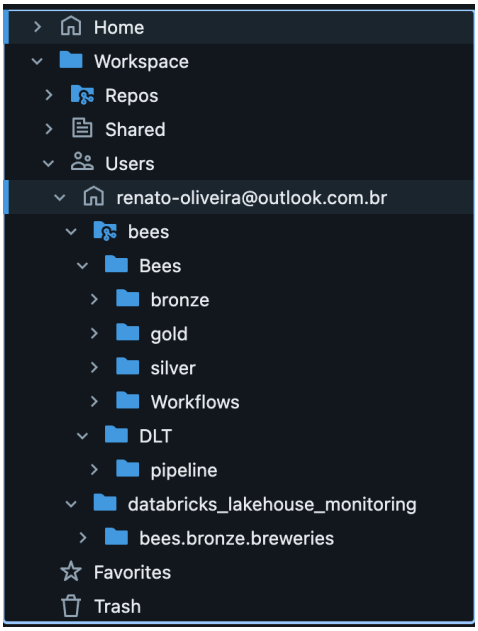
BEES Data Engineering Case – Visual Workflow & Execution Overview

This document provides a detailed visual and descriptive overview of the workflows, executions, and monitoring strategies applied in the BEES Data Engineering case developed in Databricks.

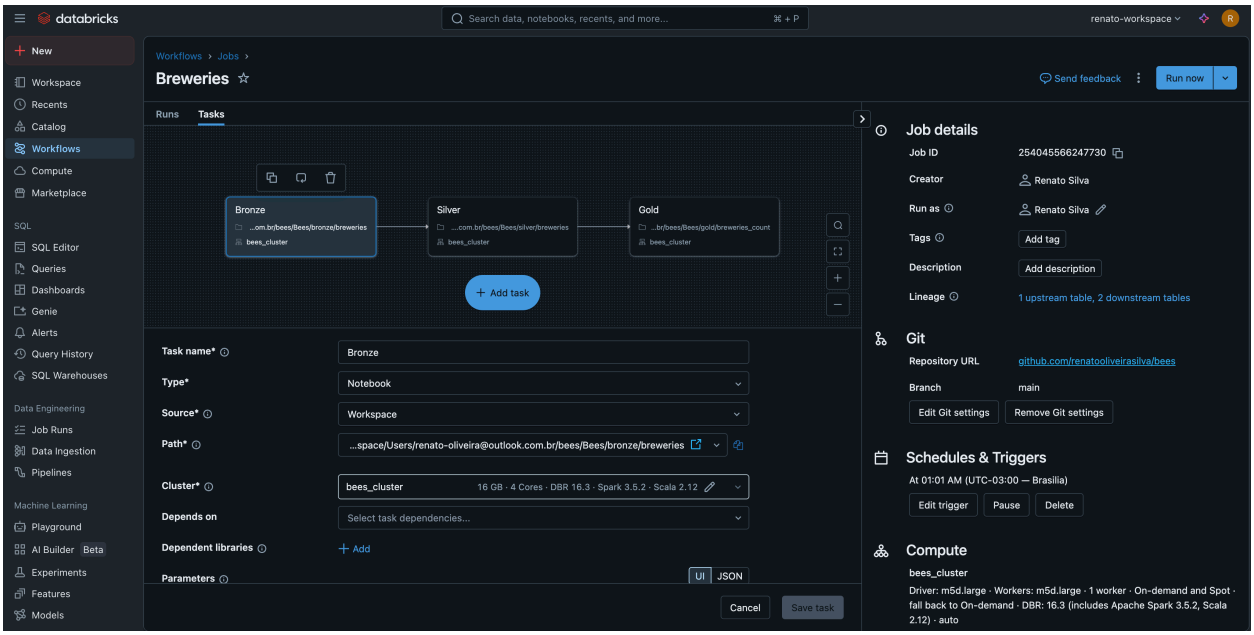
📁 Project Structure Visualization

Below is the visual representation of the structured folder architecture within the Databricks workspace. This organization follows data lakehouse principles, separating ingestion (raw - bronze), transformation (cleaned - silver), and final analytics (aggregated - gold) stages.



