Create a pipeline in **Azure Data Factory** that orchestrates the execution of the **Azure Databricks** notebooks, automating data processing and analysis tasks.

**Data Overview**

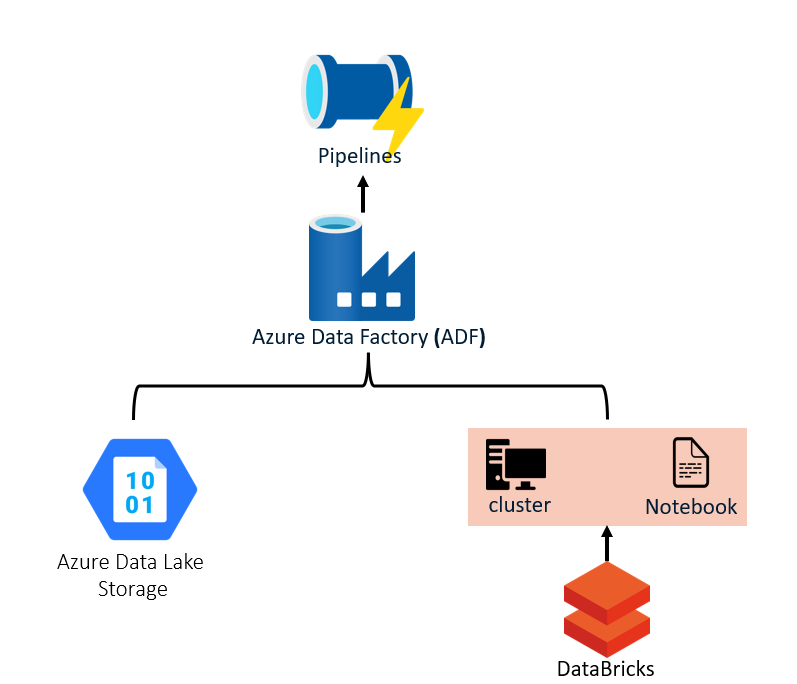
The dataset used in this project provides comprehensive statistics on global health, covering various diseases, treatments, and outcomes across multiple countries and years. It includes key metrics such as disease prevalence, incidence, mortality rates, and healthcare infrastructure, offering valuable insights for epidemiological research and health policy development.

**Key Attributes of the Dataset:**

* **Country:** The name of the country where the health data was recorded.
* **Year:** The year in which the data was collected.
* **Disease Name:** The name of the disease or health condition tracked.
* **Disease Category:** The category of the disease (e.g., Infectious, Non-Communicable).
* **Prevalence Rate (%):** The percentage of the population affected by the disease.
* **Incidence Rate (%):** The percentage of new or newly diagnosed cases.
* **Mortality Rate (%):** The percentage of the affected population that dies from the disease.
* **Age Group:** The age range most affected by the disease.
* **Gender:** The gender(s) affected by the disease (Male, Female, Both).
* **Population Affected:** The total number of individuals affected by the disease.
* **Healthcare Access (%):** The percentage of the population with access to healthcare.
* **Doctors per 1000:** The number of doctors per 1000 people.
* **Hospital Beds per 1000:** The number of hospital beds available per 1000 people.
* **Treatment Type:** The primary treatment method for the disease (e.g., Medication, Surgery).
* **Average Treatment Cost (USD):** The average cost of treating the disease in USD.
* **Availability of Vaccines/Treatment:** Whether vaccines or treatments are available.
* **Recovery Rate (%):** The percentage of people who recover from the disease.
* **DALYs:** Disability-Adjusted Life Years, a measure of disease burden.
* **Improvement in 5 Years (%):** The improvement in disease outcomes over the last five years.
* **Per Capita Income (USD):** The average income per person in the country.
* **Education Index:** The average level of education in the country.
* **Urbanization Rate (%):** The percentage of the population living in urban areas.

This dataset offers a multidimensional view of global health, allowing for in-depth analysis of disease patterns, healthcare access, and socio-economic factors influencing health outcomes. It serves as a foundation for various analyses, from identifying trends in disease prevalence to understanding the role of healthcare infrastructure in recovery rates.

**3.Architecture diagram**



**How it works:**

**Source Data File Name**

The project uses a single dataset, **Global Health Statistics.csv**, which provides detailed health statistics across various countries and years. This dataset includes information on disease prevalence, incidence, mortality rates, treatment costs, healthcare access, and other health-related metrics.

**Workflow:**

**1. Data Processing with Databricks**:

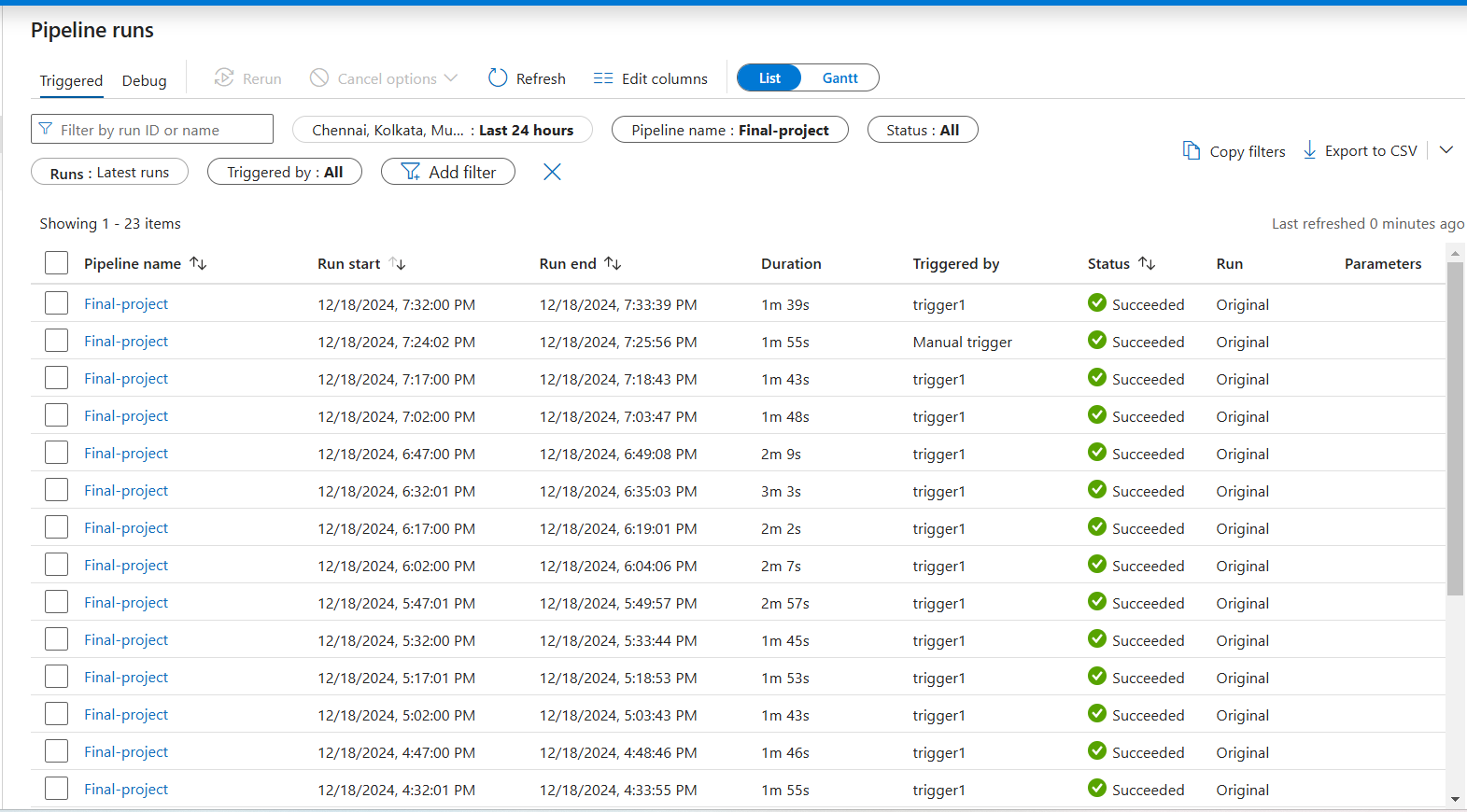
* **Azure Databricks** is used to perform data transformations and analytics in a serverless environment, reducing infrastructure management overhead.
* **Databricks Notebook 1** is responsible for the initial data cleaning and ingestion, using PySpark to load and clean the data from the Data Lake, and saving it as a **Delta table** for consistency.
* **Databricks Notebook 2** performs visualization.

