Data Engineering Exercise: Building an ETL Pipeline with Apache Airflow

Objective

Develop a scalable and automated data pipeline using Apache Airflow to manage the ETL process of loading data from Google Cloud Storage (GCS) to BigQuery.

Setup Guidelines

Prerequisites

- Ensure you have access to Google Cloud Platform with billing set up.
- Ensure `gcloud` and `bq` CLI tools are installed and authenticated.

Predefined Values

- **GCP Project ID**: `zhaw-data-engineering-2023`
- **Service Account Name**: `de-2023-service-account`
- **Display Name**: `Data Engineering 2023 Service Account`
- **Key Path**: `de-2023-service-account-key.json`
- **Cloud Composer Environment Name**: `de-2023-airflow-env`
- **Location for Cloud Composer**: `europe-west6` (Switzerland nearby region)
- **Zone for Cloud Composer**: `europe-west6-a`
- **Disk Size for Cloud Composer**: `20GB`
- **Machine Type for Cloud Composer**: `composer-n1-standard-2`

Commands

```
# Create a new GCP project
# Make sure to replace [BILLING_ACCOUNT_ID] with your billing account ID
gcloud projects create zhaw-da-[SHORTNAME]-2023 --name="ZHAW Data
 ↪  Engineering 2023"
gcloud beta billing projects link zhaw-data-engineering-2023
 ↪  --billing-account=[BILLING_ACCOUNT_ID]

# Set the GCP project
gcloud config set project zhaw-data-engineering-2023
```

```
# Enable necessary APIs
gcloud services enable bigquery.googleapis.com
gcloud services enable storage-api.googleapis.com
gcloud services enable composer.googleapis.com

# Create a service account
gcloud iam service-accounts create de-2023-service-account --display-name
 ↪  "Data Engineering 2023 Service Account"

# Create and download a JSON key for the service account
gcloud iam service-accounts keys create de-2023-service-account-key.json
 ↪  --iam-account de-2023-service-account@zhaw-data-engineering-
 ↪  2023.iam.gserviceaccount.com

# Assign roles to the service account
gcloud projects add-iam-policy-binding zhaw-data-engineering-2023
 ↪  --member="serviceAccount:de-2023-service-account@zhaw-data-engineering-
 ↪  2023.iam.gserviceaccount.com"
 ↪  --role="roles/bigquery.user"
gcloud projects add-iam-policy-binding zhaw-data-engineering-2023
 ↪  --member="serviceAccount:de-2023-service-account@zhaw-data-engineering-
 ↪  2023.iam.gserviceaccount.com"
 ↪  --role="roles/bigquery.dataEditor"
gcloud projects add-iam-policy-binding zhaw-data-engineering-2023 \
    --member="serviceAccount:de-2023-service-account@zhaw-data-engineering-
     ↪  2023.iam.gserviceaccount.com"
     ↪  \
    --role="roles/bigquery.datasetCreator"
gcloud projects add-iam-policy-binding [YOUR_PROJECT_ID] \
    --member="serviceAccount:de-2023-service-account@zhaw-data-engineering-
     ↪  2023.iam.gserviceaccount.com"
     ↪  \
    --role="roles/bigquery.admin"

gcloud projects add-iam-policy-binding zhaw-data-engineering-2023
 ↪  --member="serviceAccount:de-2023-service-account@zhaw-data-engineering-
 ↪  2023.iam.gserviceaccount.com"
 ↪  --role="roles/storage.objectAdmin"

# Create a GCS bucket
# Make sure to replace [SHORTNAME] with your unique short name
gsutil mb -p zhaw-data-engineering-2023 -l europe-west6
 ↪  gs://zhaw-de-2023-[SHORTNAME]-data-bucket/
```

```
# Uploading Files to Cloud Shell:
# Before using the gsutil cp commands, make sure your files
 ↪  (crime_robbery.csv, crime_burglary.csv,
# and optionally schema.json) are uploaded to your Cloud Shell environment.
# You can do this by clicking on the three-dotted menu in the upper right
 ↪  corner of your Cloud Shell window
# and selecting "Upload file". Navigate to your file location on your local
 ↪  machine and select the file(s) to upload.

# After you have uploaded your files to Cloud Shell, use the following
 ↪  commands to move them to your GCS bucket.

# Make sure to replace [SHORTNAME] with your unique short name
gsutil cp crime_robbery.csv gs://zhaw-de-2023-[SHORTNAME]-data-bucket/
gsutil cp crime_burglary.csv gs://zhaw-de-2023-[SHORTNAME]-data-bucket/

# (Optional) Upload schema file to the GCS bucket
gsutil cp schema.json gs://zhaw-de-2023-[SHORTNAME]-data-bucket/

# Grant required permissions to Cloud Composer service account
gcloud projects add-iam-policy-binding [YOUR_PROJECT_ID] \
    --member=serviceAccount:service-[YOUR_PROJECT_NUMBER]@cloudcomposer-
     ↪  accounts.iam.gserviceaccount.com
     ↪  \
    --role=roles/composer.admin

# Grant required permissions to Cloud Composer service account
gcloud projects add-iam-policy-binding [YOUR_PROJECT_ID] \
    --member=serviceAccount:service-[YOUR_PROJECT_NUMBER]@cloudcomposer-
     ↪  accounts.iam.gserviceaccount.com
     ↪  \
    --role roles/composer.ServiceAgentV2Ext


# Create a Cloud Composer environment (optional)
gcloud composer environments create de-2023-airflow-env \
    --location=europe-west6 \
    --image-version=composer-2.4.5-airflow-2.5.3 \
    --environment-size=small \
    --scheduler-cpu=1 \
    --scheduler-memory="4G" \
    --worker-cpu=1 \
```

```
--worker-memory="4G" \
--min-workers=1 \
--max-workers=2
--service-account
 ↪  "serviceAccount:service-[YOUR_PROJECT_NUMBER]@cloudcomposer-
 ↪  accounts.iam.gserviceaccount.com"
```