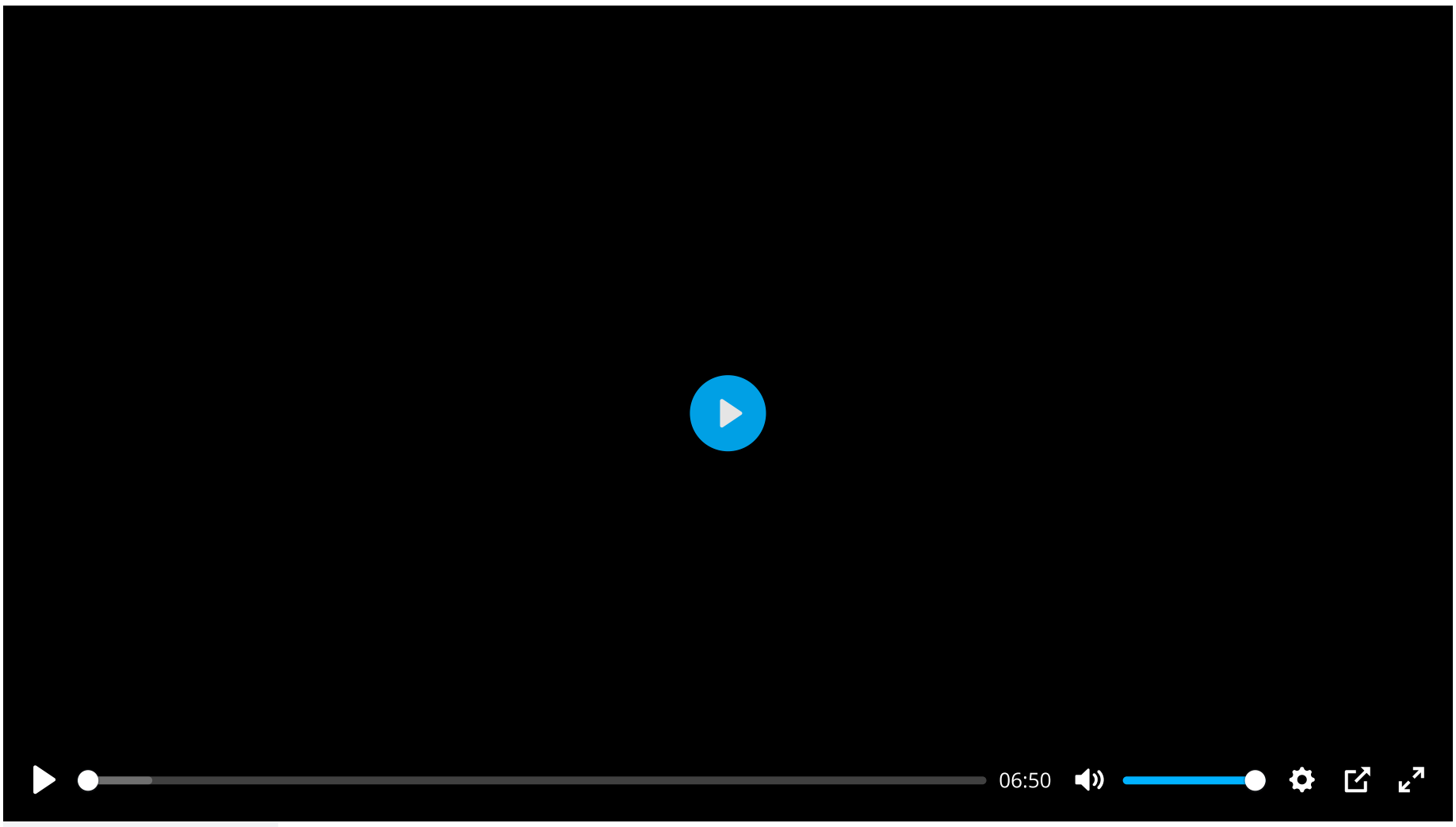
8.5 Настройка конфигурации оркестратора и мониторинг



Презентация к уроку 8.5.3

Текстовая расшифровка видео:

ВЕТВЛЕНИЕ ТАСКОВ

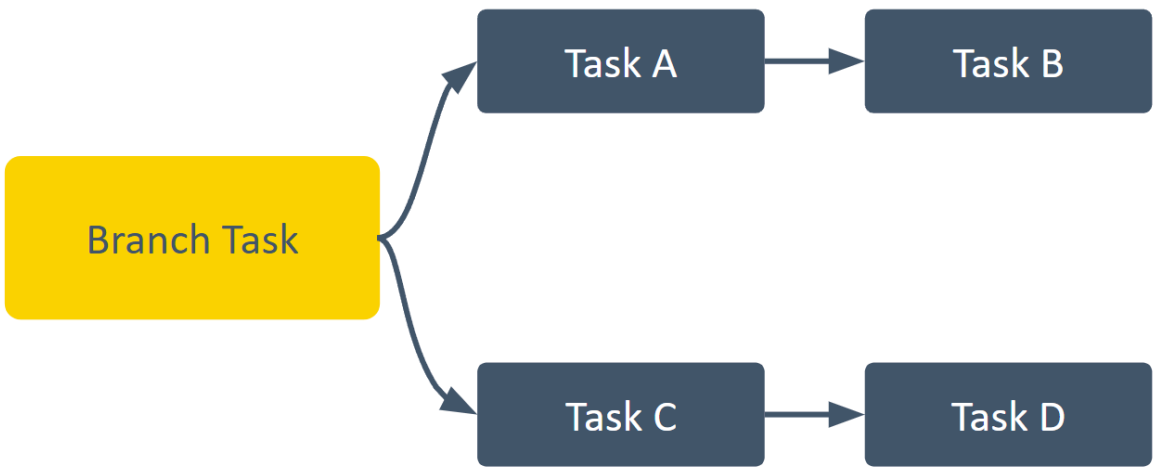
План:

- BranchOperator (ветвление тасков по условию);
- Реализация BranchOperator;
- Условия ветвления;
- Trigger Rules;
- Ветвление в WebUI;
- Разбор примера.

BranchOperator (ветвление тасков по условию)

Если нам нужно исполнение цепочки определенных тасков в зависимости от условия, мы можем использовать BranchOperator.

BranchOperators бывают разными. В их основе может лежать Python-оператор или какой-либо другой, но они возвращают по своему исполнению ID task, который должен исполниться следующим.



Реализация BranchOperator

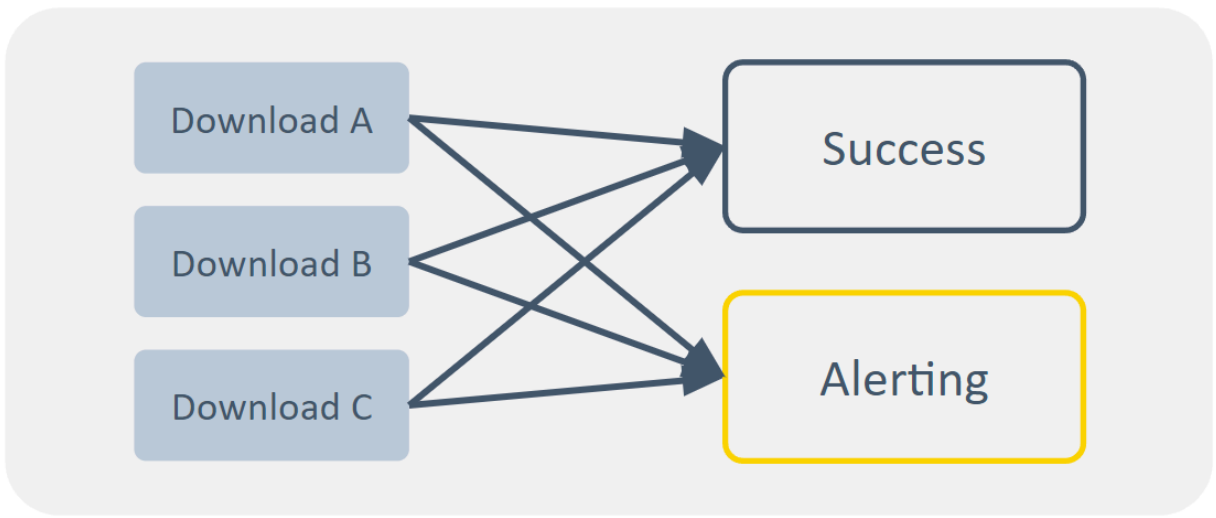
Перед вами пример реализации:

```
def branch_func(**kwargs):
    xcom_value = int(kwargs['ti'].xcom_pull(task_ids='start_task'))
    if xcom_value >= 5:
        return 'continue_task'
    else:
        return 'stop_task'

start_op = BashOperator(task_id='start_task',bash_command="echo 10",dag=dag)
branch_op = BranchPythonOperator(task_id='branch_task',provide_context=True,python_callable=branch_fun
continue_op = DummyOperator(task_id='continue_task', dag=dag)
```

В коде есть некий BranchOperator, который реализует функцию branch_func. Эта функция получает значение Xcom и на основании его значения принимает решение: либо продолжается ветка по айдишнику таска «continue_task», либо по ветке «stop_task», если Xcom меньше пяти.

