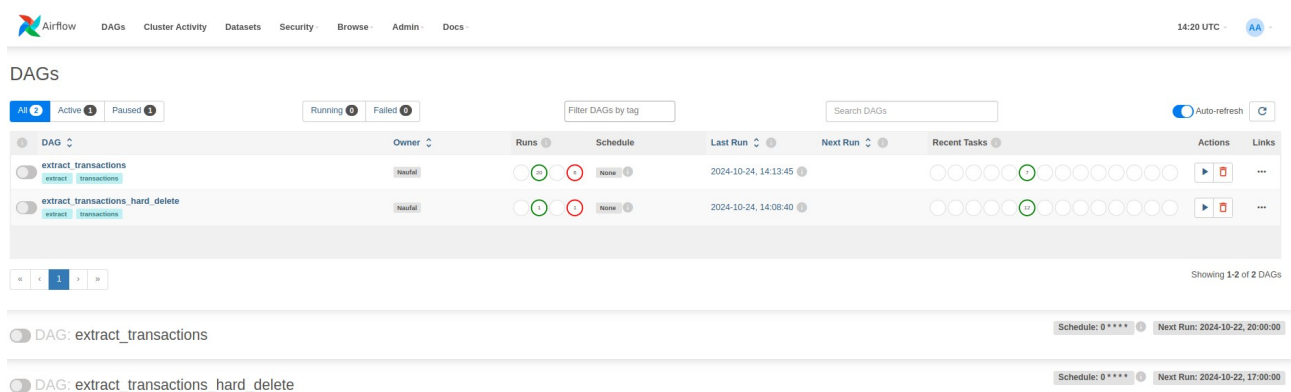


# Lion Parcel Technical Test

For this technical test, I use Airflow as Data Orchestration tools and Google BigQuery as Data Warehouse.

Before creating pipelines, the first thing I do is setup dummy data source which I already packed up in [migrate.up.sql](#) file. In this migration file, I create some trigger to handle auto update fields and hard delete case. For hard delete case, the trigger will catch the deleted records and move it to archive table. The archive table will be used as source to synchronize table in data source and data warehouse.

Pipeline results should be like this and run every hour in minute 0



I separate case in two dags:

- **extract\_transactions** for soft delete case.
- **extract\_transactions\_hard\_delete** for hard delete case.

## Soft Delete Case

Below is screenshot of Airflow Pipeline with soft delete case.



Explanation:

**init\_table:** Task for initializing new table, preventing from error table not exists

**date\_eval:** Custom python operator to evaluate date input. By using this task, we can easily do the backfill by triggering dag run using start and end date parameter

**extract\_to\_gcs:** Extracting records from data source (which this case is postgresql) to Google Cloud Storage using JSON format

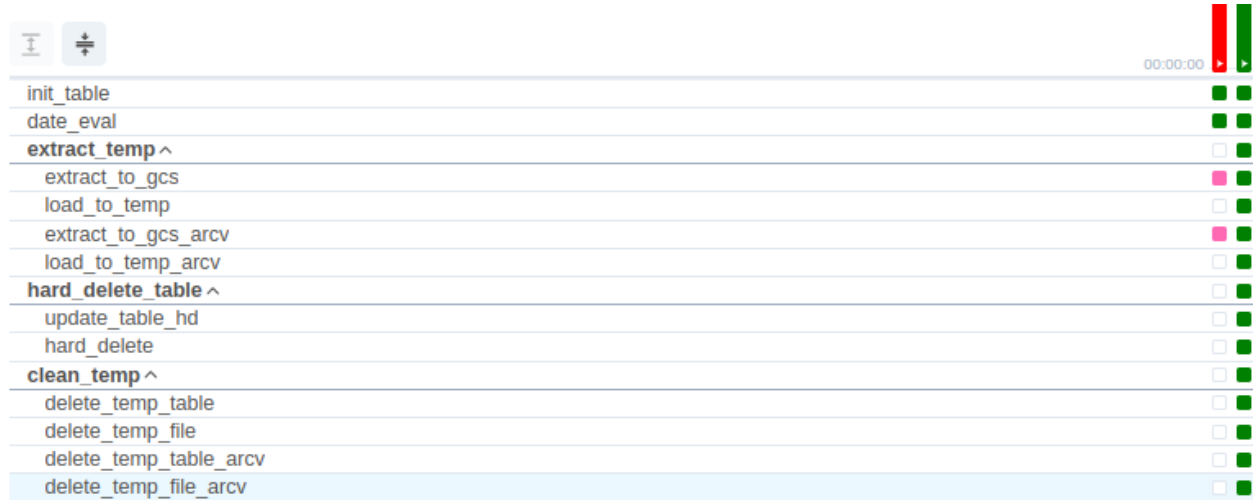
**load\_to\_temp:** Loading extracted JSON to temporary table in BigQuery

**update\_temp:** Updating records in table using merge. For this case, deleted\_at field will be filled following the data source.

**clean\_temp:** Task group to delete temporary file and table.

## Hard Delete Case

Below is screenshot of Airflow Pipeline with hard delete case.



init_table	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
date_eval	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>extract_temp ^</b>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
extract_to_gcs	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
load_to_temp	<input type="checkbox"/>	<input checked="" type="checkbox"/>
extract_to_gcs_arcv	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
load_to_temp_arcv	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<b>hard_delete_table ^</b>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
update_table_hd	<input type="checkbox"/>	<input checked="" type="checkbox"/>
hard_delete	<input type="checkbox"/>	<input checked="" type="checkbox"/>
<b>clean_temp ^</b>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
delete_temp_table	<input type="checkbox"/>	<input checked="" type="checkbox"/>
delete_temp_file	<input type="checkbox"/>	<input checked="" type="checkbox"/>
delete_temp_table_arcv	<input type="checkbox"/>	<input checked="" type="checkbox"/>
delete_temp_file_arcv	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Not too different with soft delete case, but I added some task to handle hard delete.

**extract\_to\_gcs\_arcv:** Extracting deleted records from archive table to Google Cloud Storage using JSON format

**load\_to\_temp\_arcv:** Loading extracted JSON to temporary table in BigQuery

**hard\_delete:** Delete records from main table using merge. If key in merge is match, the records will be deleted.

For how to setup the environment and run the pipeline, I already write a setup documentation in [README.md](#) file.