# Модуль 3. Управление потоком операций (xFlow).

## В этом модуле

Конвейеры состоят из операций, ходом выполнения которых необходимо управлять: контролировать последовательность выполнения, осуществлять мониторинг, своевременно устранять ошибки, версионировать сами конвейеры и результаты их работы. Операции в конвейере объединяются с использованием направленного ациклического графа (DAG, Directed Acyclic Graph). Последовательное и параллельное выполнение операций в последовательности, заданной в DAG, создает поток (flow) действий и появляющихся артефактов. Существуют различные инструменты для работы с таким потоком операций, с условным названием xFlow. Два таких инструмента, AirFlow и MLFlow мы рассмотрим в этом модуле.

## Содержание юнитов

|  |
| --- |
| 1. Постановка задачи |
| 1. Настройка окружения проекта |
| 1. Настройка Apache Airflow для автоматизации сбора данных и обучения |
| 1. Настройка MLFlow для автоматизации мониторинга |
| 1. Задание для самостоятельной работы |

# Модуль 2. Юнит 5. Постановка задачи

## Введение

В этом юните описана практическая задача, которую вы будете решать с применением Apache Airflow и MLFlow в следующих юнитах 6-8.

## Содержание

Предположим, что в нашем проекте понадобилось собирать информацию о рейтинге фильма или телепередачи из доступных источников. Такими источниками могут быть:

* классический датасет отзывов на фильмы IMDB,
* youtube.ru,
* wikipedia,
* тематические сайты,
* тематические группы в мессенджерах,
* и многие другие источники…

Актуальность сбора информации связана с тем, что рейтинги постоянно обновляются, а также могут выходить новые фильмы и передачи. Интерфейсы, по которым может быть получена эта информация, тоже различны, это могут быть:

* реляционные и нереляционные базы данных,
* стандартные датасеты, например, в xml или csv формате,
* API к агрегаторам новостей, социальным сетям, мессенджерам,
* GraphQL интерфейс DBpedia (API для доступа к информационным базам wikipedia),
* RSS каналы
* html страницы

В зависимости от типа интерфейса для интеграции с соответствующим источником информации в программном обеспечении создается соответствующий коннектор, поддерживающий интерфейс для интеграции с источником данных.

Для реализации задачи сбора информации о рейтингах фильмов нам необходимо создать управляемый конвейер, состоящий из операций запуска коннекторов для разных источников по расписанию. Коннекторы будут запускаться в определенное время, брать из конфигурационного файла перечень фильмов, для которых необходимо получить информацию о рейтинге, делать опрос доступных источников и сохранять информацию в базу данных. Далее будет запускаться препроцессинг полученных данных, преобразование в числовые признаки, векторы эмбеддингов, на основании которых алгоритм регрессии будет предсказывать рейтинг фильма в будущем.

Для автоматизации запуска коннекторов для сбора информации по расписанию мы будем использовать Apache Airflow. Также Apache Airflow мы используем для сохранения собранной информации в базе данных в «сыром» виде. Для упрощения задачи ограничимся чтением информации из youtube. Следующая операция, которая войдет в автоматизированный конвейер Apache Airflow это предобработка данных и запуск обучения и тестирования модели. Для мониторинга вышеупомянутых операций мы используем MLFlow.

## Тест

1. Какие инструменты MLOps вы будете применять в этом модуле? (0.25)
   1. **MLFlow**
   2. ClearML
   3. dvc
   4. **Airflow**
2. С каким источником информации будет выполняться интеграция в задании? (0.25)
   1. **youtube**
   2. twitter
   3. yandex
   4. rbc.ru
3. Какие интерфейсы могут быть использованы для получения данных из внешних источников? (0.25)
   1. **SQL запросы**
   2. **API и JSON**
   3. ICMP
   4. **GraphQL**
4. Какие задачи автоматизирует Airflow в конвейере проекта? (0.25)
   1. **сбор данных**
   2. **обработка данных**
   3. **обучение модели**
   4. мониторинг и сохранение артефектов обучения модели

## Итоги/выводы

В этом юните сформулирована задача, которую вы будете решать в следующих юнитах, и даны рекомендации по ее решению.