Условия ветвления

Исполнение тасков по некоторым условиям может использоваться не для BranchOperator'а, а для любых тасков в зависимости от их статуса. По умолчанию таски переходят к следующему, когда все они выполнены успешно – **all success.** Можно изменить это поведение. Рассмотрим пример:



У нас есть три таска: «Download A», «Download B», «Download C», которые при выполнении «Успешно» переходят к ветке **Success**. В случае, когда хотя бы один из них «Неуспешно» выполняется Alerting (какое-то предупреждение).

Trigger Rules

Trigger Rules около десяти:

- **all_success**

Ставится по умолчанию: если Task A и Task B успешны, то переходим к Task C.

- **all_failed**

- **all_done**

Все таски должны быть либо успешными, либо пропущенными (розовая рамочка), тогда мы перейдем к Task C.

- **one_success**

Последующий таск выполняется, когда хотя бы один из предыдущих был выполнен успешно.

- **one_failed**

Хотя бы один из предыдущих тасков был зафейлен (это наиболее подходит для нашего примера выше).

- **none_failed**

Хотя бы один из предыдущих тасков должен быть не failed. Он может быть пропущенным.

- **none_failed_min_one_success**

Из всех предыдущих тасков хотя бы один таск должен быть выполнен успешно.

- **none_skipped**

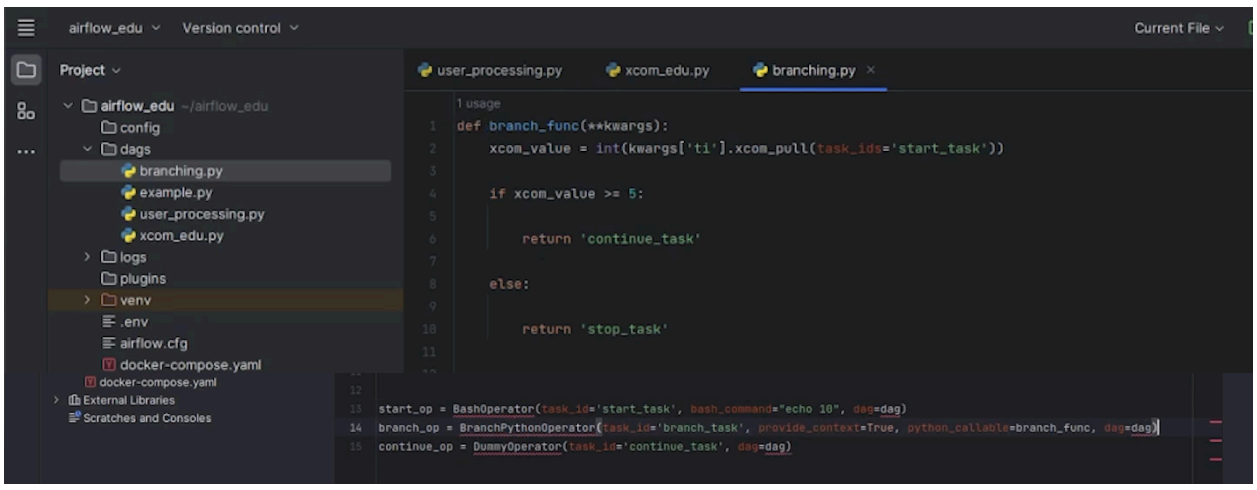
Ни один из предыдущих тасков не должен быть пропущен.

Ветвление в WebUI

Больше информации о branching можно получить в статье: https://docs.astronomer.io/learn/airflow-branch-operator

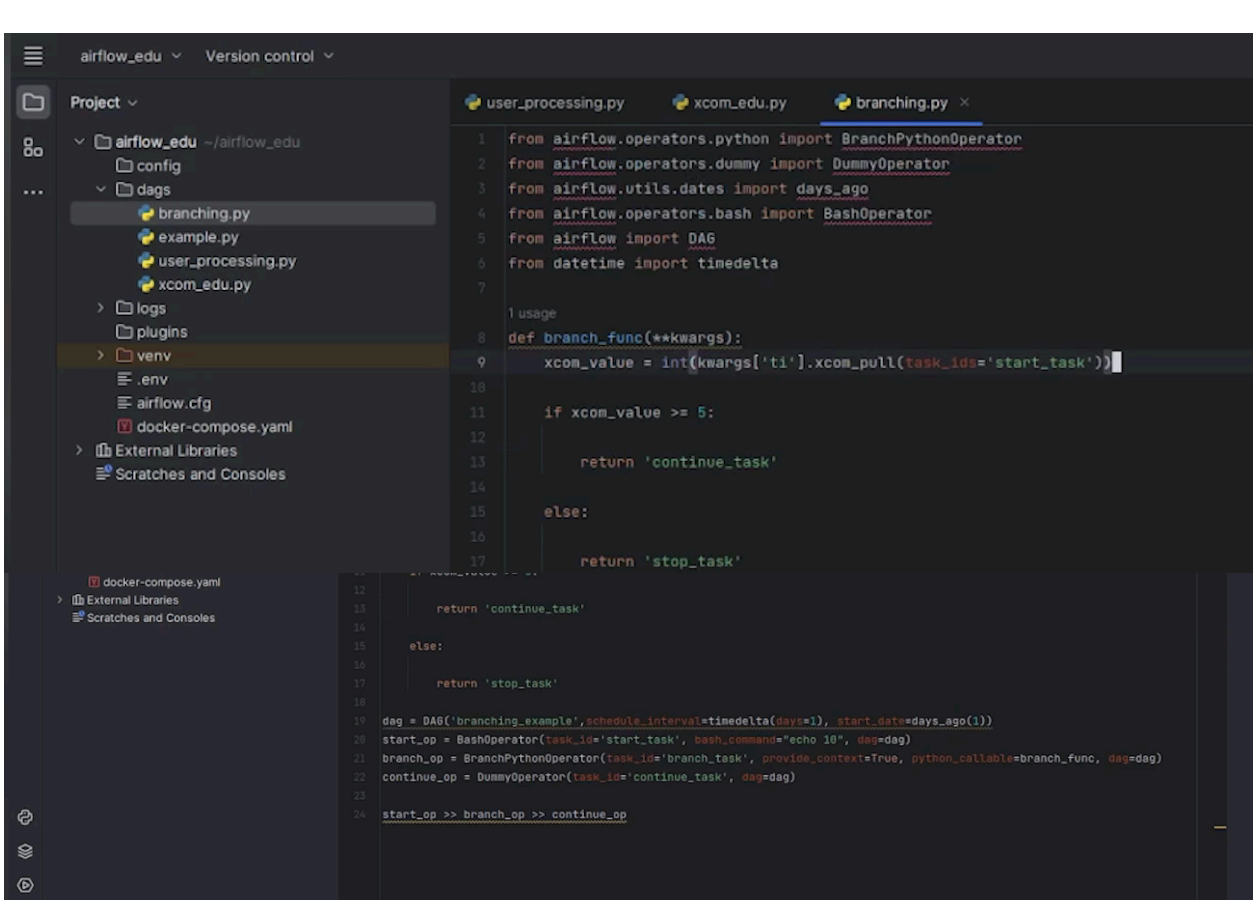
Разбор примера

Код, который мы будем рассматривать, находится в ваших дополнительных материалах.



Здесь есть функция «Branch_fanc», реализующая branching.

Добавим необходимые импорты:



Дедлайн 04 августа, 23:59 Мск

```
File  Edit  View  Search  Terminal  Help
asya@asya-Aspire-XC-886:~/airflow_edu$ cd dags
asya@asya-Aspire-XC-886:~/airflow_edu/dags$ ls -lh
total 20K
-rw-rw-r-- 1 asya  asya  830 Jun 28 09:55 branching.py
-rw-rw-r-- 1 asya  asya   84 Jun 28 09:28 example.py
drwxrwxr-x 2 50000 root 4.0K Jun 28 09:57 __pycache__
-rwxrwxrwx 1 50000 root 2.2K Jun 26 11:22 user_processing.py
-rw-rw-r-- 1 50000 root  583 Jun 26 12:03 xcom_edu.py
asya@asya-Aspire-XC-886:~/airflow_edu/dags$ chmod 777 branching.py
asya@asya-Aspire-XC-886:~/airflow_edu/dags$
```