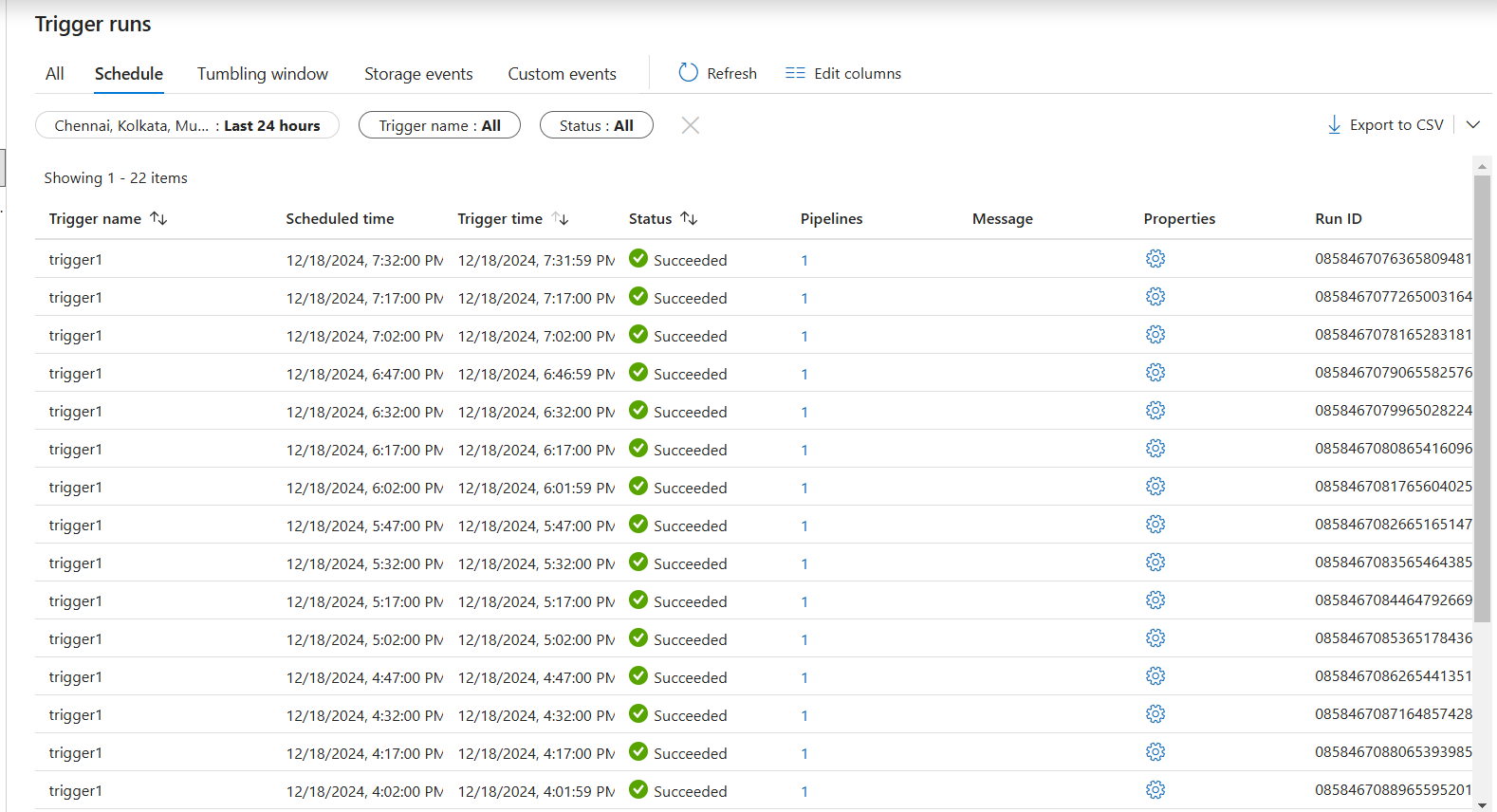
**2. Pipeline Orchestration with Azure Data Factory**:

* A **single pipeline** in **Azure Data Factory** orchestrates the execution of the two Databricks notebooks sequentially.
* The first notebook (**Notebook 1**) runs to clean the data, and once completed, the second notebook (**Notebook 2**) start running. These notebooks are connected within the same pipeline to ensure sequential execution.
* The **trigger** is created to schedule or manually start the pipeline. A manual trigger allows ad-hoc execution, while a scheduled trigger enables periodic executions (e.g., daily, weekly).

**3. Monitoring and Optimization**:

* **Azure Data Factory’s monitoring tools** track pipeline execution, displaying detailed logs, error reports, and runtime information.

****

****

**Azure Resources Used for this Project**

1. **Azure Data Factory (ADF)**
   * **Purpose**: A cloud-based data integration service that enables you to create, schedule, and orchestrate data workflows.
   * **Usage**: Azure Data Factory is used to automate the movement and transformation of data. It helps in creating pipelines that connect to various data sources, transform the data (if necessary), and load it into Azure Data Lake or Azure Databricks for further processing. It also helps in orchestrating the overall data pipeline and managing tasks across various services.
2. **Azure Databricks**
   * **Purpose**: An Apache Spark-based analytics platform optimized for the cloud, designed for big data processing, machine learning, and data engineering workflows.
   * **Usage**: Azure Databricks will be used for performing data transformations, running analytics, and applying machine learning models on the ingested data. It can directly access data stored in Azure Data Lake, process it on-demand, and generate insights or perform deep analysis.
3. **Azure Data Lake Storage**
   * **Purpose**: A highly scalable and secure data lake solution to store large amounts of unstructured and structured data.
   * **Usage**: Data Lake will serve as the central repository for both raw and processed data. Azure Data Factory can be used to load data into Data Lake from various sources, and Azure Databricks will then process and transform the data stored here for analytics or reporting purposes. You can organize data into containers for easier management and analysis.

**Project Requirements & Tasks Performed**

**Project Requirements:**

**1. Azure Services**

* **Azure Data Factory (ADF):** To orchestrate data workflows, execute Databricks notebooks, and manage pipeline executions.
* **Azure Databricks:** To perform serverless data processing, including data transformation and analysis, using notebooks.
* **Azure Data Lake:** To store raw input data and processed output data efficiently for scalable access.

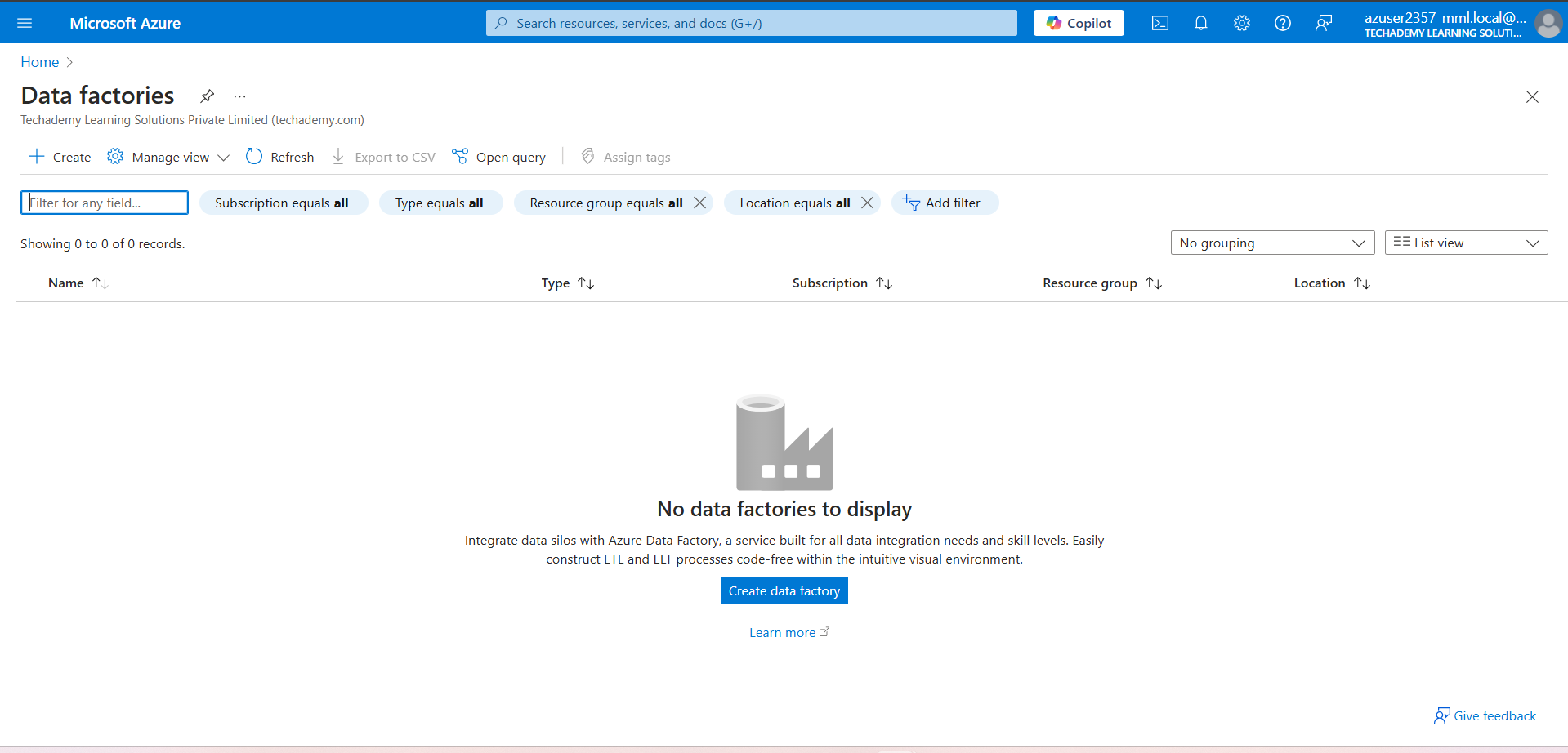
**2. Input Data**

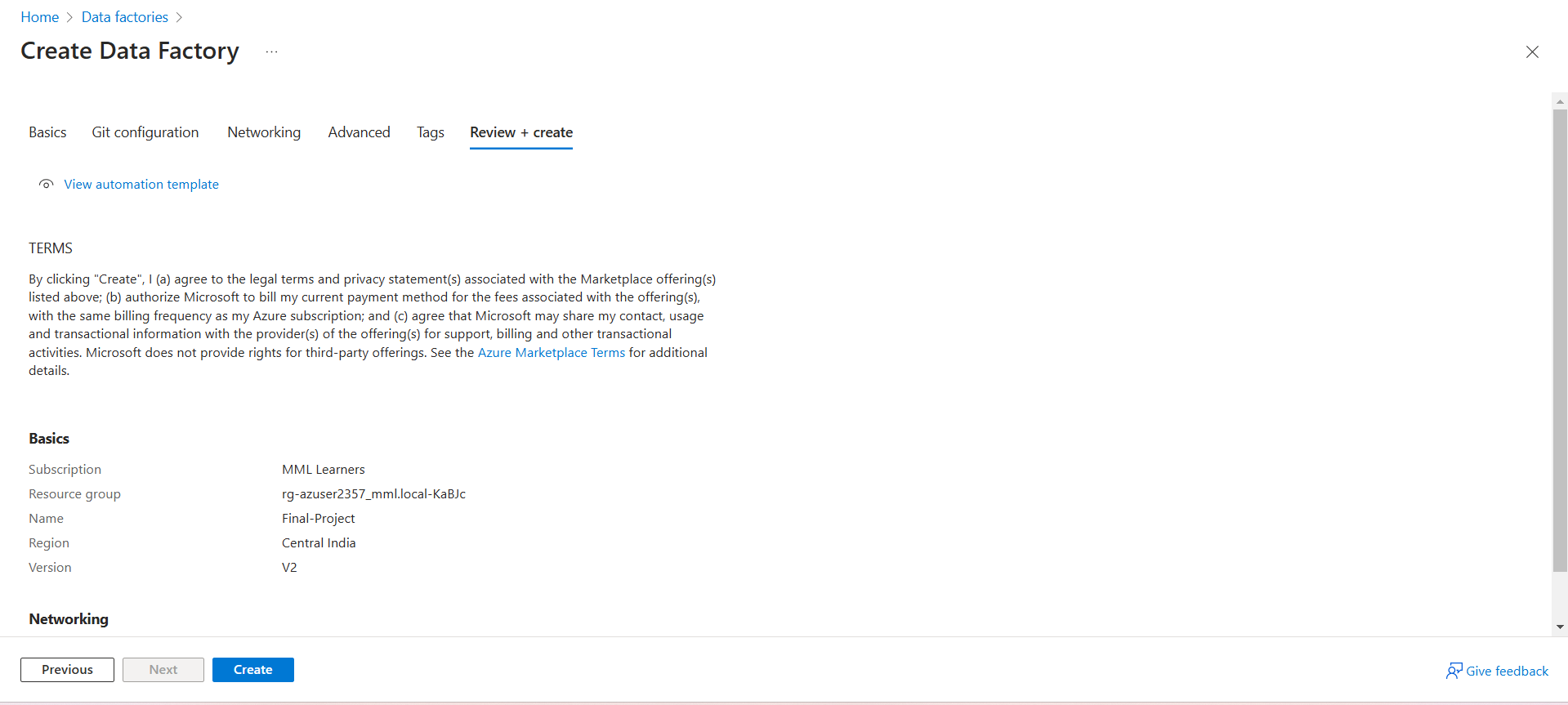
* **File Format:** CSV
* **Source:** [Kaggle](https://www.kaggle.com/datasets/malaiarasugraj/global-health-statistics)
* **Description:** The CSV file contains raw data to be ingested, processed, and transformed into meaningful insights.

