Contributing to Apache Airflow

Kaxil Naik
Airflow Committer and PMC Member
OSS Airflow Team @ Astronomer
Who am I?

- Airflow Committer & PMC Member
- Manager of Airflow Engineering team @ Astronomer
  - Work full-time on Airflow
- Previously worked at DataReply
- Masters in Data Science & Analytics from Royal Holloway, University of London
- Twitter: https://twitter.com/kaxil
- Github: https://github.com/kaxil/
- LinkedIn: https://www.linkedin.com/in/kaxil/
Agenda

- My Journey with Airflow
- How to start contributing to Airflow
- Contribution Workflow
- Continue committing regularly & steadily to become Airflow Committer. (including talking about current Guidelines of becoming a Committer)
- Different mediums of communication (Dev list, users list, Slack channel, Github Discussions etc)
- Reviewing PRs, Testing and voting on releases
- Dev Environment
- Number of commits does not matter as much as the experience and actual code
My Journey
Motivation to contribute!

Copy files from one Google Cloud Storage Bucket to other using Apache Airflow

Problem: I want to copy files from a folder in Google Cloud Storage Bucket (e.g Folder1 in Bucket1) to another Bucket (e.g Bucket2). I can't find any Airflow Operator for Google Cloud Storage to copy files.

https://stackoverflow.com/q/47452879/5691525
I just found a new operator in contrib uploaded 2 hours ago:

https://github.com/apache/incubator-airflow/blob/master/airflow/contrib/operators/gcs_to_gcs.py called GoogleCloudStorageToGoogleCloudStorageOperator that should copy an object from a bucket to another, with renaming if requested.
But it didn’t work …
Dear Airflow maintainers,

Please accept this PR. I understand that it will not be reviewed until I have checked off all the steps below!

JIRA

- My PR addresses the following Airflow JIRA issues and references them in the PR title. For example, "[AIRFLOW-XXX] My Airflow PR"
  - https://issues.apache.org/jira/browse/AIRFLOW-1842

Description

- Here are some details about my PR, including screenshots of any UI changes:
  - Renamed the super class from `GoogleCloudStorageOperatorToGoogleCloudStorageOperator` to `GoogleCloudStorageToGoogleCloudStorageOperator`.

This fixes the error `NameError: global name 'GoogleCloudStorageOperatorToGoogleCloudStorageOperator' is not defined`

Tests
My First PR - Fixes Typo

[AIRFLOW-1842] Fixed the Super class name for the operator #2810

---

```
@@ -51,7 +51,7 @@ def __init__(self,
              delegate_to=None,
              *args,
              **kwargs):
-        super(GoogleCloudStorageOperatorToGoogleCloudStorageOperator, self).__init__(*args, **kwargs)
+        super(GoogleCloudStorageToGoogleCloudStorageOperator, self).__init__(*args, **kwargs)

      self.source_bucket = source_bucket
      self.source_object = source_object
      self.destination_bucket = destination_bucket
```
My First PR - I didn’t follow Guidelines !!

<table>
<thead>
<tr>
<th>User</th>
<th>Comment</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>bolkedebriun</td>
<td>Please adhere to the commit guidelines. Thanks!</td>
<td>23 Nov 2017</td>
</tr>
<tr>
<td>kaxil</td>
<td>I have updated the title and updated the JIRA issue and everything needed to confirm to the guidelines.</td>
<td>23 Nov 2017</td>
</tr>
<tr>
<td>bolkedebriun</td>
<td>Please check your commit message. It does not refer the Jira issue and your commit message does not follow the guidelines. After that is fixed it can be merged.</td>
<td>24 Nov 2017</td>
</tr>
<tr>
<td>kaxil</td>
<td>Have edited it an created a new pull request at #2812</td>
<td>24 Nov 2017</td>
</tr>
<tr>
<td>bolkedebriun</td>
<td>Please close this issue.</td>
<td>26 Nov 2017</td>
</tr>
</tbody>
</table>

https://media.giphy.com/media/KSKvdT1YGCoJk0m69/giphy.gif

I TRIED.
My First “Merged” PR/commit

[AIRFLOW-1842] Fixed Super class name for the gcs to gcs copy operator

Fixed incorrect Super class name in gcs to gcs copy operator from 'GoogleCloudStorageOperatorToGo' to 'GoogleCloudStorageToGoogleCloudStorageOperator'.

Closes #2812 from kaxil/patch-2

2 files changed, 1 insertion, 1 deletion

diff --git a/airflow/contrib/operators/gcs_to_gcs.py b/airflow/contrib/operators/gcs_to_gcs.py
--- a/airflow/contrib/operators/gcs_to_gcs.py
+++ b/airflow/contrib/operators/gcs_to_gcs.py
@@ -51,7 +51,7 @@ def __init__(self,
             delegate_to=None,
             +args, 
             **kargs):
@@ -54, +14 @@
-    super(OperatorToGoogleCloudStorageOperator, self).__init__(+args, ++kargs)
+    super(GoogleCloudStorageToGoogleCloudStorageOperator, self).__init__(+args, ++kargs)
     self.source_bucket = source_bucket
     self.source_object = source_object
     self.destination_bucket = destination_bucket

http://gph.is/15RTH5O
Slowly & Steadily started adding more contributions
What did I learn while working on OSS

- Writing unit-tests
- Improved Coding skills
- Got to know many companies & devs across the globe
- Improved communication skills
  - Commit messages & PR descriptions
  - Email threads on dev list
  - Presentations
Became Airflow Committer & (P)PMC Member

The @ApacheAirflow is happy to announce new committer & PPMC member @kaxil! Please welcome him aboard! bit.ly/2FXWdw2

https://twitter.com/ApacheAirflow/status/993950478785490945
Steered Release for Airflow 1.10.2

From: Kaxil Naik <k...@gmail.com>
To: de...@airflow.apache.org
Subject: [VOTE] Airflow 1.10.2rc1
Date: 2019/01/17 00:04:25
List: dev@airflow.apache.org

Hey all,

I have cut Airflow 1.10.2 RC1. This email is calling a vote on the release, which will last for 72 hours. Consider this my (binding) +1.

Airflow 1.10.2 RC1 is available at:
https://dist.apache.org/repos/dist/dev/airflow/1.10.2rc1/

*apache-airflow-1.10.2rc1-source.tar.gz* is a source release that comes with INSTALL instructions.
*apache-airflow-1.10.2rc1-bin.tar.gz* is the binary Python "sdist" release.

Public keys are available at:
https://dist.apache.org/repos/dist/release/airflow/KEYS

Only votes from PMC members are binding, but members of the community are encouraged to test the release and vote with "(non-binding)".

Please note that the version number excludes the ‘rc’ string, so it’s now simply 1.10.2. This will allow us to rename the artifact without modifying the artifact checksums when we actually release.

Changes since 1.10.2b2:
[AIRFLOW-3601] Update notice to 2019 (#4503)
[AIRFLOW-3609] Update pop-up message when deleting DAG in RBAC UI (#4505)
[AIRFLOW-3608] Fix bug to set state of a task for manually-triggered DAGs (#4504)
[AIRFLOW-3607] Update Contributing Guide - Git Hooks (#4512)
[AIRFLOW-3426] Correct Python Version Documentation Reference (#4259)
[AIRFLOW-3603] Replace psycop2-binary by psycopg2 (#4508)
[AIRFLOW-2665] Add instructions to install SSH dependencies
[AIRFLOW-XXX] Clean up installation extra packages table (#3750)
[AIRFLOW-XXX] Remove redundant space in Kerberos (#3860)
[AIRFLOW-3808] Add extras group for google auth to setup.py. (#3917)
[AIRFLOW-XXX] Add Kubernetes Dependency in Extra Packages Doc (#4281)

From: Kaxil Naik <k...@gmail.com>
To: de...@airflow.apache.org
Subject: Airflow 1.10.2 is released
Date: 2019/01/23 01:18:38
List: dev@airflow.apache.org

Dear Airflow community,

I’m happy to announce that* Airflow 1.10.2* was just released.

The source release, as well as the binary "sdist" release, are available here:
https://dist.apache.org/repos/dist/release/airflow/1.10.2/

We also made this version available on PyPI for convenience (`pip install apache-airflow`):
https://pypi.python.org/pypi/apache-airflow

The documentation is available on:
- https://airflow.apache.org/
- https://airflow.apache.org/1.10.2/
- https://airflow.readthedocs.io/en/1.10.2/

Find the CHANGELOG here for more details:

Cheers,
*Kaxil*
Became Leading Airflow Committer in Feb 2021
You are next!!
How to start contributing?
How to start contributing?

- Contributing Guidelines: CONTRIBUTING.rst
- Contributing Quick Start Guide: CONTRIBUTORS_QUICK_START.rst
- Good First Issues: https://github.com/apache/airflow/contribute
Contribution Workflow

1. Find the issue you want to work on
2. Setup a local dev environment
3. Understand the codebase
4. Write Code & add tests if it’s a code change
5. Run tests locally
6. Create PR and wait for reviews
7. Address any suggestions by reviewers
8. Nudge politely if your PR is pending reviews for a while
Finding issues to work on
Finding issues to work on

- Start small: the aim should be to understand the process
- Bugs / features impacting you or your work
- Documentation Issues (including Contribution Guides)
  - Missing or outdated info, typos, formatting issues, broken links etc
- Good First Issues: [https://github.com/apache/airflow/contribute](https://github.com/apache/airflow/contribute)
- Other open GitHub Issues: [https://github.com/apache/airflow/issues](https://github.com/apache/airflow/issues)
Finding issues to work on - Open Unassigned Issues

If the issue is open and un-assigned, comment you want to work on it.

A committer will assign that issue to you. Then it is all yours.
Finding issues to work on - Improving Documentation

- If you faced an issue with docs, better fix it for future readers
- Documentation PRs are the best first contributions
- Missing or outdated info, typos, formatting issues, broken links etc
- No need of writing unit tests
- Examples:
  - [https://github.com/apache/airflow/pull/16275](https://github.com/apache/airflow/pull/16275)
  - [https://github.com/apache/airflow/pull/13462](https://github.com/apache/airflow/pull/13462)
  - [https://github.com/apache/airflow/pull/15265](https://github.com/apache/airflow/pull/15265)
Setup a local dev environment
SetUp Local Development Environment

- Fork [Apache Airflow](https://github.com/apache/airflow) repo & clone it locally
- Install pre-commit hooks ([link](https://github.com/pre-commit/pre-commit-hooks)) to detect minor issues before creating a PR
  - Some of them even automatically fix issues e.g. ‘black’ formats python code
  - Install pre-commit framework: `pip install pre-commit`
  - Install pre-commit hooks: `pre-commit install`
- Use `breeze` - a wrapper around docker-compose for Airflow development.
  - Mac Users: Increase resources available to [Docker for Mac](https://github.com/apache/airflow/blob/main/BREEZE.rst#prerequisites)
  - Check Prerequisites: [https://github.com/apache/airflow/blob/main/BREEZE.rst#prerequisites](https://github.com/apache/airflow/blob/main/BREEZE.rst#prerequisites)
  - Setup autocomplete: `./breeze setup-autocomplete`
SetUp Local Development Environment - Breeze

- Airflow CI uses **breeze** too so it allows reproduction locally

- Allows running Airflow with different environments (different Python versions, different Metadata db, etc):
  - ./breeze --python 3.6 --backend postgres --postgres-version 12

- You can also run a local instance of Airflow using:
  - ./breeze start-airflow --python 3.6 --backend postgres

- You can then access the Webserver on **http://localhost:28080**
Welcome to your true based running Airflow environment (courtesy of Breeze).

To stop Airflow and exit term, just type 'stop_airflow'.

setup@breeze:/opt/airflow# airflow scheduler

2021-07-07 01:07:29,832 (scheduler_job.py:62) INFO - Starting the scheduler
2021-07-07 01:07:29,833 (scheduler_job.py:69) INFO - Processing each file at most 5 times

Warning: This is a development server. Do not use it in a production deployment.
Use a production WSGI server instead.
* Debug mode: off
* Environment: production

import os
import logging

os.environ['AIRFLOW_LOGGING_LEVEL'] = 'INFO'

app = Flask('Airflow')
app.config['ENVIRONMENT'] = 'production'
app.config['DEBUG'] = False

app.run(debug=True)

Environment Variables:
- AIRFLOW_LOGGING_LEVEL (default: CRITICAL)
- AIRFLOW_ENV (default: production)
- AIRFLOW_DEBUG (default: False)

Running the Unicorn server with:
- Workers: 4 sync
- Host: 8.8.8.8000
- Timeout: 120
- Logfiles: Access LogFormat:

INFO:root@breeze:/opt/airflow# airflow webserver

2021-07-07 01:07:29,524 (app.py:465) INFO - Pulling up the DogBag from /dev/null

Running the Unicorn server with:
- Workers: 4 sync
- Host: 8.8.8.8000
- Timeout: 120
- Logfiles: Access LogFormat:

INFO:root@breeze:/opt/airflow# airflow webserver

2021-07-07 01:07:29,524 (app.py:465) INFO - Pulling up the DogBag from /dev/null

Run on http://0.0.0.0:8783/ (Press CTRL+C to quit)

info fesenkor2.0.13: The platform "linux" is incompatible with this module.
info fesenkor2.0.13: It is an optional dependency and failed compatibility check. Excluding it from installation.
info fesenkor2.0.13: The platform "linux" is incompatible with this module.
info fesenkor2.0.13: It is an optional dependency and failed compatibility check. Excluding it from installation.

[34] Linking dependencies...
warning * > onenamed-bootstrap-date-timepicker@0.17.49 has unmet peer dependency bootstrap@3.3.3.
warning * > onenamed-bootstrap-date-timepicker@0.17.49 has unmet peer dependency moment@2.18.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency core@5.9.14.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency moving@5.9.14.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency mobx@6.0.4.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency react@16.8.4 | 17.0.0.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency styled-components@5.1.1.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency react-dropzone@11.12 | 17.0.0.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency react-draggable@2.12 | 17.0.0.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency react-native@0.12 | 17.0.0.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency react-select@5.1.7.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency react@16.8.4 | 17.0.0.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency styled-components@5.1.1.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency mobx@6.0.4.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency react@16.8.4 | 17.0.0.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency react-select@5.1.7.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency react@16.8.4 | 17.0.0.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency styled-components@5.1.1.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency react@16.8.4 | 17.0.0.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency react-select@5.1.7.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency react@16.8.4 | 17.0.0.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency styled-components@5.1.1.
warning * > redo2@0.6.0-c5.54 has unmet peer dependency react@16.8.4 | 17.0.0.
Understand the Codebase
Understand the Codebase

- **apache/airflow** is mono-repo containing:
  - Apache Airflow Python package
  - More than 60 Providers (Google, Amazon, Postgres, etc)
  - Container image
  - Helm Chart

- Each of these items are released and versioned separately
Understand the Codebase

- Do not try to understand the entire codebase at once
- Get familiar with the directory structure first
- Dive into the source code related to your issue
- Similar to: If you are moving to a new house, you would try to first get familiar with your immediate neighbours and then others. (unless you have memory like Sheldon Cooper !!!)
<table>
<thead>
<tr>
<th>Area</th>
<th>Paths (relative to the repository root)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Airflow Docs</td>
<td>docs/apache-airflow</td>
</tr>
<tr>
<td>Stable API</td>
<td>airflow/api_connexion</td>
</tr>
<tr>
<td>CLI</td>
<td>airflow/cli</td>
</tr>
<tr>
<td>Webserver / UI</td>
<td>airflow/www</td>
</tr>
<tr>
<td>Scheduler (incl. Dag Parsing)</td>
<td>airflow/jobs/scheduler_job.py</td>
</tr>
<tr>
<td>Executors</td>
<td>airflow/executors</td>
</tr>
<tr>
<td>DAG Serialization</td>
<td>airflow/serialization</td>
</tr>
<tr>
<td>Helm Chart (&amp; it’s tests)</td>
<td>chart</td>
</tr>
<tr>
<td>Container Image</td>
<td>Dockerfile</td>
</tr>
<tr>
<td>Tests</td>
<td>tests</td>
</tr>
<tr>
<td>Area</td>
<td>Paths (relative to the repository root)</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>Providers</td>
<td>airflow/providers</td>
</tr>
<tr>
<td>Core Operators</td>
<td>airflow/operators</td>
</tr>
<tr>
<td>Core Hooks</td>
<td>airflow/hooks</td>
</tr>
<tr>
<td>Core Sensors</td>
<td>airflow/sensors</td>
</tr>
<tr>
<td>DB Migrations</td>
<td>airflow/migrations</td>
</tr>
<tr>
<td>ORM Models (Python Class -&gt; DB Tables)</td>
<td>airflow/models</td>
</tr>
<tr>
<td>Secrets Backend</td>
<td>airflow/secrets</td>
</tr>
<tr>
<td>Configuration</td>
<td>airflow/configuration.py</td>
</tr>
<tr>
<td>Permission Model</td>
<td>airflow/www/security.py</td>
</tr>
<tr>
<td>All Docs (incl. docs for Chart &amp; Container image)</td>
<td>docs</td>
</tr>
</tbody>
</table>
Understand the Codebase - Areas

