

Airflow

<https://airflow.apache.org/docs/apache-airflow/1.10.10/>

1. Скачал контейнер из <https://github.com/puckel/docker-airflow>
2. Распаковал, перешел в папку, где он был распакован и запустил:

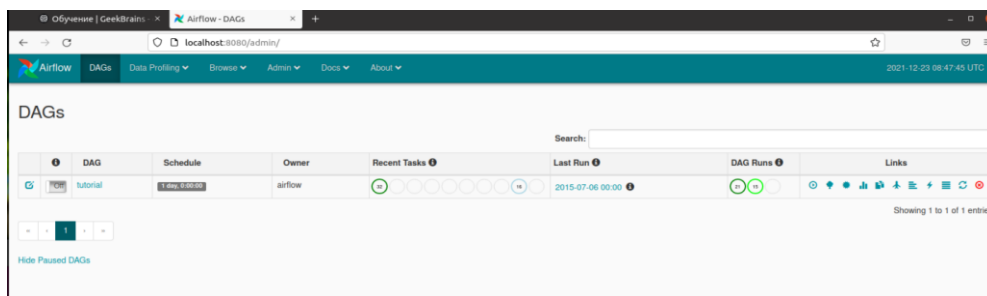
`docker-compose -f docker-compose-LocalExecutor.yml up -d`

`docker exec -it docker-airflow-master_webserver_1 bash`

```
max@max:~/Downloads/docker-airflow-master$ docker-compose -f docker-compose-LocalExecutor.yml up -d
docker-airflow-master_postgres_1 is up-to-date
docker-airflow-master_webserver_1 is up-to-date
max@max:~/Downloads/docker-airflow-master$ docker exec -it docker-airflow-master_webserver_1 bash
airflow@daf131963fb3:~$
```

3. В `docker-compose-LocalExecutor.yml` посмотрел проброшенный порт (8080) для аэифлоу. Открыл вебморду в браузере

`localhost:8080`



4. Создаю ПИтоновский скрипт и заливаю его в папку dags https://github.com/stavrtin/Hadoop/blob/main/Less_6_Airflow/hw_6.py

```
1 from airflow import DAG
2 #from airflow.operators.bash_operator import BashOperator
3 from airflow.operators.python_operator import PythonOperator, BranchPythonOperator
4 from airflow.models import Variable
5 from random import randint
6
7 from datetime import datetime, timedelta
8
9
10 default_args = {
11     'owner': 'airflow',
12     'depends_on_past': False,
13     'start_date': datetime(2015, 6, 1),
14     'retries': 1,
15     'retry_delay': timedelta(minutes=5),
16     'trigger_rule': 'none_failed'
17 }
18
19 dag = DAG(
20     dag_id='hw_7',
21     default_args=default_args,
22     schedule_interval=None,
23     catchup = False
24 )
25
26 def task_1_f(**kwargs):
27     rand_step = randint(1,3)
28     rand_step = 3
29     if rand_step == 1 or rand_step == 2:
30         next_step = eval(Variable.get('hw_etl')).get(rand_step)
31     else:
32         next_step = eval(Variable.get('hw_etl')).get(rand_step)
33     next_step = ['task_6']
34     return next_step
35
36 def task_2_f(x, **kwargs):
37     print(x)
38
39 task_1 = BranchPythonOperator(
40     task_id = 'task_1',
41     python_callable=task_1_f,
42     dag=dag
43 )
```

```
44
45 task_2 = PythonOperator(
46     task_id = 'task_2',
47     python_callable=task_2_f,
48     op_kwargs = {'x': 2},
49     dag=dag
50 )
51
52 task_3 = PythonOperator(
53     task_id = 'task_3',
54     python_callable=task_2_f,
55     op_kwargs = {'x': 3},
56     dag=dag
57 )
58
59 task_4 = PythonOperator(
60     task_id = 'task_4',
61     python_callable=task_2_f,
62     op_kwargs = {'x': 4},
63     dag=dag
64 )
65
66 task_5 = PythonOperator(
67     task_id = 'task_5',
68     python_callable=task_2_f,
69     op_kwargs = {'x': 5},
70     trigger_rule = "one_success",
71     dag=dag
72 )
73
74 task_6 = PythonOperator(
75     task_id = 'task_6',
76     python_callable=task_2_f,
77     op_kwargs = {'x': 6},
78     trigger_rule = "all_done",
79     dag=dag
80 )
81
82 #all success - no
83 #all failed - no
84
85
```

```
71 dag=dag
72
73 task_6 = PythonOperator(
74     task_id = 'task_6',
75     python_callable=task_2_f,
76     op_kwargs = {'x': 6},
77     trigger_rule = "all_done",
78     dag=dag
79 )
80
81 #all success - no
82 #all failed - no
83
84 # upstream failed - pyraercs
85 # all done - zapaborano !!!!
86 # one failed
87 # one success
88 # none failed
89 # none failed_or_skipped
90 # none skipped
91 # skipped
92 # dummy
93
94 #none failed_or_skipped - pyraercs
95
96
97
98
99
100
101 task_1 >> [task_3, task_4] >> task_5
102 task_1 >> task_2 >> task_6
103 task_5 >> task_6
```

5. В резултате:

