Item 216
git_comments:

1. [START tutorial] [START import_module]
2. [END extract]
3. [END instantiate_dag]
4. [END default_args]
5. [START transform]
6. [START extract]
7. [START instantiate_dag]
8. **comment:** [START documentation]
   label: documentation
9. The DAG object; we'll need this to instantiate a DAG
10. **comment:** [END documentation]
    label: documentation
11. **comment:** [END tutorial]
    label: documentation
12. [START default_args] These args will get passed on to each operator You can override them on a per-task basis during operator initialization
13. Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
14. [END load]
15. pylint: disable=missing-function-docstring
16. [START load]
17. [START main_flow]
18. [END main_flow]
19. [END import_module]
20. [END transform]
21. [START tutorial] [START import_module]
22. [END default_args]
23. [START instantiate_dag]
24. Operators; we need this to operate!
25. [END load_function]
26. The DAG object; we'll need this to instantiate a DAG
27. **comment:** [END documentation]
    label: documentation
28. **comment:** [END instantiate_dag] [START documentation]
    label: documentation
29. **comment:** [END tutorial]
    label: documentation
30. [END extract_function]
31. [START default_args] These args will get passed on to each operator You can override them on a per-task basis during operator initialization
32. Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.
33. [END transform_function]
34. [START transform_function]
35. [START load_function]
36. pylint: disable=missing-function-docstring
37. [START main_flow]
38. [END main_flow]
39. [START extract_function]
40. [END import_module]

git_commits:

1. **summary:** Airflow tutorial to use Decorated Flows (#11308)
   message: Airflow tutorial to use Decorated Flows (#11308) Created a new Airflow tutorial to use Decorated Flows (a.k.a. functional DAGs). Also created a DAG to perform the same operations without using functional DAGs to be compatible with Airflow 1.10.x and to show the difference. * Apply suggestions from code review It makes sense to simplify the return variables

being passed around without needlessly converting to JSON and then reconverting back. * Update tutorial_functional_etl_dag.py Fixed data passing between tasks to be more natural without converting to JSON and converting back to variables. * Updated dag options and task doc formating Based on feedback on the PR, updated the DAG options (including schedule) and the fixed the task documentation to avoid indentation. * Added documentation file for functional dag tutorial Added the tutorial documentation to the docs directory. Fixed linting errors in the example dags. Tweaked some doc references in the example dags for inclusion into the tutorial documentation. Added the example dags to example tests. * Removed multiple_outputs from task defn Had a multiple_outputs=True defined in the Extract task defn, which was unnecessary. - Removed based on feedback. Co-authored-by: Gerard Casas Saez <casassg@users.noreply.github.com> Co-authored-by: Kaxil Naik <kaxilnaik@gmail.com> Co-authored-by: Ash Berlin-Taylor <ash_github@firemirror.com>
label: code-design

github_issues:

1. **title:** [AIP-31] Update Airflow tutorial to use functional DAGs
   body: **Description** Change Airflow tutorial (or create new tutorial) using the new functional layer. **Use case / motivation** Help users onboard Airflow by reducing the complexity of the tutorial by adopting the new functional DAG definition layer. **Related Issues** Blocked by #8057
2. **title:** [AIP-31] Update Airflow tutorial to use functional DAGs
   body: **Description** Change Airflow tutorial (or create new tutorial) using the new functional layer. **Use case / motivation** Help users onboard Airflow by reducing the complexity of the tutorial by adopting the new functional DAG definition layer. **Related Issues** Blocked by #8057
3. **title:** [AIP-31] Update Airflow tutorial to use functional DAGs
   body: **Description** Change Airflow tutorial (or create new tutorial) using the new functional layer. **Use case / motivation** Help users onboard Airflow by reducing the complexity of the tutorial by adopting the new functional DAG definition layer. **Related Issues** Blocked by #8057
   label: documentation
4. **title:** [AIP-31] Update Airflow tutorial to use functional DAGs
   body: **Description** Change Airflow tutorial (or create new tutorial) using the new functional layer. **Use case / motivation** Help users onboard Airflow by reducing the complexity of the tutorial by adopting the new functional DAG definition layer. **Related Issues** Blocked by #8057
5. **title:** [AIP-31] Update Airflow tutorial to use functional DAGs
   body: **Description** Change Airflow tutorial (or create new tutorial) using the new functional layer. **Use case / motivation** Help users onboard Airflow by reducing the complexity of the tutorial by adopting the new functional DAG definition layer. **Related Issues** Blocked by #8057
6. **title:** [AIP-31] Update Airflow tutorial to use functional DAGs
   body: **Description** Change Airflow tutorial (or create new tutorial) using the new functional layer. **Use case / motivation** Help users onboard Airflow by reducing the complexity of the tutorial by adopting the new functional DAG definition layer. **Related Issues** Blocked by #8057
   label: documentation

github_issues_comments:

1. @casassg @turbaszek Can one you create a Github issue for the "dag" decorator -- looks like that is the only other pieces missing for Airflow 2.0 with regards to Functional DAGs apart from other tickets mentioned in https://github.com/apache/airflow/labels/AIP-31
2. I think the guide should present a functional and functional approach. These are two different approaches and have different use cases. This is not a 1 to 1 alternative.
3. **body:** This specific issue is about the main tutorial. I think that one benefits from having a unique style (I prefer functional as its easier). We should add a guide for both styles though
   label: documentation
4. Vikram has volunteered to tackle this

github_pulls:

1. **title:** Airflow tutorial to use functional DAGs
   body: Created a new Airflow ETL tutorial to use functional DAGs. Also created a DAG to perform the same operations without using functional DAGs to be compatible with Airflow 1.10.x and to show the difference between the "functional" and the "classic" ways side by side to illustrate the differences. I tried to make this simple and not rely on any operators. I will create the html documentation page to cover this as soon as I get confirmation from @casassg and @turbaszek that this is in the right general direction. Also tagging @ashb for general review. closes: #9041 <!-- Thank you for contributing! Please make sure that your code changes are covered with tests. And in case of new features or big changes remember to adjust the documentation. Feel free to ping committers for the review! In case of existing issue, reference it using one of the following: How to write a good git commit message: http://chris.beams.io/posts/git-commit/ --> --- **^ Add meaningful description above** Read the **[Pull Request Guidelines](https://github.com/apache/airflow/blob/master/CONTRIBUTING.rst#pull-request-guidelines)** for more information. In case of fundamental code change, Airflow Improvement Proposal ([AIP](https://cwiki.apache.org/confluence/display/AIRFLOW/Airflow+Improvements+Proposals)) is needed. In case of a new dependency, check compliance with the [ASF 3rd Party License Policy](https://www.apache.org/legal/resolved.html#category-x). In case of backwards incompatible changes please leave a note in [UPDATING.md](https://github.com/apache/airflow/blob/master/UPDATING.md).

github_pulls_comments:

1. [The Build Workflow run](https://github.com/apache/airflow/actions/runs/291706945) is cancelling this PR. It has some failed jobs matching ^Pylint$,^Static checks$,^Build docs$,^Spell check docs$,^Backport packages$,^Checks: Helm tests$,^Test OpenAPI*.
2. @vikramkoka I think this a good example 👍
3. **body:** Looks good. I would try to use multiple outputs better instead of json dumping and such. That feels against what we have been trying to do. Lets return dictionary and load the number directly
   label: code-design
4. **body:** A thought: should add somewhere the `get_current_context` function, just to showcase?
   label: code-design
5. > Looks good. I would try to use multiple outputs better instead of json dumping and such. That feels against what we have been trying to do. Lets return dictionary and load the number directly Thank you both for the detailed, prompt feedback! I really appreciate it and will make the changes.
6. **body:** (note that you may need to clean up code, my code suggestions was mostly to point how to use it) Also, may be interesting to add a unit test that executes the DAG end to end?
   label: test
7. > (note that you may need to clean up code, my code suggestions was mostly to point how to use it) > > Also, may be interesting to add a unit test that executes the DAG end to end? Yes, absolutely. Will make the changes and keep updating the PR, just so that you have the visibility.
8. > A thought: should add somewhere the `get_current_context` function, just to showcase? Your point is well taken and I should have stated up front, that I would like to add one more example which is more complex which contains both the 'get_current_context', as well as a BashOperator to demonstrate how to use functional Dags with non-python (more classic) operators.
9. > Also, may be interesting to add a unit test that executes the DAG end to end? Plus +1 for that, this should be simple as adding this DAG id to: https://github.com/apache/airflow/blob/b0fcf675595494b306800e1a516548dc0dc671f8/tests/test_example_dags_system.py#L24-L31
10. [The Workflow run](https://github.com/apache/airflow/actions/runs/296218970) is cancelling this PR. It has some failed jobs matching ^Pylint$,^Static checks$,^Build docs$,^Spell check docs$,^Backport packages$,^Checks: Helm tests$,^Test OpenAPI*.
11. [The Workflow run](https://github.com/apache/airflow/actions/runs/297764608) is cancelling this PR. It has some failed jobs matching ^Pylint$,^Static checks$,^Build docs$,^Spell check docs$,^Backport packages$,^Checks: Helm tests$,^Test OpenAPI*.
12. [The Workflow run](https://github.com/apache/airflow/actions/runs/297870967) is cancelling this PR. It has some failed jobs matching ^Pylint$,^Static checks$,^Build docs$,^Spell check docs$,^Backport packages$,^Checks: Helm tests$,^Test OpenAPI*.
13. [The Workflow run](https://github.com/apache/airflow/actions/runs/297929320) is cancelling this PR. It has some failed jobs matching ^Pylint$,^Static checks$,^Build docs$,^Spell check docs$,^Backport packages$,^Checks: Helm tests$,^Test OpenAPI*.
14. [The Workflow run](https://github.com/apache/airflow/actions/runs/297982033) is cancelling this PR. It has some failed jobs matching ^Pylint$,^Static checks$,^Build docs$,^Spell check docs$,^Backport packages$,^Checks: Helm tests$,^Test OpenAPI*.
15. [The Workflow run](https://github.com/apache/airflow/actions/runs/297983787) is cancelling this PR. It has some failed jobs matching ^Pylint$,^Static checks$,^Build docs$,^Spell check docs$,^Backport packages$,^Checks: Helm tests$,^Test OpenAPI*.
16. Ran tests local (this time on the _right_ commit) so merging.