⚙️ Databricks Workflow Job

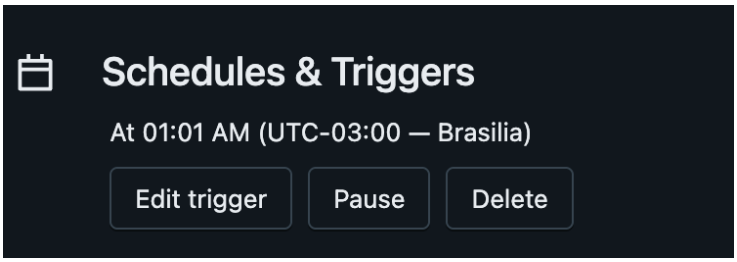
The following image shows the Databricks Workflow Job created to orchestrate the pipeline. This job coordinates notebook execution, defines task dependencies, and ensures proper execution order.



The JSON configuration used to create this job is available in the project's GitHub repository.

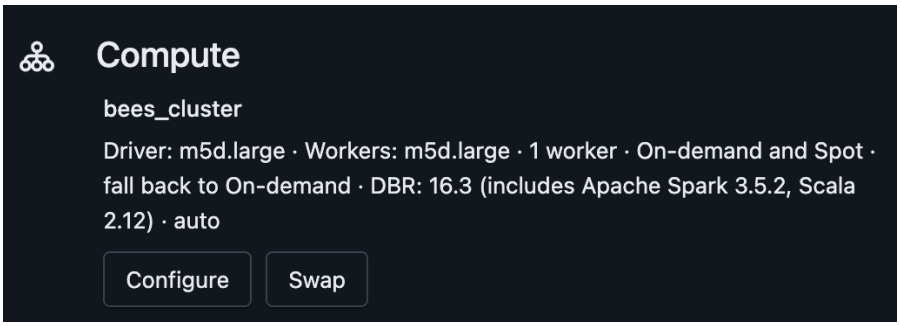
⏰ Scheduled Job Simulation

A simulation of the scheduled execution was set up to mimic a production-like environment. This helps demonstrate how automated, recurring executions would work over time.



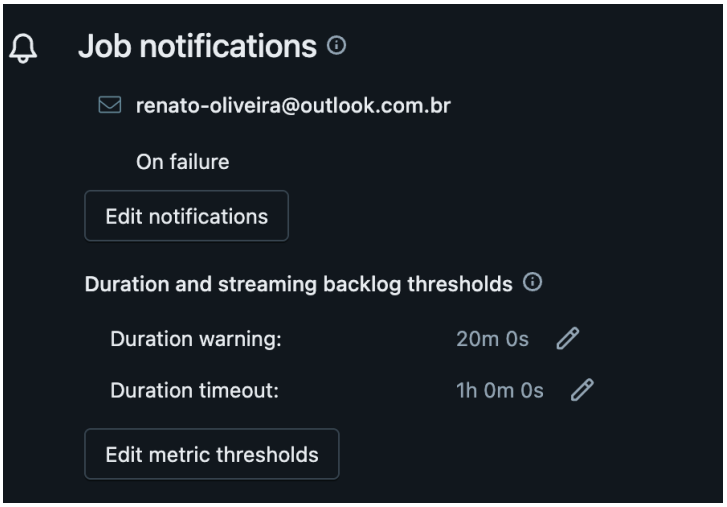
💰 Cluster Configuration

A job cluster configuration was used, as recommended by Databricks. This setup ensures compute resources are only active during pipeline execution, optimizing cost efficiency.



