

Министерство науки и высшего образования Российской Федерации



Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования

*«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»*

(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»

Ю.С. Белов

**ТЕХНОЛОГИИ РАСПОЗНАВАНИЯ ОБЪЕКТОВ В ОБЛАЧНОЙ
СИСТЕМЕ VK CLOUD – ML Platform**

методические указания к лабораторной работе

по дисциплине

«Облачные технологии»

Калуга - 2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Настоящие методические указания составлены в соответствии с программой проведения лабораторных работ по курсу «Облачные технологии» на кафедре «Программное обеспечение ЭВМ, информационные технологии» факультета информатики и управления Калужского филиала МГТУ им. Н.Э. Баумана.

Методические указания, ориентированные на студентов бакалавриата по направлению подготовки 09.03.04 «Программная инженерия», содержат краткую теоретическую часть, описывающую работу с облачным сервисом VK Cloud.

Методические указания составлены в расчете на всестороннее ознакомление студентов с основами работы с VK Cloud.

ЦЕЛЬ И ЗАДАЧИ РАБОТЫ, ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ЕЕ ВЫПОЛНЕНИЯ

Целью выполнения лабораторной работы является приобретение практических навыков по работе с сервисом VK Cloud.

Основными задачами выполнения лабораторной работы являются:

- Ознакомиться с системой VK Cloud.
- Изучить основы распознавания объекта.
- Научиться распознавать объекты на изображении.

Результатами работы являются:

- Разработанная программа согласно варианту задания
- Подготовленный отчет

ХАРАКТЕРИСТИКА ОБЪЕКТА ИЗУЧЕНИЯ, ИССЛЕДОВАНИЯ

MLOps Platform от VK Cloud представляет собой набор сервисов и инструментов, предназначенных для решения типовых задач Data Science и ML команд.

MLOps можно определить, как дисциплину или набор практик, целью которых является деплой и поддержка сервисов на основе ML-моделей. Данная дисциплина родилась на стыке сфер Machine Learning, DevOps и Data Engineering.

MLOps задачи обычно включают в себя:

- Проведение экспериментов с данными;
- Тренировку ML-моделей;
- Деплой полученных ML-моделей в различные среды, такие как, например, development и production;
- Мониторинг поведения моделей в production;
- Обновление эксплуатируемых ML-моделей.

Для решения MLOps задач VK Cloud предоставляет следующие сервисы:

- JupyterHub — инструмент для работы с Jupyter Notebook в многопользовательском режиме;
- MLflow — инструмент для решения задач трекинга и версионирования экспериментов, ML моделей.

MLflow и Cloud ML Platform

Сервис MLflow – облачное решение для автоматизации работы с моделями машинного обучения. MLflow поможет специалистам по данным управлять проведением экспериментов, сравнивать ML-модели и их результаты, а также воспроизводить тесты.

Облачная платформа Cloud ML Platform объединяет преднастроенные сервисы JupyterHub и MLflow для проведения экспериментов с данными и работы с моделями машинного обучения. Это полностью готовая среда для создания ML-решений, которую можно развернуть в облаке в несколько кликов.

Данный компонент платформы MLflow позволяет компаниям реализовать концепцию MLOps для стандартизации разработки моделей, сокращения времени вывода их в продакшен, эффективного трекинга и контроля версий. С помощью ML Platform дата-инженеры и дата-сайентисты смогут самостоятельно разворачивать среду для экспериментов и построения моделей, а также масштабировать вычислительные мощности под любые нагрузки. Это упрощает процесс проверки гипотез и повышает скорость получения результатов.

На платформе VK Cloud сервис MLflow интегрирован с JupyterHub. В будущем появится интеграция с объектным S3-совместимым хранилищем и Managed-базами данных.

В отличие от Kubeflow, MLflow может работать без Kubernetes. Но при этом MLflow умеет упаковывать модели в Docker-образы, чтобы потом их можно было развернуть в Kubernetes.

Соответственно, Cloud ML Platform от VK позволяет разделить роли и зоны ответственности в команде, настроить процесс решения задач и отслеживать результаты экспериментов участников. Дата-инженеры смогут развернуть среду для работы с данными и построения моделей, самостоятельно и быстро масштабировать вычислительные мощности под любые задачи и нагрузки. Специалистам по Data Science платформа даст возможность гибко и оперативно проводить эксперименты с данными, формулировать и проверять гипотезы.

В Cloud ML Platform доступны инструменты для управления доступом к данным и компонентам среды, а также для интеграции платформы с корпоративными системами аутентификации, логирования и аудита

пользователей. Благодаря этому администраторы системы и специалисты по безопасности обеспечат контроль за работой с данными и смогут оперативно анализировать изменения в проекте.

Итак, MLflow состоит из нескольких компонентов.

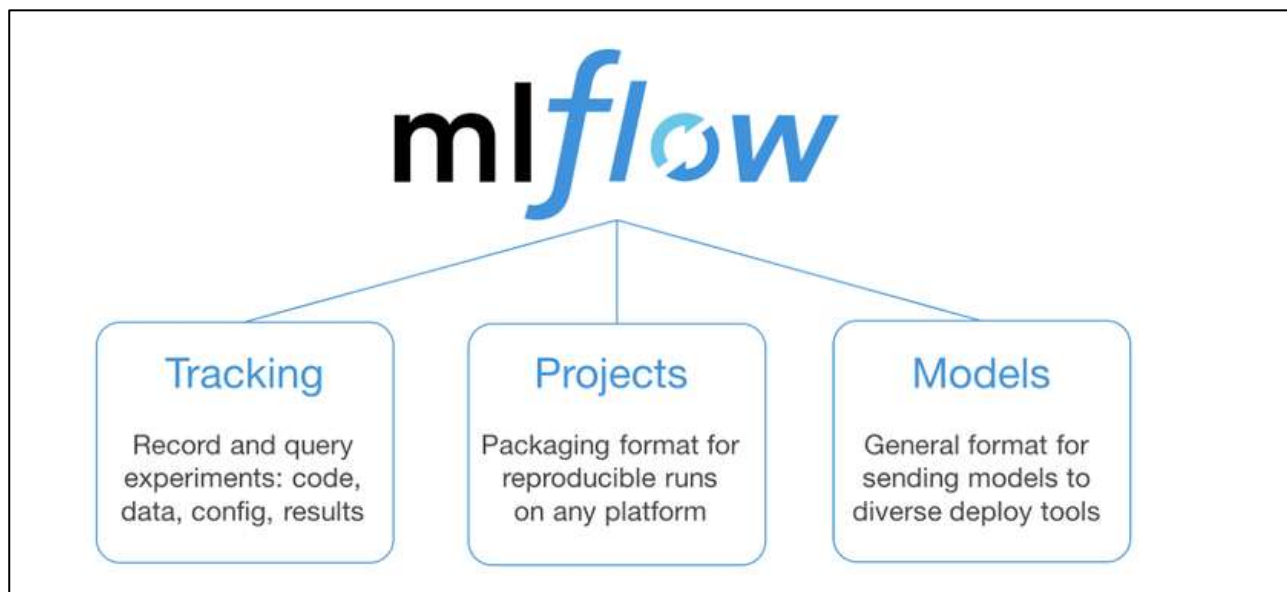


Рис. 1. Возможности MLflow

MLflow Tracking. Это удобный UI, в котором можно просматривать артефакты: графики, примеры данных, датасеты. Также можно смотреть метрики и параметры моделей. У MLflow Tracking есть API для разных языков программирования, с помощью которого можно логировать метрики, параметры и артефакты. Поддерживаются Python, Java, R, REST.

В MLflow Tracking есть две важных концепции: runs и experiments.

- Run — это единичная итерация эксперимента. Например, вы задали параметры модели и запускаете ее на обучение. На этот единичный запуск появится новая запись в MLflow Tracking. При изменении параметров модели будет создан новый run.
- Experiment позволяет сгруппировать несколько запусков (Run) в одну сущность, чтобы их можно было удобно просматривать.

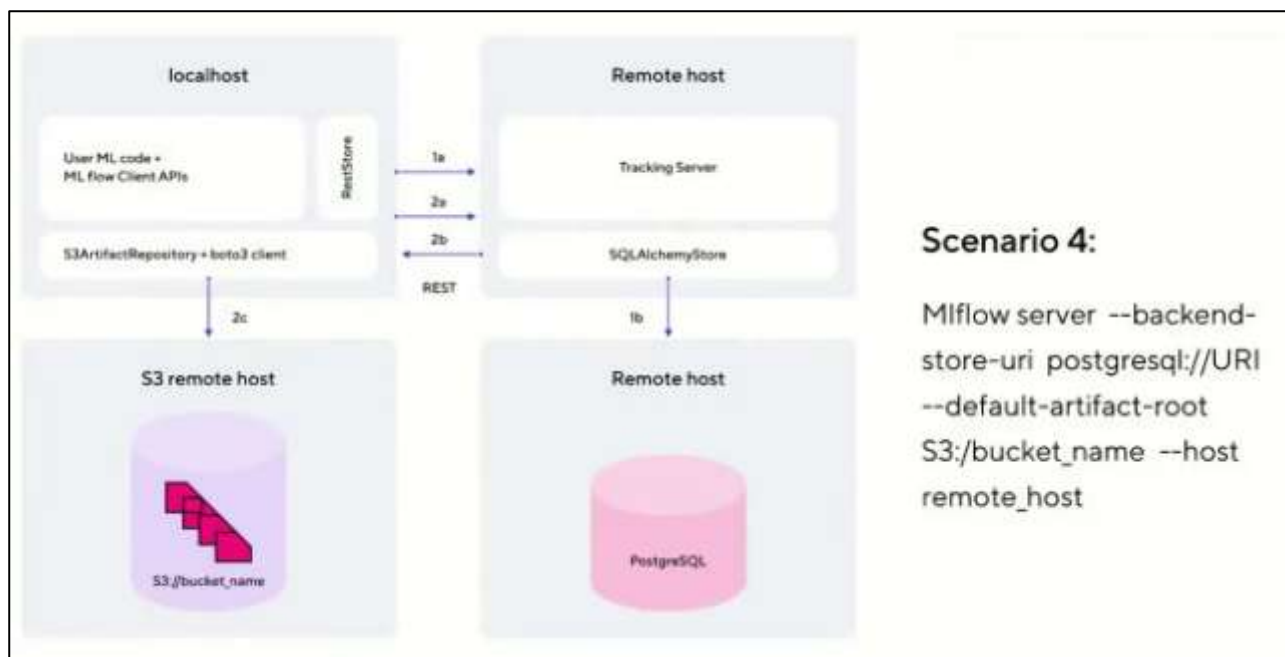


Рис. 2. Один из возможных сценариев использования MLflow Tracking

В этом варианте есть хост, на котором разворачиваются JupyterHub. Он взаимодействует с Tracking Server, размещенным на отдельной виртуальной машине в облаке. Для хранения метаданных об экспериментах используется PostgreSQL, который мы развернем как сервис в облаке. А все артефакты и модели хранятся отдельно в объектном хранилище S3.

MLflow Models. Этот компонент отвечает за упаковку, хранение и публикацию моделей. Он представляет собой концепцию flavor. Это своего рода обертка, которая позволяет использовать модель в различных инструментах и фреймворках без необходимости дополнительных интеграций. Например, можно использовать модели из Scikit-learn, Keras, TensorFlow, Spark MLlib и других фреймворков.

Также MLflow Models позволяет делать модели доступными по REST API и упаковывать их в Docker-образ для последующего использования в Kubernetes.

MLflow Registry. Этот компонент представляет собой центральный репозиторий моделей. Он включает в себя UI, который позволяет добавлять

теги и описание для каждой модели. Также он позволяет сравнивать разные модели между собой, например, чтобы увидеть отличия в параметрах.

MLflow Registry управляет жизненным циклом модели. В контексте MLflow есть три стадии жизненного цикла: Staging, Production и Archived. Также есть поддержка версионности. Все это позволяет удобно управлять всей выкаткой моделей.

MLflow Projects. Это способ организации и описания кода. Каждый проект — директория с набором файлов, чаще всего это пайплайны. Каждый проект описывается отдельным файлом MLProject в формате yaml. В нем указываются имя проекта, окружение и entrypoints. Это позволяет воспроизводить эксперимент в другом окружении. Также есть CLI и API для Python.

При помощи MLflow Projects можно создавать модули, которые представляют собой некие переиспользуемые шаги. Затем эти модули можно встраивать в более сложные пайплайны, что позволяет стандартизировать их.

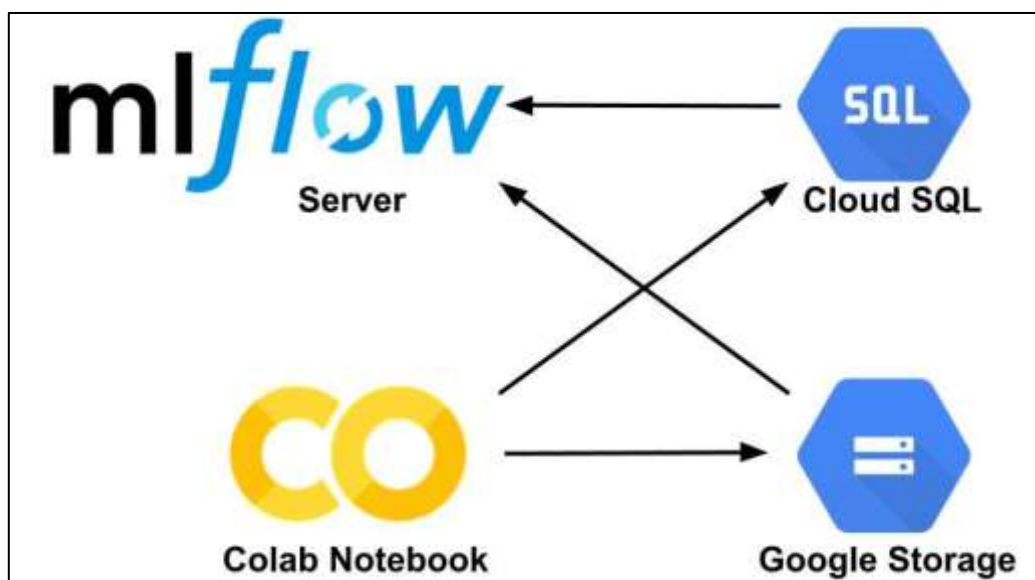


Рис. 3. Взаимодействие Colab с MLflow

Регистрация

Итак, для того чтобы приступить к работе с ML Platform в VK Cloud, необходимо перейти на официальный сайт - <https://mcs.mail.ru/> (Рис. 4).

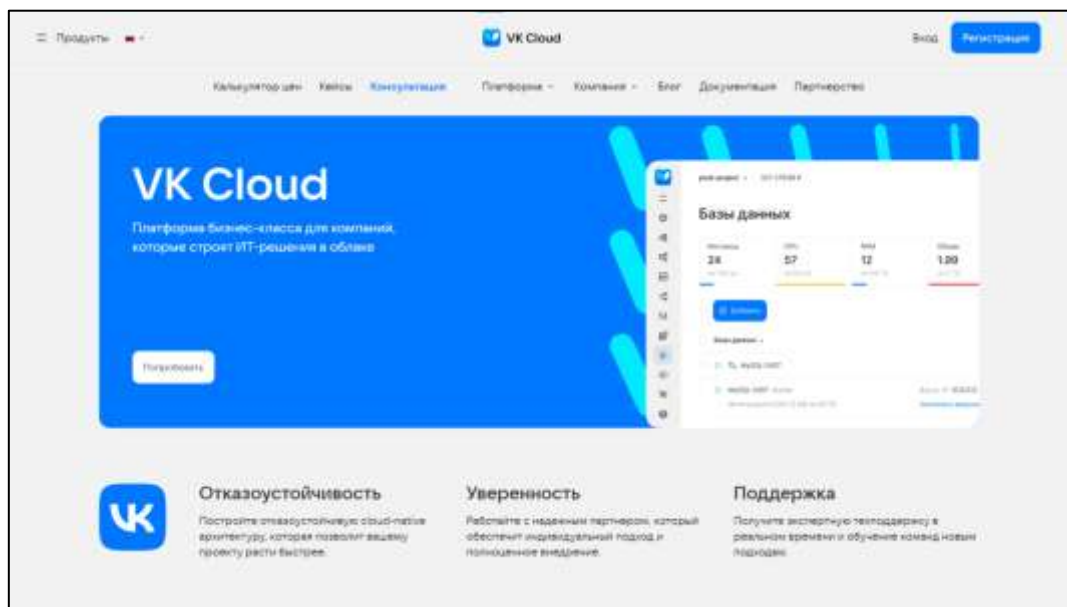


Рис. 4. Главная страница VK Cloud

Затем, на главной странице, в правом верхнем углу необходимо нажать на кнопку «Регистрация», после чего в появившемся окне, ввести Ваши данные и зарегистрироваться, нажав на «Создать аккаунт» (Рис. 5).

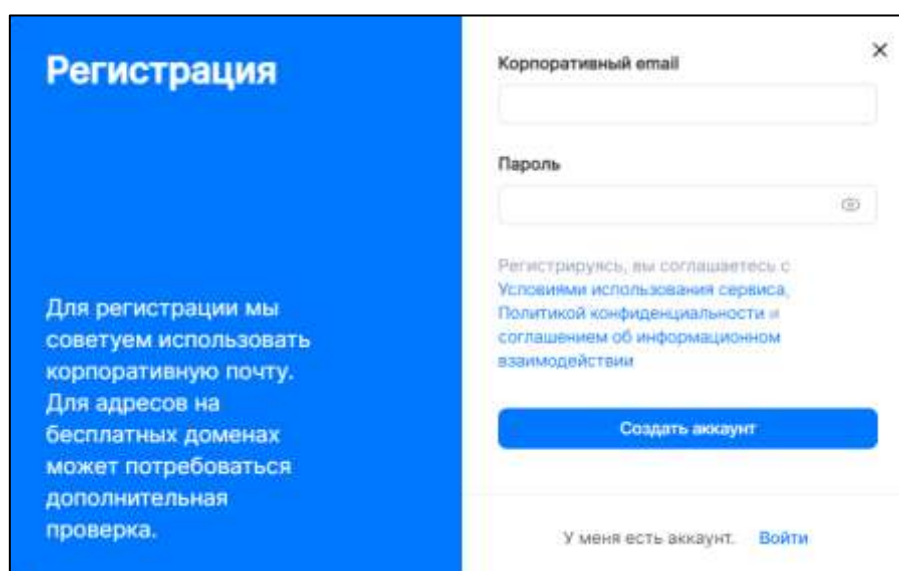


Рис. 5. Окно регистрации

Далее, вы увидите новое окно, в котором необходимо выполнить три этапа, для того, чтобы получить бонусную сумму на баланс вашего аккаунта, для дальнейшей возможности тестирования функционала VK Cloud и непосредственной работы с ML Platform (Рис. 6).

Сначала Вам необходимо подтвердить вашу почту, путем взаимодействия с сообщением от VK Cloud, которое уже должно прийти на указанный Вами адрес (в случае, если письмо не пришло, рекомендуется проверить папку «Спам»).

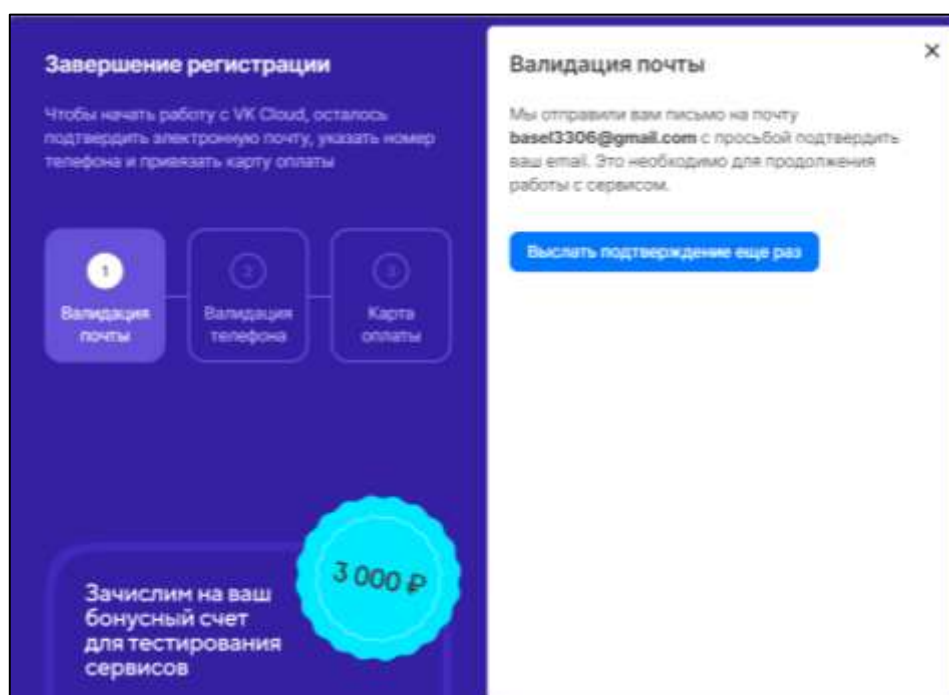


Рис. 6. Окно подтверждения почты

После успешного подтверждения Вашей почты, Вы увидите новое окно, в котором необходимо привязать Ваш телефон к созданному аккаунту (Рис. 7).

После ввода телефона, необходимо нажать на «Запросить код подтверждения», после чего появится новое поле, куда необходимо ввести пришедший Вам на телефон код.

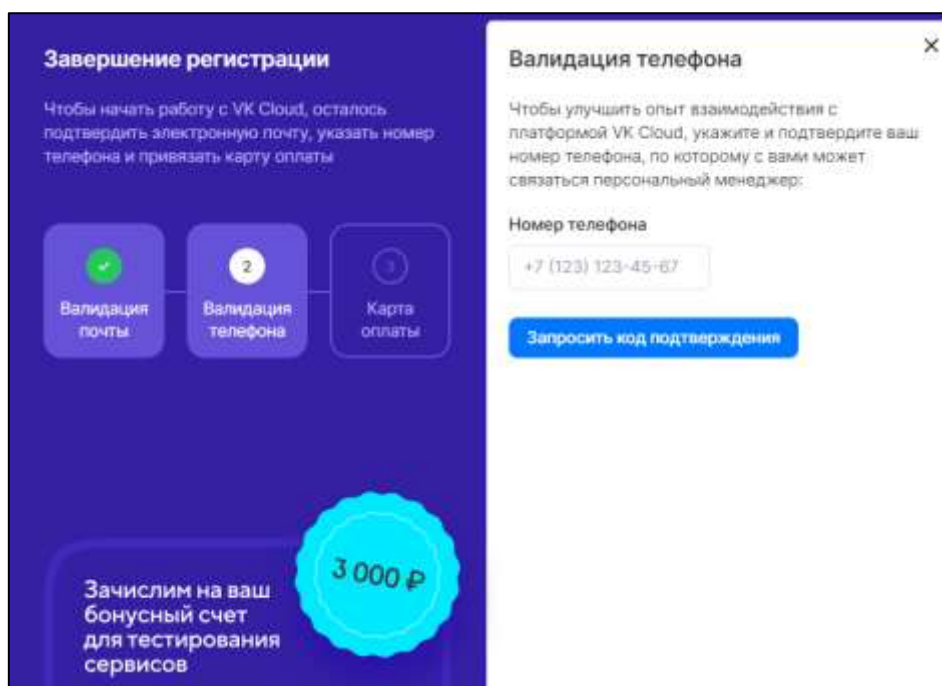


Рис. 7. Окно подтверждения номера телефона

После успешного подтверждения номера телефона, откроется следующий и последний этап завершения регистрации. В нем Вам необходимо привязать карту, обязательно имеющую на своем счету 1 рубль, для проверки возможности проведения транзакций. Данное действие необходимо для возможности тестирования инструментов VK Cloud.

