UDACITY

‹ Return to "Data Engineering Nanodegree" in the classroom

Data Pipelines with Airflow

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Well done on completing this project, good luck for your future projects 👍

General

DAG can be browsed without issues in the Airflow UI

The DAG opens up as expected and there are no visible error messages seen 👍

The dag follows the data flow provided in the instructions, all the tasks have a dependency and DAG begins with a start_execution task and ends with a end_execution task.

The graph view of DAG shows that all tasks have a dependency and it follows the data flow required 💯

Dag configuration

DAG contains default_args dict, with the following keys:

- Owner
- Depends_on_past
- Start_date
- Retries
- Retry_delay
- Catchup

At the beginning of the DAG's code a default args dict is defined with the above-mentioned parameters set to correct values ✅

The DAG object has default args set

The DAG object has a binding to defaults args, well done!

The DAG should be scheduled to run once an hour

The DAG schedule is set to hourly as required by specification ✔️

Staging the data

There is a task that to stages data from S3 to Redshift. (Runs a Redshift copy statement)

The task correctly stages data from s3 to redshift 👍

Instead of running a static SQL statement to stage the data, the task uses params to generate the copy statement dynamically

The task contains a set of parameters that are used to define for example the S3_key, S3_bucket, target table that generate the copy statement dynamically 😃

The operator contains logging in different steps of the execution

In the operators code logging.info is used to inform about the progress of the staging load

The SQL statements are executed by using a Airflow hook

The connection to the database is created by using a hook provided by Airflow and the connection is defined in Airflow’s connection list ✔

Loading dimensions and facts

Dimensions are loaded with on the LoadDimension operator

Separate functional operator for dimensions are loaded with on the LoadDimension operator 👏

Facts are loaded with on the LoadFact operator

Perfectly done!

Instead of running a static SQL statement to stage the data, the task uses params to generate the copy statement dynamically

The parameters are used to add some dynamic functionality to the operators and allow to run various SQL statements instead of hard coded SQL statement

The DAG allows to switch between append-only and delete-load functionality

The dimension operator allows to switch between append and truncate-insert

Data Quality Checks

Data quality check is done with correct operator

Data quality check is done with a operator that runs a check on the fact or dimension table(s) after the data has been loaded.

The DAG either fails or retries n times

Great job!

Operator uses params to get the tests and the results, tests are not hard coded to the operator
The operator for example allows to pass an array of checks and runs them in a loop ie tests are not hard coded to the operator 💯

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review