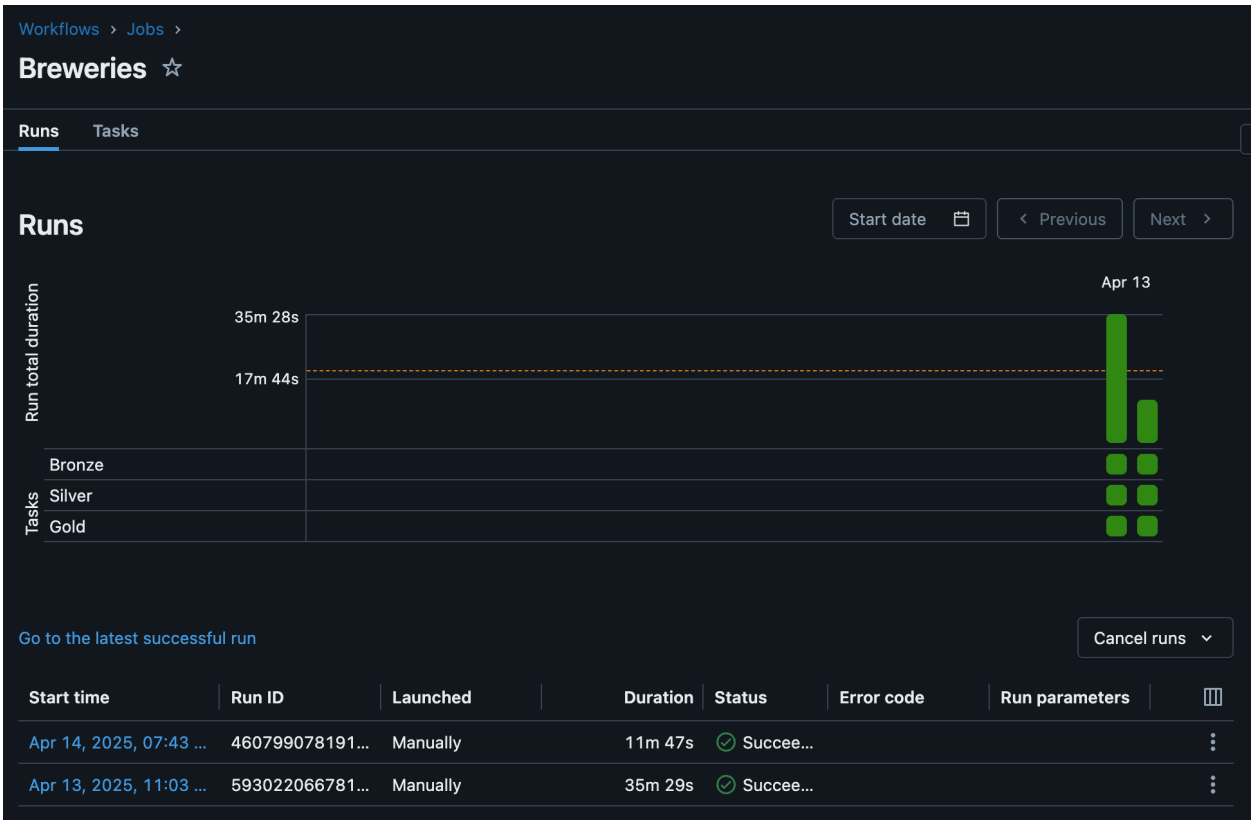
📡 Failure Monitoring & Alerts

To improve observability and operational awareness, email alerts were configured for failed runs. The job was also configured with a 1-hour timeout and an early warning alert set at the 20-minute mark.

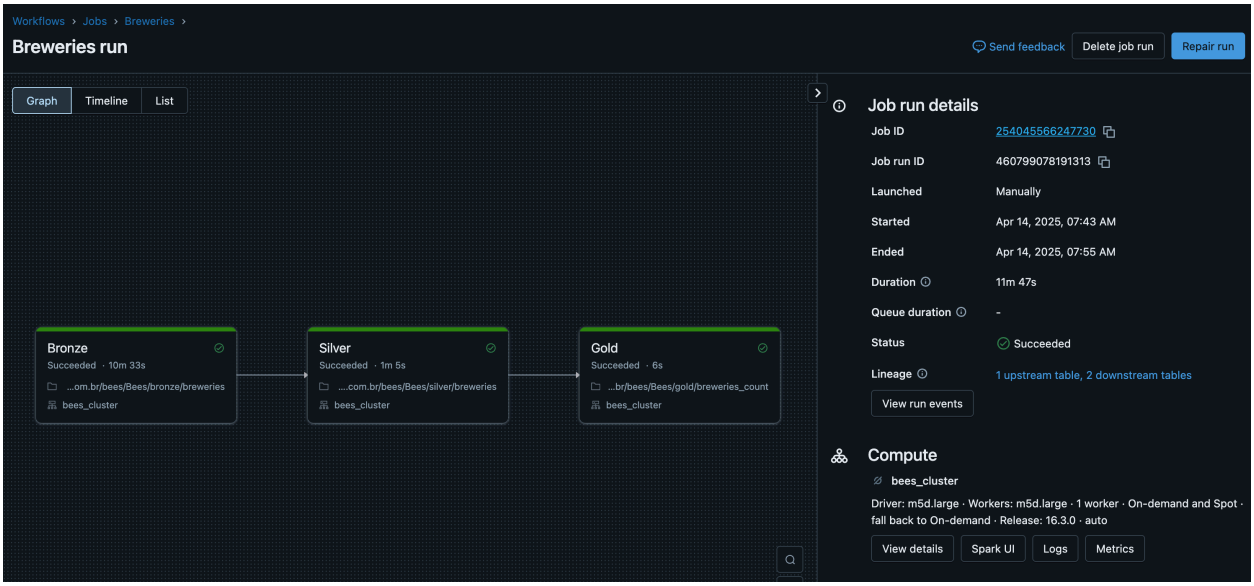


📊 Execution Example & Lineage

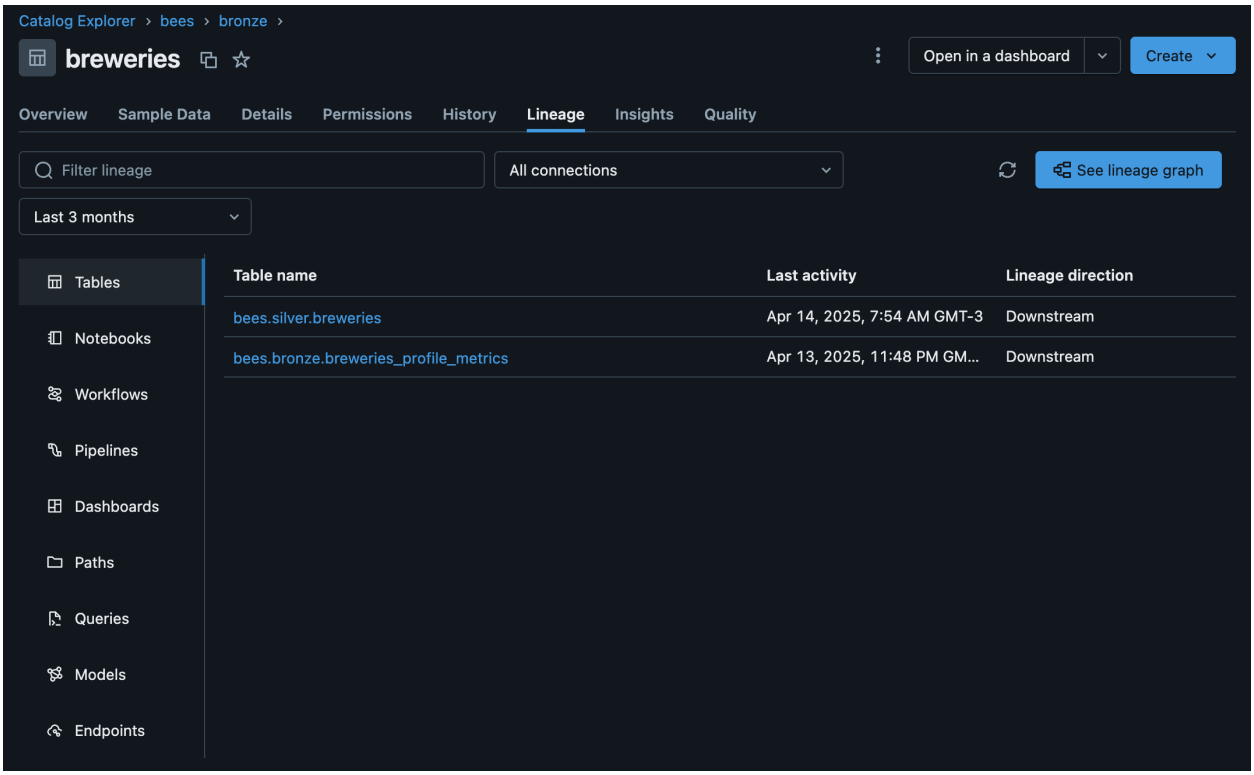
Below are visual examples of two pipeline runs, showcasing task status and execution details. Following that, a data lineage graph highlights how data progresses from the ingestion layer to the final curated tables.