Видим код:

Parsed at: 2023-06-28, 13:58:23

```python
from airflow.operators.python import BranchPythonOperator
from airflow.operators.dummy import DummyOperator
from airflow.utils.dates import days_ago
from airflow.operators.bash import BashOperator
from airflow import DAG
from datetime import timedelta

def branch_func(**kwargs):
    xcom_value = int(kwargs['ti'].xcom_pull(task_ids='start_task'))

    if xcom_value >= 5:

        return 'continue_task'

    else:

        return 'stop_task'

dag = DAG('branching_example',schedule_interval=timedelta(days=1), start_date=days_ago(1))
start_op = BashOperator(task_id='start_task', bash_command="echo 10", dag=dag)
branch_op = BranchPythonOperator(task_id='branch_task', provide_context=True, python_callable=branch_func, dag=dag)
continue_op = DummyOperator(task_id='continue_task', dag=dag)

start_op >> branch_op >> continue_op
```

Version: v2.6.2
Git Version: .release:d2f0d100dac4a95d664309d7b04a6a6110367446

В случае, когда xcom больше пяти – **continues**, в случае, если меньше, то **stop task**.

Пробуем запустить. В коде было значение больше. В xcom'ах значение «10», которое было указано в коде, поэтому мы пошли по ветке «Continues»:

Task Instance: start_task at 2023-06-27, 00:00:00

Task Instance Details | Rendered Template | Log | XCom

XCom

Key	Value
return_value	10

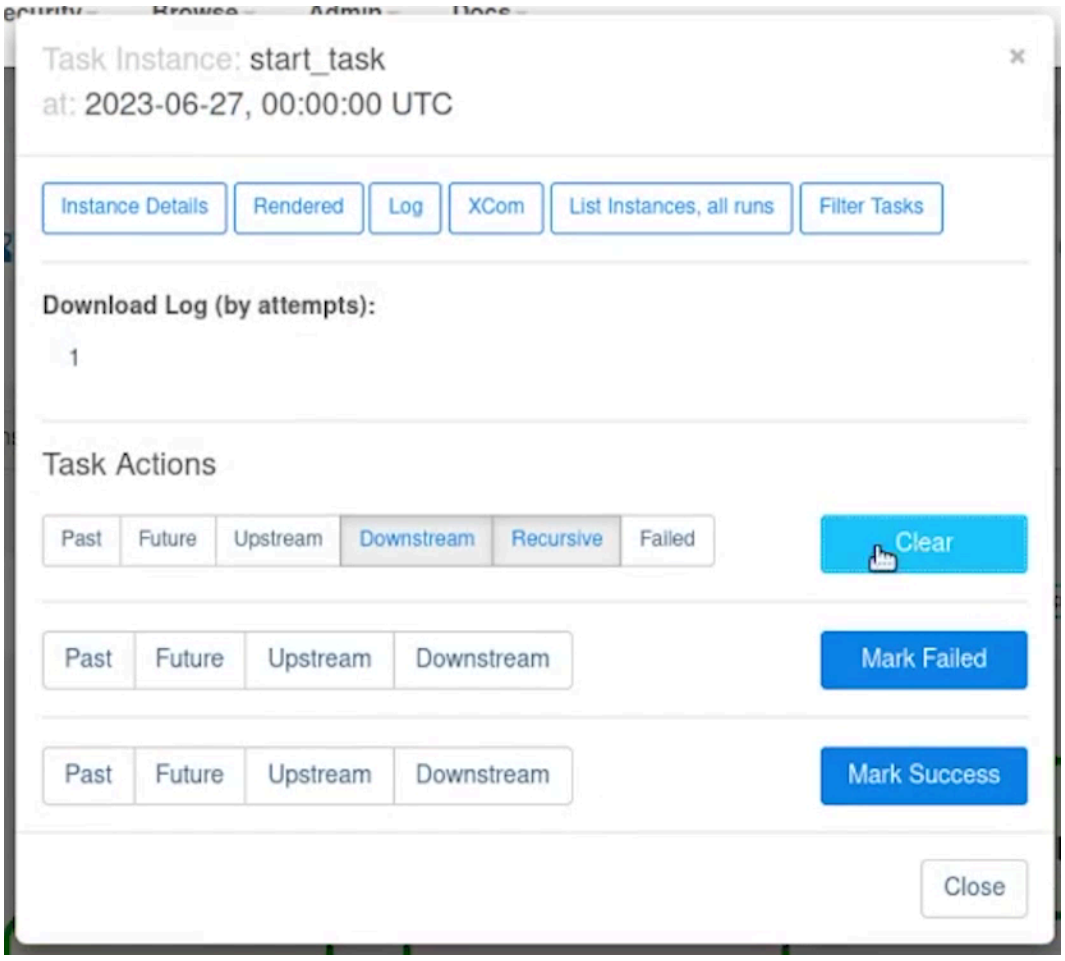
Если исправить это значение на меньшее и добавить ветку «Stop», таск сработает следующим образом:

```python
        return 'stop_task'

dag = DAG('branching_example',schedule_interval=timedelta(days=1), start_date=days_ago(1))
start_op = BashOperator(task_id='start_task', bash_command="echo 1", dag=dag)
branch_op = BranchPythonOperator(task_id='branch_task', provide_context=True, python_callable=branch_func, dag=dag)
continue_op = DummyOperator(task_id='continue_task', dag=dag)
stop_op = DummyOperator(task_id='stop_task', dag=dag)

start_op >> branch_op >> [continue_op, stop_op]
```

