How to create workflow

Organisation - Apache Airflow

Author - Kartik Khare

Overview

I got selected in Apache Airflow to work as a technical writer for the period of Aug-Nov 2019. My project was to create the docs which helped a new user in creating and running a DAG.

Project Description

Project description
In Apache Airflow, workflows are saved as a code. DAGs use operators to build complex workflow. A DAG is a collection of all the tasks you want to run, organized in a way that reflects their relationships and dependencies. A developer can describe the relationships in several ways. Task logic is saved in operators. Apache Airflow has operators that integrate with many services, but often developers need to write their own operators. Tasks can use the xcom metabase to communicate.

Expected deliverables

- A page for how to create a DAG that also includes:
    - Revamping the page related to scheduling a DAG
    - Adding tips for specific DAG conditions, such as rerunning a failed task
- A page for developing custom operators that includes:
    - Describing mechanisms that are important when creating an operator, such as template fields, UI color, hooks, connection, etc.
    - Describing the responsibility between the operator and the hook
    - Considerations for dealing with shared resources (such as connections and hooks)
- A page that describes how to define the relationships between tasks. The page should include information about:
    - >> <<
    - set_upstream/set_downstream
    - helpers method ex. Chain
- A page that describes the communication between tasks that also includes:
    - Revamping the page related to macros and XCOM

Community Bonding

The community was amazing. They already had a lot of material to onboard me. I started the process by understanding what were the requirements in the project and what should be the priorities. My deliverables were clearly defined and the mentors guided me towards completing them.

Once I got enough details and data about the project, I submitted a draft document indicating if the structure of my docs was right.

The main phase of doc development involved interacting with a lot of folks along with my mentor. My pull requests received a lot of comments which were really helpful. I polished my technical writing skills throughout the period since I had very specific and useful feedback.

Contributions

I was able to contribute 4 major documentation with more already in the pipeline.
The Airflow community really appreciated my work and I will continue to contribute even after GSoD.

Here are some of my contributions -

Title	**My contribution**	**Pull Request**	**Documentation Link**
DAG Runs	Broke Scheduling and Triggers page into this page. Added detailed documentation for how to re-run the DAG and the failed tasks Added details for how to run backfill and catchup in a DAG	https://github.com/apache/airflow/pull/6295	https://airflow.readthedocs.io/en/latest/dag-run.html
Scheduler	Broke Scheduling and Triggers page into this page. Major refactoring. Added details related	https://github.com/apache/airflow/pull/6295	https://airflow.readthedocs.io/en/latest/scheduler.html

	to pain points while scheduling the DAG for the first time		
Creating a custom Operator	Created this page from scratch. It contains details on how to create your own operator and how you can leverage various Base class functionalities to make the operator versatile as well as secure.	https://github.com/apache/airflow/pull/6348	https://airflow.readthedocs.io/en/latest/howto/custom-operator.html
Best Practices	Created this page from scratch. It was the most requested document in the Apache Airflow community with a 53% approval rate. It contains details on the precautions the user needs to take while writing a DAG as well as while deploying the DAG and airflow in production	https://github.com/apache/airflow/pull/6515	https://airflow.readthedocs.io/en/latest/best-practices.html

Project finalization

I successfully completed 2 of my deliverables and then proceeded to create the Best Practices document which was highly requested by the community and aligned with my project. The last 2 deliverable were put on hold by me after having a discussion with my mentor and community. This was because the pages made sense only after splitting up the **Concepts** page which currently exists and refactoring the resulting child page. I have taken up this task which will go on for the next month or so.

Learnings and Challenges

- Getting my grammar correct and my language neutral which is a must for a Technical writing doc.
- Engaging with community helps out a lot. They helped to clear the basic doubts and also set forward a direction in which I can then proceed.
- Google developer documentation style guide was a huge time saver. It helped me avoid mistakes which people usually make while starting.
- One of the challenges was to get familiar with the style guide of the documentation e.g. always highlighting Airflow specific keywords in red with grey background. I was able to learn the specific styles from the Pull Request comments as well as studying existing documentation thoroughly.
- Another challenge was to write the documentation in RST format and use sphinx theme syntax. This also took me some time to familiarize.

Future Work

Currently, Airflow documentation needs a re-structure. Some pages need to be broken up and other which need to be combined and grouped into proper sections.
I have taken up the task of splitting up the **Concepts** page.
Apart from this, I'll also be working on refactoring the **Macros** page as well as creating documentation for missing operators, hooks, sensor and executors.