# Übungsaufgabe: Zugriff auf Dateien im MinilO-Bucket

## Voraussetzungen

- Docker und Docker Compose sind installiert
- Die Airflow-Umgebung aus Übung 1 läuft
- Grundlegendes Verständnis von Python

## User Story 2: Zugriff auf Dateien im MinilO-Bucket

**Beschreibung:** Als Entwickler möchte ich einen DAG erstellen, der auf Dateien im MinilO-Bucket zugreifen kann, um die Daten aus dem Bucket in die Pipeline zu laden.

**Akzeptanzkriterien:**

- Airflow kann mit dem MinilO-Bucket kommunizieren und Dateien abrufen.
- Ein einfacher DAG lädt eine Testdatei aus MinilO herunter und speichert sie lokal zur weiteren Verarbeitung.

## Schritte zur Umsetzung:

1. **Vorbereitungen und Konfigurationen:** Erstellen Sie in der Minio Web-UI http://localhost:9001/access-keys einen Access Key und Secret Key und speichern Sie diese in einer Datei `minio_credentials.txt`. Falls noch nicht vorhanden, erstellen Sie einen Bucket mit dem Namen `testbucket` und laden Sie die Testdatei `sample_data.csv` hoch. In dieser Datei finden Sie Beispieldaten für einen Onlineshop.

   date,product_id,category,quantity,price,customer_id,region 2024-01-01,P001,Electronics,2,599.99,C101,North 2024-01-01,P002,Books,1,24.99,C102,South 2024-01-02,P003,Clothing,3,49.99,C103,East

2. **Airflow-Verbindung zu MinilO herstellen:** Öffnen Sie das Verbindungsmenü im Airflow-Web-Interface http://localhost:8080/connection/list/ und erstellen Sie eine neue Verbindung zu MinilO (siehe https://blog.min.io/apache-airflow-minio/).

   **Hinweise:**

   - Die Verbindung im Connection-Tab muss folgende Parameter beinhalten:
     - endpoint_url: `http://minio:9000`
     - AWS Access Key ID: Der Access Key aus `minio_credentials.txt`
     - AWS Secret Access Key: Der Secret Key aus `minio_credentials.txt`

3. **DAG erstellen:** Öffnen Sie die Datei `minio_dag_exercise.py` und lösen Sie die Aufgaben. Kopieren Sie die Datei in das Verzeichnis `dev-environment/dags/`. Der DAG erscheint nach einigen Sekunden im Web-Interface.

4. **DAG ausführen:** Aktivieren Sie den DAG und führen Sie ihn aus. Überprüfen Sie die Logs im Web-Interface, ob die Datei erfolgreich heruntergeladen wurde.

5. **Fragen:**

- Auf welchem Docker-Container wird der DAG ausgeführt?
- Auf welchem Docker-Container wird die Datei `sample_data.csv` gespeichert?
- Wie können Sie den Pfad zu der Datei `sample_data.csv` im Container ermitteln?

## Troubleshooting Guide

### Häufige Probleme und Lösungen

1. **DAG ist nicht im Web-Interface sichtbar**
    - Prüfen Sie, ob die Datei `minio_dag_exercise.py` im Verzeichnis `dev-environment/dags/` vorhanden ist.
    - Prüfen Sie die Logs: `docker compose -f dev-environment/docker-compose.yml logs -f airflow-worker`
2. **Datei wird nicht heruntergeladen**
    - Prüfen Sie die Logs: `docker compose -f dev-environment/docker-compose.yml logs -f airflow-worker`
    - Prüfen Sie die Verbindung zu MinIO in der Airflow-UI http://localhost:8080/connections/
    - Prüfen Sie die Logs von MinIO: `docker compose -f dev-environment/docker-compose.yml logs -f minio`
    - Prüfen Sie die Berechtigungen der Datei `sample_data.csv` in MinIO

## Validierungs-Checkliste

- [ ] DAG ist im Web-Interface sichtbar
- [ ] Datei wird heruntergeladen

## Ressourcen:

- Airflow UI
- Airflow S3 Hook
- Airflow DAG Context Manager
- Airflow S3 Load File
- Airflow DAG Documentation
- Airflow Logging