github_pulls_reviews:

1. Will we use the whole example in docs? If yes this is ok 👌
2. Should we limit this to minimum?
3. I'm personally in favor of using `None` as schedule in most of examples because when users turn on one the example DAG scheduler will not try to create multiple runs
4. Should we move this to DAG invocation?
5. We should use `@task()` decorator instead of `PythonOperator`, that was the main point of functional DAGs
6. If I'm not mistaken we should be able to do: ``` @task(doc_md="here goes the docs") def extract(**kwargs): ... ```
7. If we want to show the functional approach in full swing we should use: ```python from airflow.operators.python import get_current_context def extract(): ctx = get_current_context() ti = ctx['ti'] ... ```
8. Ok, I see it 😄
9. If I'm not mistaken we should be able to do: ``` @task(doc_md="here goes the docs") def extract(**kwargs): ... ```
10. **body:** I'm wondering if it would be better to define the task before DAG so they can be reusable? No strong opinion here
    label: code-design
11. **body:** And there will be indentation unless we do: ```suggestion #### Transform task A simple Transform task which takes in the collection of order data and computes the total order value. """ ```
    label: code-design
12. I would return directly the dictionary and use `multiple_outputs=True` instead. See: https://github.com/apache/airflow/blob/master/tests/operators/test_python.py#L357
13. ```suggestion @dag.task(multiple_outputs=True) ```
14. ```suggestion return {"total_order_value": total_order_value} ```
15. ```suggestion load(order_summary["total_order_value"]) ```
16. **body:** Yes, I was planning on using the whole example in the docs, unless there were objections to that.

label: documentation
17. I am fine with that.
18. That sentiment makes sense to me, I am fine with that. Will make the change
19. That makes sense. I will make the change.
20. I just did a commit with these changes for both files.
21. I just did a commit with these changes for both files.
22. I just did a commit with these changes for all tasks in both files.
23. I would prefer to leave it inside the DAG, if that's ok with you.
24. ```suggestion ``` Since we are using `@dag.task`
25. **body:** These two lines are the source of the `expected an indented block (comment)` error -- I don't know if we can indent them without messing up the inclusion in docs. Hmmm
    label: code-design
26. This one doesn't need multiple_outputs
27. **body:** Thanks @casassg , got a little confused on this when reading the code in PythonOperator - changed this now.
    label: code-design
28. **body:** You can do ```suggestion @dag.task ``` which is cleaner imo
    label: code-design
29. ```suggestion @dag.task ```
30. Would be nice to have this point to https://github.com/apache/airflow/blob/master/docs/concepts.rst#functional-dags and https://github.com/apache/airflow/blob/master/docs/concepts.rst#functional-dags more specifically. Also, it may be interesting to link the AIP-31 for more insight into this and potentially my talk and Jonathan's article as learn more resources? CC @turbaszek for thoughts
31. I think this creates mypy issues https://github.com/apache/airflow/pull/10930#discussion_r489375827 :<
32. +1 for linking AIP and this article for more insight: https://medium.com/databand-ai/aip-31-airflow-functional-dag-definition-b34852a632d0
33. **body:** I added the links to the AIP and to the Functional DAGs section of the Concept doc. I was not sure if it was ok to add a link to an external site which has a paywall (Medium) from the Apache open source docs, so did not add that.
    label: documentation

jira_issues:

jira_issues_comments: