SQL Airflow pipelines

We will start by describing a simple DAG that will perform a series of simple steps in a Postgres table (create the table, load some data, count them and, finally, drop or delete the table randomly). The implementation of all DAGs are placed on the `dags` directory of the git repository and, in this particular case, the sql comands and even the csv data will be placed in a separate `sql` subdirectory:

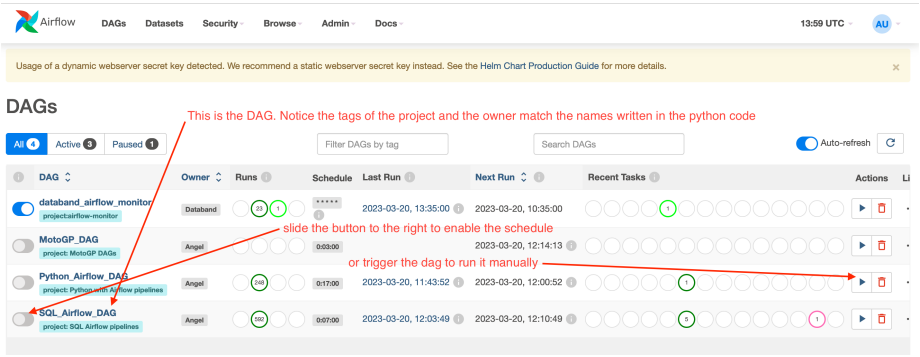
```
# You may need to change the cd command in order to be in the right directory
# Execute this cell

cd ../dags/
ls -l sql*
```

If everything went well, you will see an output similar to this:

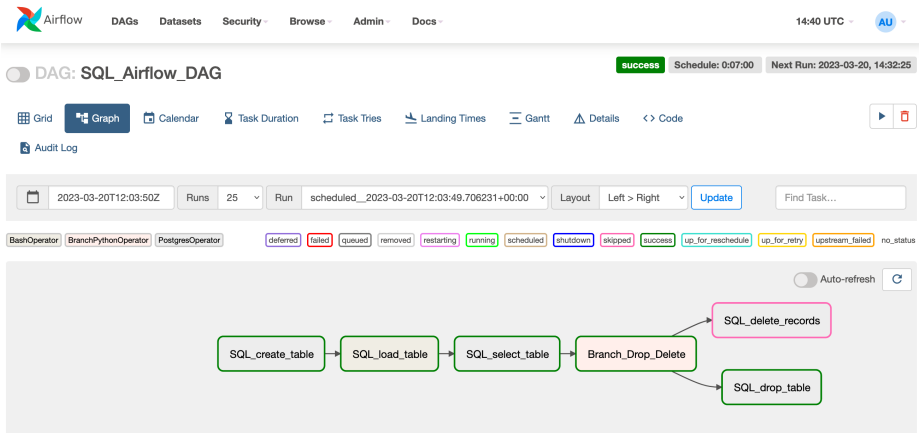
```
# Do not execute this cell. Just for information
-rw-r--r--  1 Angel  wheel  3372 Mar 20 15:09 sql_airflow_dag.py

sql:
total 312
-rw-r--r--@ 1 Angel  wheel  137845 Mar 13 11:06 motogp.csv
-rw-r--r--@ 1 Angel  wheel     181 Mar 13 11:06 motogp_create_table.sql
-rw-r--r--  1 Angel  wheel      21 Mar 13 11:06 motogp_delete_table.sql
-rw-r--r--  1 Angel  wheel      20 Mar 13 11:06 motogp_drop_table.sql
-rw-r--r--  1 Angel  wheel     469 Mar 13 11:06 motogp_load_table.py
-rw-r--r--@ 1 Angel  wheel      30 Mar 13 11:06 motogp_select_table.sql
```

More interesting than the contents of the `*.sql` files (which are simple sql statements) is the file sql_airflow_dag.py. Even if you are not a python programmer or have no Airflow skills, it is advisable to review the stucture of the code to understand what and how the DAG will do:

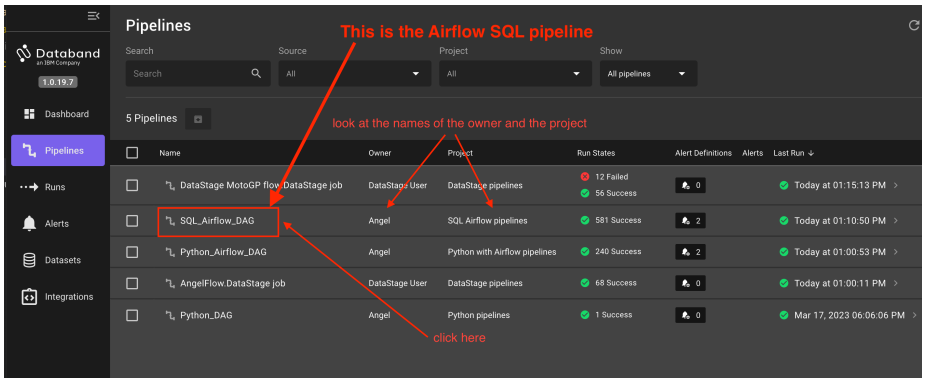
As we copied this file to the Airflow containers in the previous section of this workshop, the DAG will be visible on the Airflow console.



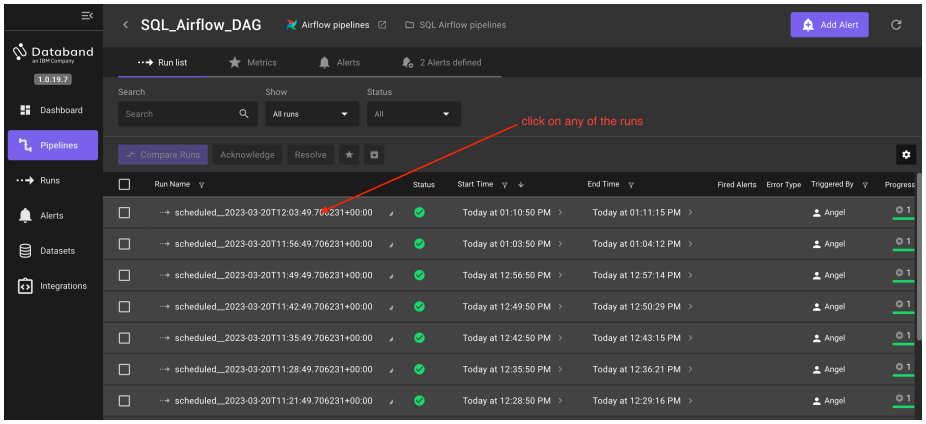
Follow the instructions of the picture above to run the DAG and click on the name of the DAG to see a graphical representation (note that the `graph` tab is highlighted)



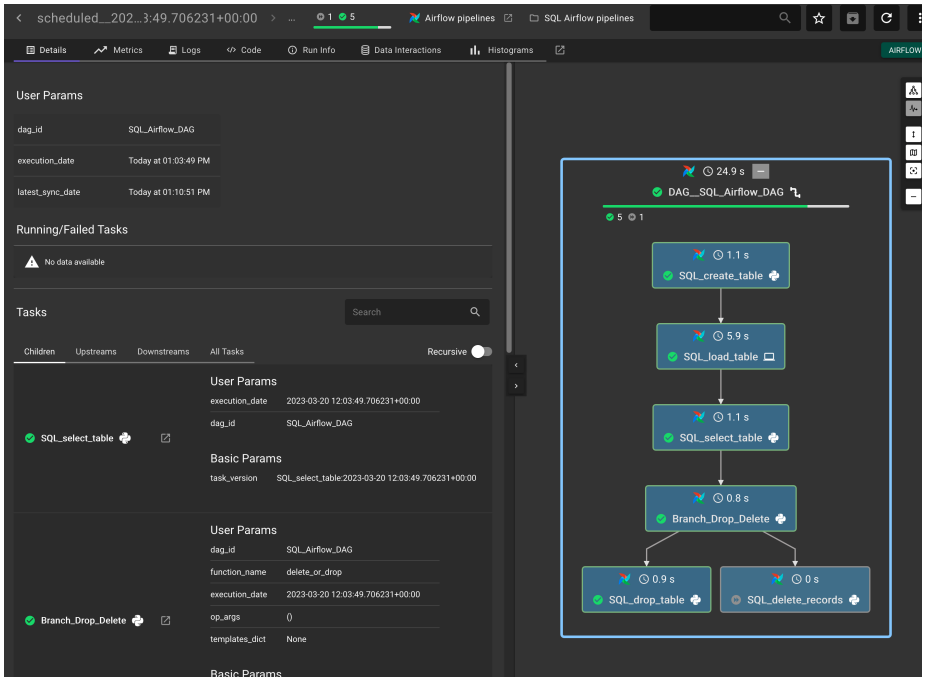
If you open the Databand main interface and navigate to the Pipelines menu on the left, you will see a list of all the pipelines, including the one we are focusing now, labeled as `SQL_Airflow_DAG`



If you click on the name of the pipeline, all the executions of this pipeline will be listed:



To see the deatils of each run, click on anyone of them:



It is important to remark that this DAG has no sign of Databand at all, i.e. we didn't write special line in the code and nothing implies that it will be monitored by Databand. Actually, it will be scheduled and run by Airflow, which will capture the execution data as any other DAG. The execution data will be pulled by Databand to display it as a pipeline.

The information collected by Databand will include the elapsed runtimes of each

task and its return codes. This is a basic start that will be enhanced in the next chapters where we will see more valuable information.

Next Section: Python pipelines

Previous Section: Preparation

Return to main