ETL Weather Data Pipeline - Technical Documentation

1. Overview

This ETL pipeline extracts weather data from various sources (CSV, JSON, Google Sheets, REST API), transforms the data into a unified schema with temperature, humidity, wind speed, and timestamp, and loads it into a MongoDB collection. The pipeline ensures no duplicate data is inserted by comparing the latest timestamp already in the database.

2. Technologies Used

- Python
- pandas
- requests
- pymongo
- schedule
- MongoDB (Atlas or local)
- GitHub Actions for CI/CD

3. ETL Process

The ETL process consists of:

Extract:
- CSV files using pandas
- JSON files and APIs using requests
- Google Sheets (CSV format) - REST API (OpenWeatherMap)

Transform:
- Temperature unit conversion (F to C)
- Timestamp normalization to UTC ISO format
- Weather impact score calculation based on temp, humidity, and wind

Load:

- Insert only records newer than the latest timestamp in MongoDB to prevent duplicates.

4. Configuration

The configuration file (config/db_config.json) stores MongoDB credentials and collection info. This allows secure and centralized configuration management.

5. Scheduling

Using the `schedule` module, the ETL runs daily at a defined time (e.g., 2:00 AM). The scheduler script continuously checks and triggers the ETL job.

6. CI/CD Integration

GitHub Actions runs the ETL pipeline automatically on push and pull requests. It installs dependencies, validates Python code, and runs the ETL script to ensure data freshness.

7. MongoDB Schema

Each document in MongoDB follows this structure:

```
{
  'source': 'CSV',
  'timestamp': ISODate('...'),
  'location': 'City Name',
  'temperature_c': float,
  'humidity': int,
  'wind_speed': float,
  'weather_score': float
}
```

ETL Weather Data Pipeline - Technical Documentation

8. Duplicate Prevention Logic

The load function checks the greatest timestamp in MongoDB and only inserts records with a newer timestamp.

This prevents inserting previously stored records and avoids redundancy.

9. How to Run

- Manual: `python etl_pipeline.py`
- Scheduled: `python scheduler.py`
- CI/CD: via GitHub Actions on push

10. Task Remaining

- CI/CD Pipeline (remaining due to time passed out)
- Open API Code implemented but not tested asking for key.