How to Execute Tasks According To Criteria Using Branching

Choose the path

Use Case

- Let's say we have a task fetching a record from a PostgreSQL's table where the value corresponds to the database currently activated in our system. Then, we have multiple tasks where only one should be executed according to this value. How could we do that?
- One way to do this would be to create one DAG for each possible value. Obviously, even if you factorize very well your code, at the end you would end up with many DAGs having to be managed by the scheduler which can dramatically reduce your performances. Also, it can be just impossible if you have too many values for the associated record.
- Again, Apache Airflow has the solution which is Branching.

Definition

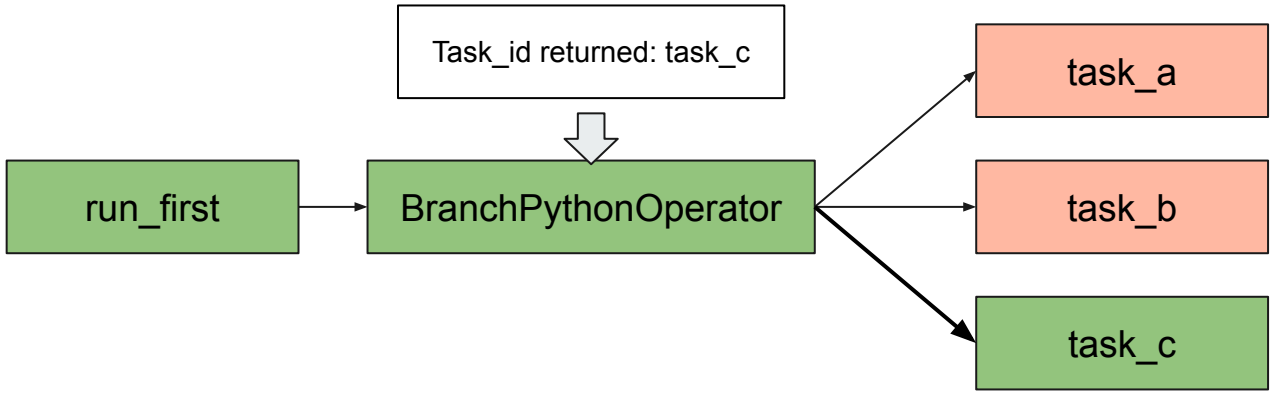
Branching is the mechanism allowing your DAG to choose between different paths according to the result of a specific task.

To do this we will use the BranchPythonOperator.

How Does It Work?

- The BranchPythonOperator is like the PythonOperator except that it expects a python_callable function that returns a task_id. In other words, the function passed to the parameter python_callable must return the task_id corresponding to the task which will be executed next.
- All other paths are skipped and only the path leading to the task with the corresponding task_id will be followed.
- The task_id returned by the Python function has to be referencing a task directly downstream from the BranchPythonOperator task.

Schema


Task_id returned: task_c
run_first
BranchPythonOperator
task_a
task_b
task_c

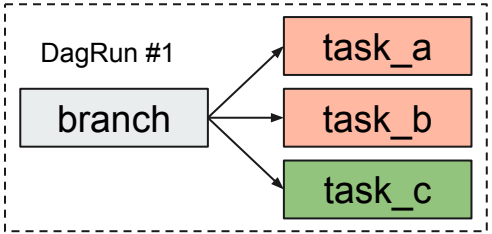
Depends_on_past and Branching

- You can use the property `depends_on_past` at the task level. It means this task will run only if the same task instance succeed in the previous DagRun. If there is no previous DagRun, the task will be triggered. Now you know that, there is no point to use `depends_on_past=True`on downstream tasks from the BranchPythonOperator, as skipped status will invariably lead to block tasks that depend on their past successes.

Depends_on_past and Branching

Let's imagine we have the following DAG where all downstream tasks have the property `depends_on_past=True` .So this is the first DagRun:

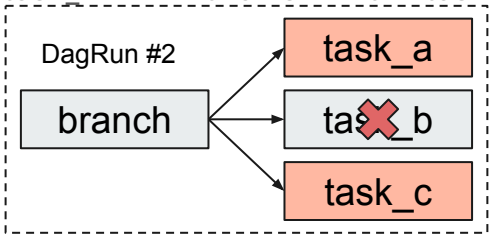
The task_c is chosen and all other tasks are "skipped".



If we run a second time the DAG and now the chosen task by the BranchPythonOperator is task_b what do you think it is going to happen?

Depends_on_past and Branching

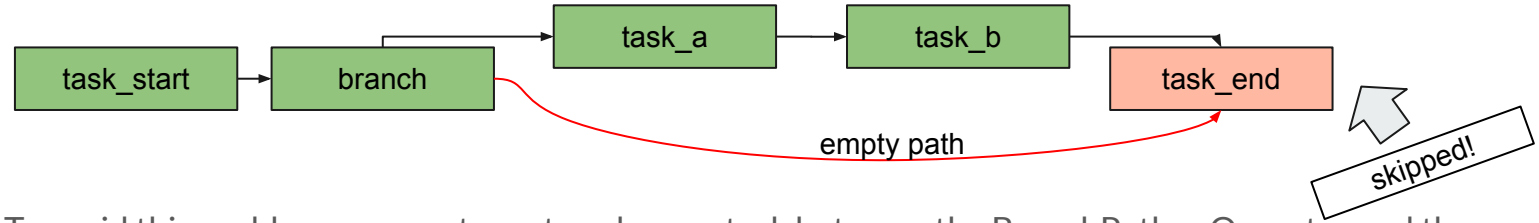
task_b will not run since the taskInstance of task_b from the previous DagRun did not succeed.



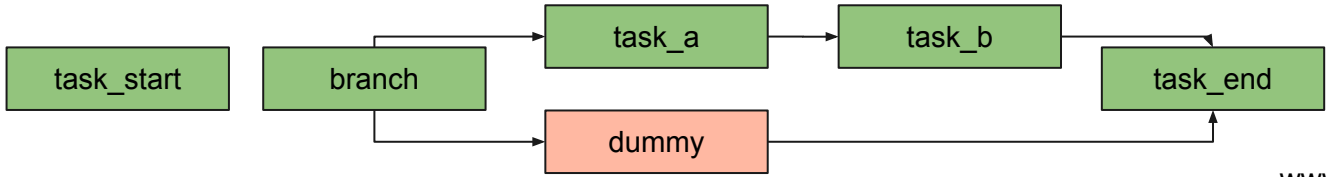
If you run a third time the DAG and now the BranchPythonOperator choose to go to task_a, the task_a will not be triggered as well and so you just have totally locked your data pipeline.

Important Notes

- If you want to skip some tasks, keep in mind that you can't have an empty path. If so, you have to make a dummy task. For example if you have a DAG like the following, the task_end will be skipped.



- To avoid this problem you must create a dummy task between the BranchPythonOperator and the task_end like this:



Coding Time!

Let’s go!