

Ad Pipeline: Project Setup Guide

This document provides step-by-step instructions to set up the Ad Data Pipeline environment, configure credentials, and run the system.

0. Account Setup

This project requires **Supabase** (Database) and **Confluent Cloud** (Kafka). Use your GitHub account for single sign-on.

A. Supabase (The Database)

1. Go to supabase.com.
2. Click "Start your project".
3. **IMPORTANT:** Click "Continue with GitHub". Authorize it.
4. Click "New Project"
 - **Organization:** Select your default org (created automatically).
 - **Name:** ad-pipeline-db (or similar).
 - **Database Password:** **WRITE THIS DOWN!** You cannot see it again. (e.g., MySecurePass123!).
 - **Region:** Choose one close to you (e.g., Singapore or East US).
5. Click "Create new project" and wait ~2 minutes for it to setup.

B. Confluent Cloud (The Kafka Stream)

1. Go to confluent.cloud.
2. Click "Login" or "Try Free".
3. **IMPORTANT:** Click "Continue with GitHub".
4. Create Cluster
 - Select "Basic" (Free tier).
 - Provider: **AWS**.
 - Region: Match your Supabase region if possible (e.g., Singapore).
 - Click "Launch Cluster".

1. Prerequisites

- Python 3.9+ installed (python --version)

- Git installed
- VS Code (Recommended)

2. Dependencies

Open your terminal in the project folder and install the required libraries:

pip **install** -r requirements.txt

Note: This installs pandas, sqlalchemy, streamlit, confluent-kafka, and toml.

3. Configuration (Secrets)

CRITICAL: We do not store passwords in the code. We use a secrets file.

A. How to get keys?

1. Database (Supabase)

- Go to [Supabase Dashboard](#).
- Select your project -> **Project Settings** -> **Database**.
- Host/User/Port are under "Connection parameters". Password is what you set when creating the project.

2. Kafka (Confluent Cloud)

- Go to [Confluent Cloud](#).
- **Bootstrap Server**
: Go to
Cluster Settings -> Endpoints
.
 - *Format Look-alike:* pkc-12345.region.provider.confluent.cloud:9092
 - **Action:** Copy the entire address including the port (:9092).
- **API Keys**
: Go to
"API Keys"
tab -> Create Key ->
Global Access
.
 - **Sasl Username** = The "Key" (e.g., K7...)
 - **Sasl Password** = The "Secret" (long string)

B. Setup

1. Create a folder named secrets in the root directory.
2. Create a file named secrets.toml inside it (secrets/secrets.toml).
3. Paste the following content (replace with your actual keys):

```
[kafka]
bootstrap_servers = "pkc-312o0.ap-southeast-1.aws.confluent.cloud:9092"
sasl_username = "YOUR_KAFKA_API_KEY"
sasl_password = "YOUR_KAFKA_API_SECRET"
```

```
[database]
user = "postgres"
password = "YOUR_DB_PASSWORD"
host = "db.your-project.supabase.co"
port = "5432"
dbname = "postgres"
```

4. Running the Pipeline

You can run the pipeline components individually or via the Dashboard.

Step 0: Initial Setup (Run Once)

Before running the pipeline for the first time, you must create the database tables.

```
python apply_schema.py
```

Output: "Schema applied successfully."

Option A: The Dashboard (Recommended)

Launch the UI:

```
streamlit run app.py
```

- Navigate to "Upload & ETL".
- Click "Run ELT Pipeline".
- Watch the progress bars.

Option B: Manual Scripts

Run these in order:

1. **Ingestion:** `python ingestion.py` (Loads CSVs)
2. **Standardization:** `python standardization.py` (Cleans & Quarantines)
3. **Modeling:** `python etl_modeling.py` (Updates Star Schema)
4. **Verification:** `python verify_pipeline.py` (Checks logic)

5. Defense Demo Tips

- **Show the Secrets:** Open secrets/secrets.toml (briefly!) to prove you aren't hardcoding passwords.
- **Show the Utils:** Open utils.py to show how you load them safely.
- **Run Verification:** detailed checks in verify_pipeline.py prove the data is correct.

6. Troubleshooting

- **"Module not found":** Did you run pip install -r requirements.txt?
 - **"Connection refused":** Check your internet. The DB and Kafka are in the Cloud.
 - **"KeyError: 'database'":** Your secrets.toml is missing the [database] section.
-

7. Project Structure (File Map)

Understanding what each file does:

Core Pipeline

- **ingestion.py:** Reads raw CSVs from data/raw and loads them into the Staging table.
- **standardization.py:** The main logic. Maps columns, handles currency conversion, applying quarantine logic.
- **etl_modeling.py:** Transforms clean data into the Star Schema (Fact/Dimensions).
- **app.py:** The Streamlit Dashboard. Visualizes the data and controls the pipeline.
- **kafka_producer.py:** Simulates real-time event streaming to Confluent Cloud.

Setup & Helpers

- **utils.py:** securely loads credentials from secrets/secrets.toml. Used by all scripts.
- **apply_schema.py:** Runs schema.sql to create empty tables in the database.
- **migrate_video_views.py:** (One-off) Adds the video_views column to the DB.
- **verify_pipeline.py:** Runs a battery of tests to prove data integrity.

Config

- **secrets/secrets.toml:** Your private passwords. **NEVER share this file.**
- **requirements.txt:** List of Python libraries needed to run the project.