Apache Airflow in Open Data Hub

Vedant Mahabaleshwarkar
Red Hat AI Center of Excellence

Red Hat | IBM

Introduction to Airflow

- Airflow is a platform created by community to programmatically author, schedule and monitor workflows.
- Airflow has a python SDK that can be used to define DAGs.
- Airflow has a large community, as well as some commercial offerings.
- Airflow can be run on bare-metal as well as on Kubernetes or OpenShift clusters.

Airflow Python SDK sample

```
<snip>
dag = DAG(
    dag_id='example_python_operator', default_args=args,
    schedule_interval=None)
<snip>
run_this = PythonOperator(
    task_id='print_the_context',
    provide_context=True,
    python_callable=print_context,
    dag=dag)

for i in range(10):
    '''
    Generating 10 sleeping task, sleeping from 0 to 9 seconds
    respectively
    '''
    task = PythonOperator(
        task_id='sleep_for_'+str(i),
        python_callable=my_sleeping_function,
        op_kwargs={'random_base': float(i)/10},
        dag=dag)

    task.set_upstream(run_this)
```

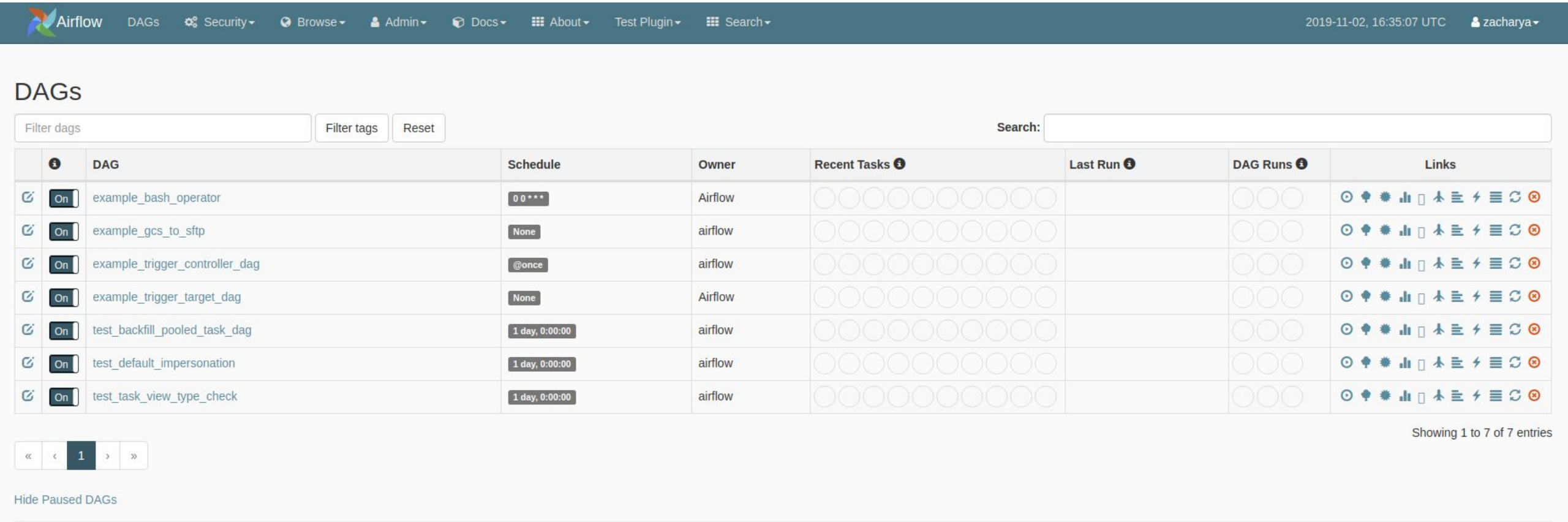
Features in Apache Airflow

- **Repeatability**: Airflow has the ability to schedule workflows to run repeatedly like a cron job.
- **Failure alerting**: Airflow has some nice features like email alerts that can be defined for workflow failures.
- Ability to **process historical data**: Airflow allows backfill for DAGs and also stores information on the history of the DAGs run.
- Parallel processing and **dependency management**: Tasks in the workflows can be set to have upstream dependencies, tasks with no dependencies can execute in parallel.

