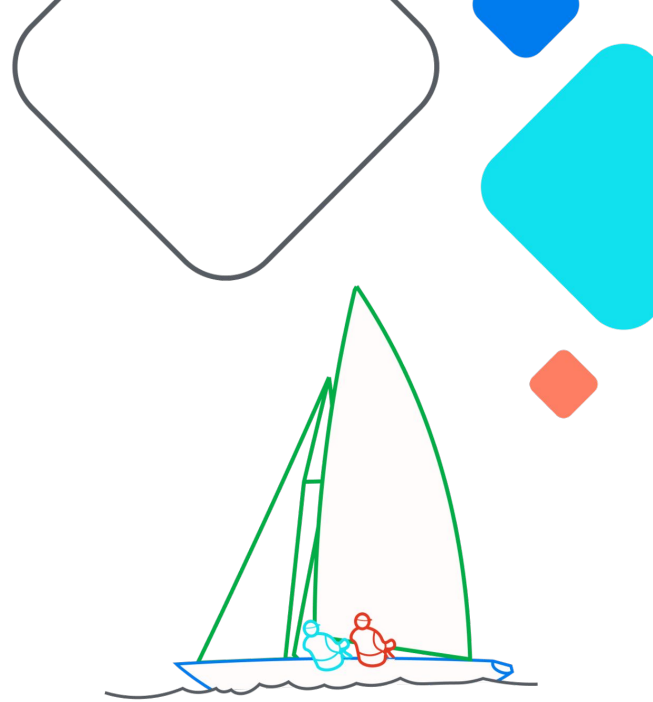


Testing Airflow Dags with dagtest

Victor Chiapaikao
Aldo Orozco

Etsy



 **Airflow Summit**

Let's flow together

September 19-21, 2023,
Toronto, Canada

Agenda

- Airflow @ Etsy
- User Archetypes
- The Problem
- Solutions & Inspiration
- dagtest
- Internals of dagtest
- Call to Action

Airflow @ Etsy



Airflow @ Etsy

- 4 Environments
- 2000+ Active DAGs in prod
- ~30k tasks running daily
- 100s of active users
- 7 VMs - LocalExecutor - 1.10.3 → 1 K8s cluster - KubernetesExecutor - 2.6.3



Etsy

User Archetypes



User Archetypes

- Data Engineers, Product Engineers, ML Engineers, Data Analysts / Scientists, etc.
- 2 Categories:
 - DAG Owner
 - Platform Developer



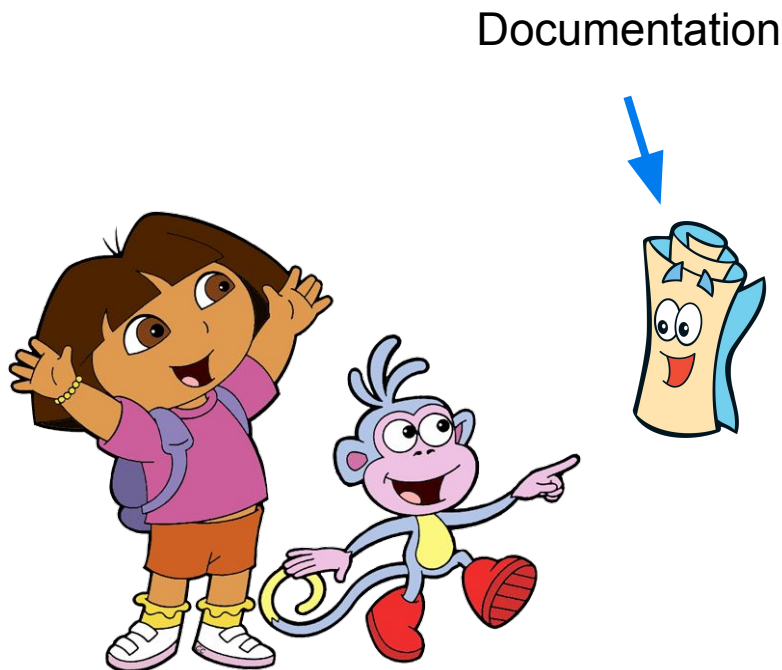
DAG Owner



Platform Developer

DAG Owners

- Majority of users (80%)
- Owns 1+ DAG(s)
- Familiar with writing dags and interacting via the UI
- Oncaller
- Checks internal and OS docs
- **May not know much about docker, k8s or python installations**



Platform Builder

- Minority (20%)
- Enabler of DAG owners
- Core maintainer or power user
- Creates new custom sensors, operators, macros, etc.
- **Might be more knowledgeable of docker / python**


















The Problem



Initial Symptoms

Commits on Apr 5, 2023

Fix base path (#12846)  gauravanand25 (ganand) committed 3 weeks ago ✓	Verified		528248d	
Fix base_path arg (#12845)  gauravanand25 (ganand) committed 3 weeks ago ✓	Verified		83e169c	
Fix args (#12844)  gauravanand25 (ganand) committed 3 weeks ago ✓	Verified		7c74dfa	
Migrate to spark job (#12841) ...  gauravanand25 (ganand) committed 3 weeks ago ✓	Verified		40c0a50	
Update settings.yaml (#12842)  mgallatin2 (mgallatin) committed 3 weeks ago ✓	Verified		32e939b	

More Symptoms

	<input type="checkbox"/>	exp_coarse_grained_base		@daily	jeskreiswinkler	5				
	<input checked="" type="checkbox"/>	exp_coarse_grained_baseb		@daily	jeskreiswinkler	5				
	<input checked="" type="checkbox"/>	exp_coarse_grained_basec		@daily	jeskreiswinkler		1	4		
	<input checked="" type="checkbox"/>	exp_coarse_grained_base_webboe		@daily	jeskreiswinkler	6				
	<input type="checkbox"/>	exp_coarse_grained_ufv22b		@daily	jeskreiswinkler	5				
	<input type="checkbox"/>	exp_coarse_grained_ufv23		@daily	jeskreiswinkler	5				
	<input type="checkbox"/>	exp_coarse_grained_ufv23_webboe		@daily	jeskreiswinkler	5				
	<input type="checkbox"/>	exp_coarse_grained_ufv24		@daily	jeskreiswinkler					
	<input type="checkbox"/>	exp_coarse_grained_ufv25		@daily	jeskreiswinkler	2	1	3		
	<input type="checkbox"/>	exp_coarse_grained_ufv26		@daily	jeskreiswinkler	5				
	<input type="checkbox"/>	exp_coarse_grained_ufv27		@daily	jeskreiswinkler	5				
	<input type="checkbox"/>	exp_coarse_grained_ufv28		@daily	jeskreiswinkler	6				
	<input checked="" type="checkbox"/>	exp_coarse_grained_ufv30b		@daily	jeskreiswinkler	5				

What About...

- `airflow dag test` cli
- Running locally (breeze)
 - Colima / Docker Desktop
 - Resource restrictions
 - Run a single DAG on the CLI
 - Permissions
- GCP Composer
 - Too many steps

Updating and testing a deployed DAG

To test updates to your DAGs in your test environment:

1. Copy the deployed DAG that you want to update to `data/test`.
2. Update the DAG.
3. Test the DAG.
 - a. [Check for syntax errors.](#)
 - b. [Check for task-specific errors.](#)
4. Make sure the DAG runs successfully.
5. Turn off the DAG in your test environment.
 - a. Go to the Airflow UI > DAGs page.
 - b. If the DAG you're modifying runs constantly, turn off the DAG.
 - c. To expedite outstanding tasks, click the task and **Mark Success**.
6. Deploy the DAG to your production environment.
 - a. Turn off the DAG in your production environment.
 - b. [Upload the updated DAG](#) to the `dags/` folder in your production environment.

The problem

- If testing is not SUPER simple, users will work around it - testing in prod or no tests at all.
- Testing is risky in production - overwrite data



Solutions & Inspiration



Solutions

- We shouldn't...
 - Require users to manually gsutil cp some dag file to some bucket
 - Require users to pull some large image down
 - Require users to pip install some large package with transitive dependencies and then setup confusing plugin steps
- Test environment should be isolated from production
- Needs to be AS SIMPLE AS POSSIBLE

Solutions

- Some inspiration:
 - Databricks
 - Spark Shell
 - Apache Beam - run pipelines locally with LocalRunner
 - Client / server architecture

To run this example in Java:

```
Direct Flink FlinkCluster Spark Dataflow Samza Nemo Jet

$ mvn compile exec:java -Dexec.mainClass=org.apache.beam.examples.WindowedWordCount \
  -Dexec.args="--inputFile=pom.xml --output=counts" -Pdirect-runner
```

To view the full code in Java, see [WindowedWordCount](#).

