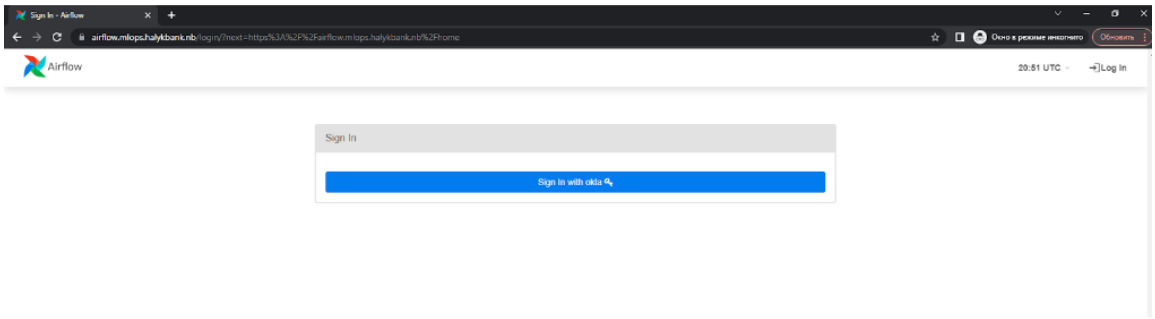
Оркестрация рабочих процессов машинного обучения

Airflow является оркестратором процессов запуска моделей. Именно Airflow ответственен за запуск скоринга моделей и их переобучения по расписанию. Основным объектом в Airflow является DAG-файл (DAG- directed acyclic graph, то есть прямой ациклический граф).

Использование веб-интерфейса Airflow

В Системе реализовано два инстанса (экземпляра) Airflow: Airflow – для отладки моделей в среде DEV и Airflow-prod – для моделей в контуре PROD.

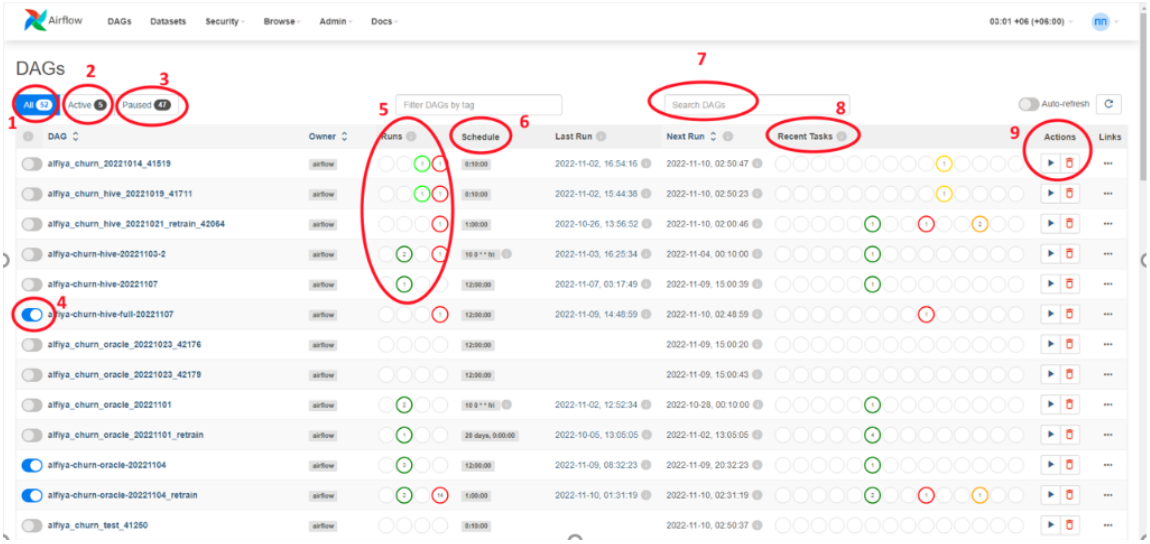
При первом переходе по данным адресам в браузере откроется страница (рисунок ниже), где нужно нажать на “Sign in with okta”, после этого откроется страница авторизации keycloak, где нужно будет ввести свои учетные данные.



После входа в Airflow (не важно в dev или prod) на экране будет основная страница интерфейса.

Главная страница Airflow представляет собой таблицу, в которой отображаются DAG-файлы. Для Airflow в среде DEV данные DAG-файлы берутся из репозитория airflow-dev (в группе gbc) в GitLab. А в этот репозиторий они попадают на третьей стадии CI/CD процесса, который инициируется любым коммитом в ветку main репозитория с моделью. Название DAG-файла для скоринга (то есть запуска скрипта main.py внутри контейнера) совпадает с именем проекта модели, а название DAG-файла для переобучения модели состоит из названия проекта модели и далее слова retrain. Отображаемые в интерфейсе DAG-файлы переобучения также берутся из репозитория airflow-dev, в который они попадают на второй стадии CI/CD процесса переобучения модели, который инициируется любым коммитом в ветку retrain.

