# Echoes of Monitoring

## Overview

**Echoes of Monitoring** is a containerized data pipeline project that integrates orchestration, distributed processing, real-time observability, and self-healing capabilities. It leverages Apache Airflow, Apache Spark, Prometheus, and Grafana in a tightly coupled yet modular Docker environment. The goal: **resilient, observable, and recoverable DAG execution.**

## Stack Used

* **Orchestration:** Apache Airflow (v3.0.3)
* **Processing Engine:** Apache Spark (v3.5.0 with Hadoop 3)
* **Monitoring:** Prometheus
* **Visualization:** Grafana
* **Containerization:** Docker, Docker Compose
* **Language:** Python 3.10

## Core Features Implemented

### 1. **Self-Healing Airflow DAGs**

* Custom PythonOperator DAGs
* Resilient decorator @recovery\_wrapper
* Automatic logging and retry logic
* Logging to both stdout and file (/opt/logs/streamline.log)

### 2. **Spark Data Pipeline**

* Reads from mounted volume /opt/input
* Repartitions and writes Parquet to /opt/output/streamline2\_run
* Built with PySpark and tested using spark-submit
* Fully dockerized Spark runtime inside Airflow worker

### 3. **Prometheus Metrics Export**

* Custom Python-based metric exposure
* files\_copied and file\_copy\_errors tracked and exported
* /metrics endpoint scraped by Prometheus
* GC, memory, and system process metrics also available

### 4. **Grafana Dashboarding**

* Metrics visualized in real-time
* Prometheus connected to Grafana
* Panels for:
  + Spark task durations
  + File processing stats
  + Python system metrics

