Using Airflow to Manage ETL Workflows

Airflow v2

MJ Tung

https://www.linkedin.com/in/mjtung/
https://github.com/mjtung

2022-Oct-29

Batch Processing

ETL Jobs

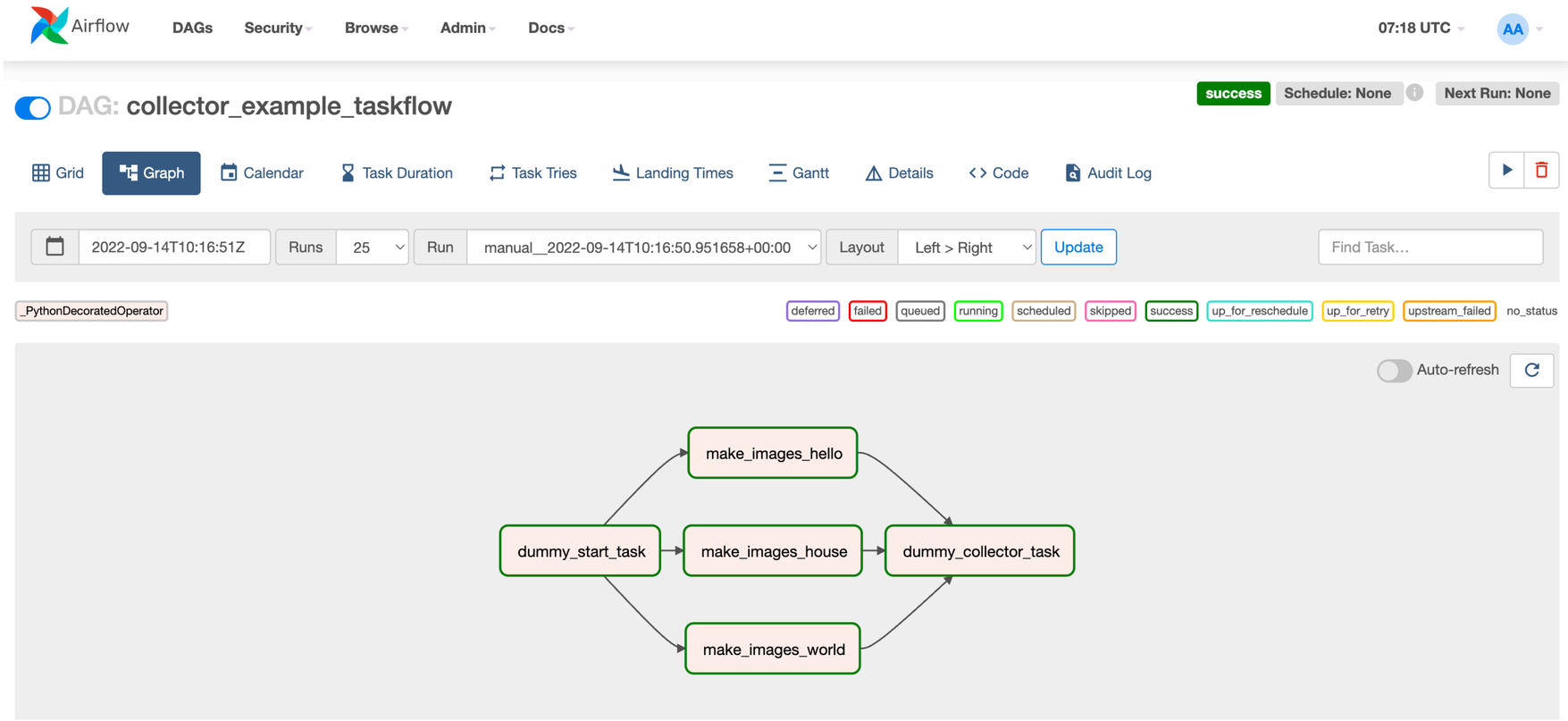
- Examples:
  - End-of-day processing
  - Downloading files
  - Calculating data for trading signals
- Challenges?
  - Scalability
  - Production support

```
# loaders
1 3 * * * /home/username/loaders/scripts/run_loaders_europe.sh > /dev/null
*/15 5 * * * /home/username/loaders/scripts/run_loaders_europe_price.sh > /
16,24 8 * * * /home/username/loaders/scripts/run_loaders_europe_price.sh >
6 6 * * * /home/username/loaders/scripts/run_loaders_europe_pms.sh > /dev/nu

# calculators
41 8 * * 1-6 /home/username/calculators/scripts/run_calculators_europe.sh >
12 13 * * 6 /home/username/calculators/scripts/run_calculators_europe_full.

# webserver
2 7 * * * /home/username/websvc/scripts/start_websvc.sh > /dev/null 2>&1
0 7 * * * /home/username/websvc/scripts/stop_websvc.sh > /dev/null 2>&1
```

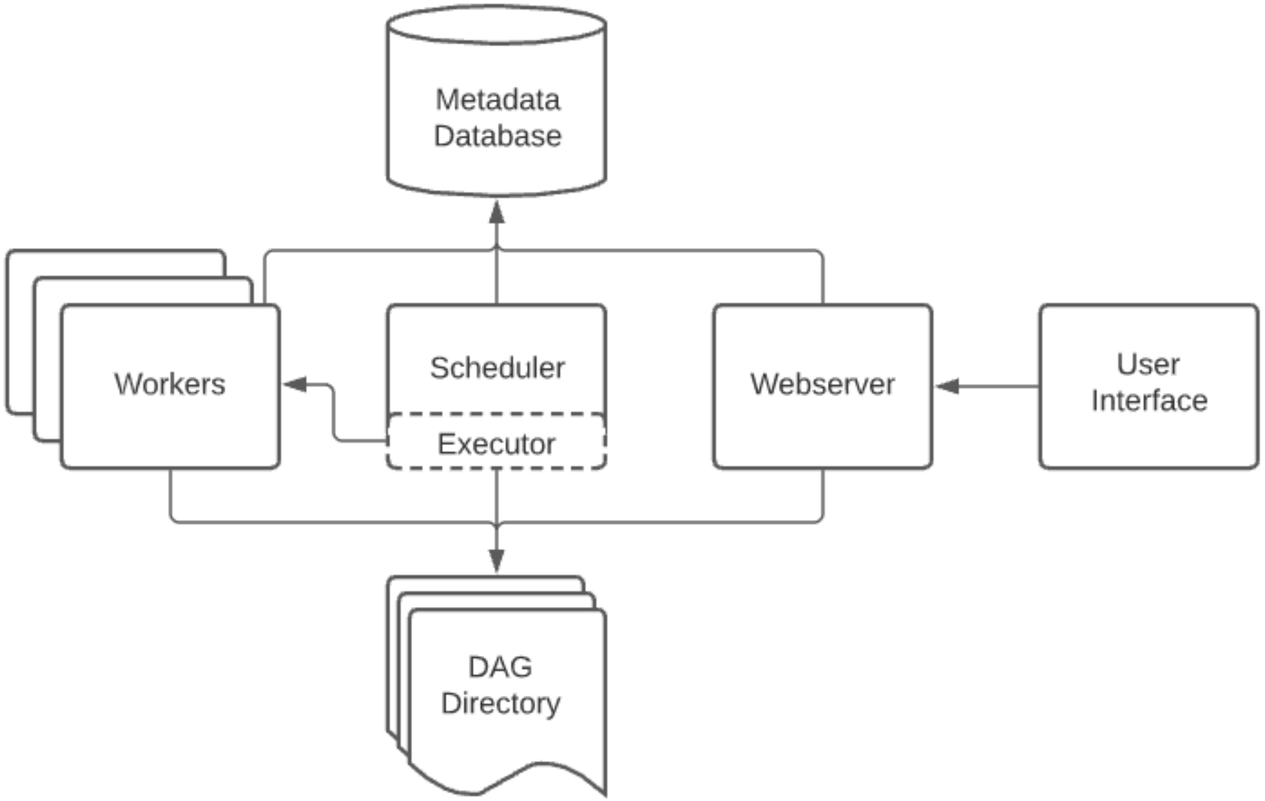
```python
import time
import os

while ~os.path.isfile('load_data_done.txt'):
    time.sleep(30)
    print('Waiting for first job to complete...')

    ...
```

Apache Airflow

DAG's: Directed Acyclic Graph



Architecture



https://airflow.apache.org/docs/apache-airflow/stable/concepts/overview.html#

Designing DAG's

Considerations when defining tasks

- Parallelism
- Atomicity
- Idempotency

Demo!

Logical Date

- Logical date / execution date: data interval covers period **before** the DAG runs
  - A DAG run is scheduled one interval **after** the start_date
  - https://airflow.apache.org/docs/apache-airflow/stable/dag-run.html?highlight=pass%20data#data-interval
  - https://airflow.apache.org/docs/apache-airflow/stable/faq.html#what-does-execution-date-mean

Other interesting bits

- External Task Sensor
- Pools
- Custom XCOM Backends

Considerations

Is Airflow right for you?

- Written in Python
- Somewhat complicated to onboard (Luigi is more lightweight)
- Tight coupling between scheduling and data logic
- Operation is easy, but maintenance is not simple
- Doesn’t suit all types of ETL (other flavours: MapReduce / Apache Beam)

References

- Data used in demo: https://www.kaggle.com/datasets/nathanlauga/nba-games
- Airflow documentation: https://airflow.apache.org/docs/apache-airflow/stable/
- Pycon HK 2018 talk on Airflow by Alejandro Saucedo: https://pycon.hk/2018/industrial-machine-learning-pipelines-with-python-airflow/