- Get expertise in a certain area before diving into a different one.

<table>
<thead>
<tr>
<th>Easy</th>
<th>Medium</th>
<th>Complex (core)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Docs</td>
<td>Webserver</td>
<td>Scheduler &amp; Dag Parsing</td>
</tr>
<tr>
<td>CLI</td>
<td>Helm Chart</td>
<td>Executors</td>
</tr>
<tr>
<td>Operators / Hooks / Sensors (Providers)</td>
<td>Dockerfile</td>
<td>Configuration</td>
</tr>
<tr>
<td>Stable API</td>
<td>Secrets Backend</td>
<td>Permission Model</td>
</tr>
<tr>
<td></td>
<td>DB Migrations</td>
<td></td>
</tr>
</tbody>
</table>
Write Code & add docs & tests
Write code, add tests and docs

- You can re-use lot of logic of the code in the “same area”
- If adding a feature, find previously merged PRs in same area to get an idea
- Check out Coding style and best practices in CONTRIBUTING.rst
- Add Tests and Docs
  - The tests directory has same structure as airflow.
  - E.g If code file is airflow/providers/google/cloud/operators/bigquery.py, tests for it should be added at tests/providers/google/cloud/operators/test_bigquery.py
  - Docs for it would be at docs/apache-airflow-providers-google/operators/cloud/bigquery.rst
Write code, add tests and docs

- You can re-use lot of logic of the code in the “same area”
  
  - e.g. if you are adding a new Operator for GCP, you can check how current GCS operators are implemented or find a PR that added a operator in Google Provider & see what all they added.

- Check out [Coding style and best practices](CONTRIBUTING.rst) in CONTRIBUTING.rst

- Add Tests and Docs
  
  - The tests directory has same structure as airflow.
  
  - E.g If code file is `airflow/providers/google/cloud/operators/bigquery.py`, tests for it should be added at `tests/providers/google/cloud/operators/test_bigquery.py`
Run tests locally
Create PR and wait for reviews
Address Reviews
Nudge for reviews
Communication Channels
Becoming a Committer
Links / References
- **Airflow**
  - Repo: [https://github.com/apache/airflow](https://github.com/apache/airflow)
  - Website: [https://airflow.apache.org/](https://airflow.apache.org/)
  - Blog: [https://airflow.apache.org/blog/](https://airflow.apache.org/blog/)
  - Documentation: [https://airflow.apache.org/docs/stable/](https://airflow.apache.org/docs/stable/)
  - Slack: [https://s.apache.org/airflow-slack](https://s.apache.org/airflow-slack)
  - Twitter: [https://twitter.com/apacheairflow](https://twitter.com/apacheairflow)

- **Contact Me:**
  - Twitter: [https://twitter.com/kaxil](https://twitter.com/kaxil)
  - Github: [https://github.com/kaxil/](https://github.com/kaxil/)
  - LinkedIn: [https://www.linkedin.com/in/kaxil/](https://www.linkedin.com/in/kaxil/)
Thank You!