To run this example in Python:

This pipeline writes its results to a BigQuery table `--output_table` parameter, using the format `PROJECT:DATASET.TABLE` or `DATASET.TABLE`.

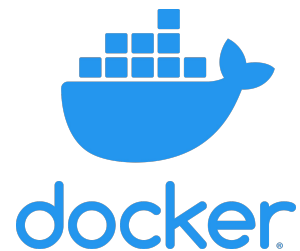
```
Direct Flink FlinkCluster Spark Dataflow Samza Nemo Jet

python -m apache_beam.examples.windowed_wordcount --input YOUR_INPUT_FILE --output_table PROJECT:DATASET.TABLE
```


Back to the 2 archetypes

Airflow Dag Owners

Airflow Platform Developers



dagtest



dagtest

- Attempts to empower users with the minimum requirements necessary to run a dag (or parts of a dag)
- Client part of client/server architecture
- Packaged as an internal PyPi package
- Tiny (13KB vs the base Airflow package's tar.gz at 11MB)
- Users send dags to an adhoc/test environment ONLY running with SA's that do not have access to Production buckets



**airflow
(11MB)**



**etsy-dagtest
(11KB)**

Testing w/ dagtest

```
pip install etsy-dagtest
```

```
dagtest path/to/my/dag.py [<execution-date>] [--dry-run]
[--ignore-dependencies] [--task] [--end-date]
```

```
vchiapaikao@7676:airflow (testing-is-fun)$ dagtest
Checking boundary-layer not installed: CHECKED
Checking boundary-layer2 is properly installed: CHECKED - boundary-layer2 - 2.2.2
Checking boundary-layer-etsy-plugin not installed: CHECKED
Checking boundary-layer-etsy-plugin2 is properly installed: CHECKED - boundary-layer-etsy-plugin
2 - 2.0.88
```

```
usage: dagtest [-h] [-e END_DATE] [-t TASK] [--dag DAG] [-r] [-i] [-p] [-c CONF] [-y]
              [--skip-version-check] [--skip-pr-tests] [-f environments/prod/dags/out.py]
              [-l ~/development/airflow/include] [-d | -m | -o]
              environments/prod/dags/dag.py [date]
```

```
dagtest: error: the following arguments are required: environments/prod/dags/dag.py
```

```
vchiapaikao@7676:airflow (testing-is-fun)$ █
```

```
vchiapaieo@7676:airflow (testing-is-fun)$ dagtest environments/dev/dags/dataeng/batch/my_first_dag.py
Checking boundary-layer not installed: CHECKED
Checking boundary-layer2 is properly installed: CHECKED - boundary-layer2 - 2.2.2
Checking boundary-layer-etsy-plugin not installed: CHECKED
Checking boundary-layer-etsy-plugin2 is properly installed: CHECKED - boundary-layer-etsy-plugin2 - 2.0.88
```

```
2023-09-13 14:12:33,804 - boundary-layer v. 2.2.2 - INFO - Loaded plugins default, etsy
Checking tool's version: PASSED
Python DAG detected
```

Execution Details:

```
=====
Run Type                | Test Run
[Username                | vchiapaieo
Skip PR Tests           | False
[DAG File Name          | my_first_dag.py
DAG ID                  | None
Execution Date Start    | 2023-09-12T00:00:00
Execution Date End      | None
Task ID                 | None
Task ID is Regex        | False
Ignore Dependencies     | False
Ignore First Depends on Past | False
Local Include Directory Path | None
Conf                    | None
Request Id              | 4b25e7b6
=====
```

```
Proceed with the test? [yes|y/no|n]: y
Uploading DAG... DONE
Running Unit Tests... SUCCESS
Initiating Test Run... STARTED
Waiting for first task to start... DONE
```

DAG my_first_dag_1694628754 logs available at:

https://web.airflow-adhoc.etsy-syseng-gke-prod.etsycloud.com/dags/my_first_dag_1694628754/grid?dag_run_id=backfill__2023-09-12T00%3A00%3A00%2B00%3A00

Please Note: These jobs will not respect the scheduler. Toggling these DAGs on will have no effect.