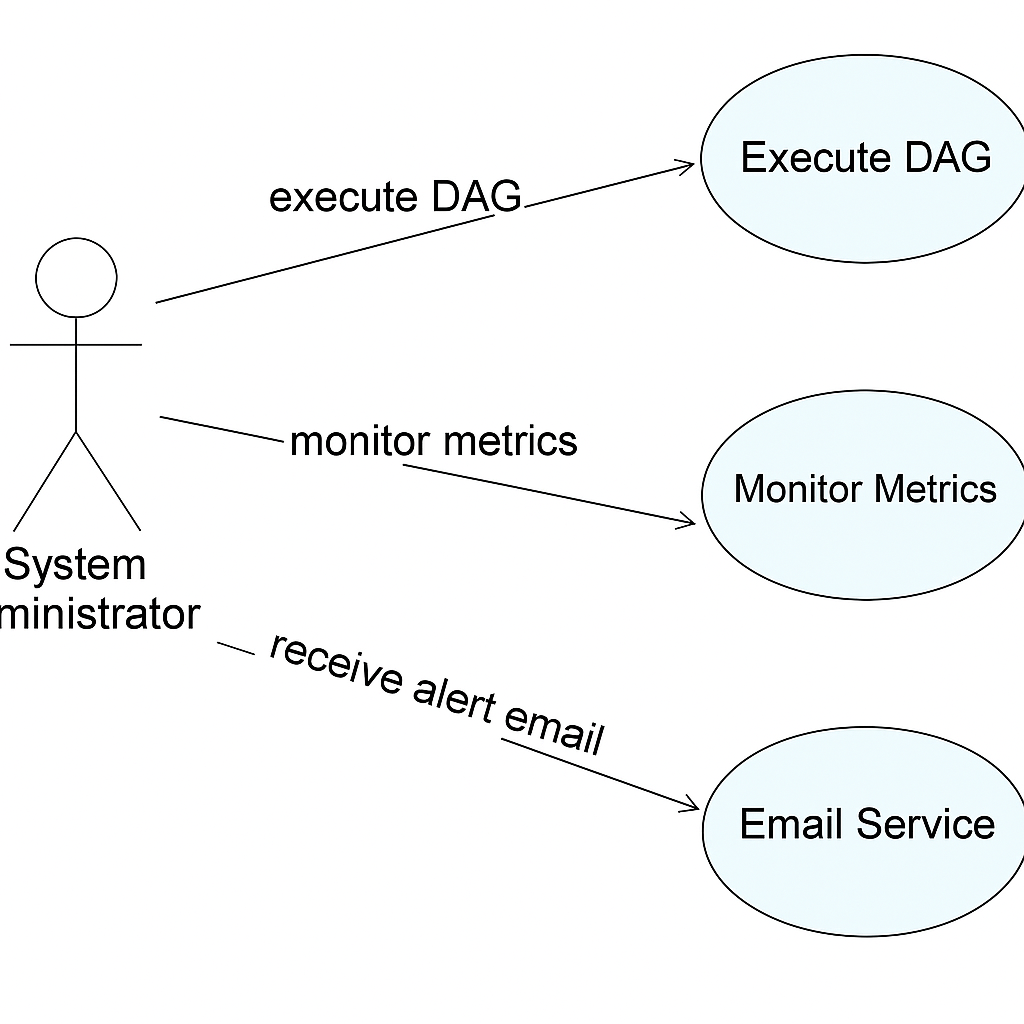
### 5. **Networking & Integrity**

* External Docker bridge network monitoring\_net
* Ensures Airflow <-> Prometheus <-> Grafana communication
* Survives container restarts by defining network in docker-compose.yml

## Workflow Execution

* Python PySpark Script.
* Error-handled via @recovery\_wrapper.
* Monitored via Prometheus.
* Triggered via Airflow DAG echoes\_of\_monitoring.

USE CASE Diagram:-



## Docker Highlights

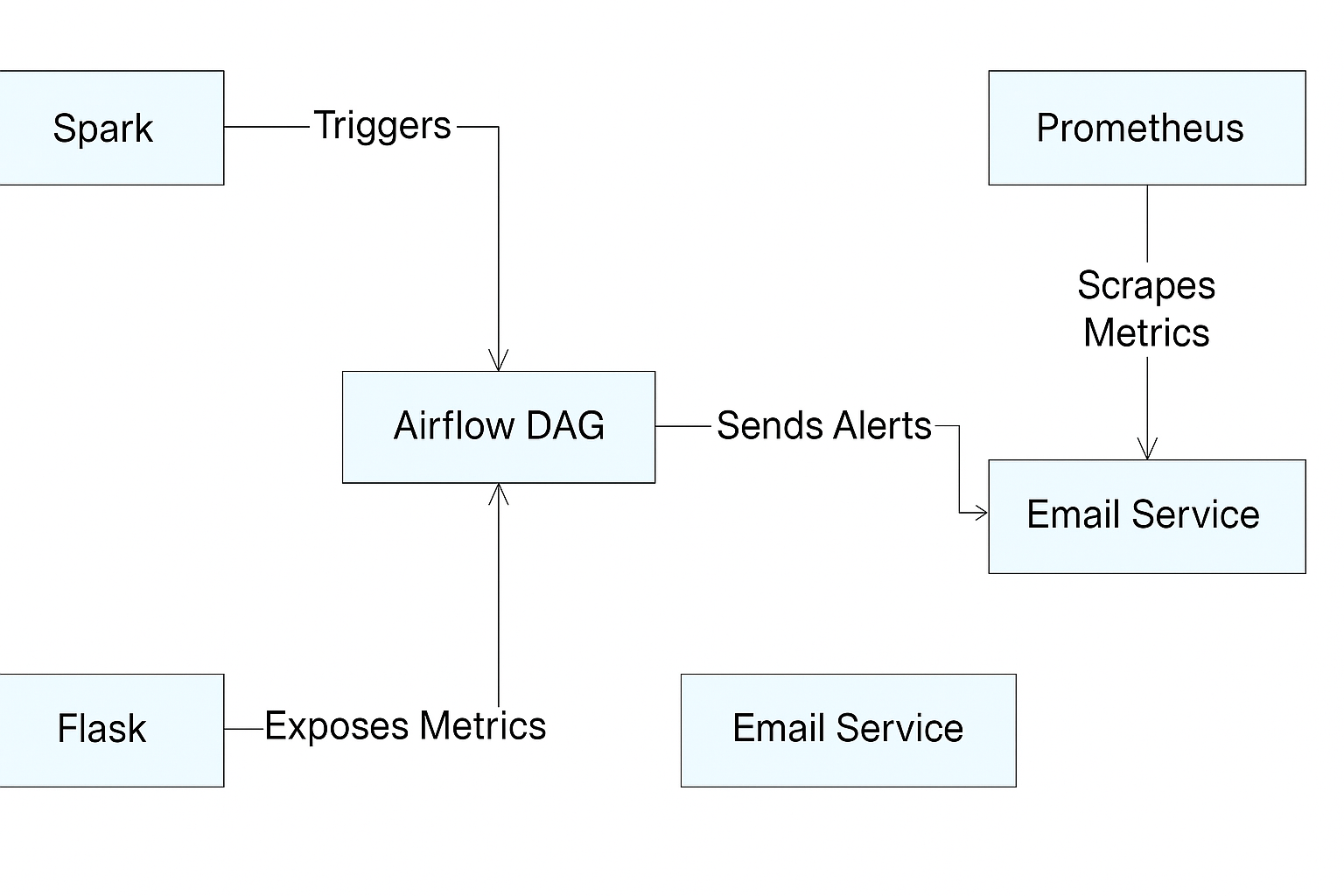
### Dockerfile

* Inherits apache/airflow:3.0.3-python3.10
* Installs:
  + Java (OpenJDK 11)
  + Spark 3.5.0
  + Python packages: pyarrow, pyspark, flask, findspark

