

# Урок 7. Airflow

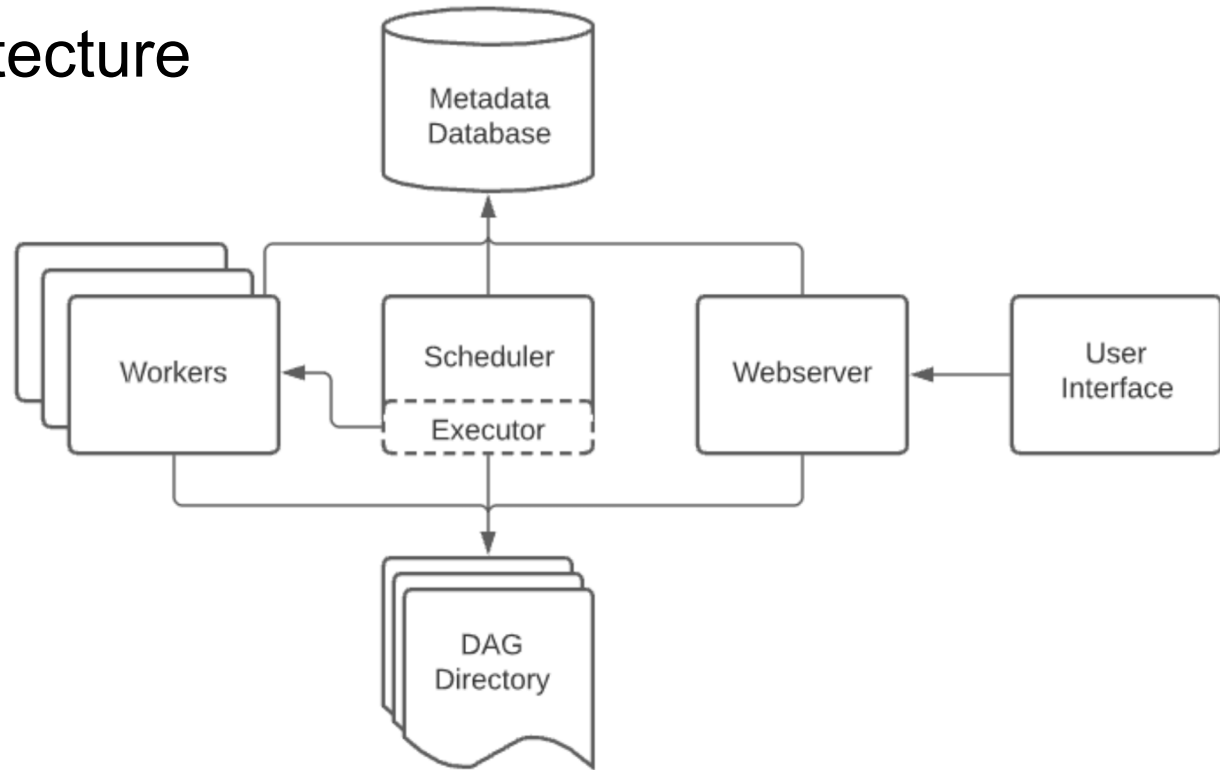
# Cron

```
# ┌─ минуты (0-59)
# │ ┌── часы (0-23)
# │ │ ┌── день месяца (1-31)
# │ │ │ ┌── месяц (1-12)
# │ │ │ │ ┌── день недели (0-6)
# │ │ │ │ │
# │ │ │ │ │
# * * * * * выполняемая команда
```

<https://crontab.guru/>

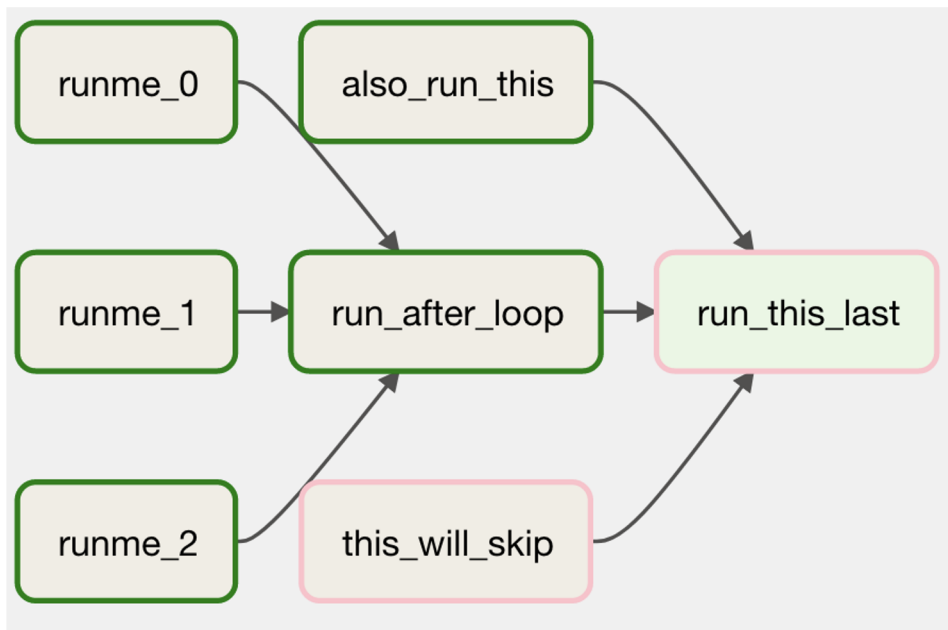
<https://tproger.ru/translations/guide-to-cron-jobs/>

# Airflow architecture



<https://airflow.apache.org/docs/apache-airflow/stable/concepts/overview.html>

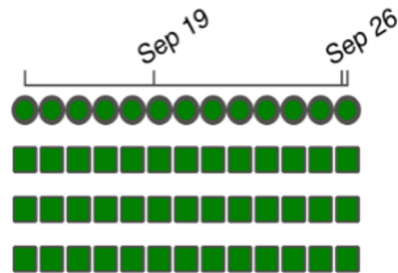
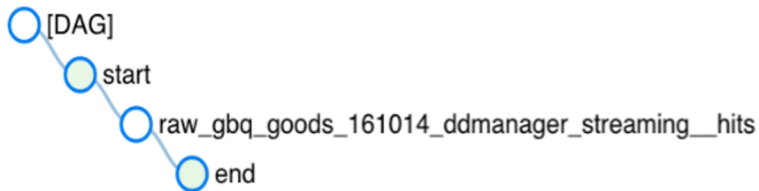
# Directed Acyclic Graph



<https://www.astronomer.io/guides/dag-best-practices>

# Идемпотентность

**Идемпотентность** ([лат.](#) *idem* — тот же самый + *potens* — способный) — свойство объекта или операции при повторном применении операции к объекту [давать тот же результат](#), что и при первом.



# Tasks

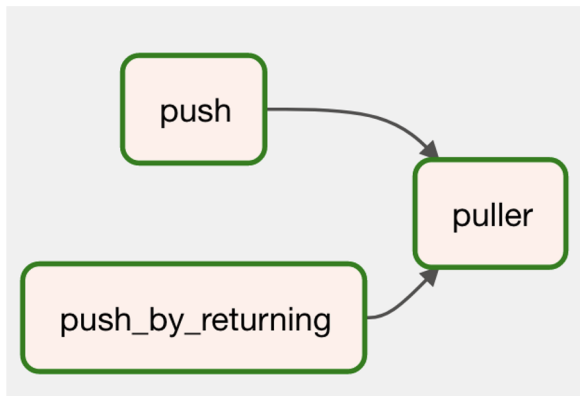
```
run_this_last = DummyOperator(
    task_id='run_this_last',
)

# [START howto_operator_bash]
run_this = BashOperator(
    task_id='run_after_loop',
    bash_command='echo 1',
)
# [END howto_operator_bash]

run_this >> run_this_last
```

- [Operators](#), predefined task templates that you can string together quickly to build most parts of your DAGs.
- [Sensors](#), a special subclass of Operators which are entirely about waiting for an external event to happen.
- A [TaskFlow](#)-decorated `@task`, which is a custom Python function packaged up as a Task.

# XComm



Task Instance: push at 2021-09-25, 03:00:00

[Task Instance Details](#)

[< > Rendered Template](#)

[≡ Log](#)

[↔ XCom](#)

## XCom

Key	Value
value from pusher 1	[1, 2, 3]

# Operator

```
with DAG("my-dag") as dag:
    ping = SimpleHttpOperator(endpoint="http://example.com/update/")
    email = EmailOperator(to="admin@example.com", subject="Update complete")

    ping >> email
```

- **BashOperator** - executes a bash command
- **PythonOperator** - calls an arbitrary Python function
- **EmailOperator** - sends an email



# Jinja templating

```
# The execution date as YYYY-MM-DD
date = "{{ ds }}"
t = BashOperator(
    task_id='test_env',
    bash_command='/tmp/test.sh ',
    dag=dag,
    env={'EXECUTION_DATE': date})
```

```
load_date = '{{yesterday_ds}}'

script = f"jobs/raw/etl_daily.py " \
        f"--load_date=\"{load_date}\" "
```

<https://airflow.apache.org/docs/apache-airflow/stable/concepts/operators.html>

# Task tree

☒ DAG: example\_bash\_operator

Tree View

Graph View

Calendar View

Task Duration

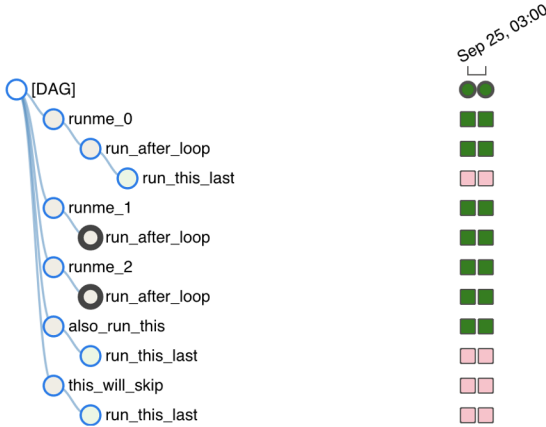
2021-09-26T00:00:00Z

Runs

25

Update

☐ BashOperator ☐ DummyOperator



# Graph view

 DAG: example\_bash\_operator

 Tree View

 Graph View

 Calendar View

 Task Duration

 Task Tries

 Landing



2021-09-26T00:00:01Z

Runs

25

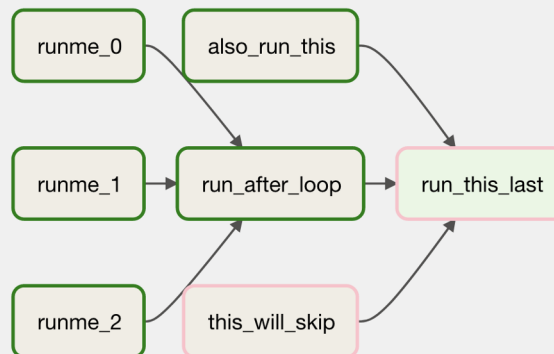
Run

scheduled\_\_2021-09-26T00:00:00+00:00

BashOperator

DummyOperator

queued



## Домашнее задание

Запустить сервисы airflow локально (airflow/docker-compose.yaml в репозитории). Поставить на ежедневную загрузку данные по адресу <http://37.139.43.86/data-nov/2021-12-01> , начиная с 2021-11-26 и посчитать ежедневные агрегаты из 1-й лекции в parquet файлы

\*писать агрегаты по дням в базу postgresql из внешнего docker-compose, которая доступна изнутри контейнеров по адресу 172.17.0.1:5432