Technical Design Document for ETL Pipeline

# Overview

This document outlines the design for an ETL (Extract, Transform, Load) pipeline for K9 care data. The pipeline retrieves data from a remote GitHub source, processes it, and stores it in a PostgreSQL database. The design also includes a Docker-based setup for orchestration using Apache Airflow.

## Components

* **ETL Pipeline Script**
  + **Purpose**: The script performs the ETL process:
    - **Extract**: Pulls data from a remote URL.
    - **Transform**: Cleans and categorizes the data.
    - **Load**: Inserts or updates the data into a PostgreSQL database.
  + **Key Functions**:
    - clean\_text(text): Removes non-alphanumeric characters from the input text.
    - pull\_data(url): Retrieves data from the specified URL.
    - filter\_data(data): Cleans and categorizes data.
    - save\_data(data, conn): Inserts or updates data in PostgreSQL.
    - main\_pipeline(): Orchestrates the ETL process.
  + **Error Handling**:
    - Handles errors in data retrieval, processing, and database operations.
    - Logs errors and ensures that database operations are either committed or rolled back as necessary.
* **Docker Configuration**
  + **Purpose**: The Docker configuration defines the environment for running the ETL pipeline and managing dependencies using Apache Airflow.
  + **Services**:
    - **Postgres**:
      * **Image**: postgres:13
      * **Purpose**: Provides the database for storing processed data.
      * **Healthcheck**: Ensures that the database is ready to accept connections.
      * **Volumes**:
        + local\_postgres\_data: Stores persistent data.
        + ./init.sql: Initializes the database schema.
    - **Scheduler**:
      * **Image**: apache/airflow:2.9.3
      * **Purpose**: Runs Airflow's scheduler to manage task execution.
      * **Command**: scheduler
      * **Dependencies**: Depends on postgres and airflow-init.
    - **Webserver**:
      * **Image**: apache/airflow:2.9.3
      * **Purpose**: Provides the web interface for Apache Airflow.
      * **Command**: webserver
      * **Ports**: Exposes port 8080.
      * **Healthcheck**: Monitors the health of the webserver.
    - **Airflow Init**:
      * **Image**: apache/airflow:2.9.3
      * **Purpose**: Initializes Airflow directories and permissions.
      * **Command**: Sets up directories and verifies Airflow installation.
  + **Volumes**:
    - local\_postgres\_data: Stores PostgreSQL data persistently.
  + **Docker Compose Configuration**:
    - Defines service dependencies and environment configurations.
    - Uses .env file for sensitive information and environment variables.

## Workflow

1. **Extraction**:
   * The pull\_data function fetches data from the provided URL.
   * Logs the status of the extraction process.
2. **Transformation**:
   * The filter\_data function processes the data by cleaning text and categorizing it.
   * Logs the status of the transformation process.
3. **Loading**:
   * The save\_data function inserts or updates the processed data into PostgreSQL.
   * Uses psycopg2 for database interactions.
   * Logs errors and manages transactions to ensure data integrity.
4. **Orchestration**:
   * Apache Airflow is used for scheduling and managing the ETL tasks.
   * The Docker setup ensures that all necessary services are up and running, and dependencies are managed effectively.

### **Security and Configuration**

* **Environment Variables**:
  + Sensitive information (e.g., database credentials) is managed using .env files.
* **Database Initialization**:
  + An SQL initialization script (init.sql) is used to set up the database schema.

### **Summary**

This design ensures a robust and scalable ETL pipeline for K9 care data, leveraging Docker and Apache Airflow for efficient task management and orchestration. The system is built to handle errors gracefully and maintain data integrity throughout the ETL process.