Testing w/ dagtest

The screenshot displays the Apache Airflow web interface. At the top, the navigation bar includes the Airflow logo, menu items for DAGs, Datasets, Security, Browse, Admin, Docs, and Submit Feedback, the current time (18:22 UTC), and a VC button. Below the navigation bar, the DAG name 'my_first_dag_1694628754' is shown with a toggle switch, a schedule of '@daily', and a next run time of '2023-09-13, 00:00:00'. A toolbar offers various views: Grid, Graph, Calendar, Task Duration, Task Tries, Landing Times, Gantt, Details, Code, and Audit Log. A dark green banner indicates 'GCP Cost Details (Beta) - Toggle to Show/Hide'. Below this, a filter bar shows the date '09/13/2023, 06:22:34 PM', a page number '25', and dropdowns for 'All Run Types' and 'All Run States', with a 'Clear Filters' button and an 'Auto-refresh' toggle. A horizontal bar contains status filters: deferred, failed, queued, removed, restarting, running, scheduled, shutdown, skipped, success, up_for_reschedule, up_for_retry, upstream_failed, and no_status. The main content area shows a DAG run for 'my_first_dag_1694628754' at '2023-09-13, 00:00:00 UTC'. On the left, a vertical bar shows task durations for 'start_task', 'middle_task1', 'middle_task2', and 'end_task'. The 'DAG Run Notes' section includes an 'Add Note' button. A table below shows the run details:

Status	success
Run ID	backfill__2023-09-12T00:00:00+00:00
Run type	backfill
Run duration	00:00:59

