Project 5 - ETL with Airflow

Goals and Expected Output:

- Can read file from Gdrive
- Can build one pipeline ETL in Airflow
- Can use Operator in Airflow

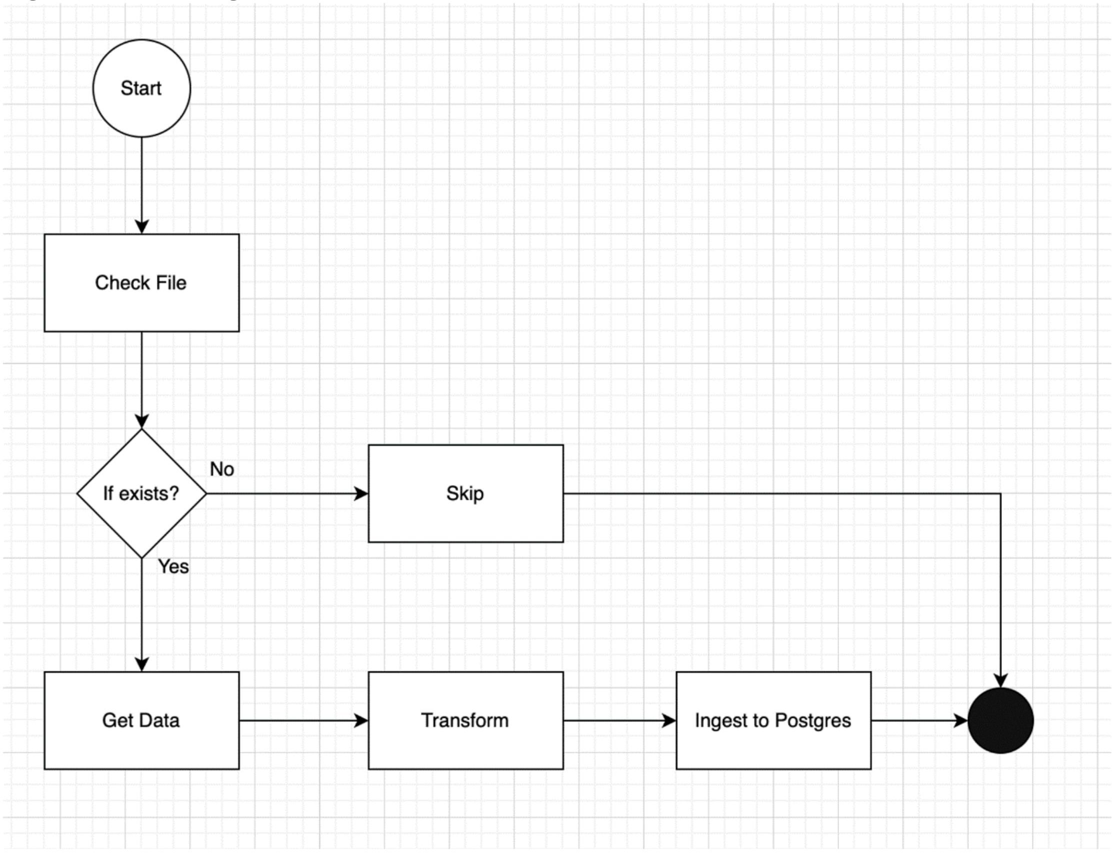
Requirement tools:

- Docker, if you want to install Airflow in local device (Optional)
  - Tutor will provide airflow-server if your local device doesn't support
- SSH to Server Airflow
  - can try using https://mobaxterm.mobatek.net/
- Python > 3.7
- VSCode
- Dbeaver

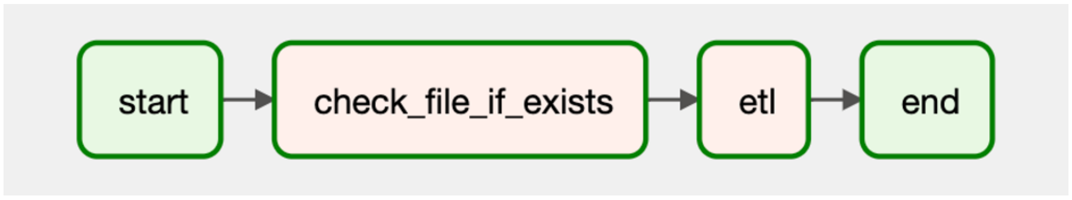
Use Case Flow:

We will build 2 task for this project

1. Check if file exists, dataset
2. Ingest data to Postgres



Flow in DAG:



Install docker on Local (Optional)

- Git clone airflow-docker
- add file `.env`, copy and paste:

```
JavaScript
# Meta-Database
POSTGRES_USER=airflow
POSTGRES_PASSWORD=airflow
POSTGRES_DB=airflow

# Airflow Core
AIRFLOW__CORE__FERNET_KEY=UKMzEm3yIuFYEq1y3-2FxPNWSVwRASpahmQ9kQfEr8E=
AIRFLOW__CORE__EXECUTOR=LocalExecutor
AIRFLOW__CORE__DAGS_ARE_PAUSED_AT_CREATION=True
AIRFLOW__CORE__LOAD_EXAMPLES=False
AIRFLOW_UID=0

# Backend DB
AIRFLOW__DATABASE__SQL_ALCHEMY_CONN=postgresql+psycopg2://airflow:airflow@postgres/airflow
AIRFLOW__DATABASE__LOAD_DEFAULT_CONNECTIONS=False

# Airflow Init
_AIRFLOW_DB_UPGRADE=True
_AIRFLOW_WWW_USER_CREATE=True
_AIRFLOW_WWW_USER_USERNAME=airflow
_AIRFLOW_WWW_USER_PASSWORD=airflow

#  Api
AIRFLOW__API__AUTH_BACKEND=airflow.api.auth.backend.basic_auth
```

- run with `docker-compose up -d`

Fixed docker-compose airflow

```
YAML
version: '2.1'
services:
  postgres:
    image: postgres:13
    container_name: postgres
    ports:
      - "5434:5432"
    healthcheck:
      test: ["CMD", "pg_isready", "-U", "airflow"]
      interval: 5s
      retries: 5
    env_file:
      - .env
    volumes:
      - postgres_airflow:/var/lib/postgresql/data

  scheduler:
    image: apache/airflow:2.3.0
    user: "${AIRFLOW_UID}:0"
    env_file:
      - .env
    volumes:
      - ./dags:/opt/airflow/dags
      - ./logs:/opt/airflow/logs
      - ./plugins:/opt/airflow/plugins
      - /var/run/docker.sock:/var/run/docker.sock
    depends_on:
```

```yaml
      postgres:
        condition: service_healthy
      airflow-init:
        condition: service_started
    container_name: airflow-scheduler
    command: scheduler
    restart: on-failure
    ports:
      - "8793:8793"

  webserver:
    image: apache/airflow:2.3.0
    user: "${AIRFLOW_UID}:0"
    env_file:
      - .env
    volumes:
      - ./dags:/opt/airflow/dags
      - ./logs:/opt/airflow/logs
      - ./plugins:/opt/airflow/plugins
      - /var/run/docker.sock:/var/run/docker.sock
    depends_on:
      postgres:
        condition: service_healthy
      airflow-init:
        condition: service_started
    container_name: airflow-webserver
    restart: always
    command: webserver
    ports:
      - "8080:8080"
    healthcheck:
      test: ["CMD", "curl", "--fail", "http://localhost:8080/health"]
      interval: 30s
      timeout: 30s
      retries: 5

  airflow-init:
    image: apache/airflow:2.3.0
    user: "${AIRFLOW_UID}:0"
    env_file:
      - .env
    volumes:
      - ./dags:/opt/airflow/dags
      - ./logs:/opt/airflow/logs
      - ./plugins:/opt/airflow/plugins
      - /var/run/docker.sock:/var/run/docker.sock
    container_name: airflow-init
    entrypoint: /bin/bash
    command:
      - -c
      - |
        mkdir -p /sources/logs /sources/dags /sources/plugins
        chown -R "${AIRFLOW_UID}:0" /sources/{logs,dags,plugins}
        exec /entrypoint airflow version
volumes:
  postgres_airflow:
      external: true
```