Airflow DAG Documentation: BigQuery Table Copy from corp\_sec to rto

# 1️⃣ Executive Summary

This Airflow DAG automates copying a BigQuery table from one dataset to another between two projects.  
  
Use case:  
- Move history or snapshot data from semantic/staging layer to RTO layer.  
- Automate regular refresh of target tables.

# 2️⃣ Goal

Source Table:  
 sandbox-corp-odin-devl-f930.corp\_sec.t\_badge\_events

Target Table:  
 sandbox-corp-odin-dev1-f930.rto.t\_badge\_events

What it does:  
- Fully replaces target table contents with source table contents on every run.

# 3️⃣ High-Level Steps

1️⃣ Airflow schedules or triggers the DAG.  
2️⃣ DAG runs a single PythonOperator task.  
3️⃣ Task initializes Google BigQuery Client.  
4️⃣ Runs BigQuery copy job from source to target.  
5️⃣ Uses WRITE\_TRUNCATE to overwrite target table.  
6️⃣ Logs success or failure in Airflow UI.

# 4️⃣ Code Overview

The Airflow DAG code is maintained in the team’s repository and is also viewable in Airflow UI.  
  
Link to DAG in Composer/Airflow:  
 [Insert your Airflow DAG URL here]  
  
The code implements:  
- 1 PythonOperator task  
- Calls Google BigQuery client  
- Runs a table copy job from the corp\_sec dataset to the rto dataset  
- Uses WRITE\_TRUNCATE disposition to fully overwrite the target table

# 5️⃣ How to Deploy in Cloud Composer

1️⃣ Save the code as:  
 bq\_copy\_t\_badge\_events\_dag.py  
  
2️⃣ Upload to Composer's DAGs bucket:  
 - Cloud Console → Composer → Your Environment → DAGs → Upload  
 - OR using gsutil:  
 gsutil cp bq\_copy\_t\_badge\_events\_dag.py gs://YOUR-COMPOSER-BUCKET/dags/  
  
3️⃣ Wait for Composer to detect the new DAG.  
4️⃣ Go to Airflow UI.  
5️⃣ Enable the DAG (toggle it on).  
6️⃣ Trigger it manually.  
7️⃣ Monitor logs in Airflow UI.

# 6️⃣ Result

✅ Target table (sandbox-corp-odin-dev1-f930.rto.t\_badge\_events) will match the source table exactly.  
✅ Data is fully overwritten on each run.  
✅ Logs and success/failure visible in Airflow UI.

# 7️⃣ Customization Options

🟣 Append instead of overwrite:  
 write\_disposition="WRITE\_APPEND"  
  
🟣 Daily schedule at 2 AM:  
 schedule\_interval='0 2 \* \* \*'  
  
🟣 Email alerts on failure:  
 default\_args = {  
 ...  
 'email': ['your-team@example.com'],  
 'email\_on\_failure': True,  
 }

# 8️⃣ Requirements

✔ Airflow 2.5.3 (Cloud Composer-compatible)  
✔ Composer’s service account must have BigQuery Data Editor on both projects  
✔ Google Cloud BigQuery Python Client (already installed in Composer)

# 9️⃣ Quick One-Liner

This Airflow DAG automates copying the t\_badge\_events table from corp\_sec dataset to rto dataset, replacing the target with the latest snapshot.

# 🔟 Summary

✅ Single-task Airflow DAG  
✅ Uses Google BigQuery Client  
✅ Overwrites target table for clean refresh  
✅ Fully managed in Composer