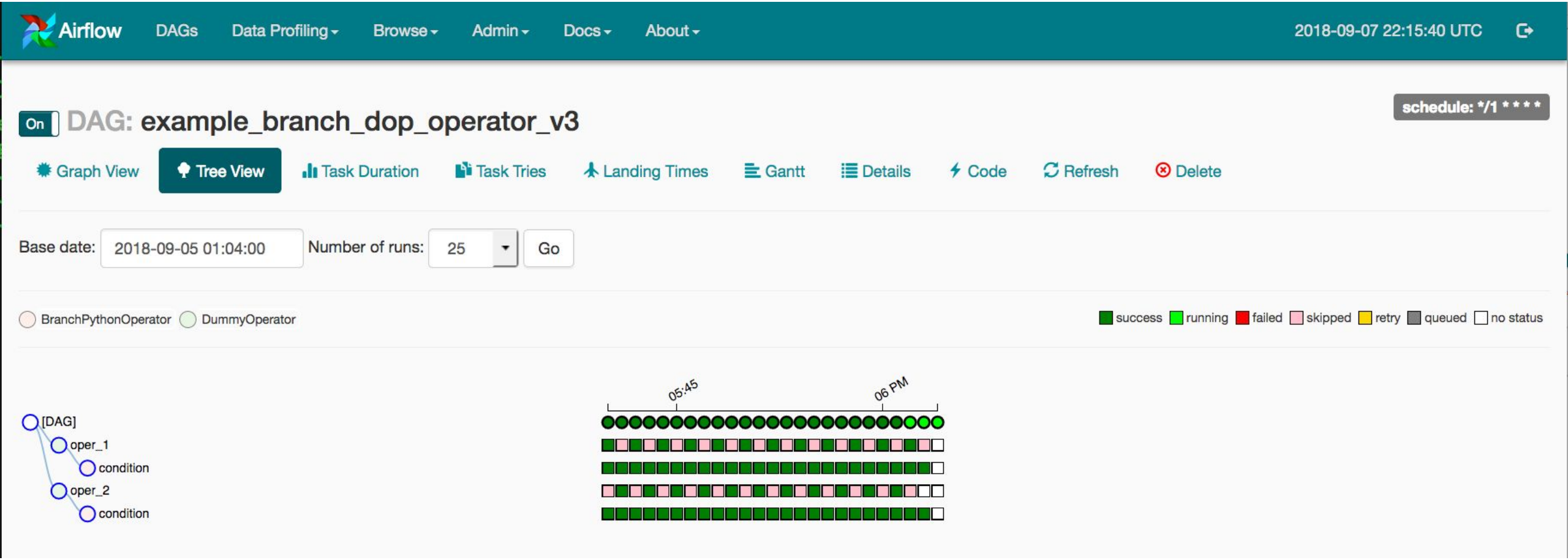
Features in Apache Airflow

- Airflow has a huge selection of Operators that make interacting with other systems easier.
  - DAGs determine how to run a workflow, operators are (usually) a atomic task in the workflow (eg: bash operator, email operator)
- Airflow has executor options like CeleryExecutor, LocalExecutor, and KubernetesExecutor.
- KubernetesExecutor is a nice feature for cloud deployments because it enables Airflow to scale out, the executor creates and deletes worker pods to execute and manage tasks for a given DAG

Airflow UI



Airflow UI


Airflow
DAGs
Data Profiling
Browse
Admin
Docs
About
2018-09-07 22:15:40 UTC
On
DAG: example_branch_dop_operator_v3
schedule: */1 * * * *
Graph View
Tree View
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Refresh
Delete
Base date:
2018-09-05 01:04:00
Number of runs:
25
Go
BranchPythonOperator
DummyOperator
success
running
failed
skipped
retry
queued
no status
05:45
06 PM
[DAG]
oper_1
condition
oper_2
condition

Airflow integration with Open Data Hub

- Open Data Hub will deploy the Airflow Operator to manage Airflow as an application.
- We are using the Airflow Operator originally developed in the GoogleCloudPlatform repository and later donated to Apache.
- The Operator creates a controller-manager pod which will be created as a part of the Open Data Hub deployment.
- Users can then install the Airflow components they need from the available options (eg: CeleryExecutor or KubernetesExecutor, Postgres deployment or MySQL deployment etc.)

DEMO