Class Notes April 27, 2020

DAG is a workflow
- Nodes - tasks
- Edges - dependencies

Represented in standard Python file (airflow DAGs folder)

Unit of work is executed a single operator
operator = single task (most are atomic)

Sometimes state needs to be shared, most of the time can
run independently

t1 >> t2 means t1 runs before t2

Trigger rule = conditions when a task can run

Scheduler is the brain of the operation
- Parses DAGs, sends to workers, coordinates with
- metadata repository, etc.

airflow list_dags
airflow list_tasks <dag> --treee
airflow test <dag_name> <task_name> <yyyy-mm-dd>
airflow backfill <dag_name> -s <yyyy-mm-dd> -e <yyyy-mm-dd>
airflow clear <dag_name> -s <yyyy-mm-dd> -e <yyyy-mm-dd>

Setup instructions on wiki.

Make sure to activate venv!

country1.py
- Very simple
- Two BashOperators -> all we need
- create table dependent on create dataset
- EVERY Airflow DAG must have start date!
  - Arbitrary in this case
- Commands in exactly the same form as on BQ console
- Move into dags folder
  - Won't run otherwise

Tasks fail. We want to backfill without corrupting the rest of the data

Life of a data engineer (a hah)

can branch things so they run in parallel
branch >> .....
branch >> .....