### docker-compose.yaml

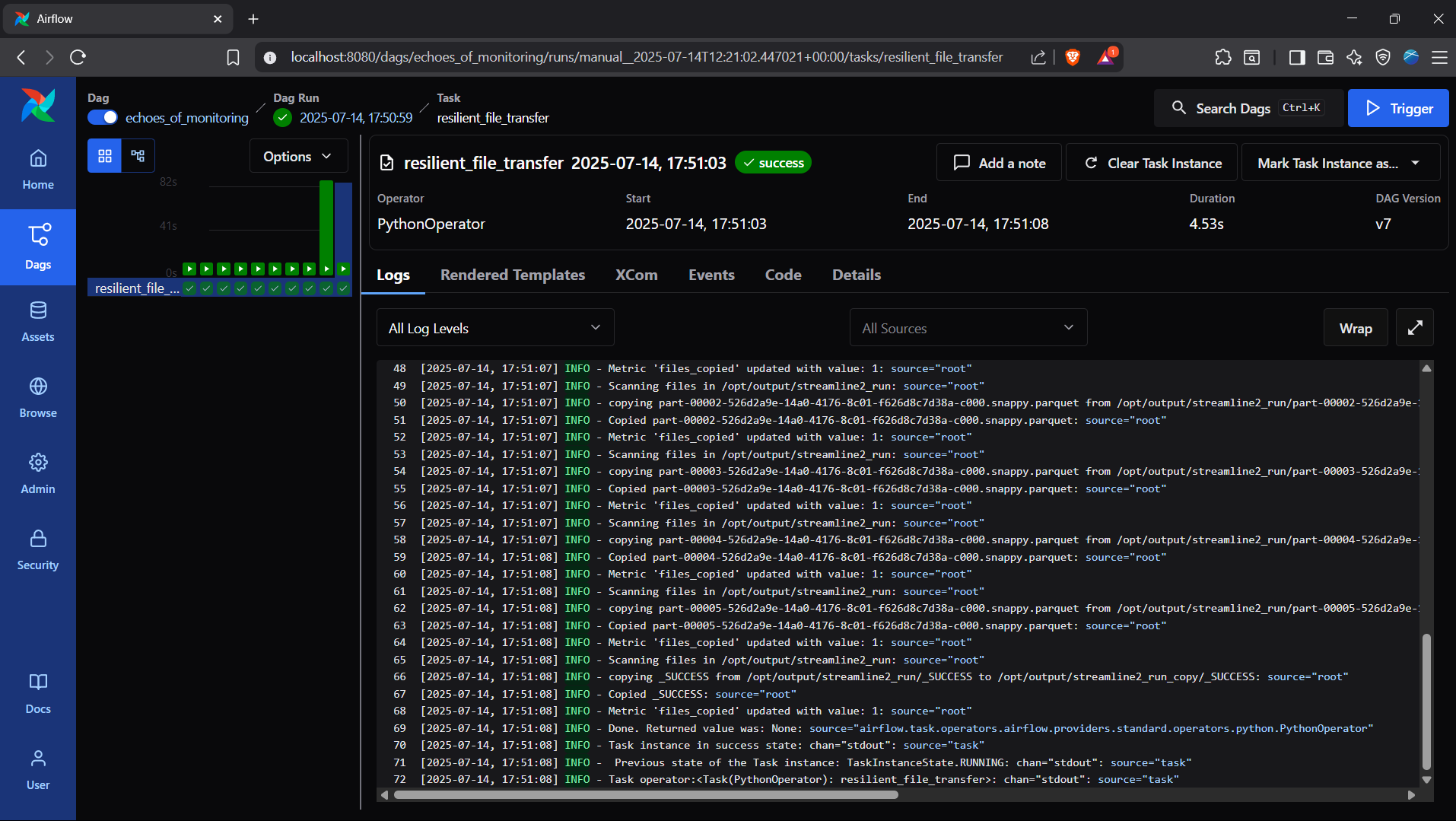
* Mounts:
  + Input: C:/usr/arind/downloads
  + Output: C:/parquet\_dir
* External network: monitoring\_net
* Separate service for Prometheus and Grafana

*UML Diagram:-*

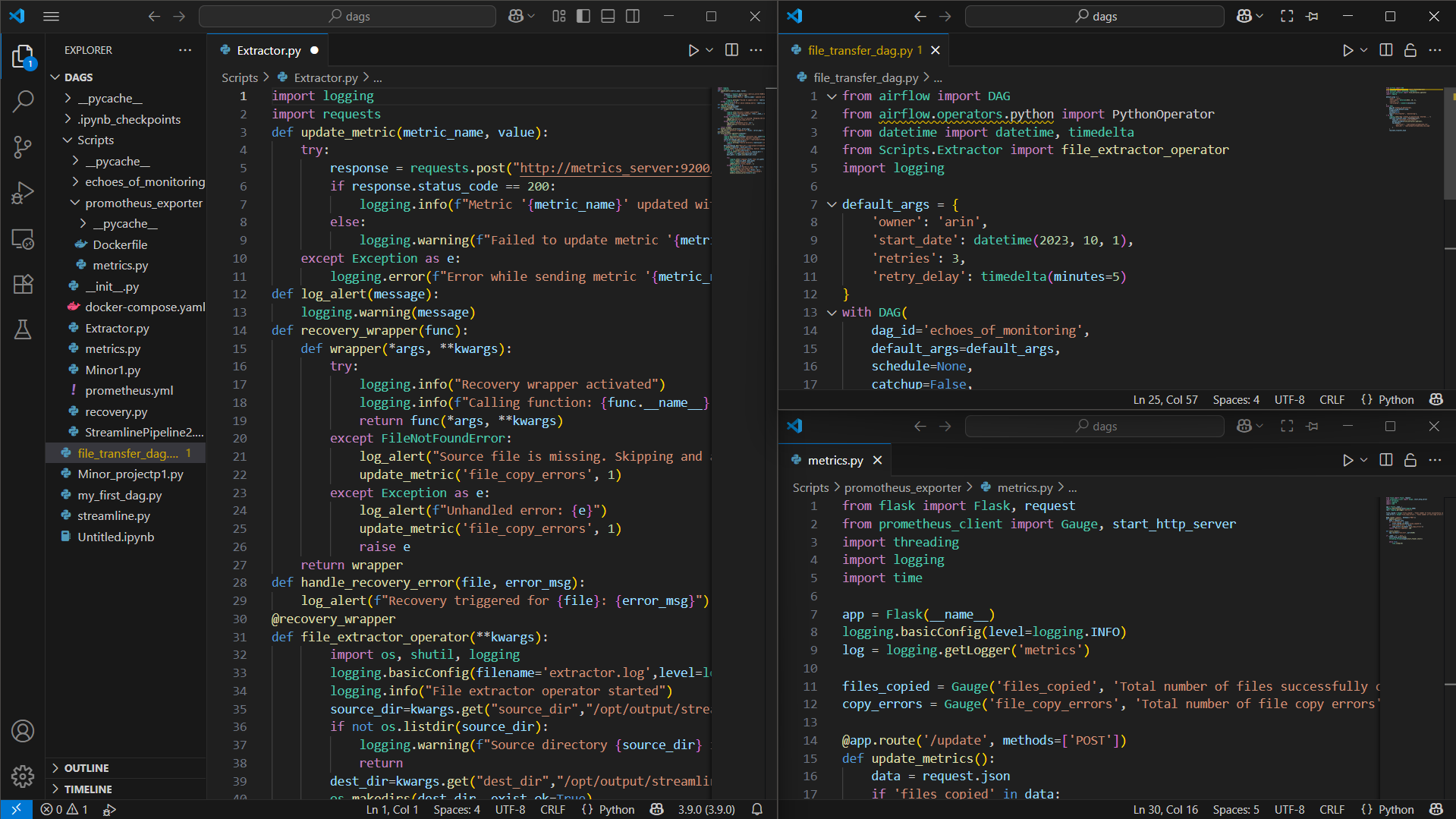


## 🎯 Achievements

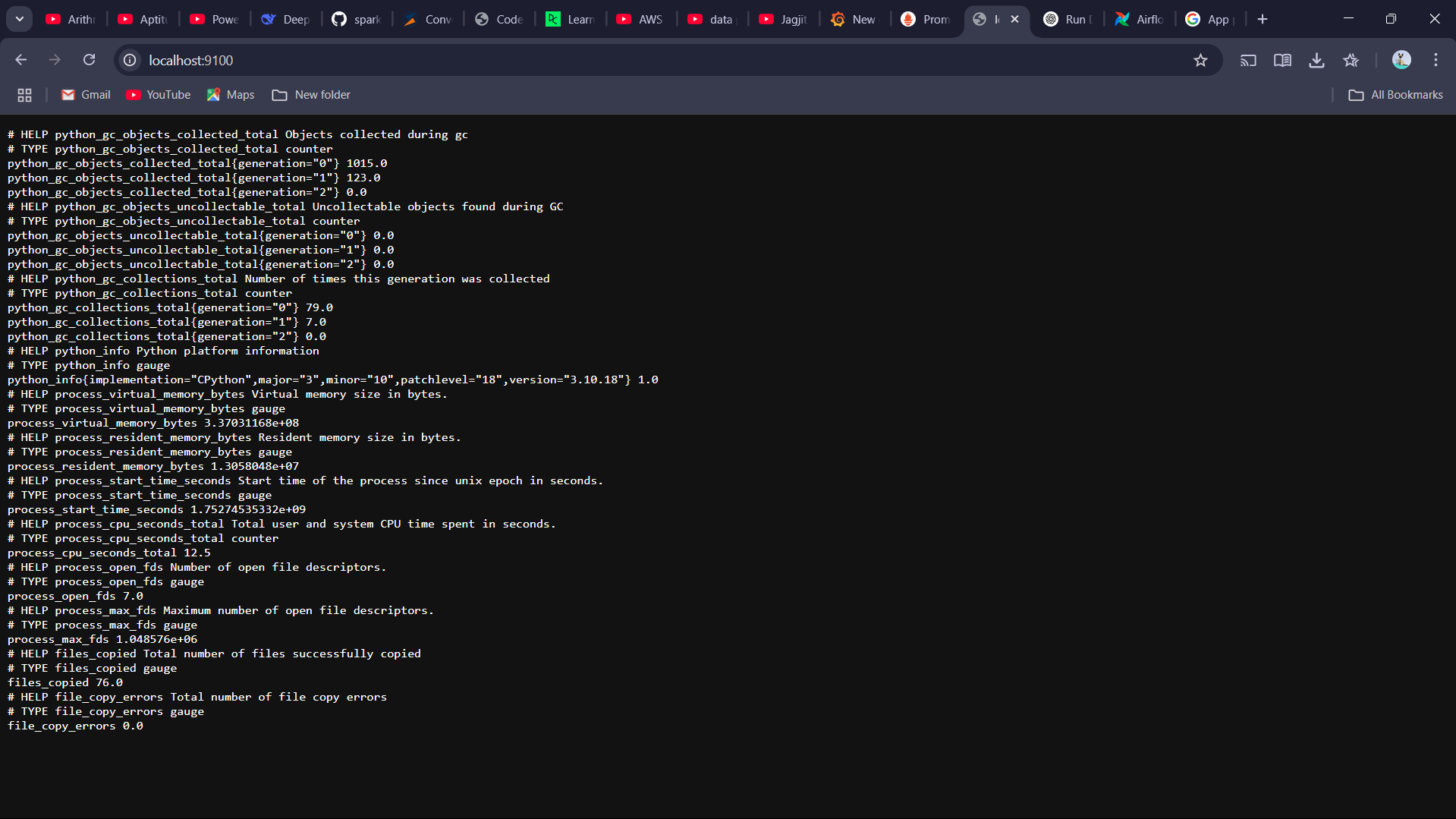
* ✅ Successfully containerized orchestration and processing workflows



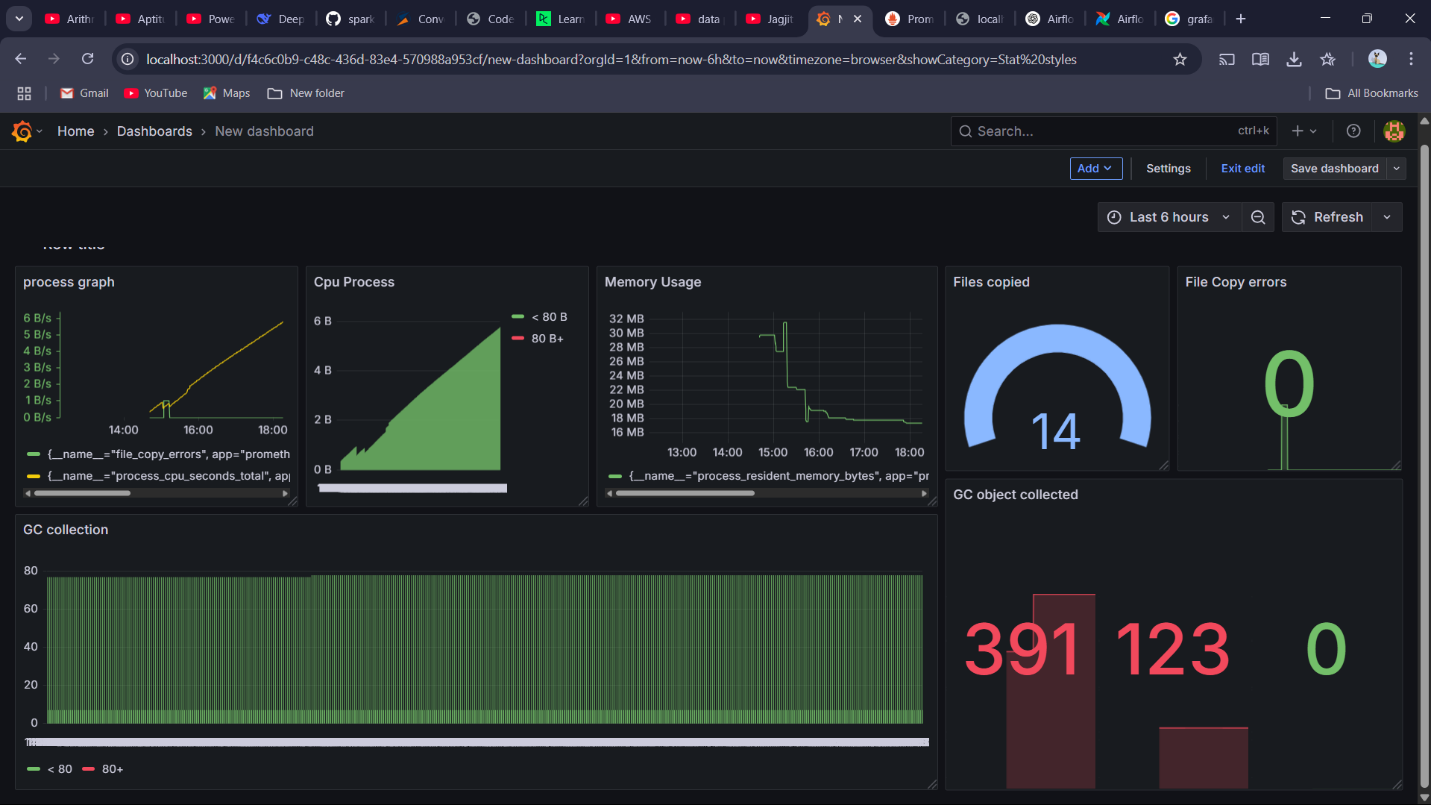
* ✅ Built resilient file ingestion and transformation logic



* ✅ Enabled monitoring with Prometheus and Grafana

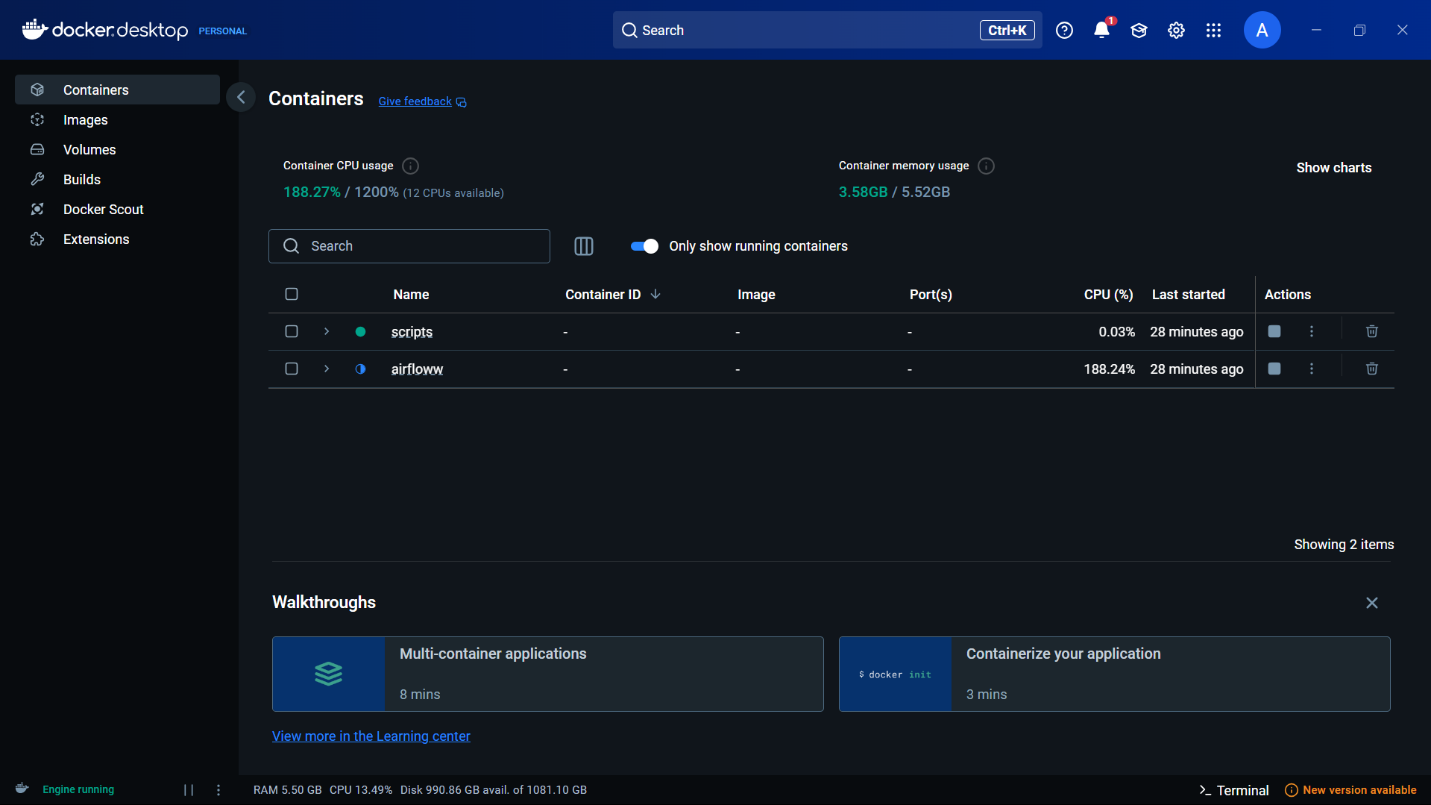


Prometheus



Grafana

* ✅ Verified communication and metric scraping over Docker bridge network



## 🚨 Potential Extensions

* ⏰ Prometheus Alert manager integration
* 🚫 Kafka-based log stream ingestion
* 📊 Historical DAG performance analysis