Apache Airflow: семь раз примерь, один раз зашедуль

Петр Ермаков
Lamoda / DataGym

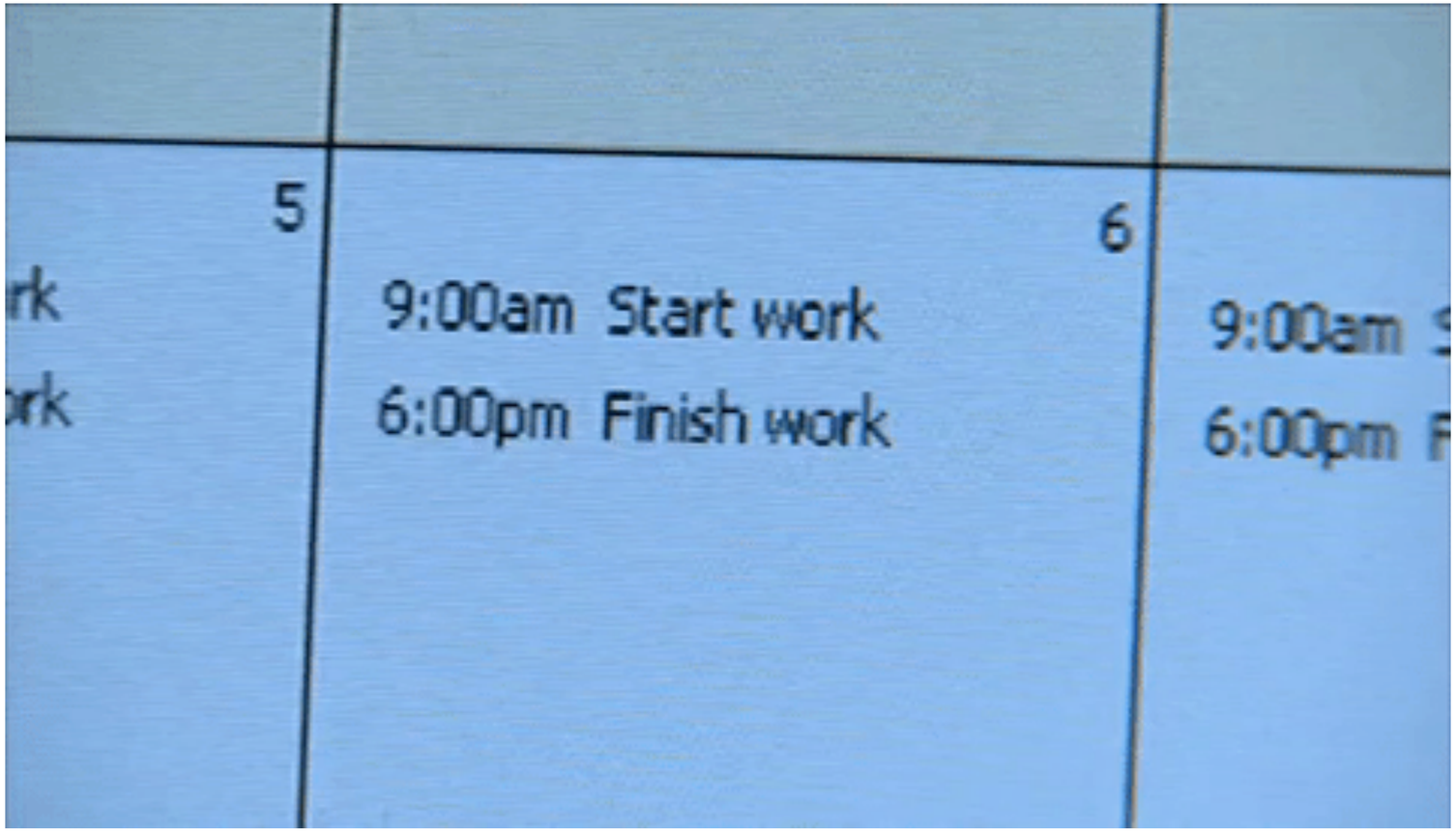
DataGym

lamoda

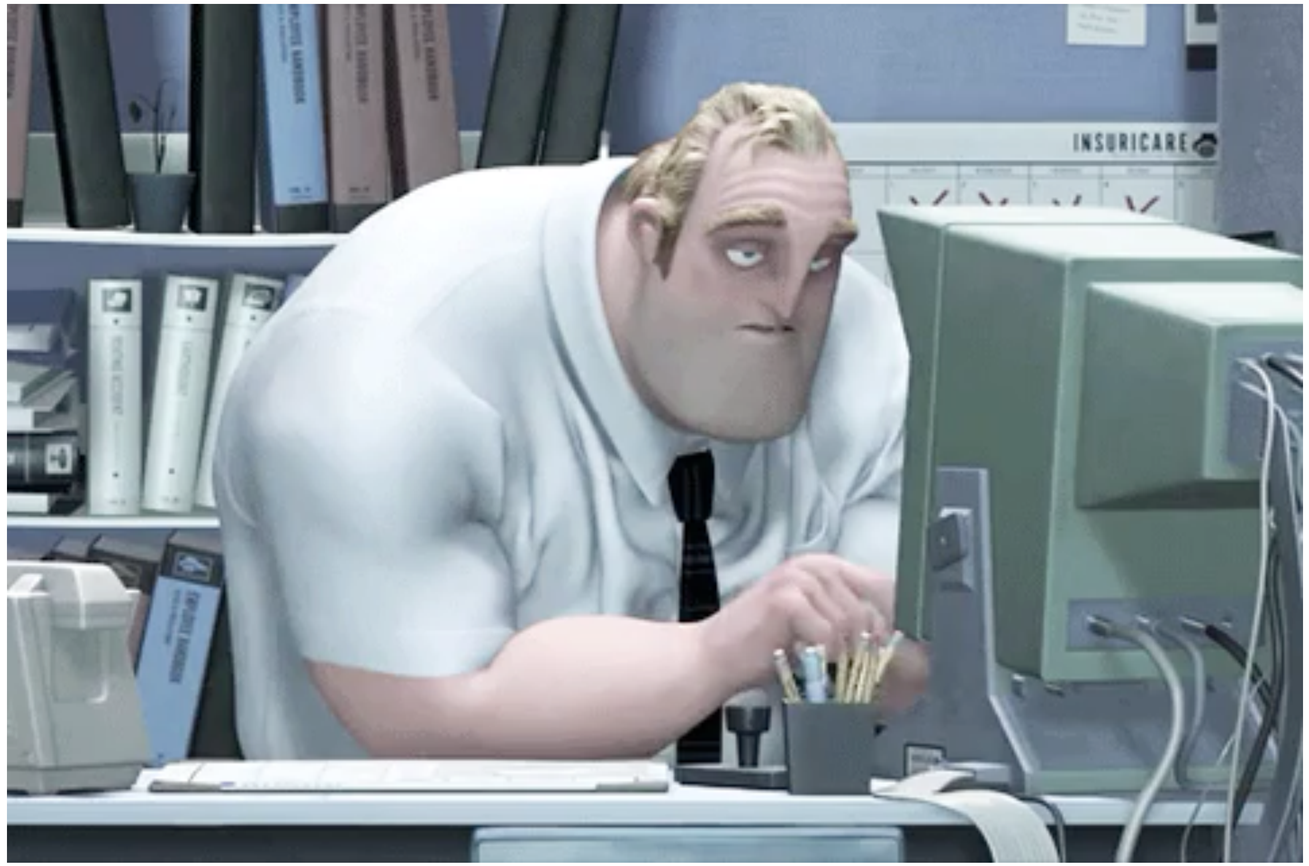
Agenda

- Мотивация
- Альтернативы
- Airflow
- Best Practices

Мотивация



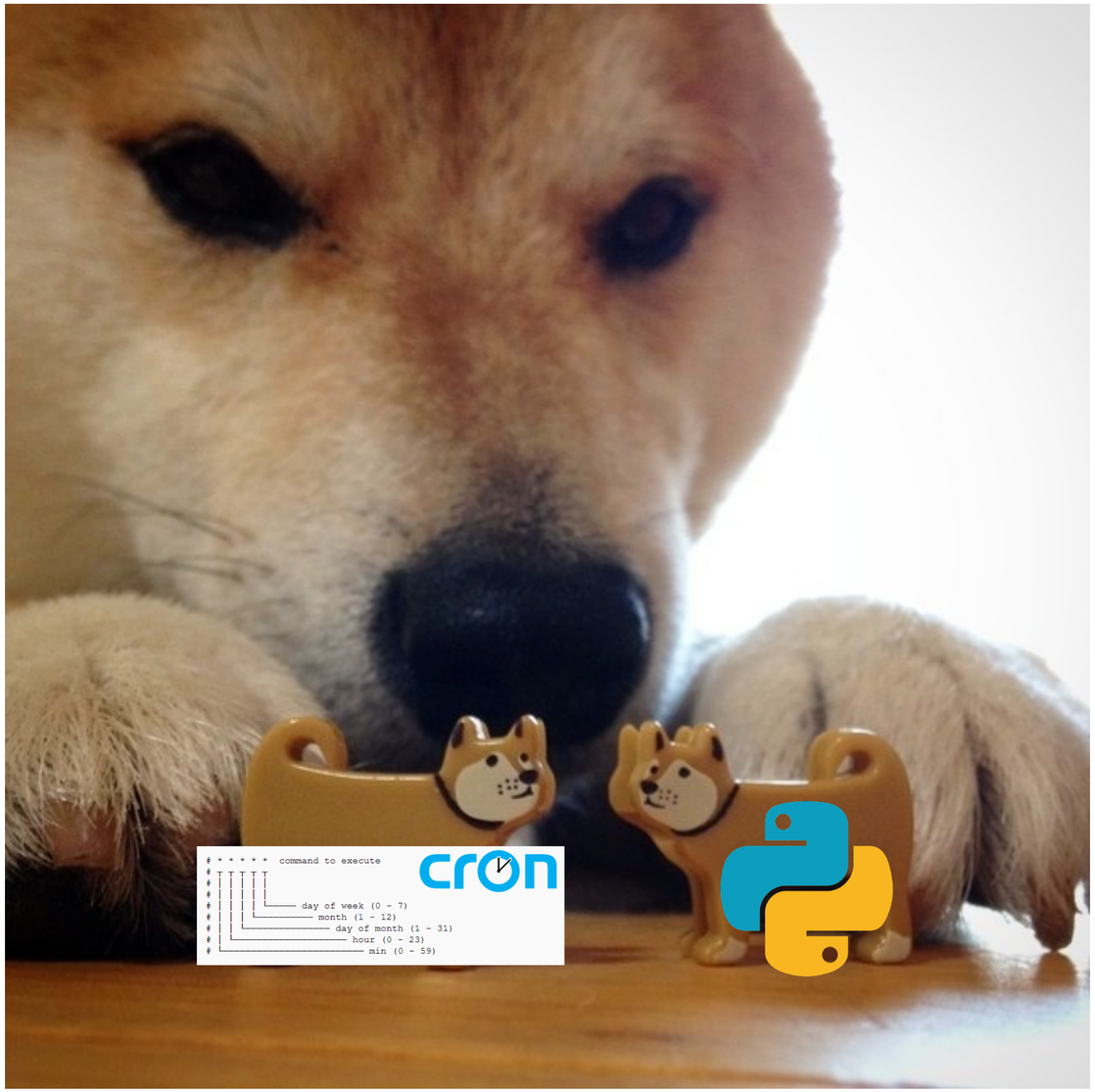
Мотивация



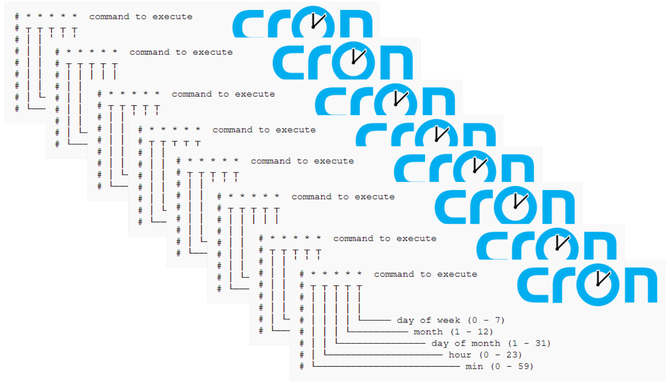
Мотивация

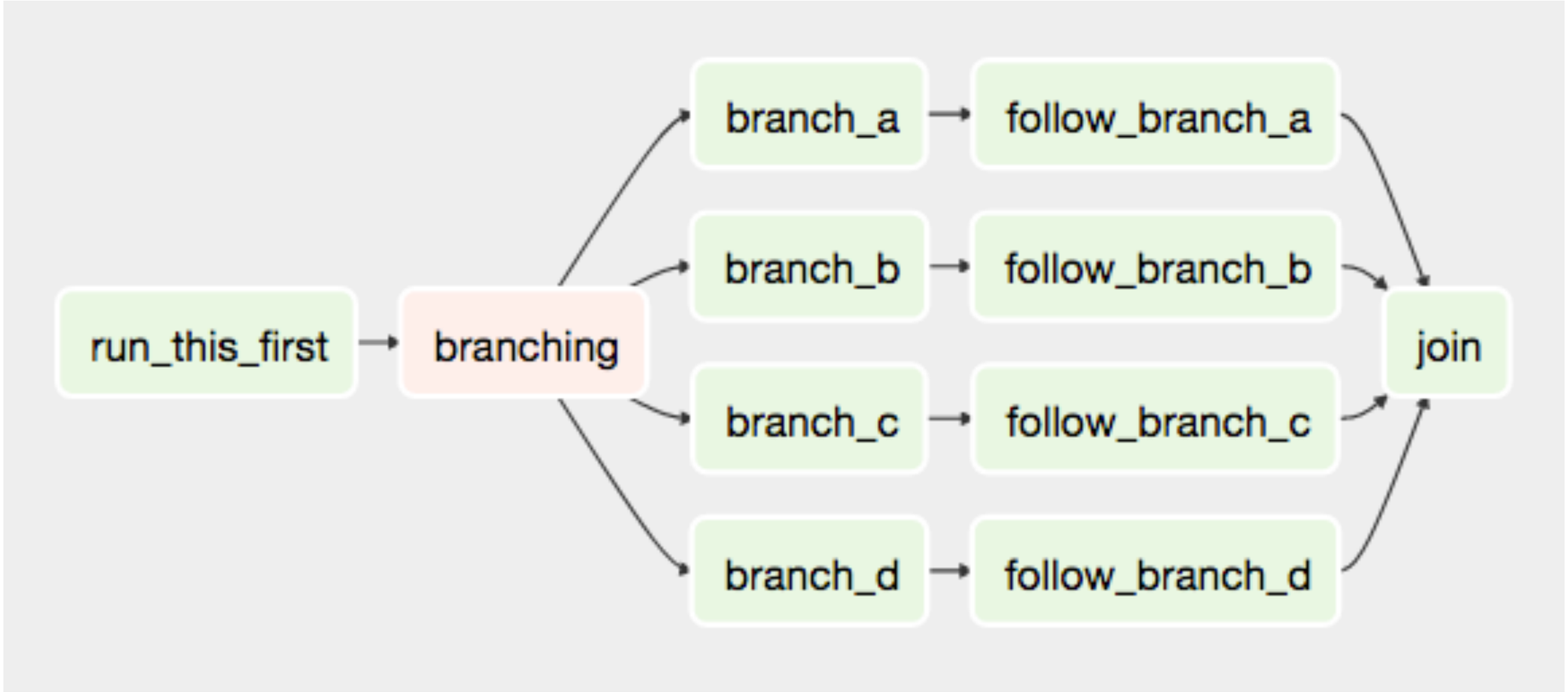
- Нужно запустить выгрузку в 8:00 из DB
- При загрузке данных в базу, сделать инференс всех новых строк?
- Если до 12:00 неактивный Slack, написать “Работаю из дома” в чат
- Найти и полайкать все новые фотографии жены, сестры, мамы, тещи
- Сделать все это, в определенном порядке и передавая состояния между задачами

Мотивация


cron
* * * * * command to execute
┬ ┬ ┬ ┬ ┬
│ │ │ │ │
│ │ │ │ └──── day of week (0 - 7)
│ │ │ └────── month (1 - 12)
│ │ └──────── day of month (1 - 31)
│ └────────── hour (0 - 23)
└──────────── min (0 - 59)

Мотивация


* * * * * command to execute
day of week (0 - 7)
month (1 - 12)
day of month (1 - 31)
hour (0 - 23)
min (0 - 59)


run_this_first
branching
branch_a
branch_b
branch_c
branch_d
follow_branch_a
follow_branch_b
follow_branch_c
follow_branch_d
join

Инструменты

- Oozie (https://github.com/apache/oozie)
- Luigi (https://github.com/spotify/luigi)
- AirFlow (https://github.com/apache/airflow)
- Azkaban (https://github.com/azkaban/azkaban)
- Conductor (https://github.com/Netflix/conductor)
- Pinball (https://github.com/pinterest/pinball)
- Chronos (https://github.com/mesos/chronos)

AirFlow

- Airflow was started in October 2014 by Maxime Beauchemin at Airbnb
- It was open source from the very first commit and officially brought under the Airbnb GitHub and announced in June 2015
- The project joined the Apache Software Foundation’s Incubator program in March 2016
- The Foundation announced Apache Airflow as a Top-Level Project in January 2019.