Internals



Test API

29 lines (19 sloc) | 1.01 KB

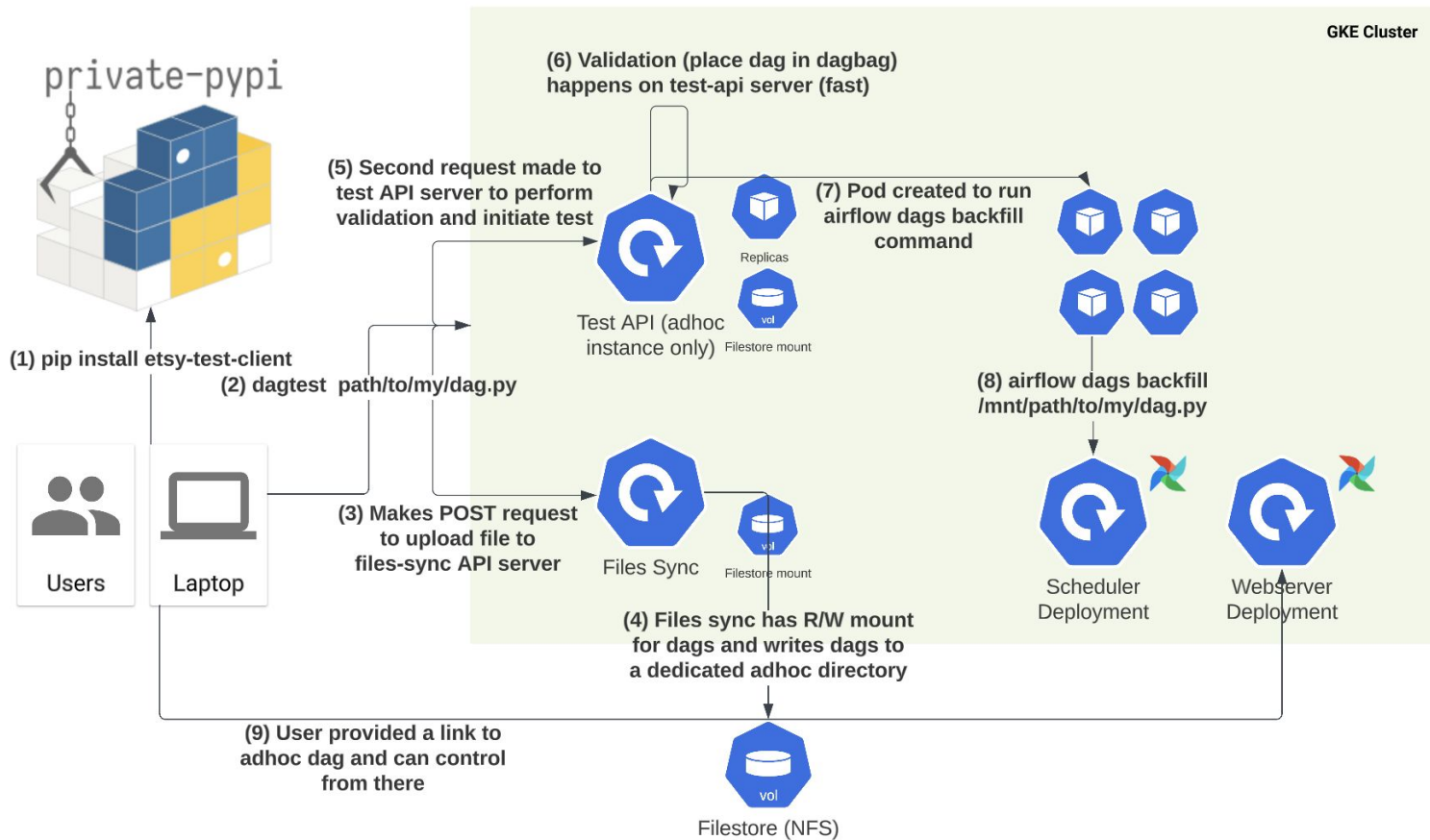
Raw

Blame



```
1 # This version should match with whatever version is set in
2 # http://airflow.apache.org/docs/apache-airflow/2.5.3/overrides.yaml#L1
3 FROM [REDACTED]/airflow:2.5.3-python3.10-etsy-0.0.5
4
5 USER root
6
7 ENV AIRFLOW_HOME=/opt/airflow
8
9 RUN export ACCEPT_EULA=Y && \
10     apt-get update && apt-get upgrade -y && \
11     apt-get install -y procs htop && \
12     rm -r /var/lib/apt/lists /var/cache/apt/archives
13
14 COPY modules/etsy-test-api $AIRFLOW_HOME/etsy-test-api
15 RUN chown -R airflow:root $AIRFLOW_HOME/etsy-test-api
16
17 COPY tests $AIRFLOW_HOME/tests
18 RUN chown -R airflow:root $AIRFLOW_HOME/tests
19
20 COPY requirements-test.txt $AIRFLOW_HOME/requirements-test.txt
21 RUN chown -R airflow:root $AIRFLOW_HOME/requirements-test.txt
22
23 USER airflow
24
25 RUN pip install -e $AIRFLOW_HOME/etsy-test-api
26
27 RUN pip install -r $AIRFLOW_HOME/requirements-test.txt
28
29 CMD ["bash", "-c", "gunicorn --bind 0.0.0.0:9107 -k gevent --workers 4 -t 120 --access-logfile - --log-level debug etsy_test_api.server:app"]
```


What happens when users call dagtest?



Environments

Test



Airflow running
KubernetesExecutor



Workload
Identity Pool



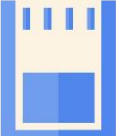
Test SA



GCS Buckets
(test)



Bigquery
Datasets
(dev)



Quotas
(test)



Filestore
(NFS Mount)

Production



Airflow running
KubernetesExecutor



Workload
Identity Pool



Prod SA



GCS Buckets
(prod)



Bigquery
Datasets
(prod)



Quotas
(prod)



Filestore
(NFS Mount)

The Outcome

- An empowered dag owner who can safely test dags outside of Production and faster iteration cycles
- PRs for dags can be merged with more confidence
- Lower likelihood that Production data is corrupted during testing
- Users uncover permissions issues in adhoc instead of Production
- If modifying a large Production dag, a single task or set of new tasks can be isolated and tested



Call to Action



Call to Action

- Gap in OSS offering
- `airflow dags test` does not work well for even slightly customized environments
- `airflow dags backfill` does not have proper REST API support
- Trigger dag functionality in REST API requires that a dag exists on some instance
- Trigger dag also must run the dag from E2E (can't test single or subset of tasks)
- Users shy away from Airflow because of the difficulty in testing / developing
- How can we build a better solution for this?



Questions?