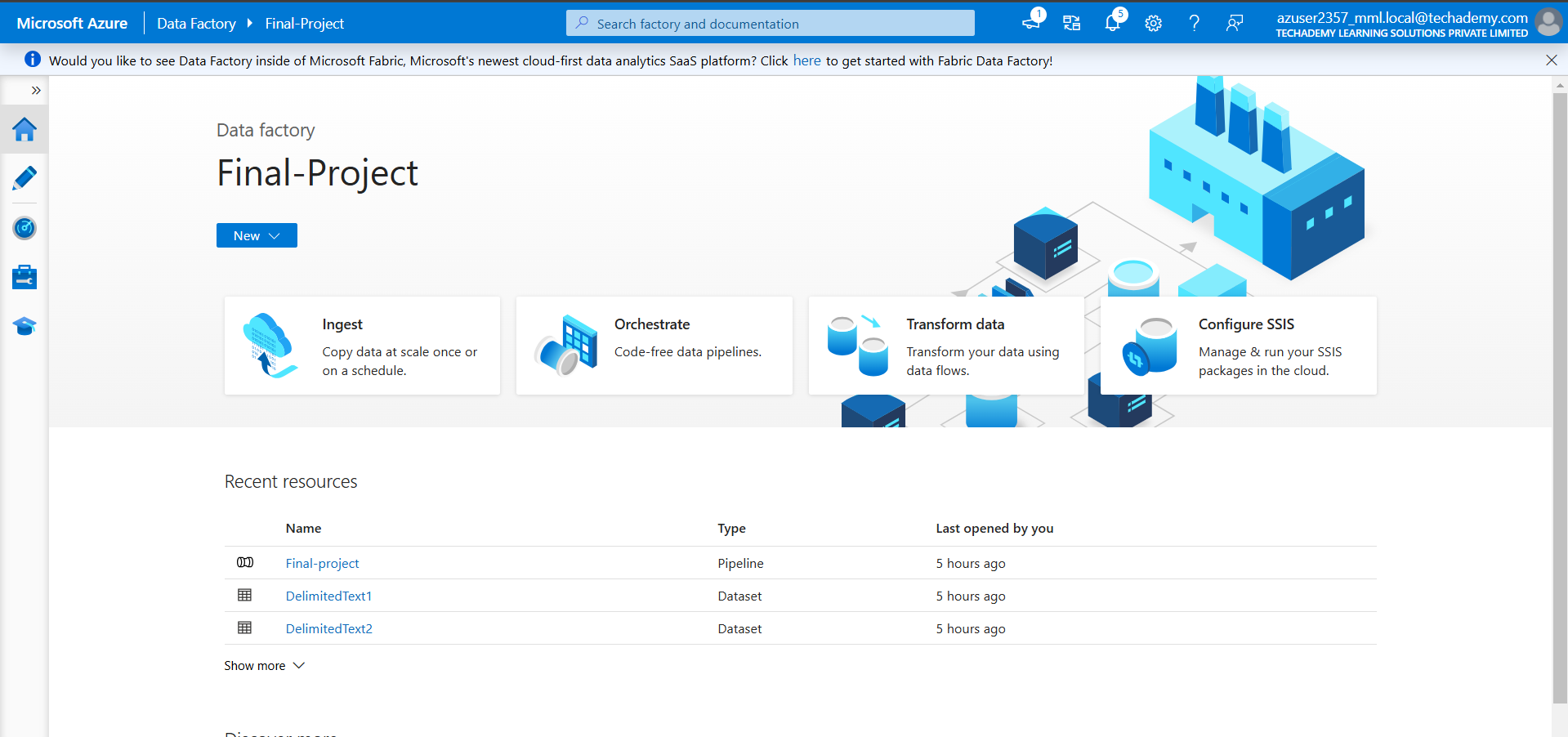
**Tasks Performed:**

**Task 1: Set Up Azure Data Factory**

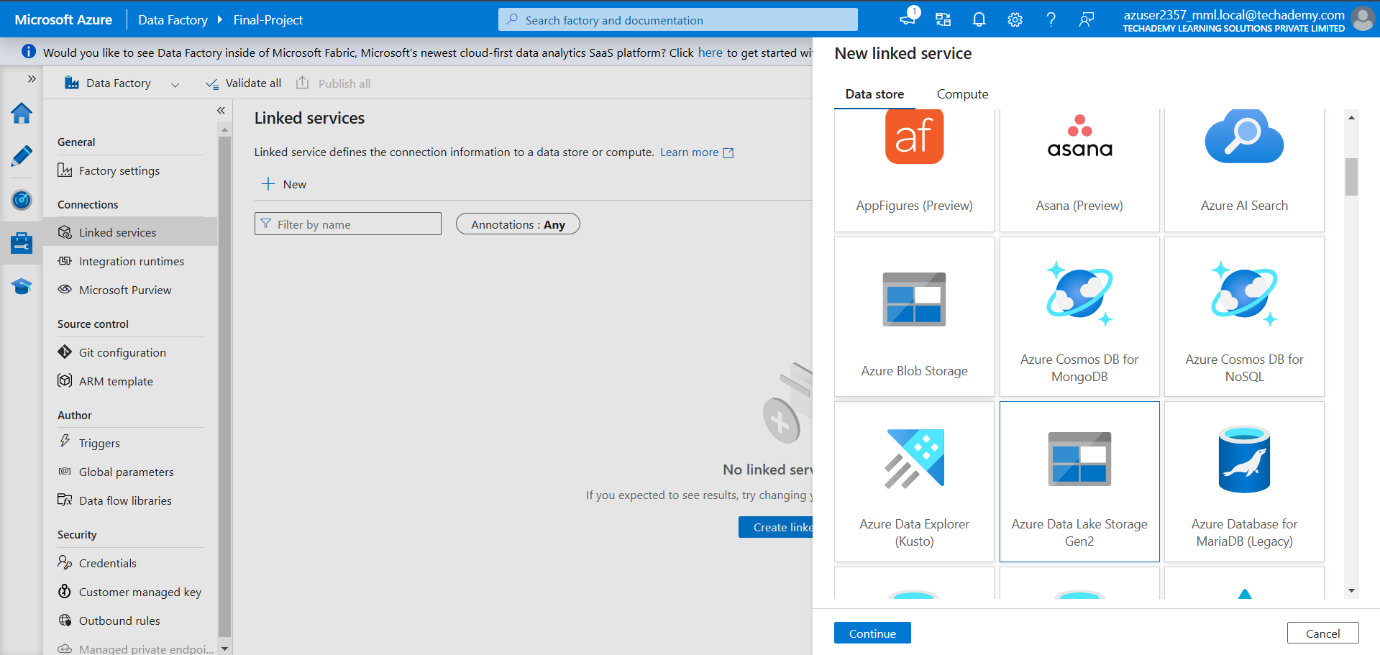
* **Create an Azure Data Factory instance**:
  + Go to the Azure portal, select **Data Factory**, and create a new instance.

