**Машинное обучение и высоконагруженные системы**

Москва, осень 2023

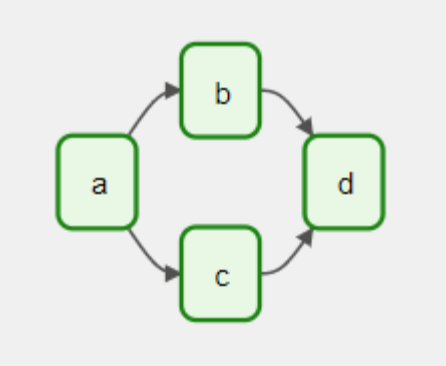
# MLOps. Начало

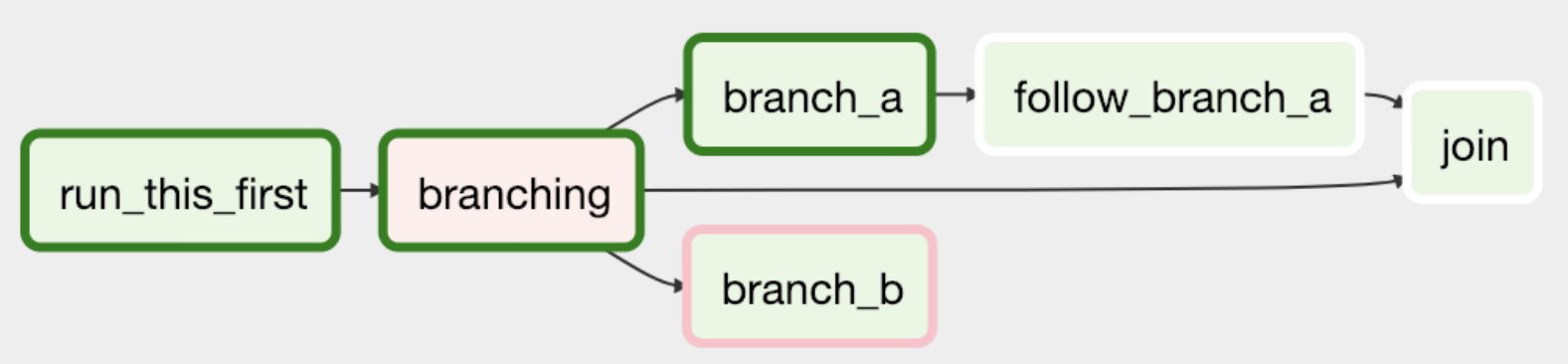

Гаврилова Елизавета,
Senior ML-engineer

# Неделя 3. Что будем обсуждать?

- Вспомним концепцию **DAG**
- Разберём основные типы **Операторов**
- Разберём, что такое **Хуки**
- Разберём, из чего состоит базовый **пайплайн обучения** модели
- Напишем свой первый **DAG**

# Концепция DAG в AirFlow




DAG — это ориентированный ациклический граф, т.е. граф с началом и концом, без циклов, но допускает параллельные пути.

Это сущность, которая объединяет задач в единый пайплайн (цепочку задач).

# Основа DAG

Параметры, описывающие работу DAG'а

**default_args** – настройки для операторов
**dag_id** – уникальный идентификатор DAG'а
**schedule_interval** – расписание запусков
**start_date** – начало запусков
**catchup** – нужно ли «догонять упущенное»
**tags** – тэги проекта

**email_on_failure** – оповещение в случае неудачи
**email_on_retry** – оповещение в случае перезапуска
**retry** – количество перезапусков в случае неудачи
**retry_delay** – временной интервал между перезапусками

Ссылка на документацию

```python
from airflow.models import DAG
from airflow.operators.python_operator import PythonOperator
from airflow.utils.dates import days_ago
from datetime import timedelta

DEFAULT_ARGS = {
    "owner" : "Elizaveta Gavrilova",
    "email" : "example@gmail.com",
    "email_on_failure" : True,
    "email_on_retry" : False,
    "retry" : 3,
    "retry_delay" : timedelta(minutes=1)
}

dag = DAG(
    dag_id = "mlops_dag_1",
    schedule_interval = "0 1 * * *",
    start_date = days_ago(2),
    catchup = False,
    tags = ["mlops"],
    default_args = DEFAULT_ARGS
)
```

# Основные типы операторов

Если представить DAG как **пайплайн** обучения модели, то Airflow оператор – это одно **звено** в пайплайне. Это классы, которые содержат логику выполнения единичной работы.

**Python operator** – выполнение python скриптов.
**Bash operator** – выполнение bash скриптов.
**Email operator** – оповещения на почту.

Ссылка на документацию

## Submodules

- airflow.operators.bash
- airflow.operators.branch
- airflow.operators.datetime
- airflow.operators.email
- airflow.operators.empty
- airflow.operators.generic_transfer
- airflow.operators.latest_only
- airflow.operators.python
- airflow.operators.smooth
- airflow.operators.subdag
- airflow.operators.trigger_dagrun
- airflow.operators.weekday

# PythonOperator

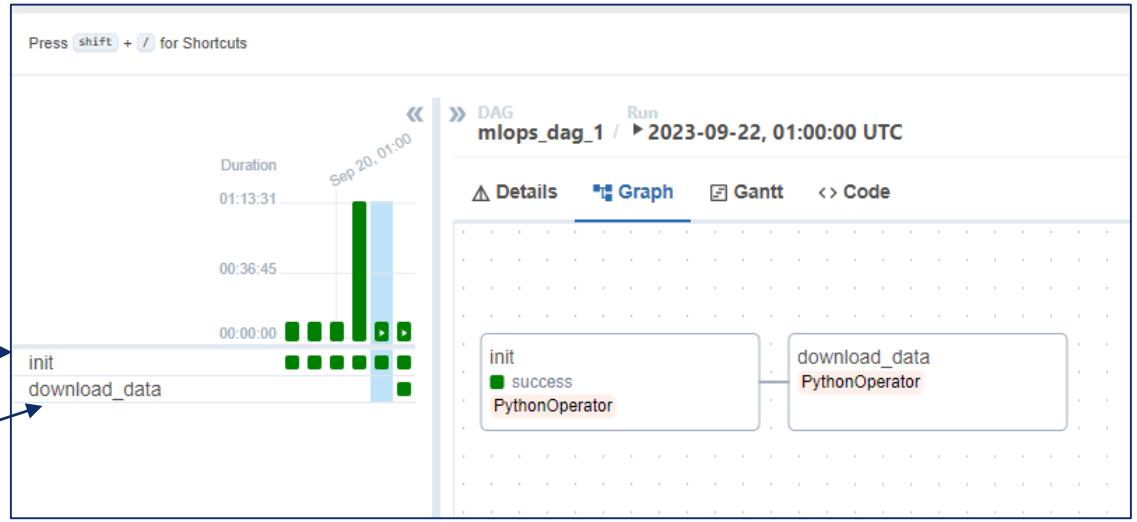
```python
def init() -> NoReturn:
    print("Hello, World")

def get_data_from_s3() -> NoReturn:
    s3_hook = S3Hook("s3_connection")
    file_path = s3_hook.download_file(
        key='dataset/california_housing',
        bucket_name='lizvladi-mlops'
    )
    print(file_path)

task_init = PythonOperator(task_id = "init",
                           python_callable = init,
                           dag = dag)

task_get_data_from_s3 = PythonOperator(task_id = "download_data",
                                       python_callable = get_data_from_s3,
                                       dag = dag)

task_init >> task_get_data_from_s3
```




[Ссылка на документацию]

# Популярные типы операторов, доступные через провайдера

Некоторые операторы не включены в базовый модуль AirFlow, но поставляются через провайдеров.

Их необходимо просто доустановить при необходимости.

Такие операторы живут внутри класса ***class*** **airflow.providers**

**Postgres operator** – выполнение запросов к postgres БД.
**SlackAPI operator** – оповещения в slack.
**Docker operator** – выполнение скриптов в докер-контейнерах.

Ссылка на документацию

- SimpleHttpOperator
- MySqlOperator
- PostgresOperator
- MsSqlOperator
- OracleOperator
- JdbcOperator
- DockerOperator
- HiveOperator
- S3FileTransformOperator
- PrestoToMySqlOperator
- SlackAPIOperator

# Хуки (Hooks)

Хуки  - это инструменты взаимодействия с различными внешними системами.

Хуки используют подключения из раздела **Admin->Connections** и решают проблему использования кред/секретов в коде.

Некоторые хуки встроены в операторы, но их можно использовать и обособленно.

Большинство популярных хуков также как и операторы требуется установить дополнительно.

Ссылка на документацию

## Submodules

- airflow.hooks.S3_hook
- airflow.hooks.base_hook
- airflow.hooks.dbapi_hook
- airflow.hooks.docker_hook
- airflow.hooks.druid_hook
- airflow.hooks.hdfs_hook
- airflow.hooks.hive_hooks
- airflow.hooks.http_hook
- airflow.hooks.jdbc_hook
- airflow.hooks.mssql_hook
- airflow.hooks.mysql_hook
- airflow.hooks.oracle_hook
- airflow.hooks.pig_hook
- airflow.hooks.postgres_hook
- airflow.hooks.presto_hook
- airflow.hooks.samba_hook
- airflow.hooks.slack_hook
- airflow.hooks.sqlite_hook
- airflow.hooks.webhdfs_hook
- airflow.hooks.zendesk_hook

# S3Hook

Для использования этого хука необходимо поставить пакет провайдера amazon
**pip install apache-airflow-providers-amazon**

```python
from airflow.providers.amazon.aws.hooks.s3 import S3Hook

s3_hook = S3Hook("s3_connection")

s3_hook.load_file(
        filename='california_housing.csv',
        key='dataset/california_housing',
        bucket_name='lizvladi-mlops'
    )

s3_hook.download_file(
        key='dataset/california_housing',
        bucket_name='lizvladi-mlops'
    )
```

Созданный нами s3_connection через **Admin->Connections**

**filename**: путь к файлу, который нужно отправить на s3
**key**: путь к месту хранения на s3
**bucket_name**: имя бакета на s3

**key**: путь к месту хранения на s3
**bucket_name**: имя бакета на s3

Ссылка на документацию

# S3: session

Сессия – это объект подключения к AWS. Сессия содержит в себе:

- креды(секреты): **aws_access_key_id** и **aws_secret_access_key**
- регион подключения **region_name**
- другие параметры, описанные в профиле **profile_name**

Документация

# boto3: resource

Ресурсы – это верхнеуровневый интерфейс библиотеки boto3. Они применяют объектно-ориентированные подход к взаимодействию с различными AWS-сервисами, в том числе S3. Ресурс – это метод сессии.

**Identifiers** – это уникальные значения, обозначающие уникальный ресурс. Например, бакет.

```
bucket = s3.bucket(name="bucket_example")
```

**Attributes** – это характеристики ресурсов. Например, content_type, last_modified, metadata, version_id.

**Actions** – это методы, позволяющие совершать действия над объектами AWS-сервисов. Например,

```
object = s3.Object(name="bucket_example")
response = object.get()
```

Документация

# S3Hook пример с использованием resource

Для использования этого хука необходимо поставить пакет провайдера amazon
**pip install apache-airflow-providers-amazon**

```
s3_hook = S3Hook("s3_connection")


file = s3_hook.download_file(key = DATA_PATH,
                             bucket_name=BUCKET)
data = pd.read_pickle(file)


session = s3_hook.get_session("ru-central1")
resource = session.resource("s3")
pickle_byte_obj = pickle.dumps(data)
resource.Object(BUCKET, DATA_PATH
                ).put(Body=pickle_byte_obj)
```

Созданный нами s3_connection через **Admin->Connections**

Использование базовых методов **S3Hook**

Создание и использование **resource** **S3Hook** используется для создания сессии

Зачем использовать resource? Функционала s3Hook не всегда хватает для выполнения нужных нам задач.

# PostgresHook

Для использования этого хука необходимо поставить пакет провайдера PostgreSQL
**pip install apache-airflow-providers-postgres**

```python
import pandas as pd
from airflow.providers.postgres.hooks.postgres import PostgresHook

# Использовать созданный PG connection
pg_hook = PostgresHook("pg_connection")
con = pg_hook.get_conn()

# Прочитать все данные из таблицы table_name
data = pd.read_sql_query("SELECT * FROM table_name", con)
```

# Этапы обучения модели(пайплайн обучения)


Описанный пайплайн является минимальным для нашего курса, но не исчерпывающим для промышленной разработки.

# Давайте поможем Jupyter Notebook стать автоматически обучаемой моделью


## Разобрали сегодня

- Основы DAG
- Операторы и хуки в AirFlow
- Этапы пайплайна обучения ML модели

## ДЗ

- Запустить свой первый DAG

## В следующей серии

- Когда моделей много...