Prefect Project: Automated Data Pipeline (3-Hour Project)

Problem Statement

Create an automated data pipeline using Prefect (open-source) that:
1. Fetches real-time weather data from a public API for a list of cities.
2. Cleans and formats the data.
3. Calculates average temperature and humidity across the cities.
4. Stores the final summarized data to a local CSV file.
5. Uses task dependencies, retries, logging, and caching.

Step-by-Step Solution

Step 1: Project Setup (30 min)

Install required libraries:
pip install prefect pandas requests

Create the following folder structure:
weather_pipeline_project/
├── weather_pipeline.py
├── cities.txt
└── output/

Step 2: Define Tasks (1 hr)

Import necessary libraries, define individual tasks such as:
- load_cities: Reads city names from a text file
- fetch_weather: Fetches weather data from a public API (with retries)
- process_weather_data: Cleans the raw data
- calculate_summary: Computes average temperature and humidity (with caching)
- save_summary: Saves summary as a CSV file

Step 3: Build and Register the Flow (1 hr)

Use Prefect's Flow context manager to build your flow explicitly:

```
with Flow('weather-data-pipeline') as flow:
    cities = load_cities('cities.txt')
    weather_data = fetch_weather.map(cities)
    processed_df = process_weather_data(weather_data)
    summary = calculate_summary(processed_df)
    save_summary(summary, 'output/weather_summary.csv')
```

Step 4: Explore Enhancements (Optional - 1 hr)

Advanced options to try:
- Store raw weather data per city in individual files
- Add scheduling to the flow
- Add unit conversion as a parameter
- Deploy to Prefect Cloud (free tier)