PriceHubble Data Pipeline Design

|  |  |
| --- | --- |
| Title | Data Pipeline Design |
| Submission date | 27-10-2023 |
| Prepared By | Shweta Shivamogga Dattatri |
| For | PriceHubble |
| Version | v.1 |

Contents

[Introduction: 3](#_Toc149307052)

[Architecture diagram 3](#_Toc149307053)

[Question1: Designing an ETL pipeline to generate the output data set 5](#_Toc149307054)

[Question2: Tool Selection Overview for data Pipeline 6](#_Toc149307055)

[Question3: Optimal Data Storage Format for Advanced Data Pipelines 7](#_Toc149307056)

[Question4: Scalability Enhancements for High-Volume data Pipelines 8](#_Toc149307057)

[Question5: Handling Unstructured User Data in Pipelines 9](#_Toc149307058)

# Introduction:

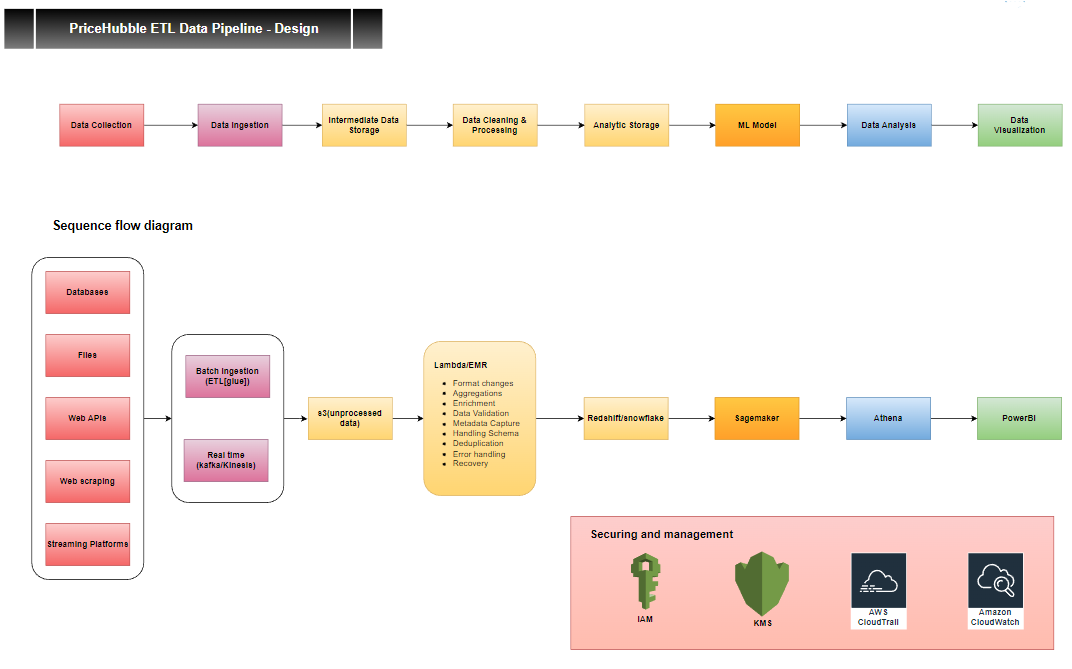
This documentation presents an in-depth look at the design, development, and deployment of an advanced data pipeline tailored for processing property-related data. The project addresses the challenges of dealing with large-scale data influx, varying data sources, and data quality inconsistencies. With a robust architectural foundation and well-thought-out design decisions, the pipeline is poised to deliver reliable and insightful results.

# Architecture diagram

**Objective:** Our goal is to build a flexible and robust data processing system that can handle both real-time and batch data, ensuring its quality and making it accessible for analysis and downstream applications.

**Components:** Our architecture revolves around a strong ETL process, using Kafka for instant data intake, AWS Glue for batch tasks, and S3 as our scalable data lake.

***Design document Link:*** *ETL Data Pipeline – Design document ??*



Architecture Overview

1. Data Ingestion

- Gathers data from diverse sources into a central storage.

* DatabaseConnector: Links and pulls from databases.
* APIFetcher: Retrieves data from web services.
* FileLoader: Imports from files and cloud.

2. Intermediate Storage (Data Lake)

Central, scalable repository for unprocessed data.

* DataLake: Our storage blueprint.
* StorageManager: Manages data activities.

3. Data Transformation

Processes and refines raw data.

* DataValidator: Ensures data meets our standards.
* DataCleaner: Tidies up and preps data.
* DataTransformer: Modifies and enriches data.
* ETLJobScheduler: Oversees ETL tasks.

4. Analytic Storage (Data Warehouse)

Designed for analysis and ML readiness.

* DataWarehouse: Our optimized storage layout.
* QueryEngine: Fast data retrieval tool.