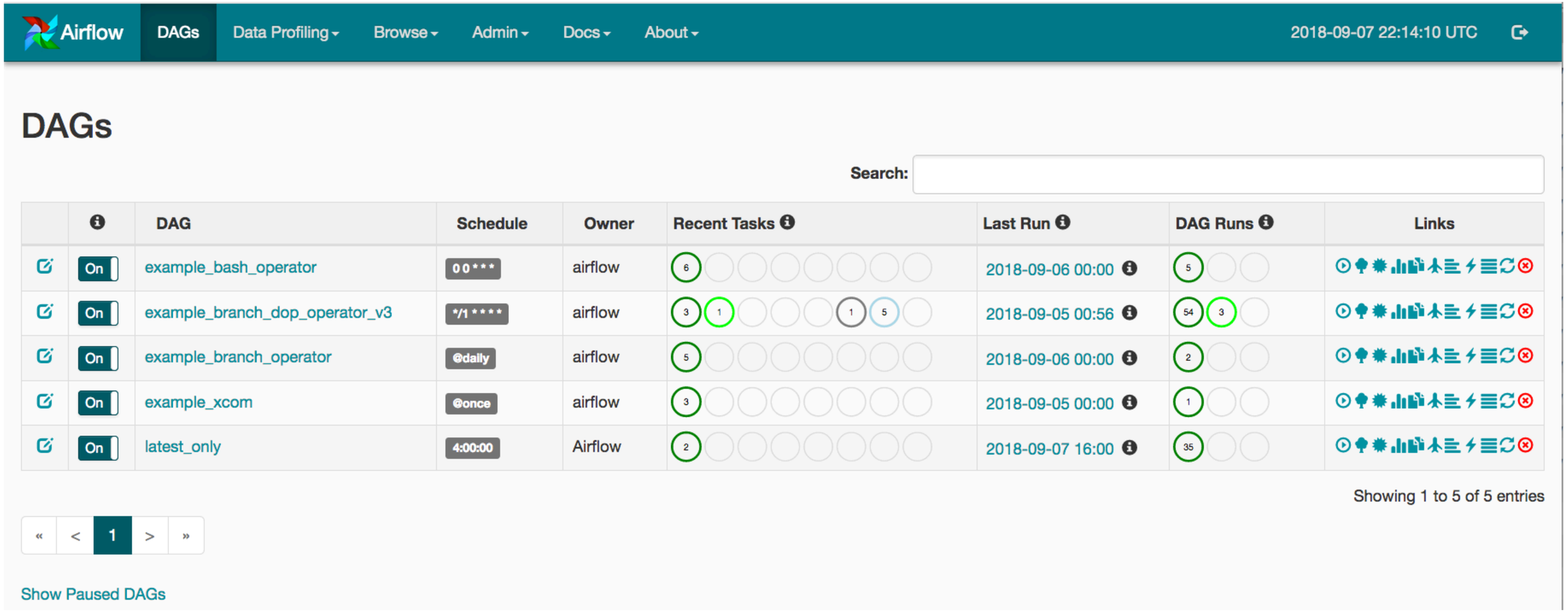
AirFlow

- Python 2.7, 3.5, 3.6, 3.7
- Flask
- Jinja
- Relationships database
- LDAP

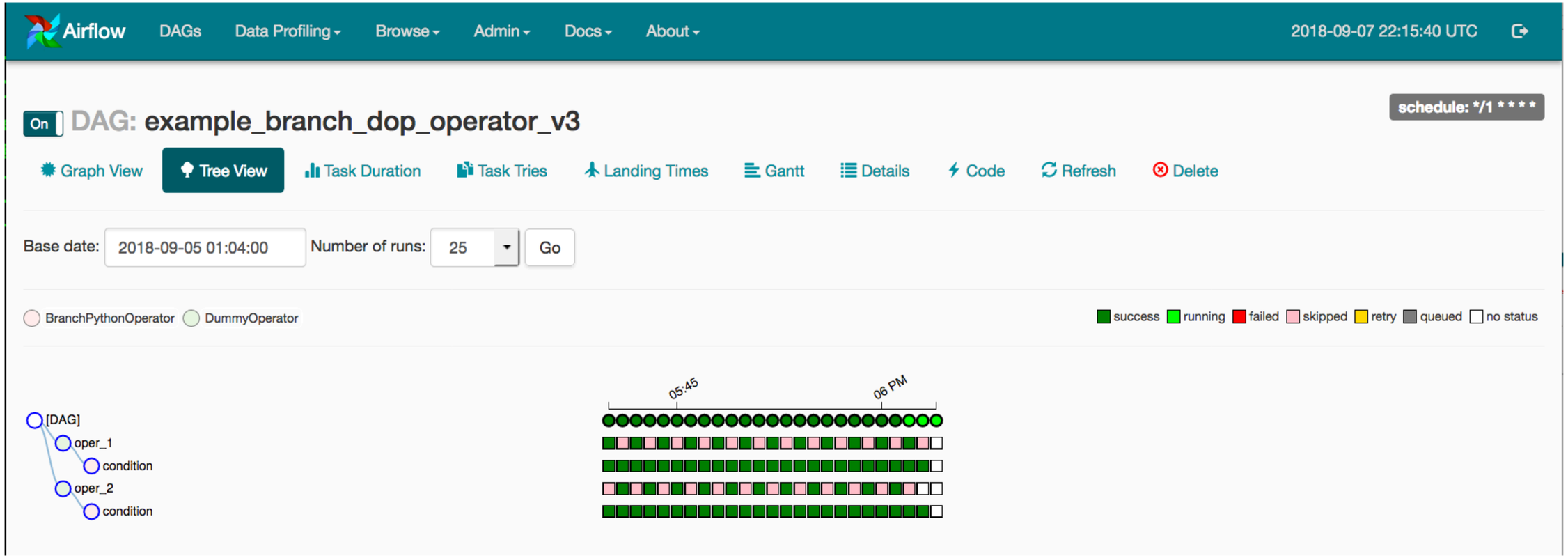
Составляющие

- Scheduler
- Webserver
- Worker

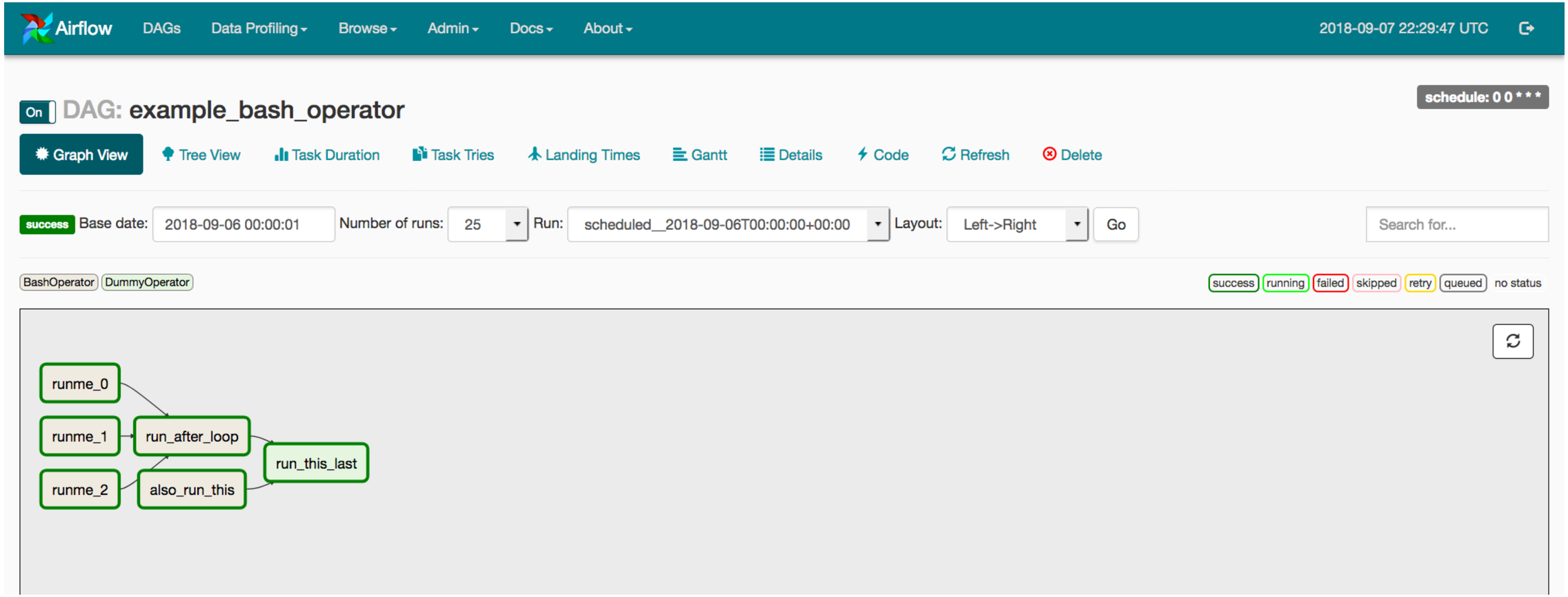
Webserver


Airflow
DAGs
Data Profiling
Browse
Admin
Docs
About
2018-09-07 22:14:10 UTC
DAGs
Search:
DAG
Schedule
Owner
Recent Tasks
Last Run
DAG Runs
Links
On
example_bash_operator
0 0 * * *
airflow
2018-09-06 00:00
On
example_branch_dop_operator_v3
*/1 * * * *
airflow
2018-09-05 00:56
On
example_branch_operator
@daily
airflow
2018-09-06 00:00
On
example_xcom
@once
airflow
2018-09-05 00:00
On
latest_only
4:00:00
Airflow
2018-09-07 16:00
Showing 1 to 5 of 5 entries
«
<
1
>
»
Show Paused DAGs

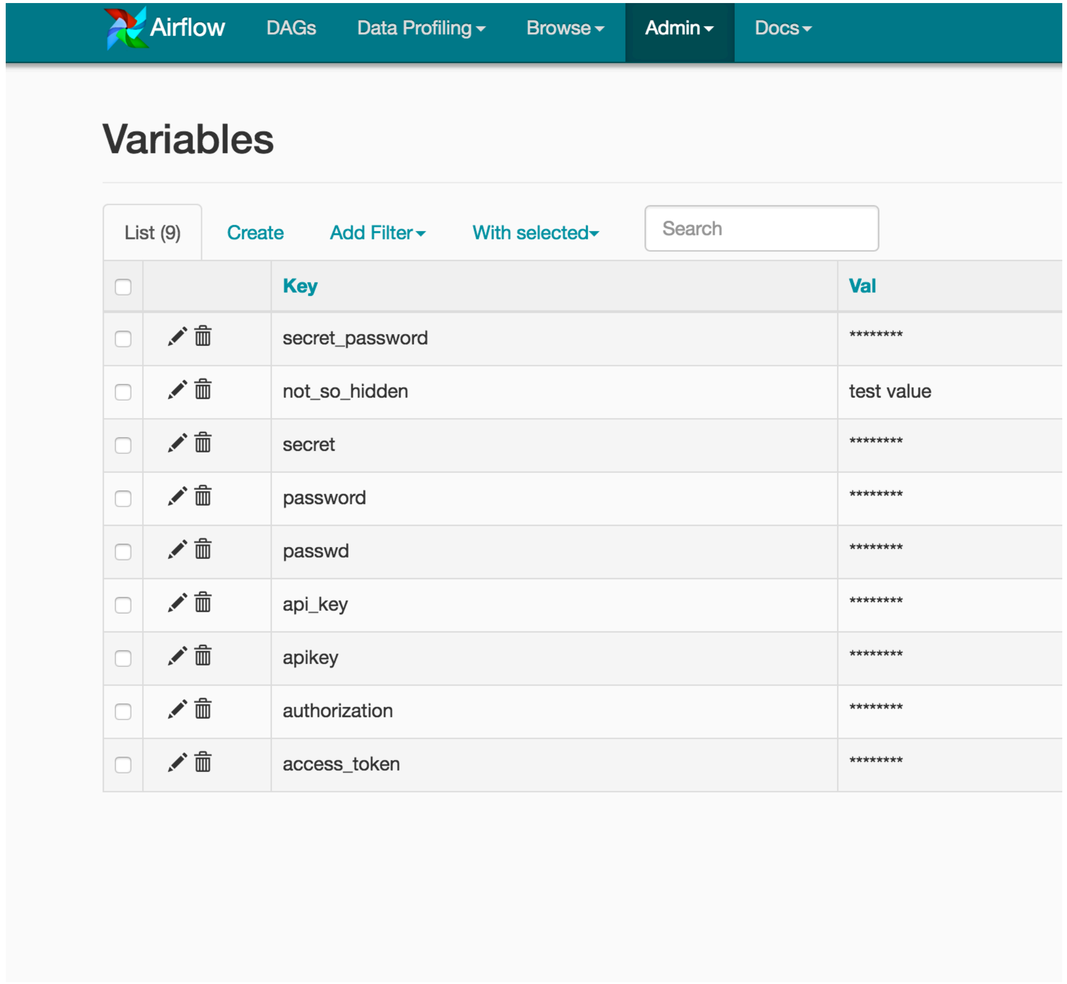
Webserver


Airflow
DAGs
Data Profiling
Browse
Admin
Docs
About
2018-09-07 22:15:40 UTC
On
DAG: example_branch_dop_operator_v3
schedule: */1 * * * *
Graph View
Tree View
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Refresh
Delete
Base date:
2018-09-05 01:04:00
Number of runs:
25
Go
BranchPythonOperator
DummyOperator
success
running
failed
skipped
retry
queued
no status
[DAG]
oper_1
condition
oper_2
condition
05:45
06 PM

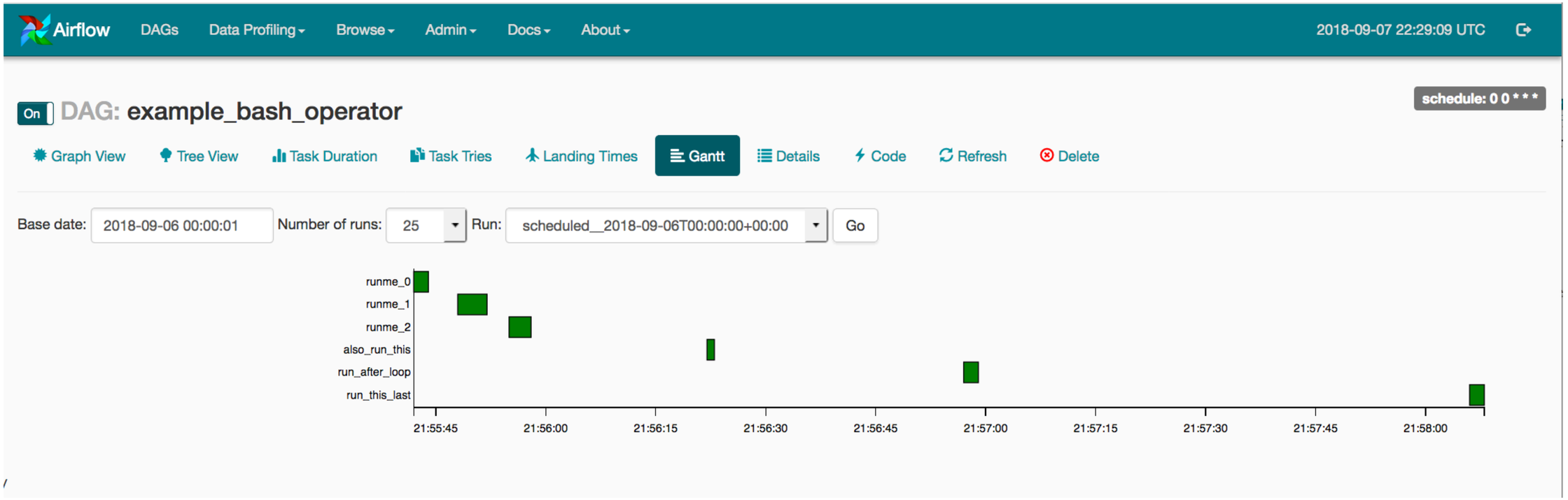
Webserver


Airflow
DAGs
Data Profiling
Browse
Admin
Docs
About
2018-09-07 22:29:47 UTC
On
DAG: example_bash_operator
schedule: 0 0 * * *
Graph View
Tree View
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Refresh
Delete
success
Base date:
2018-09-06 00:00:01
Number of runs:
25
Run:
scheduled__2018-09-06T00:00:00+00:00
Layout:
Left->Right
Go
Search for...
BashOperator
DummyOperator
success
running
failed
skipped
retry
queued
no status
runme_0
runme_1
runme_2
run_after_loop
also_run_this
run_this_last

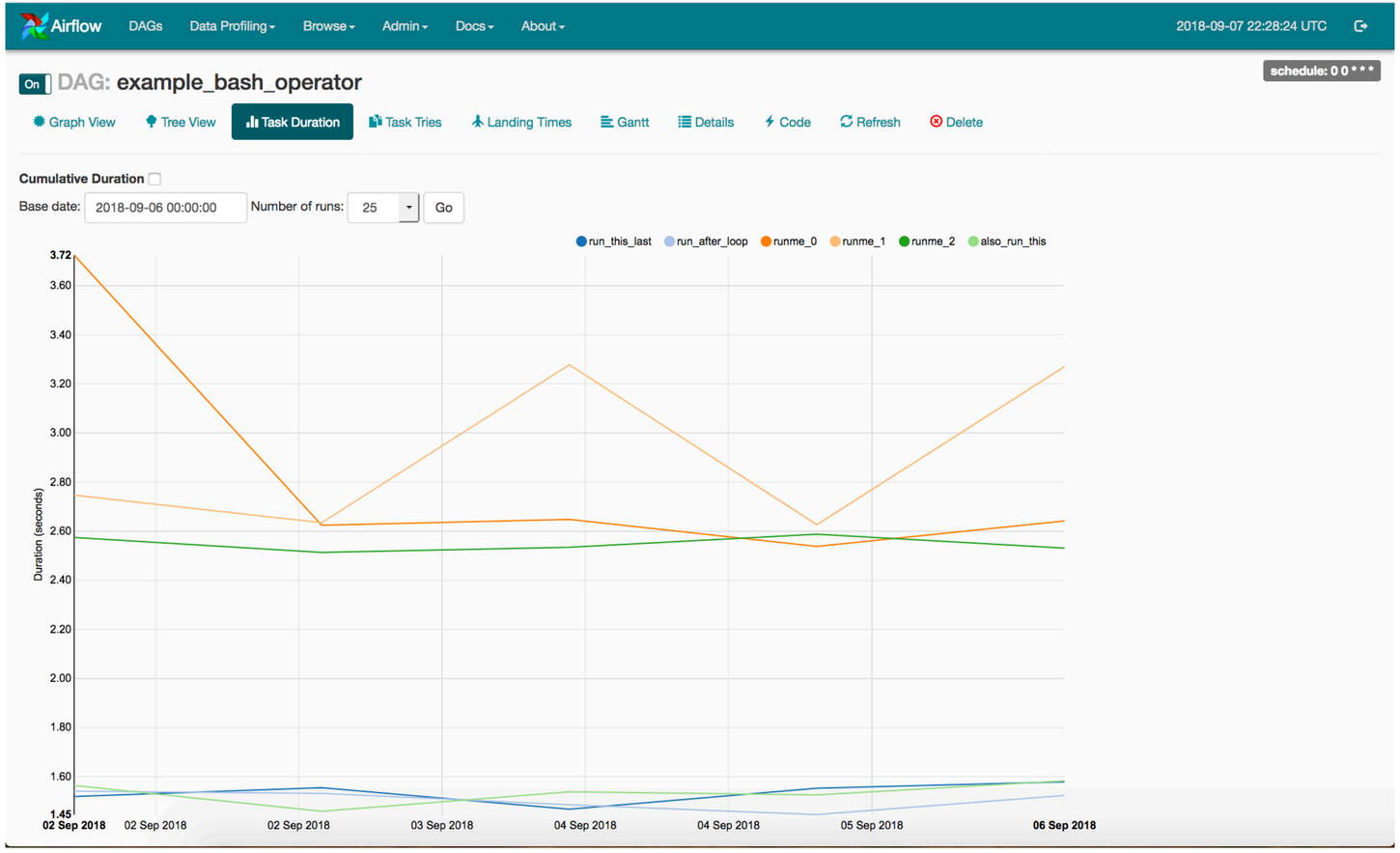
Webserver


Airflow
DAGs
Data Profiling
Browse
Admin
Docs
Variables
List (9)
Create
Add Filter
With selected
Search
Key
Val
secret_password
****
not_so_hidden
test value
secret
****
password
****
passwd
****
api_key
****
apikey
****
authorization
****
access_token
****

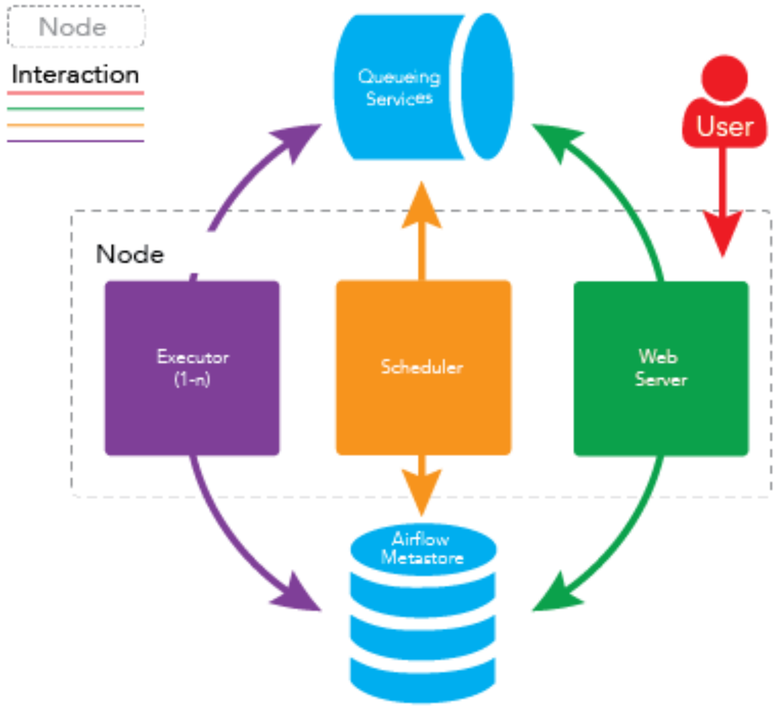
Webserver


Airflow
DAGs
Data Profiling
Browse
Admin
Docs
About
2018-09-07 22:29:09 UTC
On
DAG: example_bash_operator
schedule: 0 0 * * *
Graph View
Tree View
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Refresh
Delete
Base date:
2018-09-06 00:00:01
Number of runs:
25
Run:
scheduled__2018-09-06T00:00:00+00:00
Go
runme_0
runme_1
runme_2
also_run_this
run_after_loop
run_this_last
21:55:45
21:56:00
21:56:15
21:56:30
21:56:45
21:57:00
21:57:15
21:57:30
21:57:45
21:58:00

Webserver


Airflow
DAGs
Data Profiling
Browse
Admin
Docs
About
2018-09-07 22:28:24 UTC
On
DAG: example_bash_operator
schedule: 0 0 * * *
Graph View
Tree View
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Refresh
Delete
Cumulative Duration
Base date: 2018-09-06 00:00:00
Number of runs: 25
Go
run_this_last
run_after_loop
runme_0
runme_1
runme_2
also_run_this
Duration (seconds)
3.72
3.60
3.40
3.20
3.00
2.80
2.60
2.40
2.20
2.00
1.80
1.60
1.45
02 Sep 2018
02 Sep 2018
02 Sep 2018
03 Sep 2018
04 Sep 2018
04 Sep 2018
05 Sep 2018
06 Sep 2018

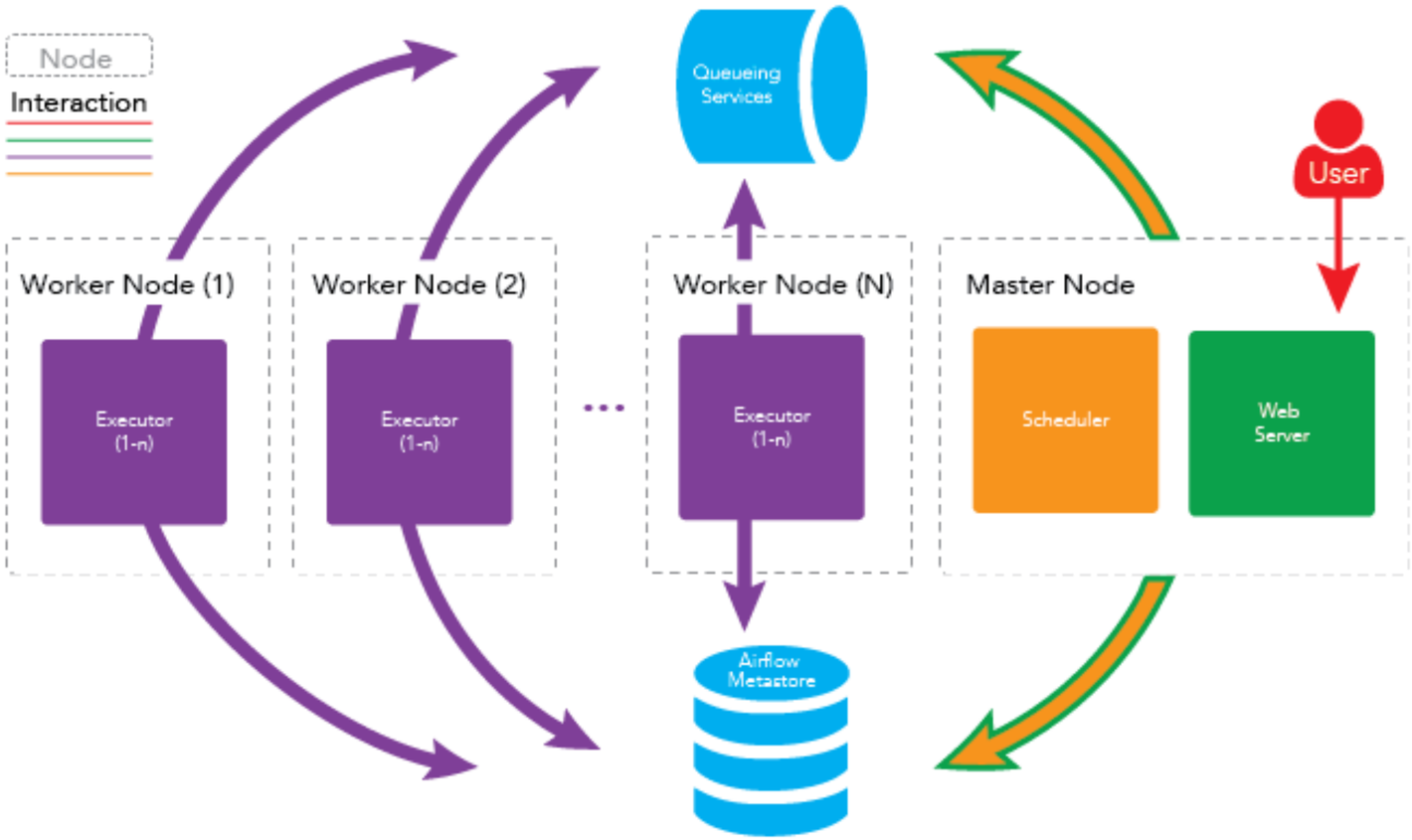
Архитектура



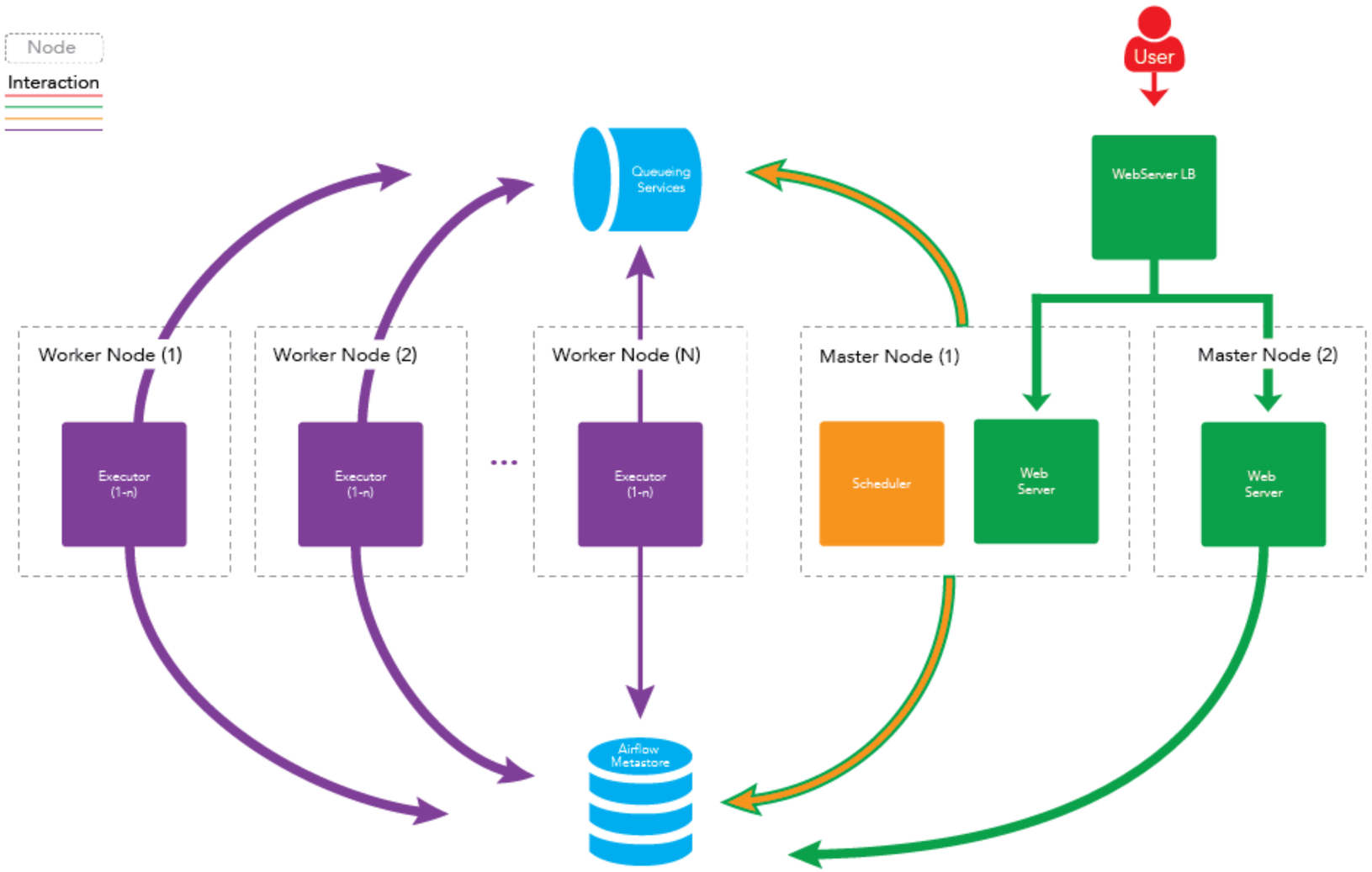
Scheduler - executors

- sequential_executor
- local_executor
- celery_executor
- dask_executor
- kubernetes_executor

Архитектура

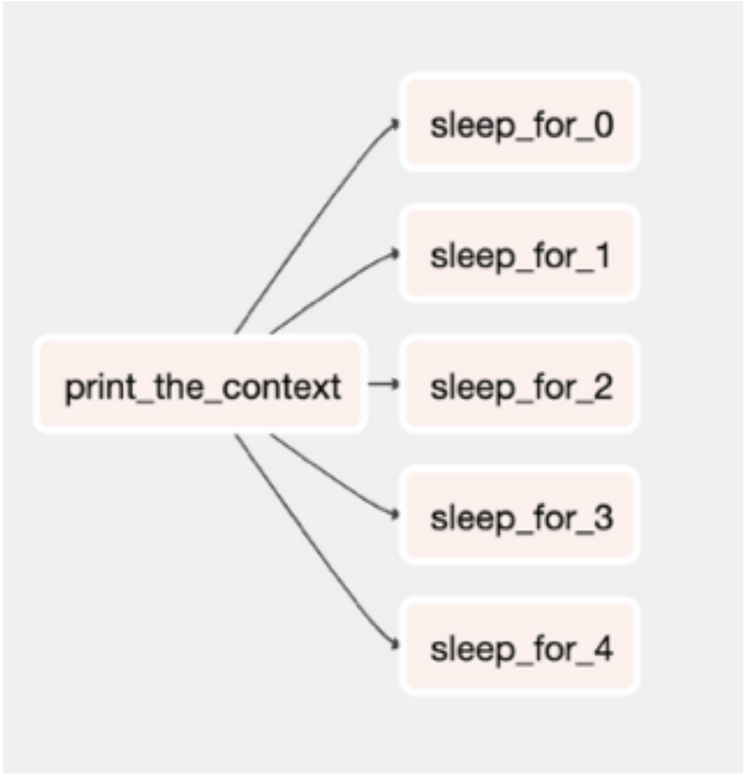


Архитектура

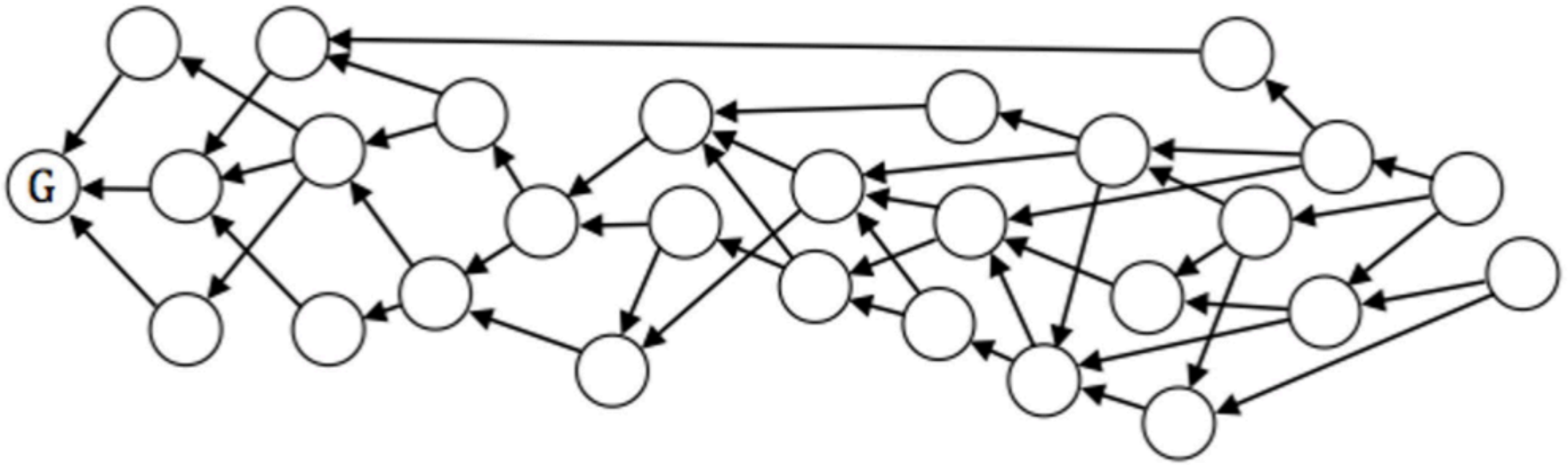


Easy

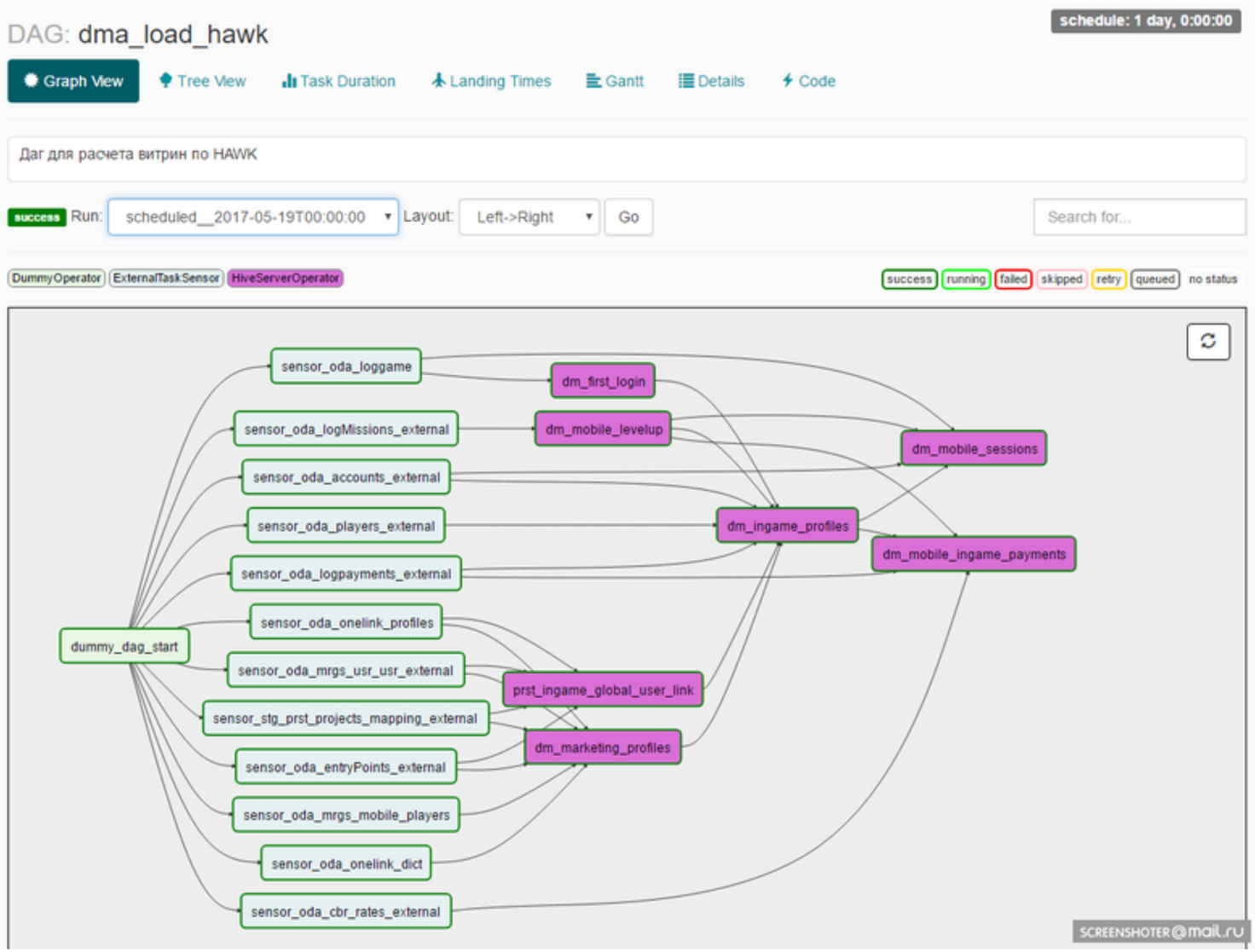
```python
import time
from pprint import pprint

import airflow
from airflow.models import DAG
from airflow.operators.python_operator import PythonOperator

args = {
    'owner': 'Airflow',
    'start_date': airflow.utils.dates.days_ago(2),
}

dag = DAG(
    dag_id='example_python_operator',
    default_args=args,
    schedule_interval="0 15 1 * *",
)


def print_context(ds, **kwargs):
    pprint(kwargs)
    print(ds)
    return 'Whatever you return gets printed in the logs'

run_this = PythonOperator(
    task_id='print_the_context',
    python_callable=print_context,
    dag=dag,
)
```



DAG



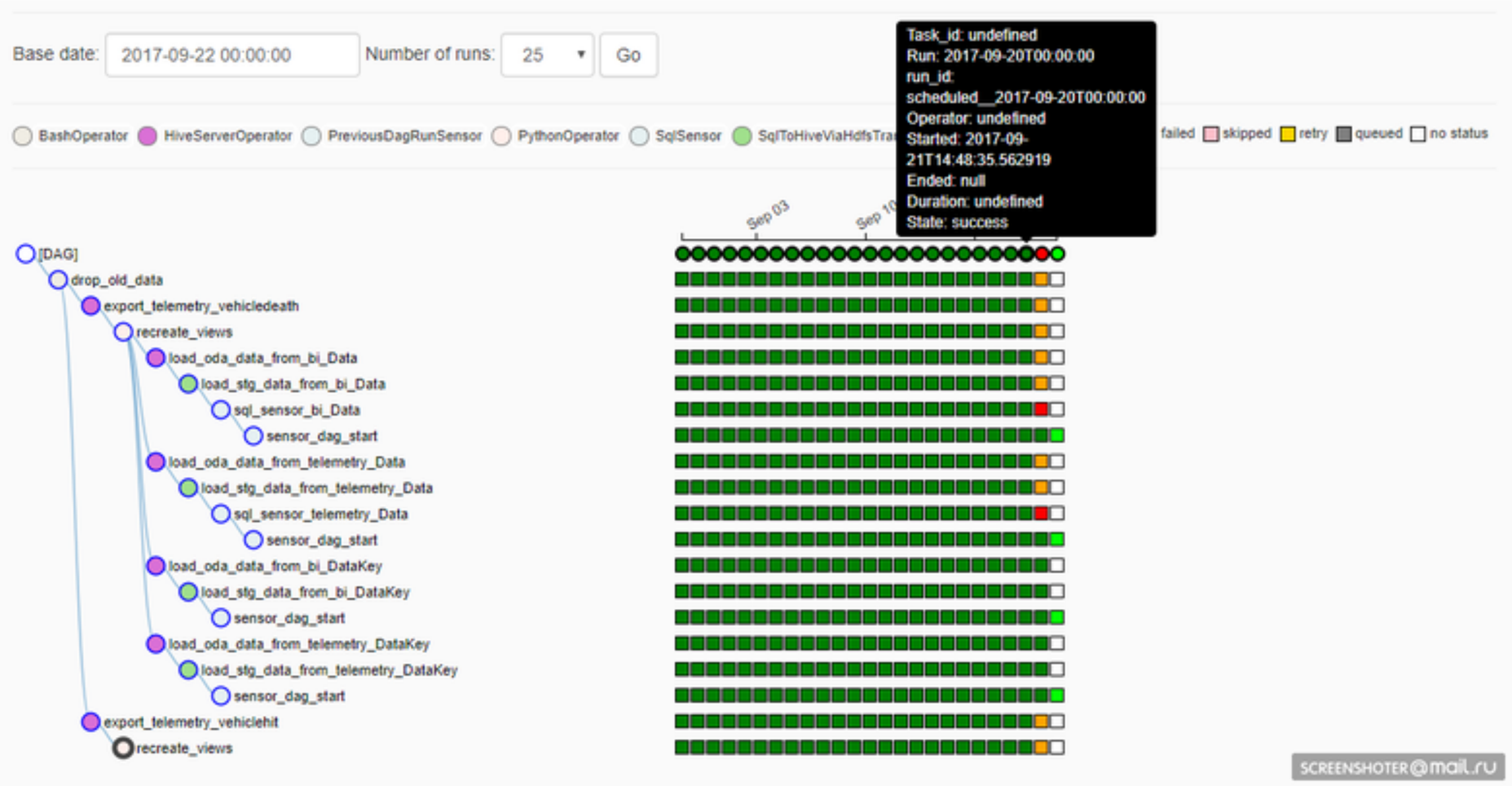
DAG


DAG: dma_load_hawk
schedule: 1 day, 0:00:00
Graph View
Tree View
Task Duration
Landing Times
Gantt
Details
Code
Даг для расчета витрин по HAWK
success
Run:
scheduled__2017-05-19T00:00:00
Layout:
Left->Right
Go
Search for...
DummyOperator
ExternalTaskSensor
HiveServerOperator
success
running
failed
skipped
retry
queued
no status
sensor_oda_loggame
dm_first_login
sensor_oda_logMissions_external
dm_mobile_levelup
dm_mobile_sessions
sensor_oda_accounts_external
sensor_oda_players_external
dm_ingame_profiles
dm_mobile_ingame_payments
sensor_oda_logpayments_external
sensor_oda_onelink_profiles
dummy_dag_start
sensor_oda_mrgs_usr_usr_external
prst_ingame_global_user_link
sensor_stg_prst_projects_mapping_external
dm_marketing_profiles
sensor_oda_entryPoints_external
sensor_oda_mrgs_mobile_players
sensor_oda_onelink_dict
sensor_oda_cbr_rates_external
SCREENSHOTER@mail.ru

Операторы

- BashOperator — оператор для выполнения bash-команды
- PythonOperator — оператор для вызова Python-кода
- EmailOperator — оператор для отправки email’а
- HTTPOperator — оператор для работы с http-запросами
- SqlOperator — оператор для выполнения SQL-кода
- Sensor — оператор ожидания события

Execution Date



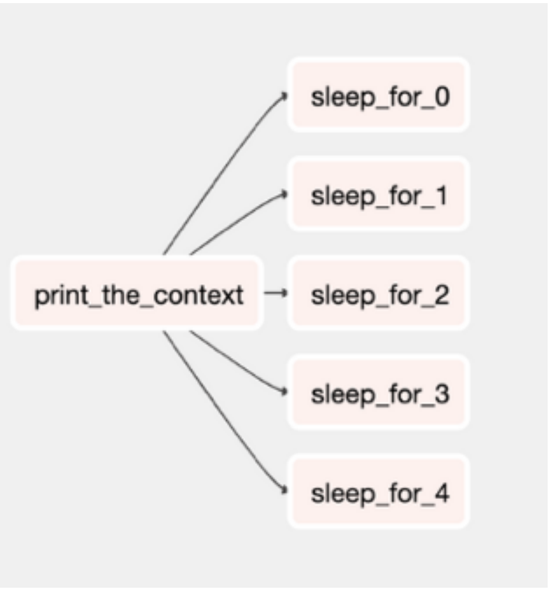
Очереди

XCom

Task Details | Rendered Template | Log | XCom

XCom

Key	Value
table_name	squads_tourney02122017_2100

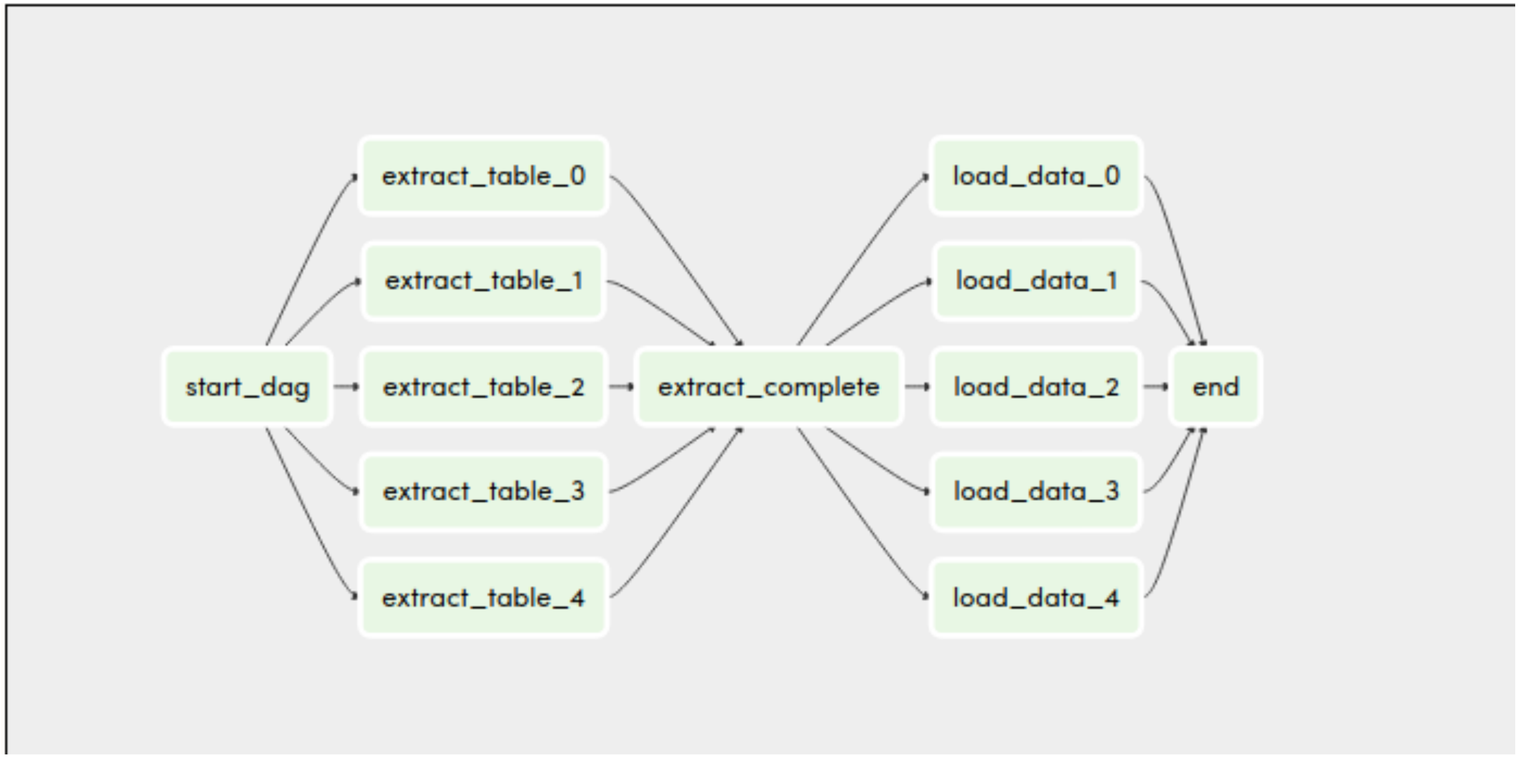


Best Practices

Идемпотентность

- $\forall x : f(f(x)) = f(x)$
- $a = a + 0 = (a + 0) + 0 = ((a + 0) + 0) + 0 =$

Гранулярность



Sensors

- ExternalTaskSensor
- HdfsSensor
- HttpSensor
- HivePartitionSensor
- SqlSensor
- TimeDeltaSensor
- TimeSensor
- WebHdfsSensor

Дополнительно

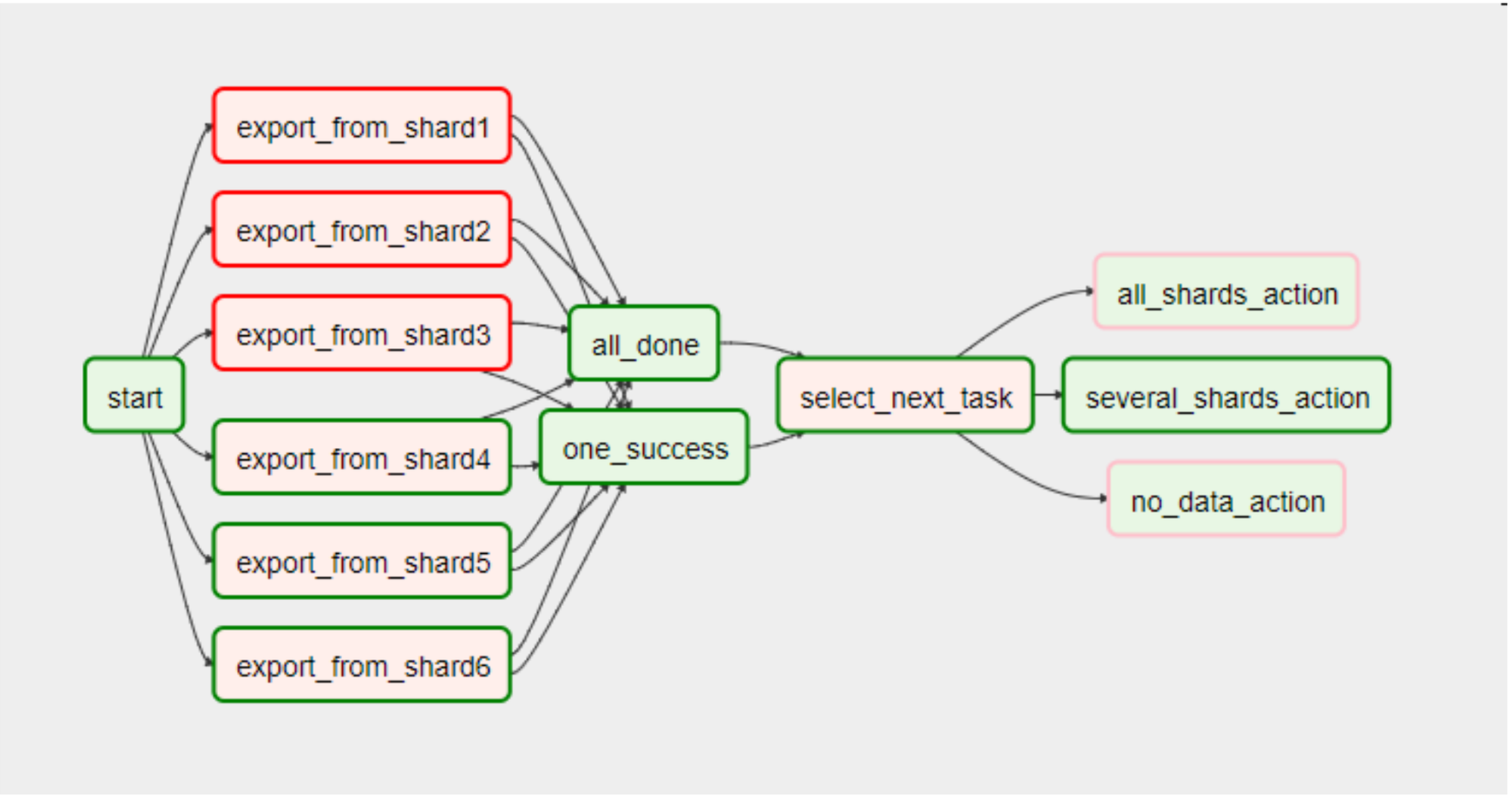
- UTC
- https://github.com/epoch8/airflow-exporter
- Нет долгим scheduler-ам