Рассмотрим основные поля главной страницы (рисунок ниже):



По умолчанию, вновь появившийся (после соответствующего CI/CD процесса) DAG-файл попадает в таблицу и является неактивным, другое название “стоящим на паузе” (серый ползунок в столбце слева). Если DAG-файл не активен, то он не будет запущен по расписанию. Если перевести этот ползунок в состояние активации или, “затриггерить” (как, например, на цифре 4 рисунка), то теперь Airflow будет отслеживать расписание запуска, заданное внутри данного файла и, если наступит условие запуска, то запустит DAG-файл.

Под цифрами 1, 2, 3 обозначены фильтры, показывающие все файлы (1), показывающие только активные файлы (2) и не активированные файлы (3).

В столбце под цифрой 6 указано расписание запуска данного файла, можно навести на это поле мышкой и увидеть более подробную информацию про расписание.

Для поиска нужного DAG-файла можно воспользоваться поиском, цифра 7.

Под цифрой 5 указано число запусков (runs) данного файла, а цветом указан статус запуска:

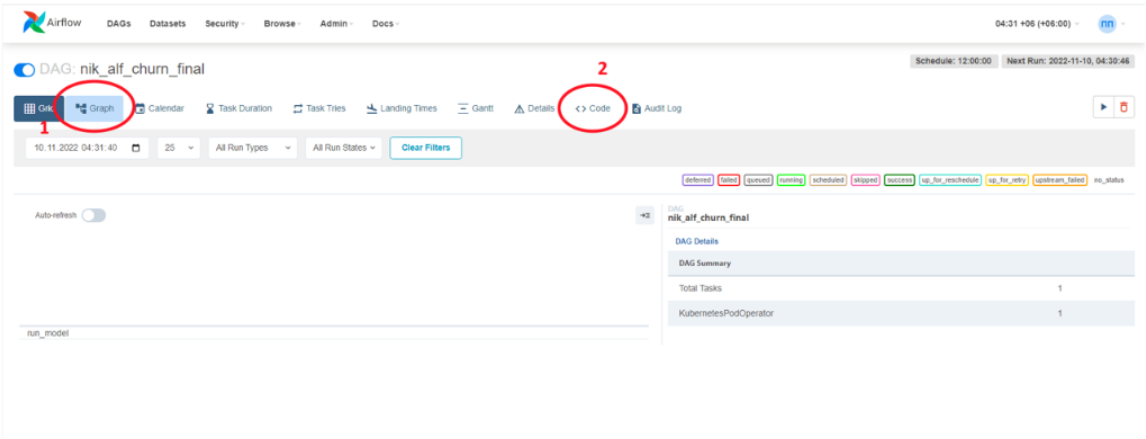
- Салатовый означает что запуск идет прямо сейчас,
- Темно-зеленый означает, что запуск прошел успешно,
- Красный - запуск окончился ошибкой,
- Светло-серый - запуск поставлен в очередь.

Под цифрой 8 указаны последние tasks (таски) запуска. В нашем случае, для скоринга модели таска тождественна рану, поскольку ран содержит одну таску. А в случае с переобучением один ран содержит четыре таски. Для упрощения можно считать, что таска — это стадия работы DAG-файла, а если более точно, то каждая таска описывается как объект KubernetesPodOperator внутри DAG-файла. Напомним, что в разделе “Описание файла dag-retrain.py” в примере кода имеется четыре объекта

KubernetesPodOperator, а в файле “dag-batch.py” описан один KubernetesPodOperator, поэтому DAG-файл скоринга содержит одну таску, а DAG-файл переобучения - четыре.

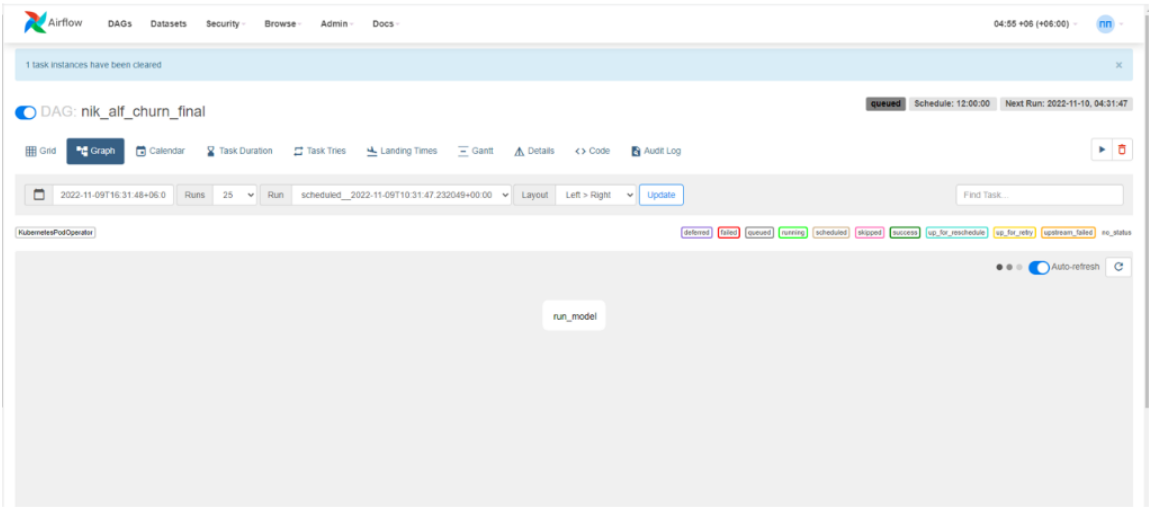
Под цифрой 9 указаны кнопка ручного запуска (не по расписанию, а прямо в данный момент) и кнопка удаления DAG-файла. Отметим, что при ручном запуске будет предложено две опции: “Trigger DAG” и “Trigger DAG w config”. В случае выбора первой, DAG-файл запустится как есть, в случае выбора второй опции, будет предложено заполнить json файл, где можно указать значение переменной, которое далее будет “проброшено” в контейнер с моделью как переменная окружения. Такой подход может быть полезен, когда нужно запустить скоринг моделью не по расписанию, а на конкретную дату.

Для активации или снятия с паузы нужного DAG-файла нужно, как было сказано выше, передвинуть ползунок в левом столбце. Далее можно “провалиться” в название данного файлы и попасть на страницу, изображенную на рисунке ниже:

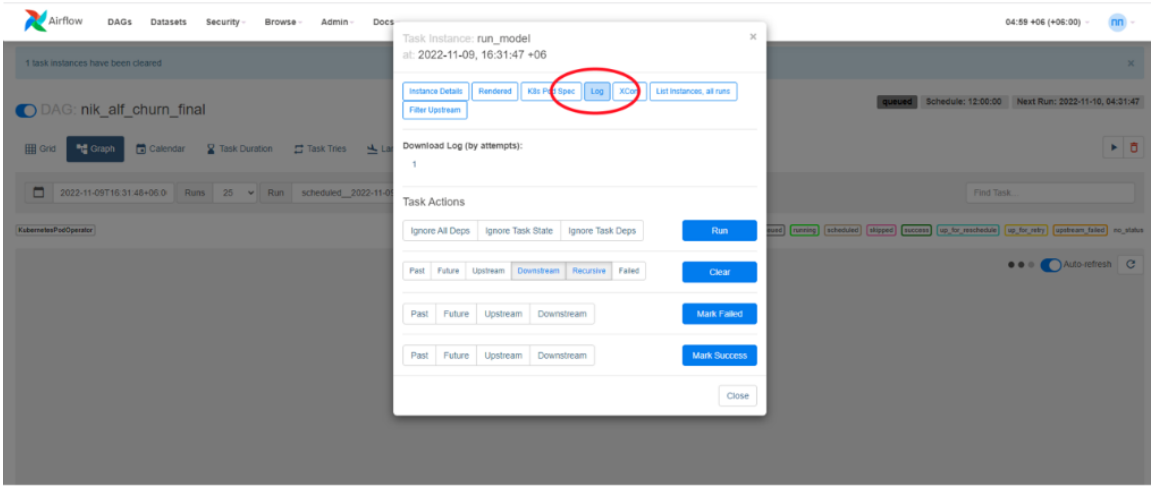


Далее нужно перейти на вкладку Graph (цифра 1), также стоит обратить на вкладку Code (цифра 2), нажав на которую можно перейти и увидеть код DAG-файла, что бывает полезно при отладке.

Перейдя на вкладку Graph на экране, будет прямоугольник “run_model” (для DAG-файла скоринга, не переобучения), рисунок «Интерфейс AirFlow2», можно отметить, что данный прямоугольник и является таской в данном DAG-файле.



Затем нужно нажать левой кнопкой мыши на “run_model” и в появившемся окошке выбрать “Log”:



В открывшемся окне будут выводиться логи запуска данного DAG-файла, с помощью которых можно смотреть вывод лога модели. Работа в Airflow была рассмотрена на примере инстанса Airflow для среды DEV. Работа в Airflow-prod полностью аналогична.