5. Feature Store

Offers consistent data features for ML tasks.

* FeatureManager: Manages feature activities.
* FeatureExtractor: Pulls and forwards features for storage.

6. ML Models

Uses features for predictions.

* ModelTrainer: Prepares models with data.
* ModelInferenceEngine: Delivers model predictions.
* ModelRegistry: Keeps track of model versions.

7. Visualization & Analysis

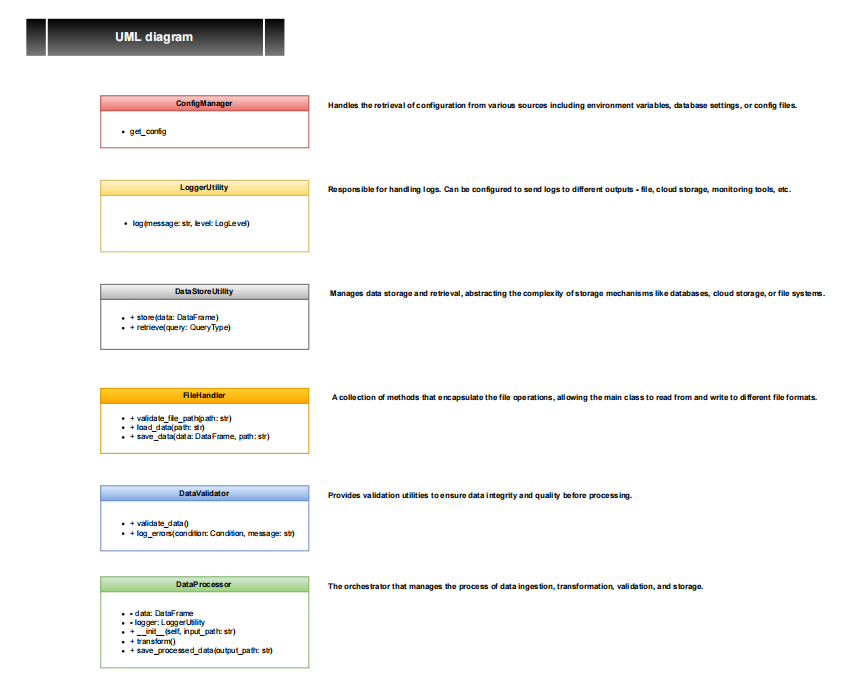
Tools to interpret results and inspect data quality.

* DashboardBuilder: Crafts visual insights.
* ReportGenerator: Summarizes findings or ML outcomes.

# Question1: Designing an ETL pipeline to generate the output data set

Code Link: <https://github.com/ShwetaProTest/pricehubble_poc>

UML diagram :



*UML document link ??*

***Attached is the detailed breakdown of UML:***

# Question2: Tool Selection Overview for data Pipeline

To build a robust, scalable, and efficient data pipeline, carefully select the tools and technologies that meet our current and future needs, considering performance, scalability, integration capability, and cost-effectiveness.

Attached is a detailed breakdown of the tools and technologies chosen for each stage of the data pipeline, along with a justification for each choice over popular alternatives.

*Tool Selection Overview for data pipeline??*

# Question3: Optimal Data Storage Format for Advanced Data Pipelines

Considering all the downstream jobs and various stakeholders who might be data analysts, data scientists, or business intelligence professionals, the Parquet format is optimal. They might be using SQL-based tools, big data platforms, or data science platforms, and Parquet plays nicely with all these.

However, for final delivery to some stakeholders or for certain applications, might convert specific subsets of this data into more appropriate formats. For instance, business users might prefer Excel or CSV for smaller reports, but for large-scale storage and analytics, Parquet is the choice.

Attached is a detailed tabular breakdown:

*Data Storage Format*

# Question4: Scalability Enhancements for High-Volume data Pipelines

Adapting the data pipeline for the influx of billions of records on a regular basis necessitated refining tool choices. The evolution in tools emphasizes scalability, efficiency, data integrity, and versatility. By comparing the initial tools with the revised set, it's evident that the goal was to ensure the pipeline remains robust, cost-efficient, and capable of handling vast datasets without compromising on performance or security.

Attached is a detailed tabular breakdown:

*Enhanced Scalable Solution*

# Question5: Handling Unstructured User Data in Pipelines

The architecture aims to transform, refine, and enrich the human-inputted and unstructured data. The added steps prioritize accuracy, privacy, user feedback, and data evolution tracking, ensuring not only data quality and usability but also compliance and continuous improvement.

By enhancing the previously built architecture with these steps, the pipeline will be better equipped to handle the challenges of unstructured and humanly inputted data, ensuring robustness, scalability, and data integrity.

Attached is a detailed tabular breakdown:

*Handling Unstructured User Data in Pipelines*