iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

CLOUD-BASED DATA PIPELINE USING PYSPARK, AIRFLOW, AND SUPERSET

Botta Kusuma Suvarna Lakshmia[1],
Lanka Yamini Swathi[2],
[1]Student, Department of Information Technology & Computer Applications,
[2]Assistant Professor, Department of Computer Science & Systems Engineering, Andhra University College of Engineering(A), Andhra University, Visakhapatnam, Andhra Pradesh – 530003

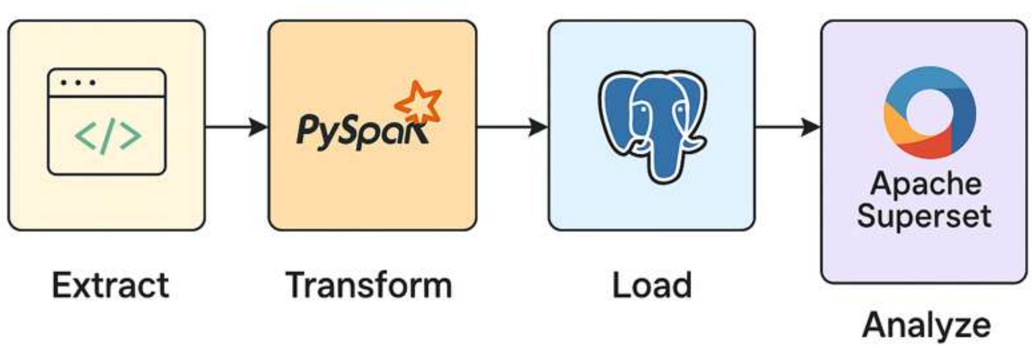
ABSTRACT
In today's data-driven ecosystem, organizations require scalable and automated solutions to extract, transform, and load (ETL) data from heterogeneous sources. This paper presents a cloud-based ETL pipeline that leverages PySpark for distributed transformation, Apache Airflow for workflow orchestration, and Apache Superset for data visualization. The pipeline ingests data from RESTful APIs and cloud storage, applies data cleaning, normalization, enrichment, and snapshot tracking, and stores the processed data in a PostgreSQL instance hosted on Google Cloud Platform (GCP). Automation of these steps enhances data consistency, reliability, and operational scalability. The system serves both academic and industrial applications where real-time reporting and decision-making are essential.

Keywords
ETL Pipeline, PySpark, Apache Airflow, Apache Superset, REST API, Google Cloud Platform, Data Engineering, Docker, PostgreSQL, Data Visualization

I. INTRODUCTION

With the exponential rise in data volume and velocity, modern organizations need robust, scalable pipelines to ingest, process, and analyze data in near real time. Traditional ETL solutions often involve manual interventions, rigid workflows, and limited scalability. To address these challenges, cloud-native and containerized ETL architectures have emerged as a promising alternative.

This project proposes a modular ETL pipeline that automates the ingestion of data from REST APIs, transformation using PySpark, and orchestration via Apache Airflow. Processed data is loaded into PostgreSQL, while Superset dashboards provide real-time insights. The entire stack is containerized using Docker and deployed on Google Cloud Platform (GCP), ensuring portability, scalability, and reliability.



II. LITERATURE REVIEW

iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

Several studies and industry solutions have highlighted the importance of automating the data pipeline lifecycle. Apache Spark is a widely adopted framework for distributed data transformation due to its speed and scalability. Apache Airflow has become a standard for managing complex ETL workflows through DAGs (Directed Acyclic Graphs) and operator-based task execution.
Apache Superset, though relatively new, offers a lightweight yet powerful alternative for business intelligence and real-time dashboarding. PostgreSQL is recognized for its open-source reliability and support for structured data analytics. Google Cloud services such as Cloud Storage and Cloud SQL provide infrastructure scalability and security for modern data operations.
This project builds upon these technologies to design and implement an end-to-end ETL solution with automation, reliability, and extensibility at its core.

3. SYSTEM ARCHITECTURE AND METHODOLOGY

The proposed ETL pipeline architecture is designed to automate the end-to-end data lifecycle — from extraction to reporting — using modular, scalable, and cloud-native tools. This methodology is broken down into distinct phases:

1. Data Extraction Layer (Ingestion)

Data is extracted from multiple sources via REST APIs for the following entities:

- **Suppliers**
- **Products**
- **Customers**
- **Sales (CSV files from GCS)**

The extraction is handled using a Python-based Extractor utility that sends HTTP GET requests to the endpoints (e.g., /v1/products, /v1/suppliers). The sales data is stored in Google Cloud Storage (GCS) and loaded using Spark's CSV reader.

2. Data Transformation Layer (Processing with PySpark)

Once extracted, the raw JSON or CSV data is converted into Spark DataFrames. The following transformations are applied:

- **Normalization of Column Names**: Trim spaces, convert to uppercase, replace spaces with underscores.
- **Data Cleaning**: Drop irrelevant columns, handle missing values if any.
- **Deduplication**: A custom Duplicate_check utility checks for primary key duplications (e.g., PRODUCT_ID, CUSTOMER_ID).
- **Snapshot Tracking**: A new column DAY_DT is added using current_date() to capture daily load dates.

The transformations are modular, ensuring reusability and consistency across all datasets.

3. Orchestration Layer (Apache Airflow)

Apache Airflow is used to orchestrate ETL workflows as Directed Acyclic Graphs (DAGs). Each DAG is responsible for one dataset (e.g., m_ingest_data_into_products), with tasks defined using @task decorators:

- Triggered manually or scheduled (e.g., daily)
- Handles retries and logging
- Logs progress and failures for monitoring

This layer ensures modular, fault-tolerant execution of each stage.

4. Storage Layer (PostgreSQL on GCP)

Processed data is written to:

- **raw.*_pre tables** – for pre-cleaned data
- **legacy.* tables** – for daily snapshots with DAY_DT included

Tables are maintained in PostgreSQL databases hosted on **Google Cloud Platform**. Two schemas are used:

- **Raw schema** for operational analysis

iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

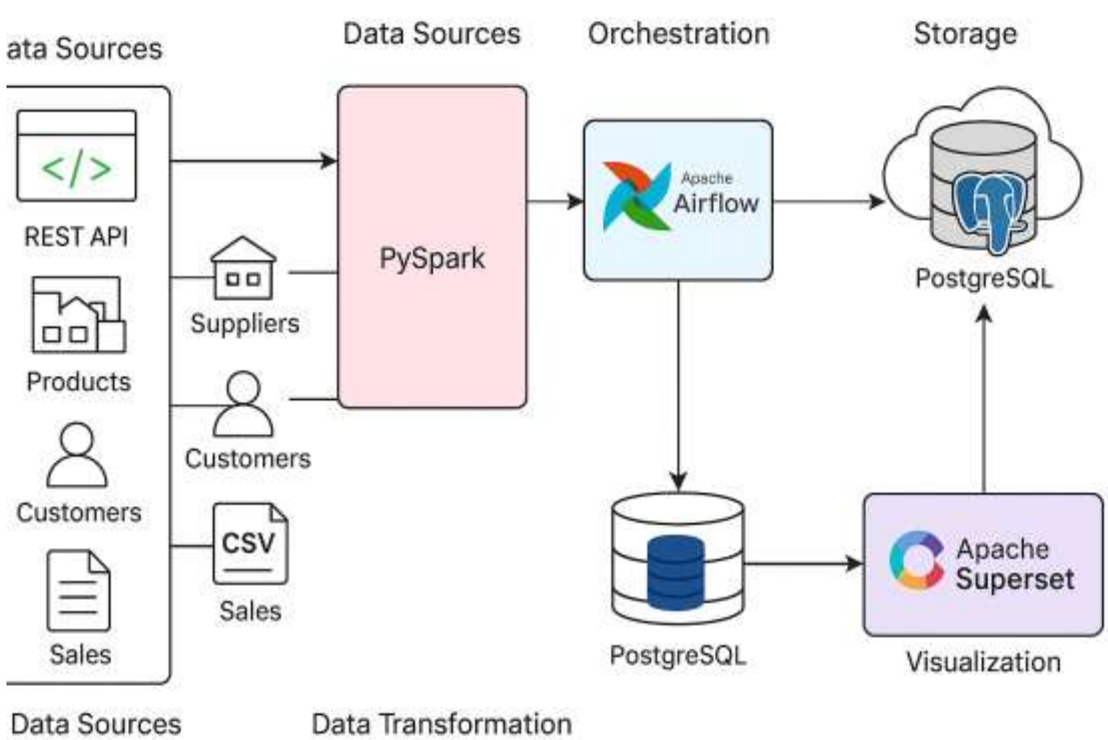
- **Legacy schema** for historical reporting

5. Visualization Layer (Apache Superset)
Apache Superset connects directly to the PostgreSQL database and is used to build:
- Dashboards for customer performance, product categories, sales trends
- KPI-based tiles such as TOP_PERFORMER, LOYALTY_TIER, etc.

This enables non-technical stakeholders to consume insights visually without accessing the underlying data.

6. Containerization and Deployment (Docker + GCP)
Each component (Airflow, PySpark, PostgreSQL, Superset) is containerized using Docker. The full pipeline is deployed on:
- **GCP Compute Engine** or **Cloud Run**
- GCS is used for storage (CSV ingestion)
- PostgreSQL is deployed on **Cloud SQL**

This enables seamless scalability and portability of the entire data stack.



4. IMPLEMENTATION

The proposed solution integrates various open-source tools within a containerized architecture to establish a robust, modular, and scalable ETL pipeline. The implementation focuses on seamless data ingestion, transformation, validation, and reporting across different components.

1. Environment Setup Using Docker
The entire ETL ecosystem is containerized using **Docker**, enabling easy deployment and environment reproducibility. The following services are containerized and orchestrated:
- **Apache Spark**: For distributed data transformation using PySpark.
- **Apache Airflow**: For orchestrating DAGs (Directed Acyclic Graphs) and task scheduling.

iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

- **PostgreSQL**: As the target data warehouse for storing structured data in raw and legacy schemas.
- **Apache Superset**: For data visualization and dashboarding.

Docker Compose is used to define multi-container applications, allowing services to communicate via networks and share volumes.

2. DAG Design with Airflow

Airflow DAGs are defined in Python, with each DAG representing a dataset pipeline (e.g., suppliers, products, customers, sales). Tasks are decorated using @task for modularity and include the following steps:

- **Data Extraction**
  Data is pulled from RESTful APIs (/v1/suppliers, /v1/products, etc.) using an Extractor utility. Sales data is read from Google Cloud Storage (GCS) in CSV format.
- **Column Normalization**
  A reusable normalize_column_names() function standardizes all Data Frame column names by:
  - Trimming whitespace
  - Converting to uppercase
  - Replacing spaces with underscores
    This ensures schema consistency across all pipelines.
- **Data Transformation & Enrichment**
  PySpark applies the following:
  - Selects necessary fields
  - Adds calculated columns if required
  - Enriches data (e.g., by joining customer/product details for reporting)
- **Snapshot Tracking**
  Each record is stamped with a DAY_DT column using current_date() to capture the load date. This allows time-series analysis and historical tracking.
- **Duplicate Checks**
  A utility class Duplicate_check ensures primary key uniqueness (e.g., PRODUCT_ID, CUSTOMER_ID, etc.) by detecting duplicates and logging alerts for manual or automatic resolution.
- **Data Loading**
  Transformed data is loaded into two PostgreSQL schemas:
  - raw.*_pre – stores cleaned but untracked data
  - legacy.* – stores time-stamped data with DAY_DT for audit and reporting

Data is loaded using a load_to_postgres() helper, which abstracts the JDBC connection and write modes (overwrite for raw, append for legacy).

3. Reporting and Visualization with Superset

Apache Superset connects to the PostgreSQL backend using SQLAlchemy. Dashboards are created for business and performance insights such as:

- Top-selling products
- Customer loyalty tiers
- Monthly/Yearly sales performance
- Regional trends for suppliers

Superset supports drag-and-drop charting, filters, and auto-refresh capabilities, making the reporting layer dynamic and interactive.

4. Cloud Integration

The system is designed to run on **Google Cloud Platform (GCP)** using the following services:

- **GCS** (Google Cloud Storage) to host sales data files
- **Cloud SQL** for PostgreSQL hosting
- **GCE** (Compute Engine) or **Cloud Run** to deploy containerized services if scaling is required

iJETRM
International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

The use of cloud storage allows automation of daily ingestion without manual file handling.

5. RESULTS AND DISCUSSION

The developed ETL pipeline effectively demonstrates the capability of automating and scaling data workflows in a cloud-based environment. Key observations include:

- ✅ **Efficient Batch Processing**: The pipeline handles daily batch loads of suppliers, products, customers, and sales data with consistency using PySpark transformations.
- 📊 **Interactive Visualization**: Superset dashboards support filtering by city, category, customer, and time periods, enabling deep data-driven insights.
- 📌 **Historical Data Tracking**: The DAY_DT field added to every load enables time-series comparisons and change tracking across snapshots.
- 🔁 **Reduced Manual Intervention**: Apache Airflow automates job scheduling, retries on failure, and logging—resulting in a highly maintainable solution.
- 🔍 **Data Quality Checks**: Duplicate primary key detection ensures accuracy and prevents downstream data corruption.
- 💻 **Modular Design**: Each DAG is independently deployable, promoting better maintainability and extensibility.
- ☁️ **Cloud Native Benefits**: Integration with GCS and PostgreSQL in the cloud allows seamless scaling and centralized storage.
- 📦 **Portability with Docker**: The containerized setup allows the entire pipeline to be deployed on any system supporting Docker.

VI. CONCLUSION AND FUTURE WORK

This paper presents a robust, modular, and scalable cloud-based ETL pipeline integrating open-source technologies like PySpark, Apache Airflow, Superset, and PostgreSQL. The architecture simplifies data ingestion, processing, validation, and visualization while adhering to best practices in data engineering.

✅ **Conclusion:**

- The pipeline successfully demonstrates the automation of multi-source data extraction and transformation.
- Apache Airflow ensures reliable orchestration of all data processing steps.
- The solution provides business users with real-time and historical visibility through Superset dashboards.
- Docker enables cross-platform deployment, making the system production-ready and scalable.

🔮 **Future Work:**

- **Real-Time Streaming Support**: Integrate Apache Kafka or Spark Streaming to handle real-time data.
- **Incremental Loading**: Implement change data capture (CDC) mechanisms to reduce data volume in daily loads.
- **Cloud-Native Storage**: Expand beyond PostgreSQL to include warehouses like BigQuery, Snowflake, or Redshift.
- **Data Validation Framework**: Add rule-based data quality validation before ingestion.
- **Monitoring Dashboards**: Implement Airflow and Superset health dashboards for operations teams.
- **Security Enhancements**: Add role-based access control (RBAC) in Superset and database encryption.

VI. CONCLUSION AND FUTURE WORK

International Journal of Engineering Technology Research & Management
(IJETRM)
https://ijetrm.com/

This paper presents a scalable, modular, and automated cloud-based ETL pipeline using PySpark, Apache Airflow, and Apache Superset. By containerizing the data engineering stack and orchestrating workflows via Airflow, the system ensures reliability, reproducibility, and maintainability. The use of Superset for data visualization enhances decision-making by providing real-time insights across sales, customer engagement, and product performance.

The key benefits achieved by the solution include:

- End-to-end automation from data ingestion to reporting
- Modular DAG structure with minimal coupling
- Historical data tracking with DAY_DT snapshot columns
- Real-time dashboards through direct PostgreSQL-Superset integration
- Seamless deployment using Docker and cloud-native services

Future Work:

To extend the capabilities of the system, the following improvements are planned:

- Integrating **real-time data streaming** using Kafka or Spark Structured Streaming.
- Enabling **incremental loading** through change data capture (CDC).
- Migrating to **cloud-native data warehouses** such as Google BigQuery or Snowflake for large-scale analytics.
- Adding **automated data validation frameworks** for quality assurance.
- Implementing **role-based access control (RBAC)** in Superset for enterprise security.

REFERENCES

1) Apache Spark, "Lightning-Fast Unified Analytics Engine," [Online]. Available: https://spark.apache.org
2) Apache Airflow, "Workflow Orchestration Platform," [Online]. Available: https://airflow.apache.org
3) Apache Superset, "Modern Data Exploration and Visualization," [Online]. Available: https://superset.apache.org
4) PostgreSQL, "The World's Most Advanced Open Source Relational Database," [Online]. Available: https://www.postgresql.org
5) Google Cloud Platform (GCP), "Cloud Computing Services," [Online]. Available: https://cloud.google.com