- Workflow lineage



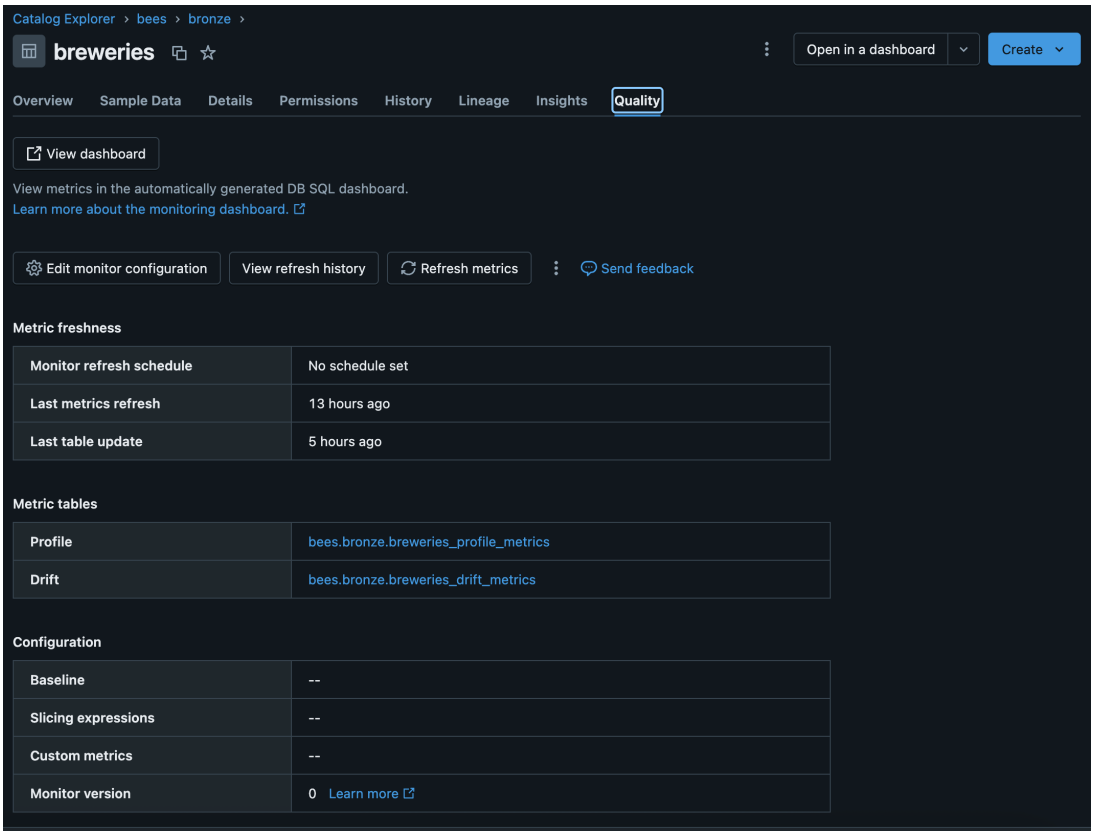
- Table lineage

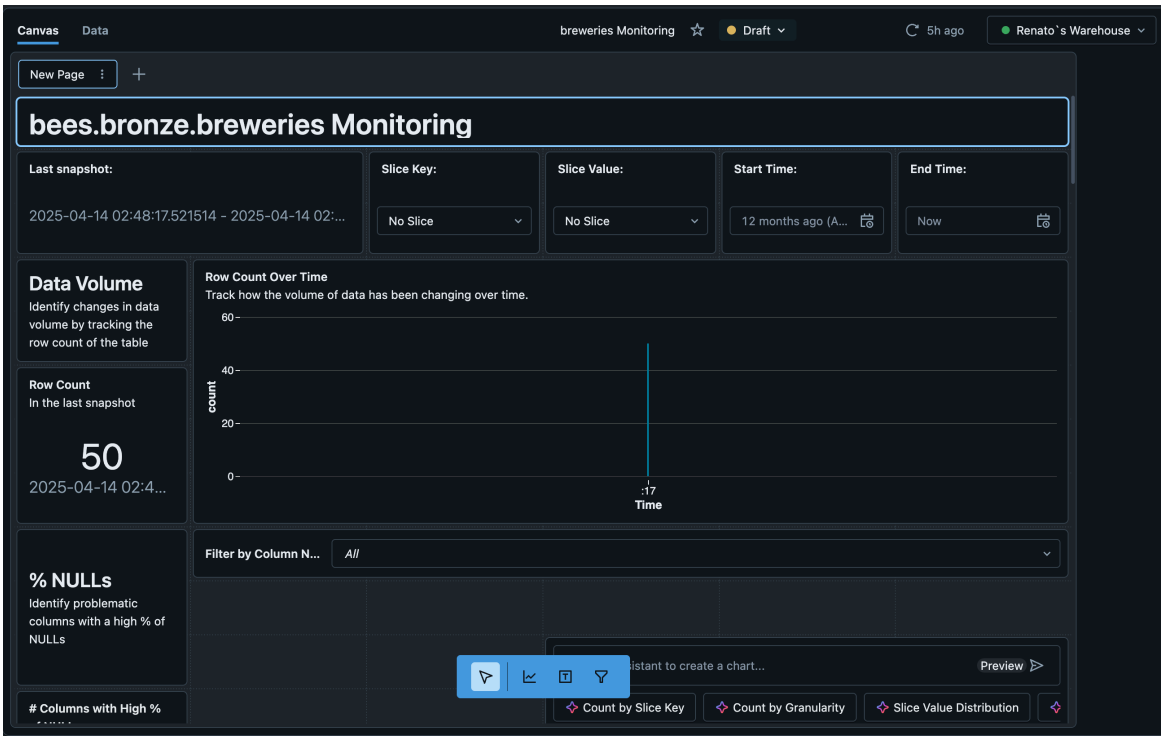


- Table lineage graph

✅ Data Quality Layer

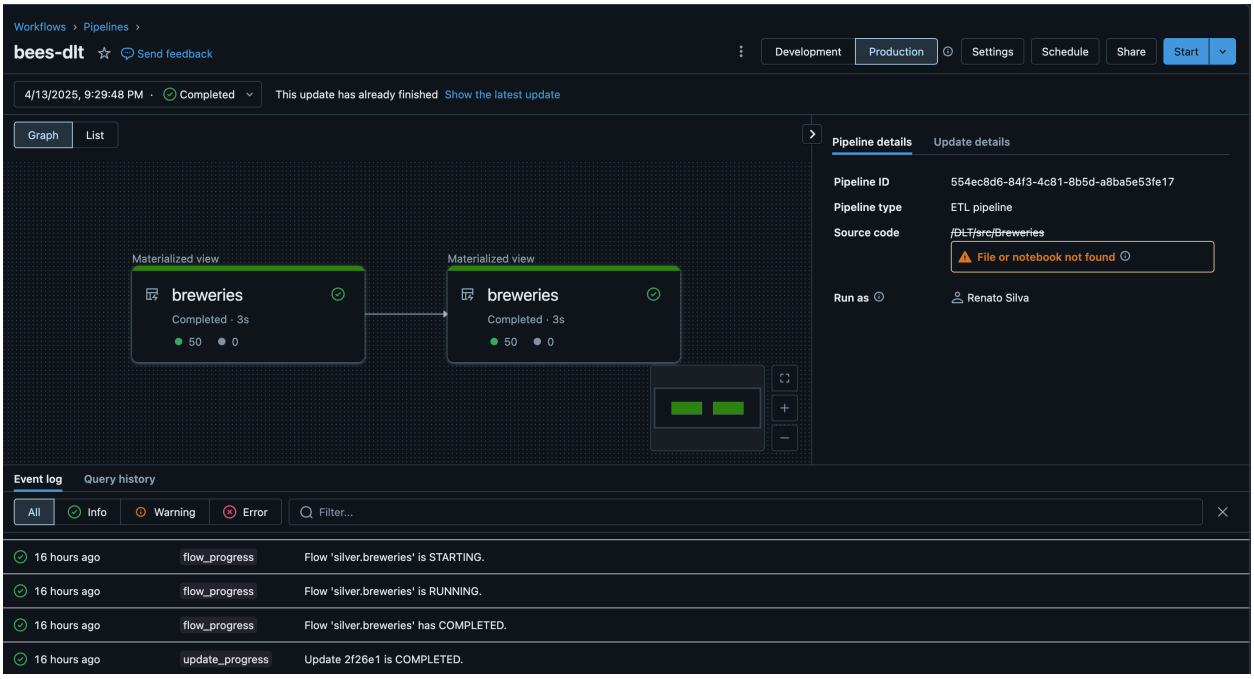
An example of the data quality monitoring layer is shown below. This dashboard provides real-time validation metrics such as null checks, unique constraints, and domain-specific validations.





🧪 Delta Live Tables (DLT) Execution

Additionally, a Delta Live Tables (DLT) pipeline was configured and successfully executed. This highlights the project's capability to support declarative pipelines using Databricks' managed features.



📌 Final Notes

This visual documentation serves to complement the GitHub README by providing concrete examples and operational insights from within the Databricks environment.