Генерация DAG-ов

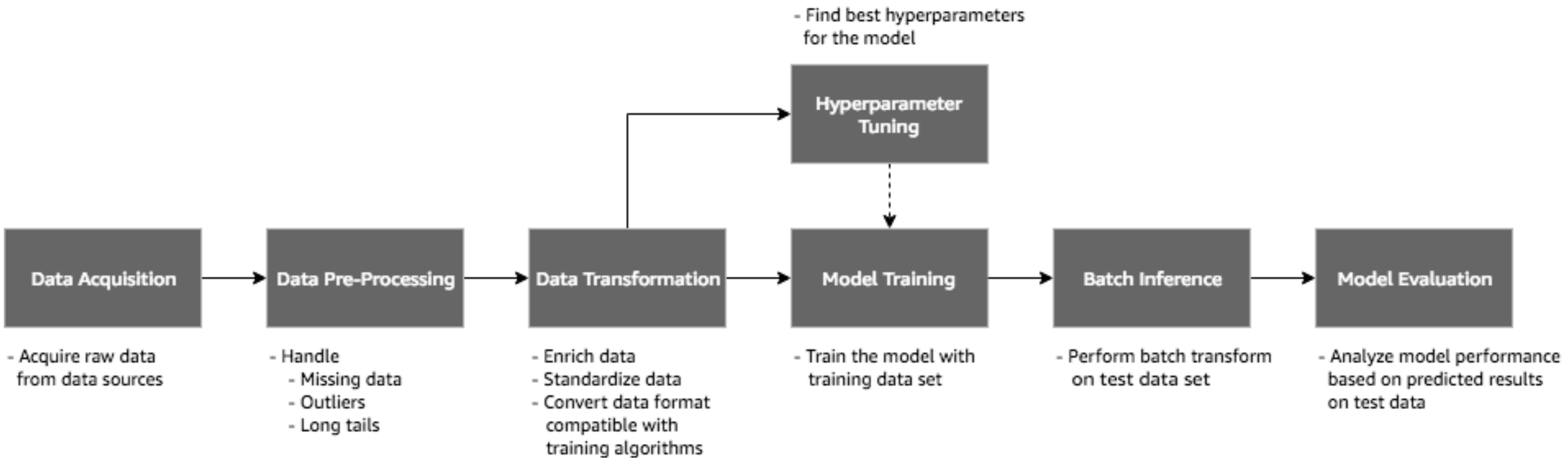
```python
def generate_dags():
    dags = []

    for hql_filename in hql_files:

        default_args = {
            'owner': 'rnd_metrics',
            'depends_on_past': False,
            'start_date': datetime.now() - timedelta(days=2),
            'retries': 1,
            'retry_delay': timedelta(minutes=5)
        }

        dag = DAG(
            dag_id=metric_name+'_dag',
            default_args=default_args,
            schedule_interval=schedule_interval,
            max_active_runs=5,
            catchup=True,
            concurrency=5
        )

        tasks = [
            PythonOperator(
                task_id=metric_name+'_operator',
                python_callable=upload_metrics,
                op_kwargs={
                    'hql_query': hql_query,
                    'metric_name': metric_name
                },
                provide_context=True,
                dag=dag
            )
        ]

        dags.append({'dag': dag, 'tasks': tasks})

    return dags
```

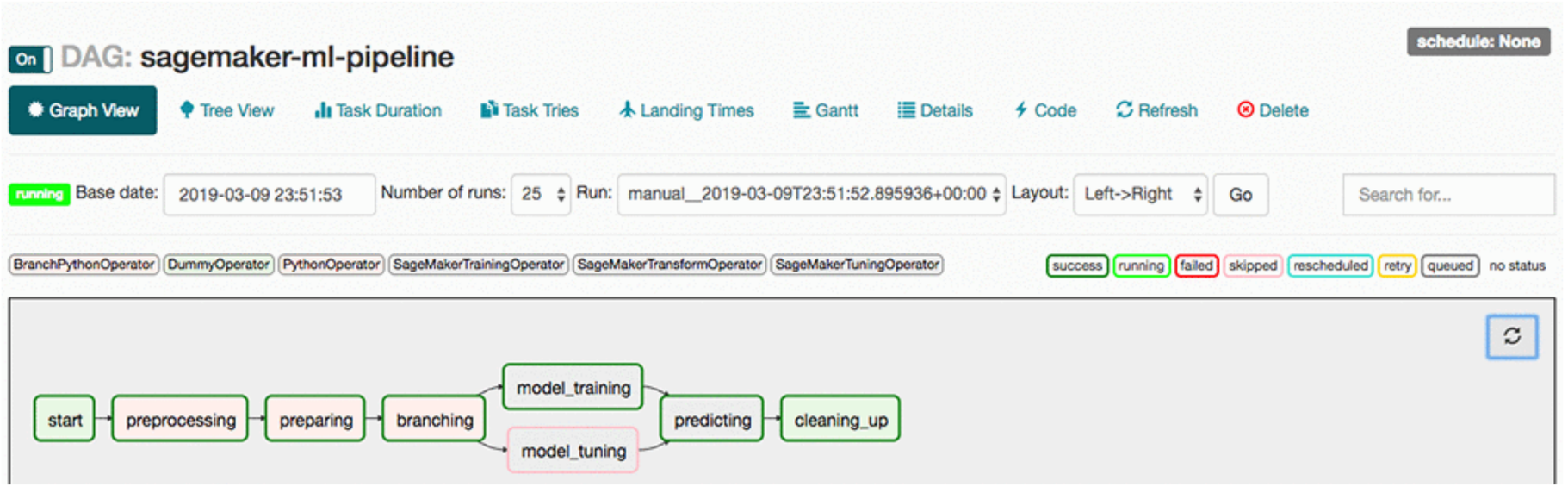
```python
for dag in generate_dags():
    globals()[dag['dag'].dag_id] = dag['dag']

    DummyOperator(
        dag=dag['dag'],
        task_id=f'dummy_{dag["dag"].dag_id}'
    ).set_upstream(dag['tasks'])
```

Ветвление


export_from_shard1
export_from_shard2
export_from_shard3
start
export_from_shard4
export_from_shard5
export_from_shard6
all_done
one_success
select_next_task
all_shards_action
several_shards_action
no_data_action

ML


- Find best hyperparameters for the model
Hyperparameter Tuning
Data Acquisition
Data Pre-Processing
Data Transformation
Model Training
Batch Inference
Model Evaluation
- Acquire raw data from data sources
- Handle
- Missing data
- Outliers
- Long tails
- Enrich data
- Standardize data
- Convert data format compatible with training algorithms
- Train the model with training data set
- Perform batch transform on test data set
- Analyze model performance based on predicted results on test data


On
DAG: sagemaker-ml-pipeline
schedule: None
Graph View
Tree View
Task Duration
Task Tries
Landing Times
Gantt
Details
Code
Refresh
Delete
running
Base date:
2019-03-09 23:51:53
Number of runs:
25
Run:
manual__2019-03-09T23:51:52.895936+00:00
Layout:
Left->Right
Go
Search for...
BranchPythonOperator
DummyOperator
PythonOperator
SageMakerTrainingOperator
SageMakerTransformOperator
SageMakerTuningOperator
success
running
failed
skipped
rescheduled
retry
queued
no status
start
preprocessing
preparing
branching
model_training
model_tuning
predicting
cleaning_up

MLcomp

Catalyst

Projects | Dags | Tasks | Models | Reports | Logs | Computers | Auxiliary

Filters

Project	Id	Name	Tasks	Created	Started	Last activity	Duration	Task status	Links	Image size	File size
severstal	51	kaggle_kernel	1	08.19 14:59:20	08.19 14:59:34	08.19 15:01:28	1 min 54 sec	1		0 byte	0 byte
severstal	50	kaggle_kernel	1	08.19 14:59:12	08.19 14:59:14	08.19 15:00:41	1 min 27 sec	1		0 byte	800 byte
severstal	49	kaggle_kernel	1	08.19 14:57:56	08.19 14:58:42	08.19 14:59:06	24 sec	1		0 byte	0 byte
severstal	48	kaggle_kernel	1	08.19 14:56:12	08.19 14:57:44	08.19 15:57:37	59 min 52 sec	1		0 byte	0 byte
severstal	47	kaggle_kernel	1	08.19 14:56:11	08.19 14:56:47	08.19 15:57:45	1 hour 0 min	1		0 byte	0 byte

Items per page: 5 1 - 5 of 35

MLcomp

Почитать и посмотреть

- https://www.astronomer.io/podcast/
- http://site.clairvoyantsoft.com/setting-apache-airflow-cluster/
- https://www.astronomer.io/guides/airflow-executors-explained/
- https://medium.com/@maximebeauchemin/functional-data-engineering-a-modern-paradigm-for-batch-data-processing-2327ec32c42a
- https://habr.com/ru/company/mailru/blog/339392/
- https://habr.com/ru/company/mailru/blog/344398/
- https://blog.twitter.com/engineering/en_us/topics/insights/2018/ml-workflows.html