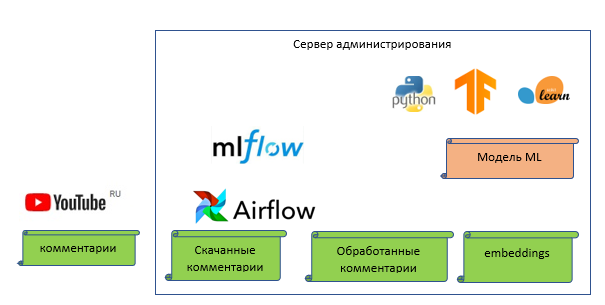
# Модуль 3. Юнит 6. Настройка окружения проекта.

## Введение

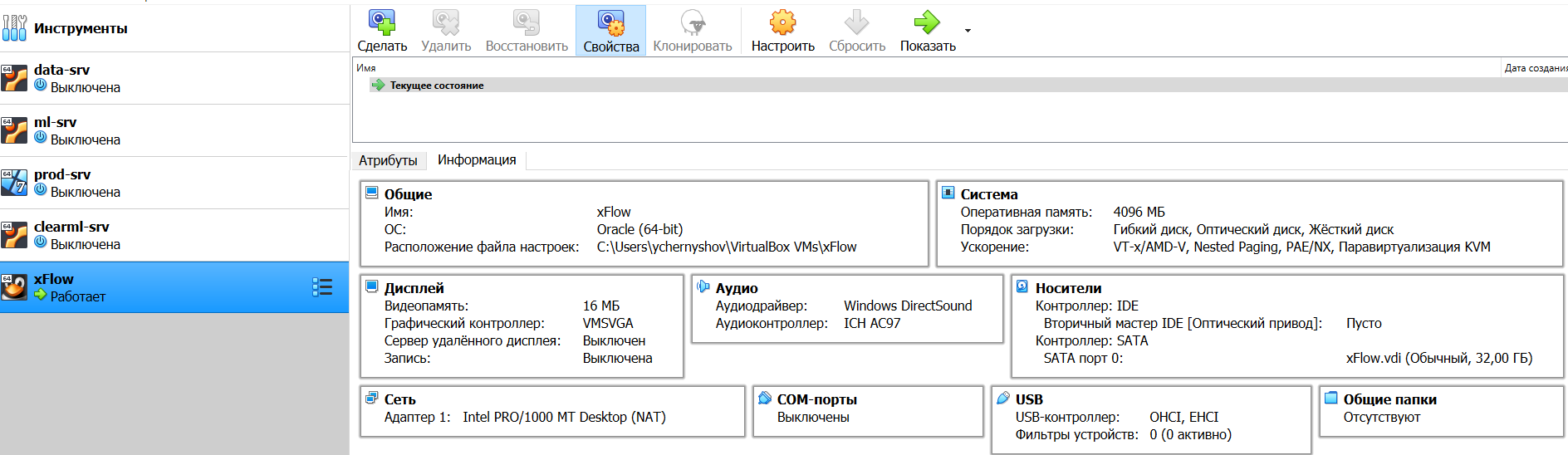
В этом юните вы настроите рабочее окружение для решения задачи: создадите необходимую инфраструктуру и установите системное и прикладное программное обеспечение.

## Содержание

Для работы вам понадобится виртуальная машина, на которую будет установлено все необходимое программное обеспечение. Главная задача в реализуемом проекте - организация автоматизированного сбора данных, обработка этих данных и первичное тестирование с использованием простых baseline-моделей. То есть вам понадобится инструмент для автоматизации операций сбора данных, в этой роли будем использовать Airflow, а также инструмент для мониторинга операция сбора данных и обучения моделей, для этого будем использовать MLFlow. В этом задании нет необходимости разделять функции и программное обеспечение на нескольких виртуальных серверах. Поэтому далее создадим одну виртуальную машину и на ней установим все необходимое программное обеспечение.



Поднимаем виртуальную машину «xflow» в virtualbox



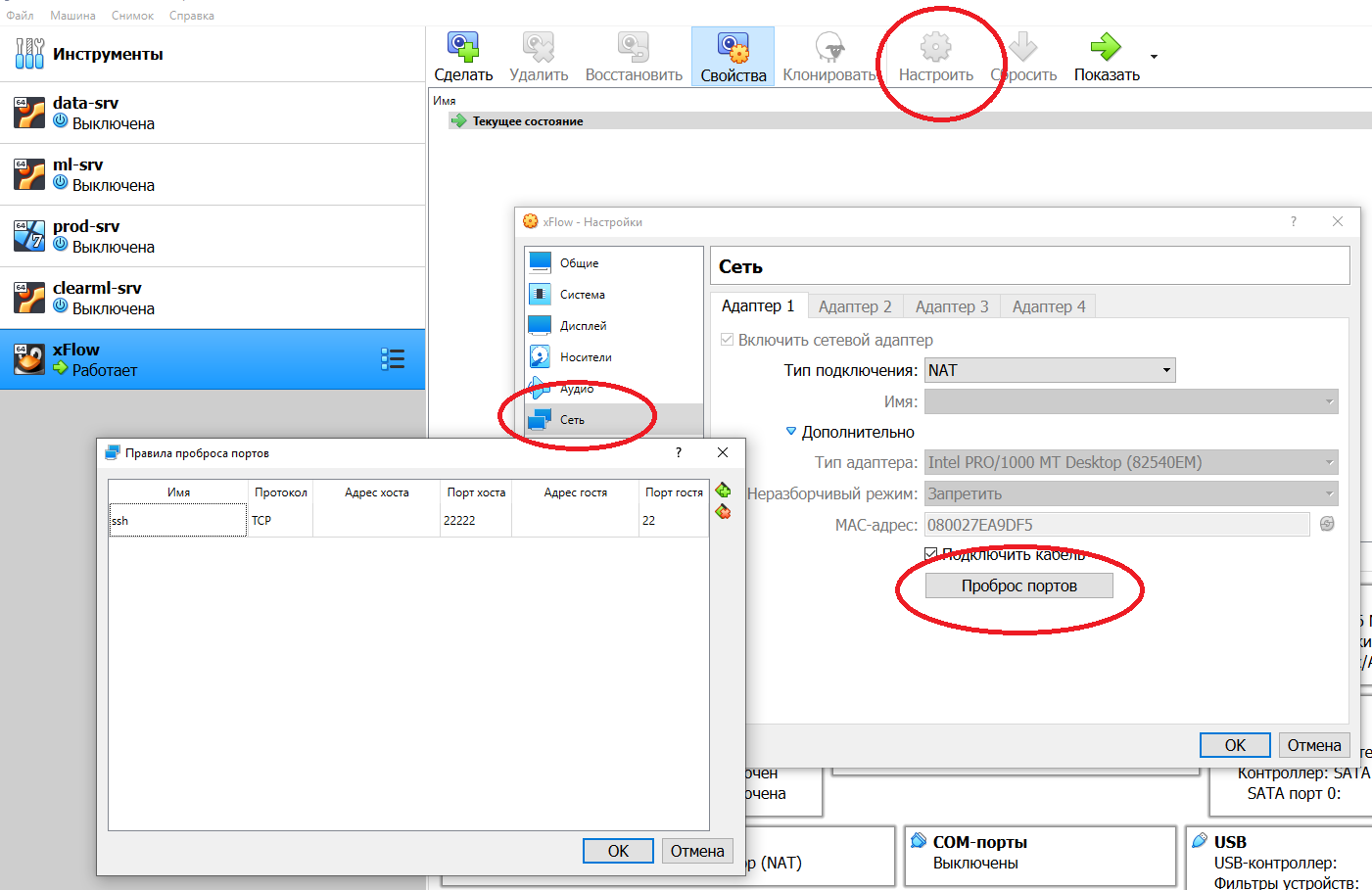
В качестве операционной системы, также, как и в предыдущих модулях, рекомендуем использовать Ubuntu 20.4.

## Установка и настройка прикладного программного обеспечения разработчика.

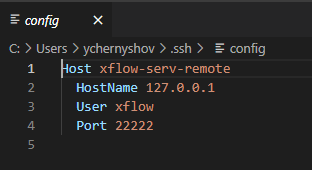
Для удобства работы нам понадобится инструмент разработчика IDE, например уже знакомы вам VS Code, который подключается удаленно к серверу через ssh и позволяет редактировать и выполнять скрипты на удаленном сервере. Для организации такой инфраструктуры требуется установить openssh-server на виртуаьлной машине xflow командой

**sudo apt install openssh-server**

затем прокинуть порт в virtualbox



После чего в конфигурационном файле ssh в VS Code прописать параметры подключения



И открыть удаленное подключение нажав на кнопку  слева внизу в интерфейсе VS Code.

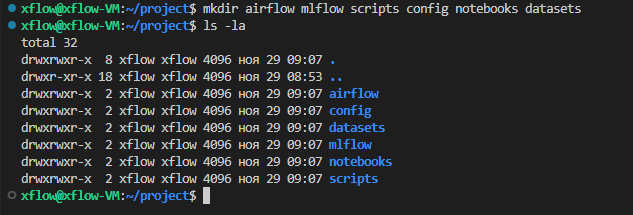
## Создание файловой структуры и виртуального окружения.

Для создания нужной конфигурации для работы вам понадобится организовать файловую структуру. Для этого в своей пользовательской директории создайте директорий проекта и в нем директории mlflow, airflow, notebooks (если планируете хранить и использовать jupyter ноутбуки), config (для конфигурационных файлов).

**mkdir project**

**cd project**

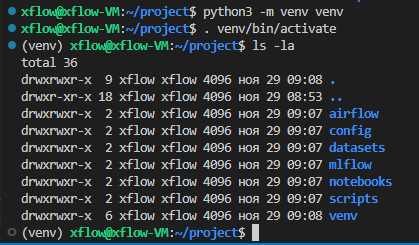
**mkdir airflow mlflow scripts config notebooks datasets**

****

Находясь в главной папке проекта создайте и активируйте виртуальное окружение venv

**python3 -m venv venv**

**source venv/bin/activate**



Теперь у вас есть виртуальное окружение, в котором вы можете установить все необходимые пакеты, в том числе airflow и mlflow.

## Установка и настройка airflow

Программное обеспечение Apache Airflow предназначено для соединения отдельных операций в единый конвейер, который описывается ациклическим направленным графом операций (DAG, Directed Acyclic Graph). В этот граф вы объедините все операции в проекте. Процесс установки Apache Airflow описан в официальной документации на сайте <https://airflow.apache.org/docs/apache-airflow/stable/installation/index.html>. Есть различные способы развернуть Airflow, например с использованием docker и helm. Мы используем самый простой, уже известный вам вариант, установку с использованием python установщика pip. Официально поддерживаемым установщиком является только pip, разработчики Apache Airflow не гарантируют работоспособность при использовании в качестве установщика poetry или pip-tools. Важно, что airflow устанавливается вместе с зависимостями, если этого не сделать, то возможна некорректная работа airflow. Вот образец команды из официальной документации на сайте, Раздел «Installation from PyPI» <https://airflow.apache.org/docs/apache-airflow/stable/installation/installing-from-pypi.html>:

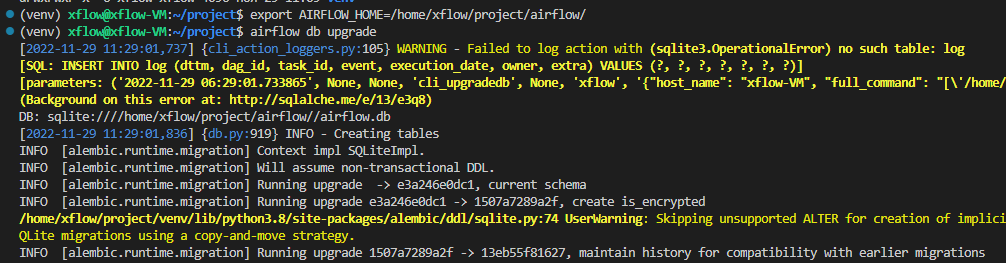
**pip install "apache-airflow[celery]==2.2.5" \**

**--constraint https://raw.githubusercontent.com/apache/airflow/constraints-2.2.5/constraints-3.8.txt**

В этом команде «2.2.5» это версия airflow, а «3.8» - версия используемого интерпретатора python. После установки необходимо установить значение переменной окружения AIRFLOW\_HOME и выполнить инициализацию базы данных.

**export AIRFLOW\_HOME=/home/xflow/project/airflow**

**airflow db init** или **airflow db upgrade**

****

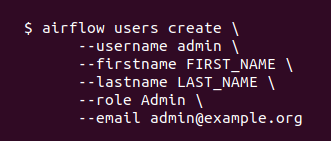
В файле airflow.cfg в разделе webserver необходимо установить значение параметров rbac, а также load\_examples

**[webserver]**

**rbac = True**

**load\_examples = False**

Создание пользователя

****

**source “venv”**

Теперь можно запустить airflow web-сервер для того, чтобы работать в графическом интерфейсе, а также scheduler, обеспечивающий выполнение задач по расписанию

**airflow webserver -p 8080**

**airflow scheduler**

Полезная команда, которая находит и останавливает процессы, содержащие в названии airflow

**kill -9 `ps aux | grep airflow | awk ‘{print $2}’`**

## Установка и настройка MLFlow

Программное обеспечение MLFlow понадобится в проекте для того, чтобы осуществлять мониторинг всех операций в проекте: сбор и обработка данных, формирование векторов эмбеддингов, обучение модели.

Для установки MLFlow необходимо выполнить следующие операции:

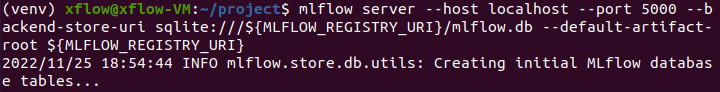
**pip install mlflow**

**export MLFLOW\_REGISTRY\_URI=mlflow**

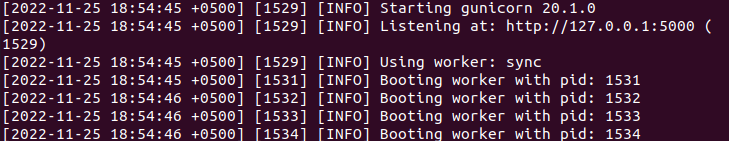
**mlflow server --host localhost --port 5000 \**

**--backend-store-uri sqlite:///${MLFLOW\_REGISTRY\_URI}/mlflow.db \**

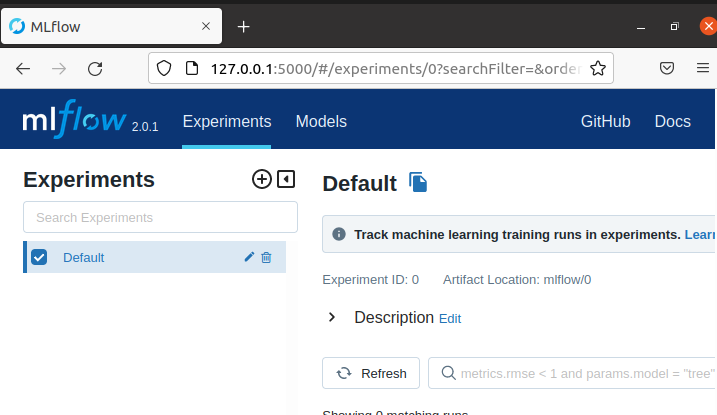
**--default-artifact-root ${MLFLOW\_REGISTRY\_URI}**

****

**…**

****

Теперь можно открыть в браузере приложение на localhost:5000 (обратите внимание, что по умолчанию используется небезопасный http, а не https)



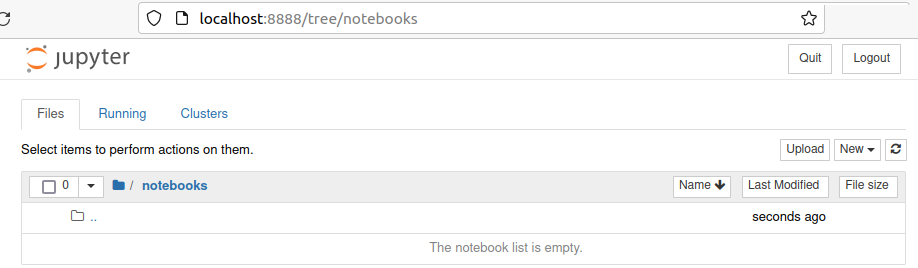
Для организации работы исследователя данных или экспериментов инженера ML понадобится jupyter notebook, давайте его также установим в системе. Находясь в созданном вами ранее виртуальном окружении установите программы jupyter с помощью команды

pip install jupyter

в том числе будет установлен jupyter notebook, который можно запустить в командной строке

**jupyter notebook**

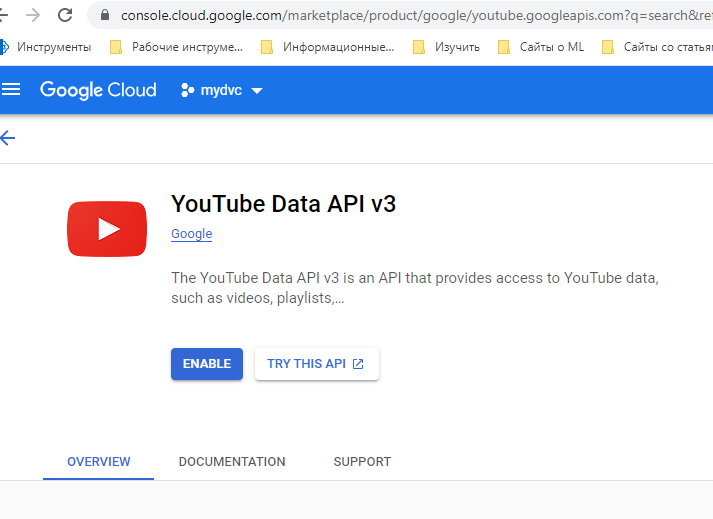
после чего появится возможность открывать jupyter ноутбуки в браузере

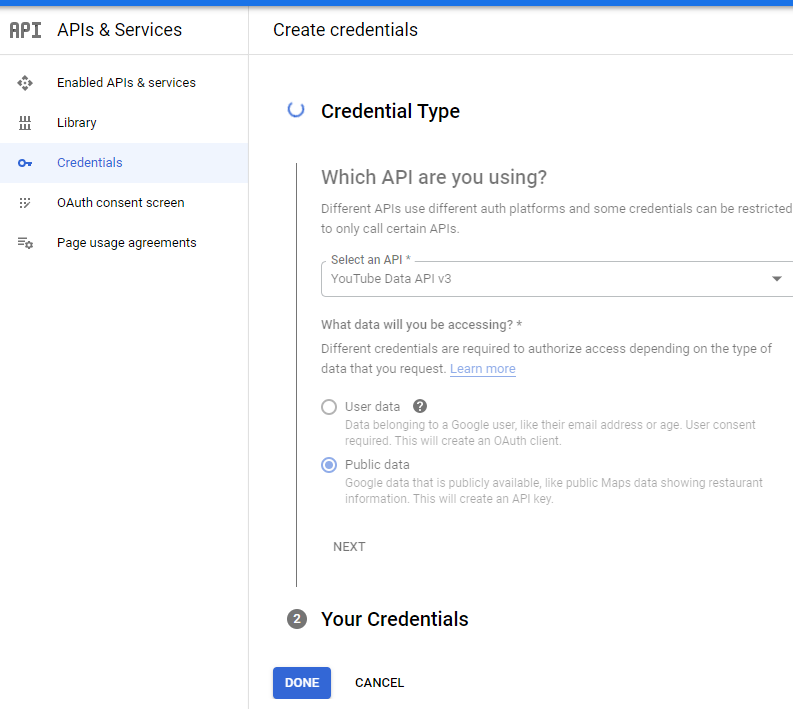


Для сбора данных с youtube понадобится библиотека python-youtube, которую можно установить в виртуальное окружение командой

**pip install python-youtube**

Также понадобится ключ для API





## Тест

1. Какая база данных используется в качестве служебной для airflow и mlflow в примере? (0.25)
   1. mysql
   2. **sqlite**
   3. postgresql
   4. oracle
2. Какая команда удалит все процессы, содержащие в названии «mlflow»? (0.25)
   1. **kill -9 `ps aux | grep airflow | awk ‘{print $2}’`**
   2. delete all mlflow
   3. kill mlflow
   4. remove mlflow ps
3. Для чего предназначен airflow sheduler? (0.25)
   1. **Контроль запуска выполнения задач**
   2. Тестирование модели
   3. Настройка базы данных
   4. Аутентификация пользователей
4. На каком номере порта открывается графический интерфейс mlflow? (0.25)
   1. 1000
   2. **5000**
   3. 5555
   4. 5050

## Итоги/выводы

В этом юните вы создали необходимую инфраструктуру и настроили окружение для дальнейшей работы, в том числе установили Apache Airflow и MLFlow.

# Модуль 3. Юнит 7. Автоматизация сбора данных с использованием Airflow.

## Введение

В этом юните вы создадите модуль для автоматизированного сбора информации с использованием Apache Airflow.

## Содержание

## Тест

1. ? (0.25)
   1. **a**
   2. b
   3. c
   4. d
2. ? (0.25)
   1. **a**
   2. b
   3. **c**
   4. d
3. ? (0.25)
   1. a
   2. **b**
   3. c
   4. d
4. ? (0.25)
   1. a
   2. **b**
   3. c
   4. d

## Итоги/выводы

В этом юните вы создали модуль автоматизированного сбора информации с использованием Apache Airflow.

# Модуль 3. Юнит 8. Мониторинг операций с использованием MLFlow.

## Введение

В этом юните вы организуете мониторинг выполнения всех операций проекта с использованием MLFlow.

## Содержание

## Тест

1. ? (0.25)
   1. a,
   2. b,
   3. **c,**
   4. d
2. ? (0.25)
   1. a
   2. b
   3. c
   4. d
3. ? (0.25)
   1. a
   2. b
   3. c
   4. d
4. ? (0.25)
   1. a
   2. b
   3. c
   4. d

## Итоги/выводы

В этом юните вы организуете мониторинг выполнения всех операций проекта с использованием MLFlow.

# Модуль 3. Юнит 9. Задача для самостоятельной работы.

## Содержание

## Тест

1. ? (0.25)
   1. **a**
   2. b
   3. c
   4. d
2. ? (0.25)
   1. **a**
   2. b
   3. **c**
   4. d
3. ? (0.25)
   1. a
   2. **b**
   3. c
   4. d
4. ? (0.25)
   1. a
   2. **b**
   3. c
   4. d

## Итоги/выводы

В этом юните сформулирована задача для самостоятельной работы и даны рекомендации по ее решению.

# Список источников

1. Видеолекция «Airflow и MLFlow автоматизаций пайплайнов Machine Learning / MLOps»

<https://www.youtube.com/watch?v=NfPf0Y770DA&t=2s> (канал miracl6)

<https://pypi.org/project/python-youtube/>

https://developers.google.com/youtube/v3/docs/