

[◀ Return to Classroom](#)[DISCUSS ON STUDENT HUB](#)

# Data Pipelines with Airflow

## REVIEW

## CODE REVIEW

## HISTORY

### Meets Specifications

Well done setting the parameters correctly as previously suggested. Congratulations! You have passed all specifications for this project. Good luck with your next Data Engineering project!

### General

DAG can be browsed without issues in the Airflow UI

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.

### Dag configuration

DAG contains default\_args dict, with the following keys:

- Owner

- Depends\_on\_past
- Start\_date
- Retries
- Retry\_delay
- Catchup

Good job setting proper arguments for your DAG. I'd like to bring your attention to this very useful configuration called `max_active_runs`. Sometimes, you may want to have exactly one task per DAG that is actively running. Perhaps when there is only one cluster of resources to be shared between the tasks. Setting `max_active_runs` to `1` would fit this requirement.

This setting is very useful to ensure we avoid any racing condition as the [documentation](#) states:

`max_active_runs` (int) – maximum number of active DAG runs, beyond this number of DAG runs in a running state, the scheduler won't create new active DAG runs

Remember that this setting would still allow for multiple DAGs, the only constraint here is one run per DAG.

The DAG object has default args set

The DAG should be scheduled to run once an hour

## Staging the data

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

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

The operator contains logging in different steps of the execution

The SQL statements are executed by using a Airflow hook

## Loading dimensions and facts

Dimensions are loaded with on the LoadDimension operator

Facts are loaded with on the LoadFact operator

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

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

## Data Quality Checks

Data quality check is done with correct operator

The DAG either fails or retries n times

Operator uses params to get the tests and the results, tests are not hard coded to the operator

 [DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)

Rate this review

START