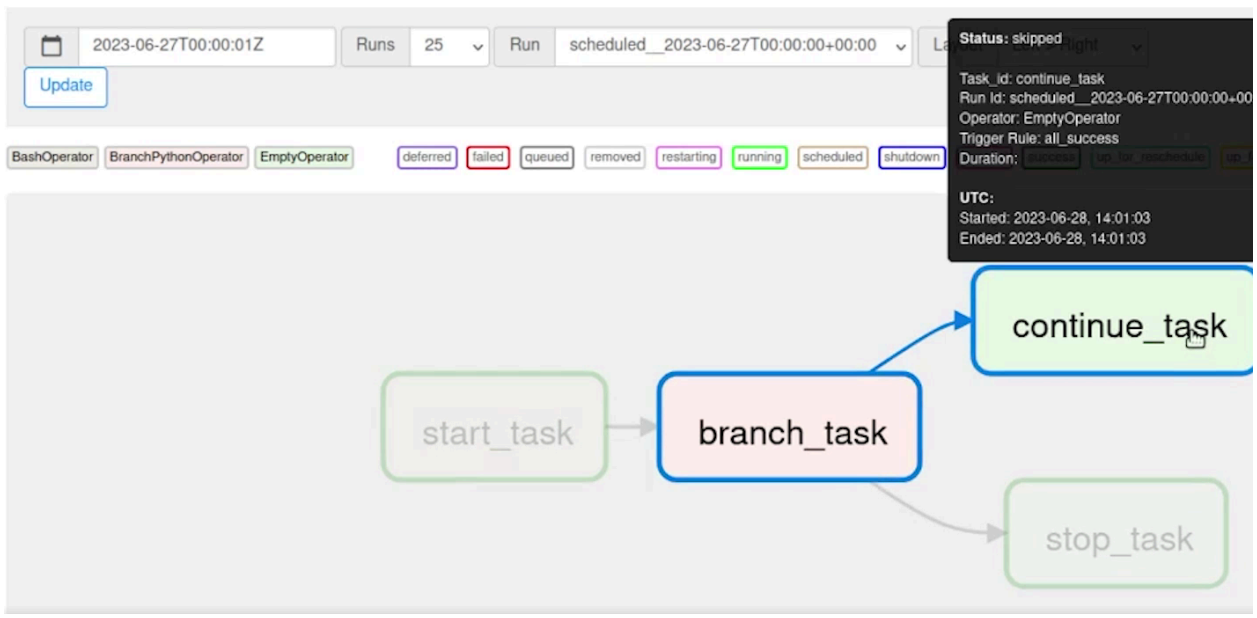
Мы сгруппировываем квадратные скобки, так как они находятся на одном уровне.

Чистим все предыдущие инстансы таска выполнения:

Дедлайн 04 августа, 23:59 Мск



Запускаем еще раз и видим **continue task skipped**:



Невыбранная ветка пропускается и отмечается розовой рамочкой.

Мы познакомились с оператором ветвления и научились его применять, также мы рассмотрели случаи, в которых данный оператор может быть полезен.

Как вам урок?

Изучил, далее

Дедлайн 04 августа, 23:59 Мск