

ETL. Семинар №4

1. Викторина
2. Блок 1

Задание 1

Создайте новый граф, добавьте в него два BashOperator, первый должен выводить на экран сообщение "Hello from Airflow", второй должен брать код из bash файла. Создайте bash файл который будет выводить сообщение "Hello from Airflow bash script processor." Создайте последовательную связь между первым и вторым оператором.

Добавьте между первым и вторым операторами еще один BashOperator который будет заканчивать свою работу в статусе skipped



30 минут

Решение:

```
from datetime import datetime
from airflow import DAG
from airflow.operators import BashOperator, PythonOperator

dag = DAG( 'hello_world' , description= 'Hello World DAG' ,
    schedule_interval= '0 12 * * *' ,
    start_date=datetime( 2023 , 1 , 1
    ), catchup= False )

hello_operator = BashOperator(task_id= 'hello_task' , bash_command='echo Hello from Airflow', dag=dag)
skipp_operator = BashOperator(task_id= 'skip_task' , bash_command='exit 99', dag=dag)
hello_file_operator = BashOperator(task_id= 'hello_file_task' ,
    bash_command='./home/airflow/airflow/dags/scripts/file1.sh', dag=dag)

hello_operator >> skipp_operator >> hello_file_operator
```

Задание 2

В графе из предыдущего задания поменяйте BashOperator на pythonOperator функционал измениться не должен.



10 минут

Решение:

```
from datetime import datetime
from airflow import DAG
from airflow.operators import BashOperator, PythonOperator
def print_hello ():
    return 'Hello world from Airflow DAG!'

def skipp():
    return 99
dag = DAG( 'hello_world' , description= 'Hello World DAG' ,
    schedule_interval= '0 12 * * *' ,
    start_date=datetime( 2023 , 1 , 1
    ), catchup= False )

hello_operator = PythonOperator(task_id= 'hello_task' , python_callable=print_hello, dag=dag)
skipp_operator = PythonOperator(task_id= 'skip_task' , python_callable=skipp, dag=dag)
hello_file_operator = BashOperator(task_id= 'hello_file_task' , bash_command='python
/home/airflow/airflow/dags/scripts/file1.py', dag=dag)

hello_operator >> skipp_operator >> hello_file_operator
```

Задание 3

Добавьте в граф `httpSensor` который будет обращаться к сайту `gb.ru`.
Отправьте в чат скриншот кода и логи работы



15 минут

Решение:

Добавить в предыдущий файл:

```
task_http_sensor_check = HttpSensor(
    task_id="http_sensor_check",
    http_conn_id="http_default",
    endpoint="",
    request_params={},
    response_check=lambda response: "httpbin" in response.text,
    poke_interval=5,
    dag=dag,
)

hello_operator >> skippp_operator >> hello_file_operator >> task_http_sensor_check
```

Задание 4

Создайте `ShortCircuitOperator` и измените граф таким образом чтобы было видно как выполняются два различных сценария в зависимости от результатов работы данного оператора.



15 минут

Решение:

```
from datetime import datetime
from airflow import DAG
from airflow.operators import BashOperator, PythonOperator, HttpSensor, ShortCircuitOperator

def visit_gb():
    #If you want to visit gb.ru site return True else False
    return False

def print_hello():
    return 'Hello world from Airflow DAG!'

def skipp():
    return 99

dag = DAG('hello_world', description='Hello World DAG',
          schedule_interval='0 12 * * *',
          start_date=datetime(2023, 1, 1),
          catchup=False)

hello_operator = PythonOperator(task_id='hello_task', python_callable=print_hello, dag=dag)
skipp_operator = PythonOperator(task_id='skip_task', python_callable=skipp, dag=dag)
hello_file_operator = BashOperator(task_id='hello_file_task', bash_command='python
/home/airflow/airflow/dags/scripts/file1.py', dag=dag)

task_http_sensor_check = HttpSensor(
    task_id='http_sensor_check',
    http_conn_id='http_default',
    endpoint='',
    request_params={},
    response_check=lambda response: "httpbin" in response.text,
    poke_interval=5,
    dag=dag,
)

visit_site = ShortCircuitOperator(
    task_id='visit_gb',
    provide_context=False,
    python_callable=visit_gb,
    dag=dag)
```

```
hello_operator >> skipp_operator >> hello_file_operator >> visit_site >> task_http_sensor_check
```