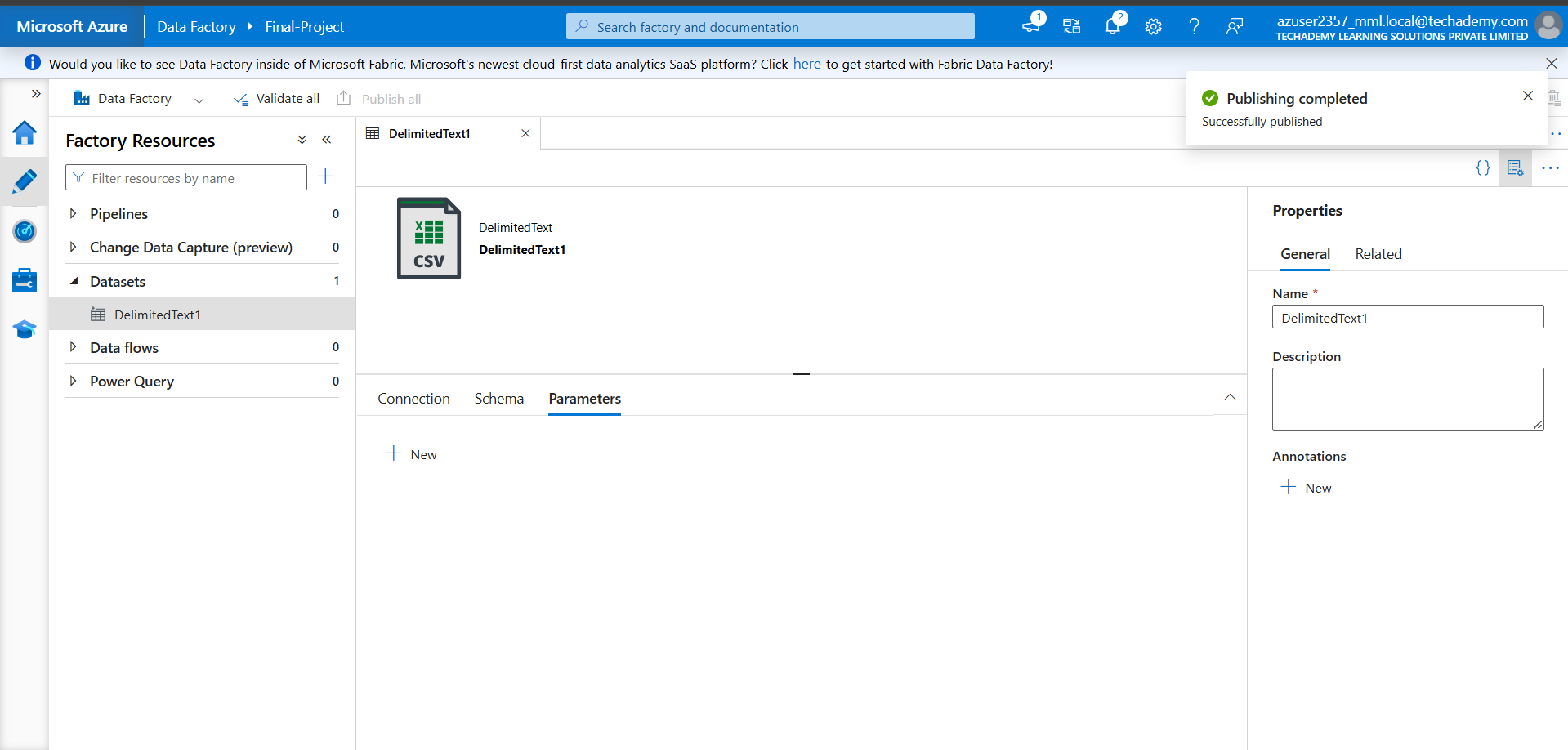






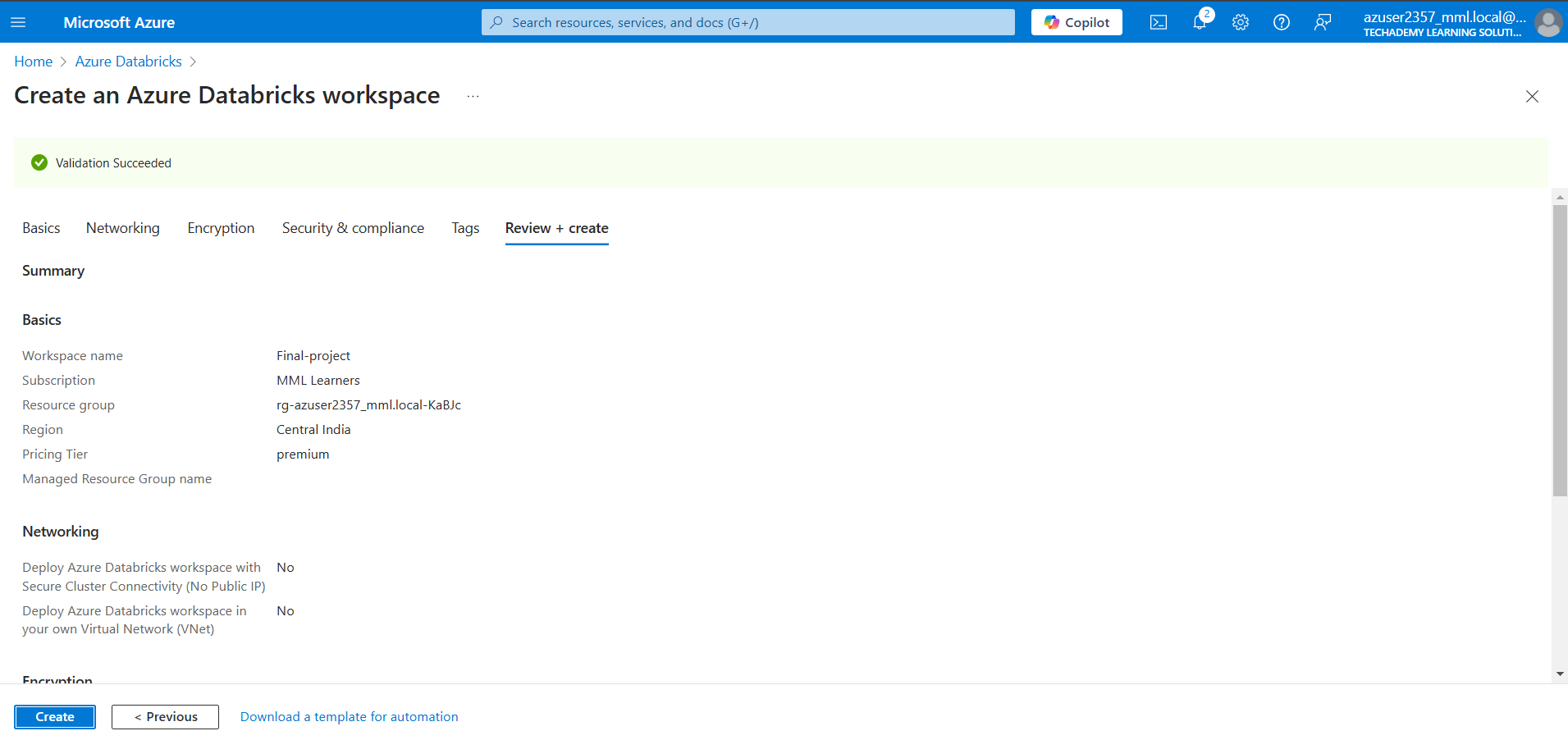
* **Set up linked services**:
  + Connect Azure Data Factory to your data sources and sinks (e.g., Azure Data Lake, Blob Storage).

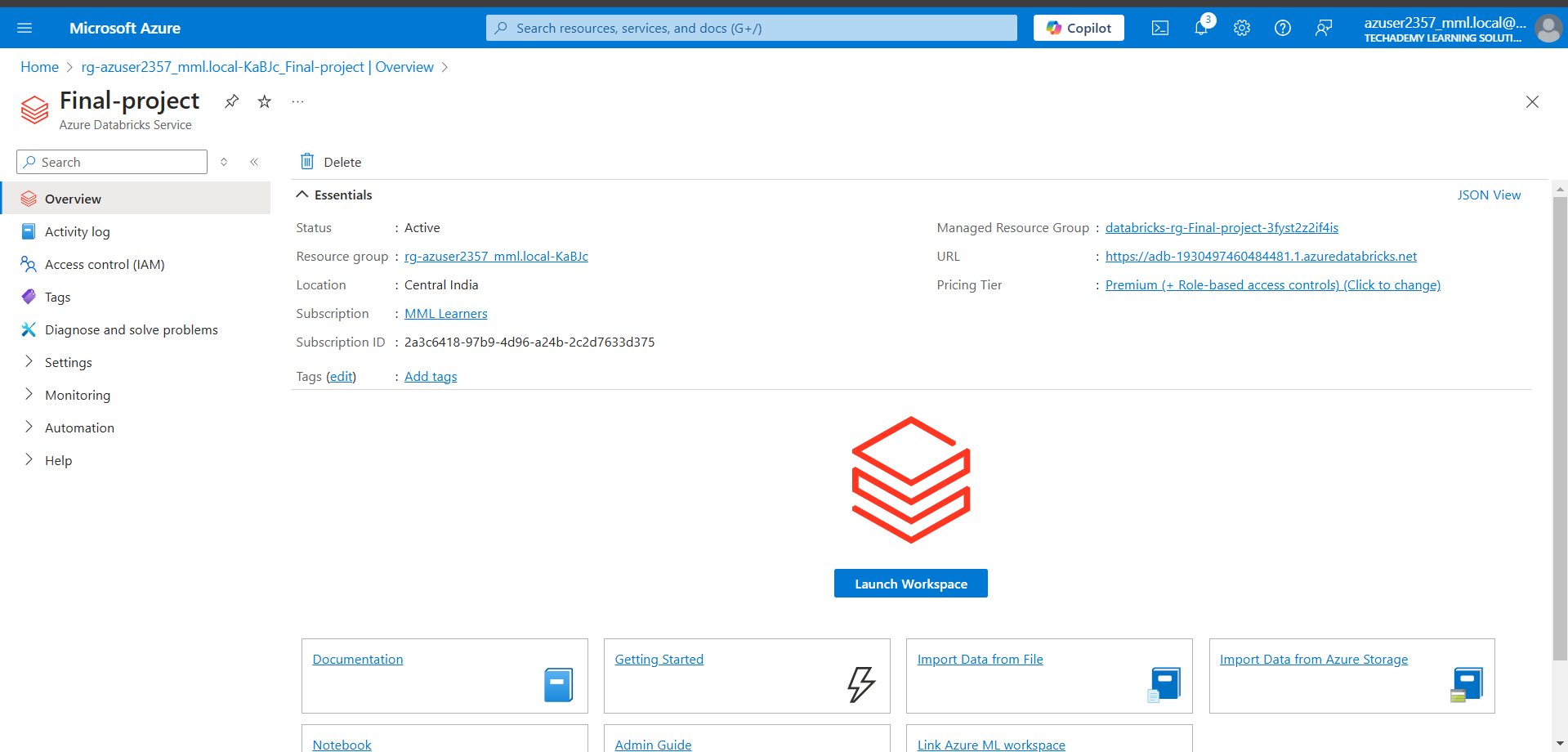




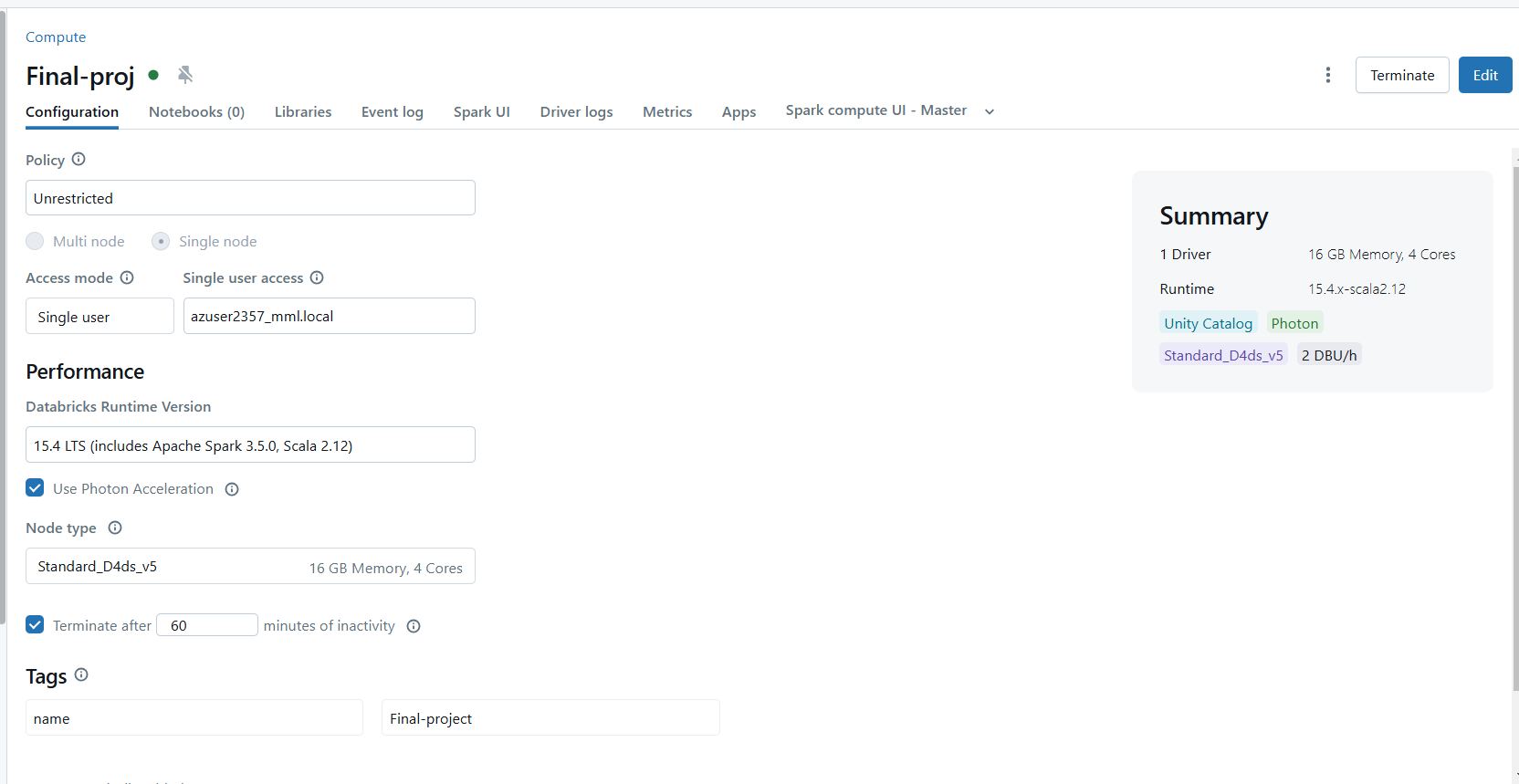
**Task 2: Set Up Azure Databricks**

* **Create an Azure Databricks workspace**:
  + In the Azure portal, create a Databricks workspace and configure it for serverless processing.



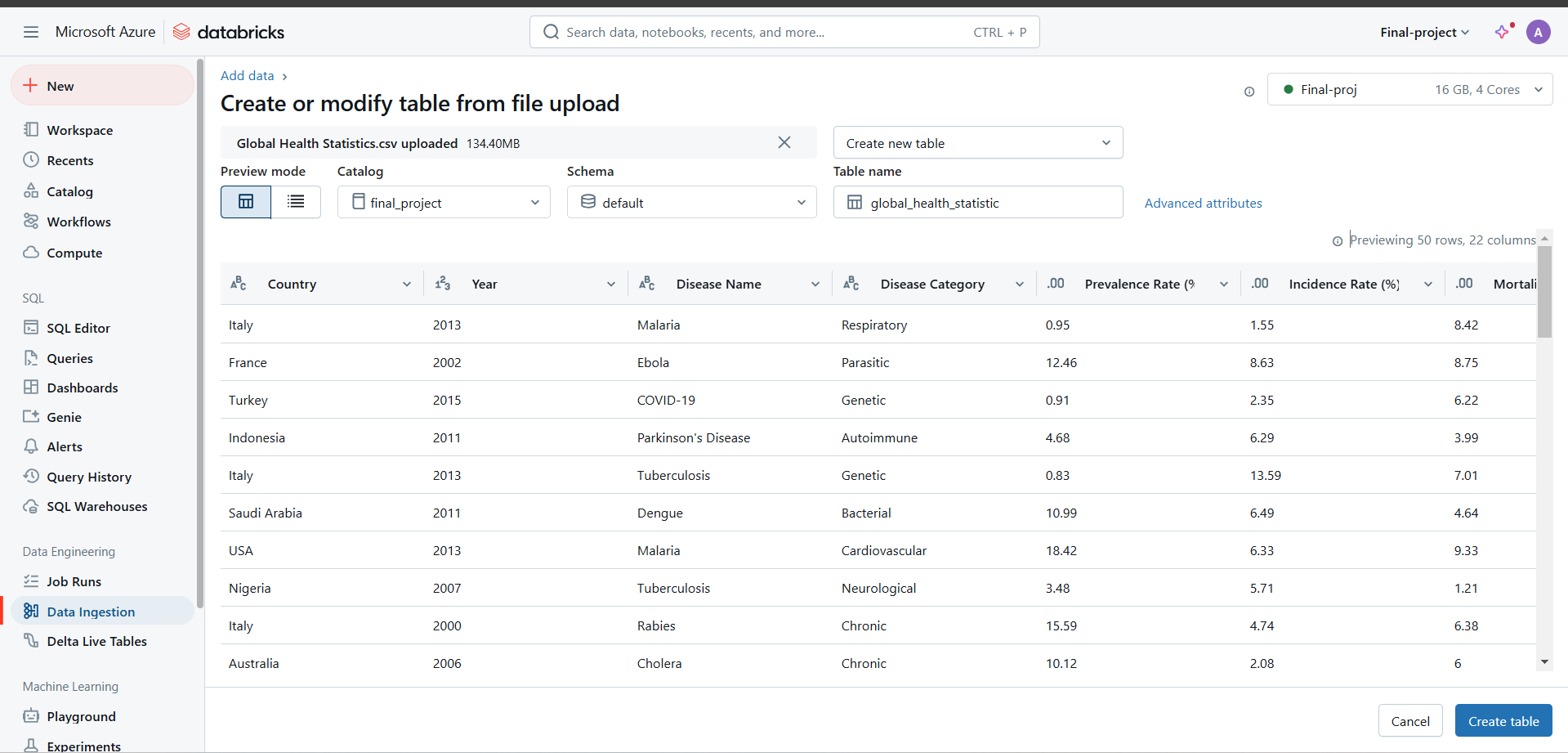


* **Cluster setup**:
  + Configure a serverless cluster or job cluster for efficient and cost-effective data processing.



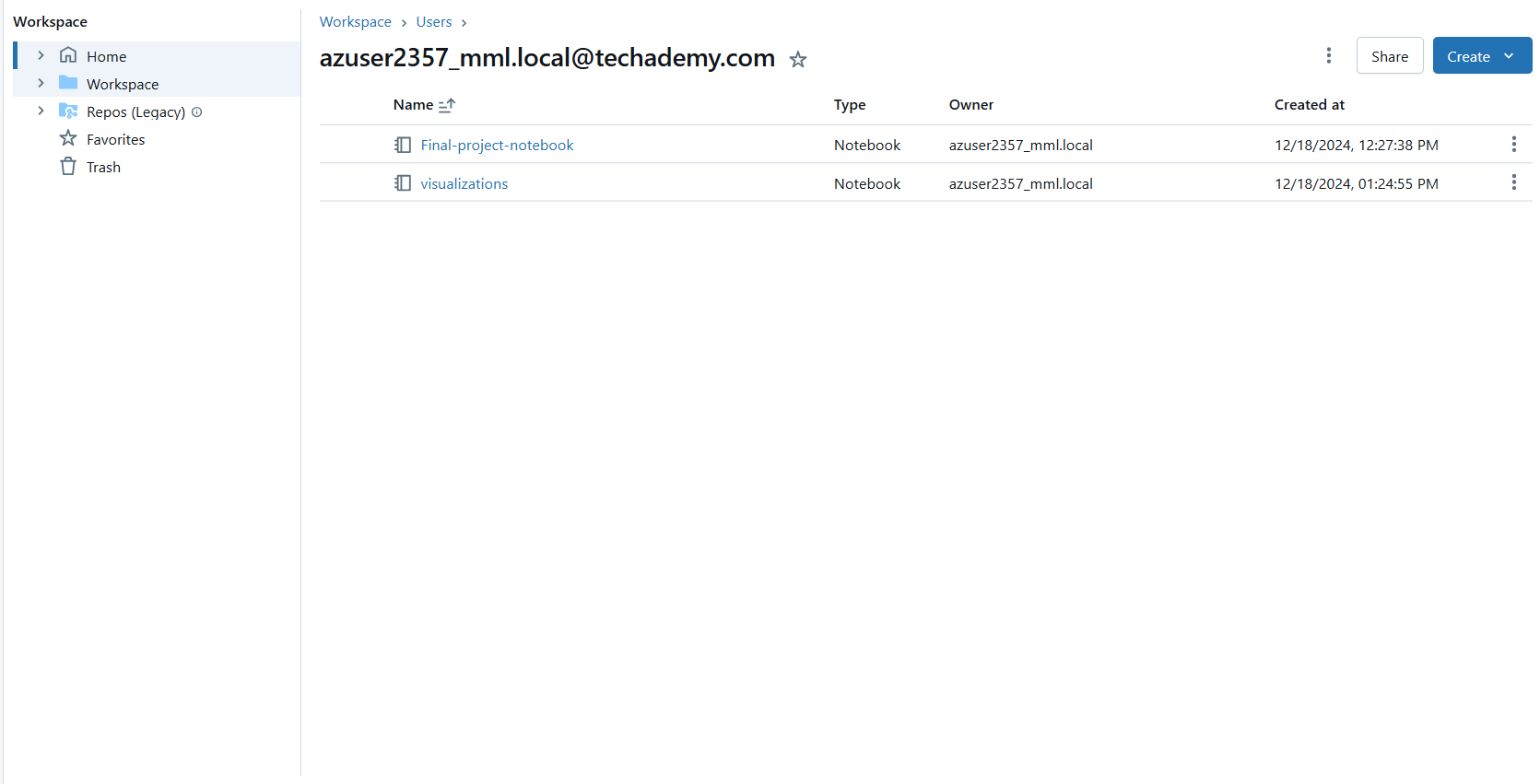
**Task 3: Add the Input Data**

* **Upload data**:
  + Add the input CSV file



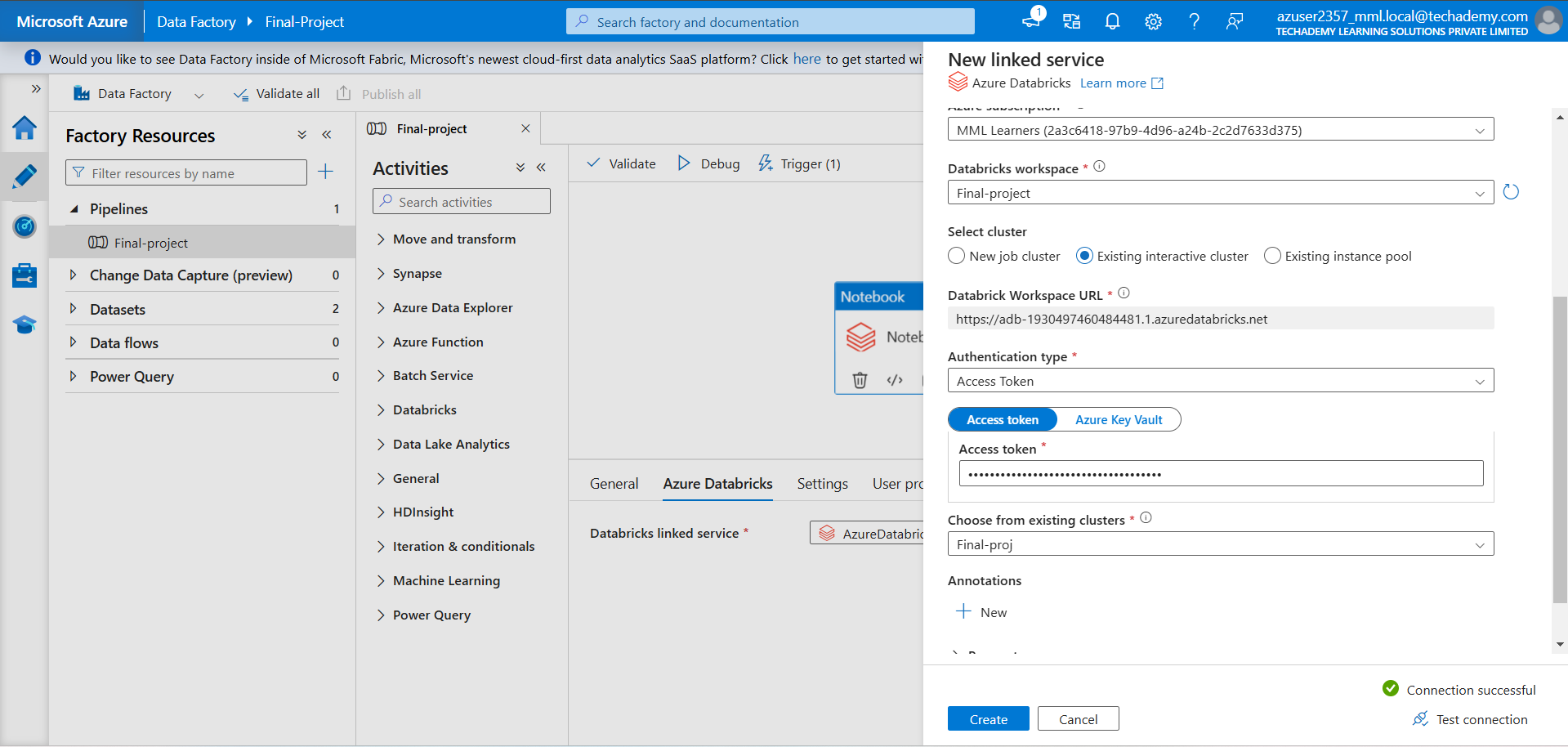
**Task 4: Create and Configure Databricks Notebooks**

* **Notebook 1**:
  + Develop a notebook for initial data ingestion, cleaning, and transformations.
* **Notebook 2**:
  + Create a second notebook for visualization.

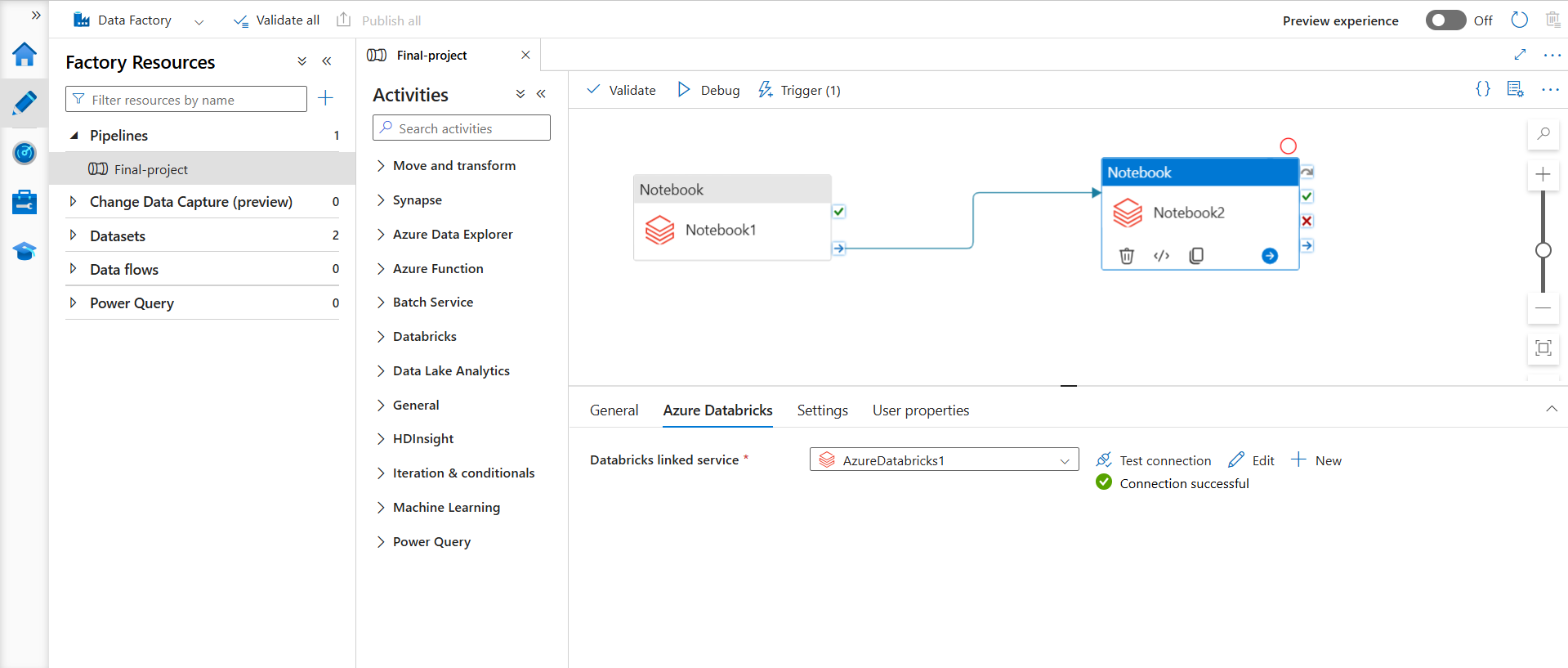


**Task 5: Design a Single Pipeline in Azure Data Factory**

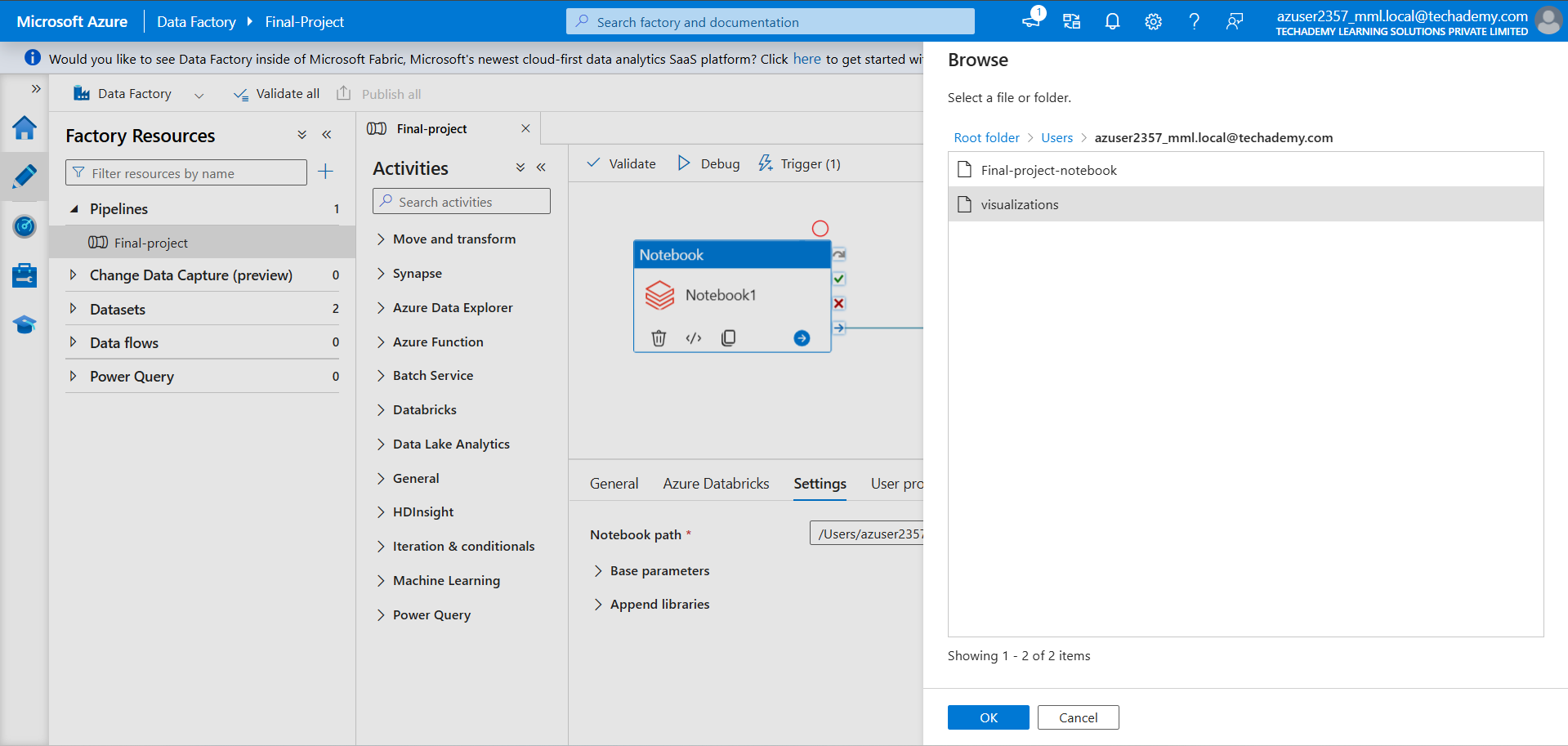
* **Create a pipeline:**
  + Open Data Factory and navigate to **Author > Pipelines > New Pipeline.**
  + Add a **Databricks Notebook** activity for **Notebook 1** and configure it.



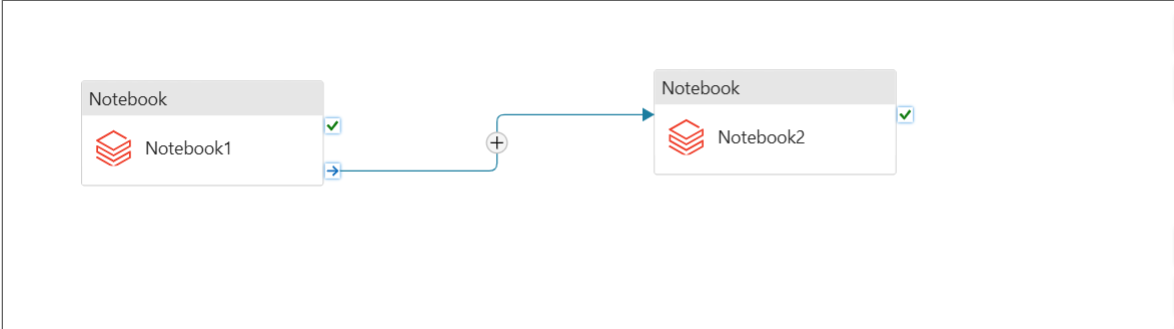
* + Add another **Databricks Notebook** activity for **Notebook 2** and configure it.



* In the activity settings, specify the notebook path for **Notebook** and configure the cluster.

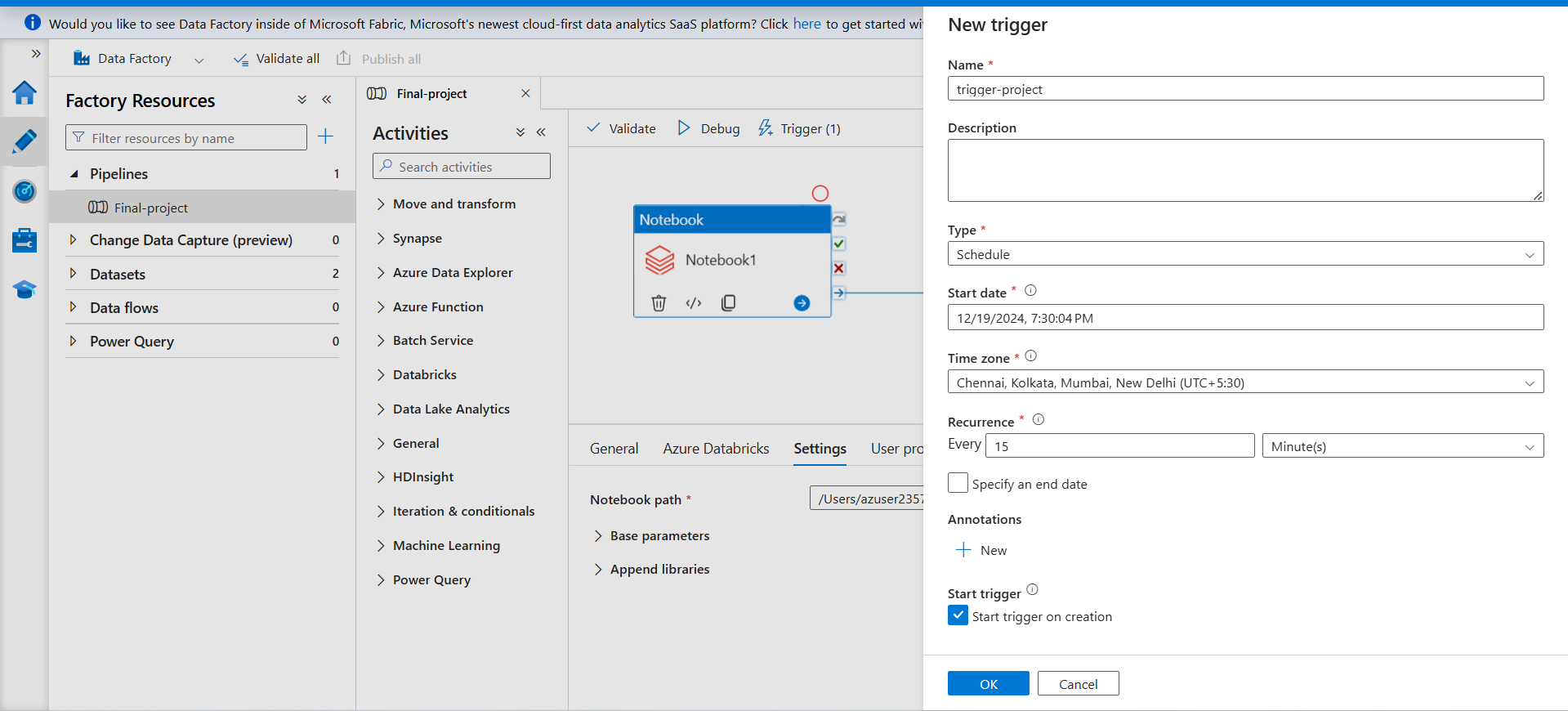


* Connect the activities to execute **Notebook 2** after **Notebook 1**.



**Task 6: Create a Trigger**

* **Add a trigger to the pipeline**:
  + Navigate to the **Triggers** tab in the pipeline editor.
  + Click on **Add Trigger** > **New/Edit**.
  + Configure the trigger type:
    - **Schedule Trigger**: Define a schedule for periodic pipeline execution



**Task 7: Monitor and Optimize**

* **Monitor pipeline execution**:
  + Go to the **Monitor** tab in Data Factory to view pipeline run status.