Выполнив все 3 этапа, Вы получите 3000 рублей на Ваш аккаунт в VK Cloud на 60 дней, для тестирования необходимых Вам инструментов. Более подробную информацию об этих возможностях Вы можете узнать в официальной документации на сайте VK Cloud.

Теперь Ваш аккаунт готов для работы с облачными сервисами от VK Cloud.

ЗАДАНИЕ И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

ML Platform – начало

Итак, после регистрации Вы попадете на главную страницу сервиса VK Cloud (рис. 8).

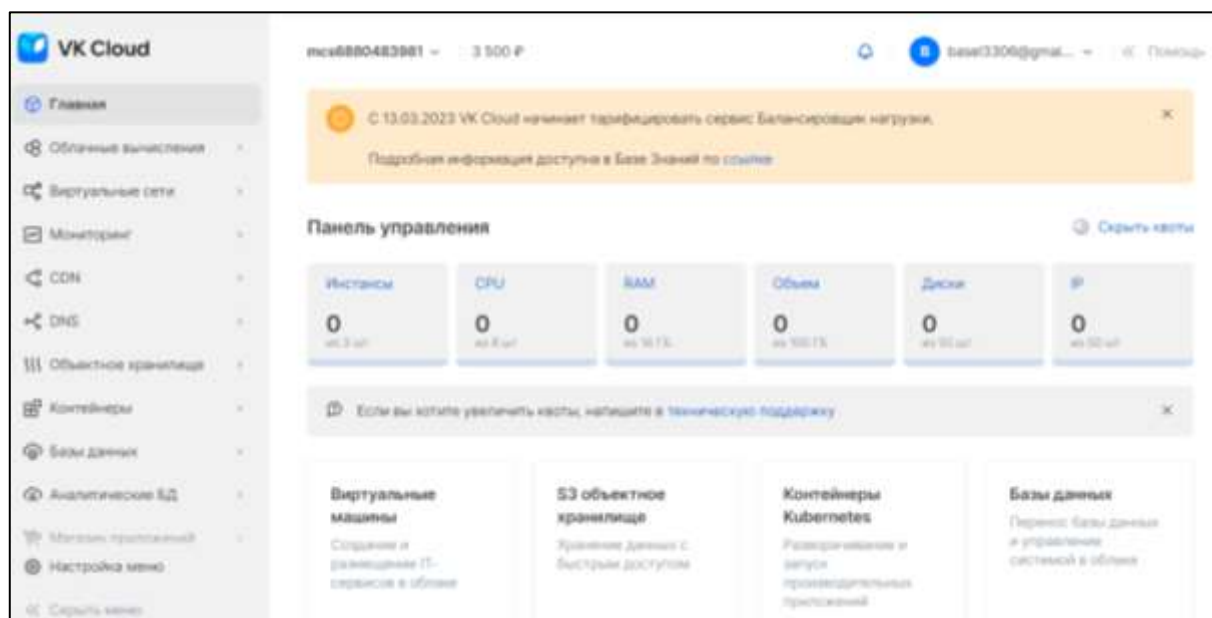


Рис. 8. Главная страница

Здесь, представлена панель управления, отображающая аппаратные ресурсы, используемые на Вашем аккаунте. Также слева располагается панель с различными сервисами. В случае, если Вам понадобится какая-либо помощь по данному сервису, справа вверху располагается кнопка «Помощь».

Приступим, найдите на панели слева пункт «ML Platform» и нажмите на него, после чего откроется главная страница выбранного Вами сервиса (Рис. 9).

На данной странице также присутствует документация по ML Platform, расположенная внизу страницы (рис.9).

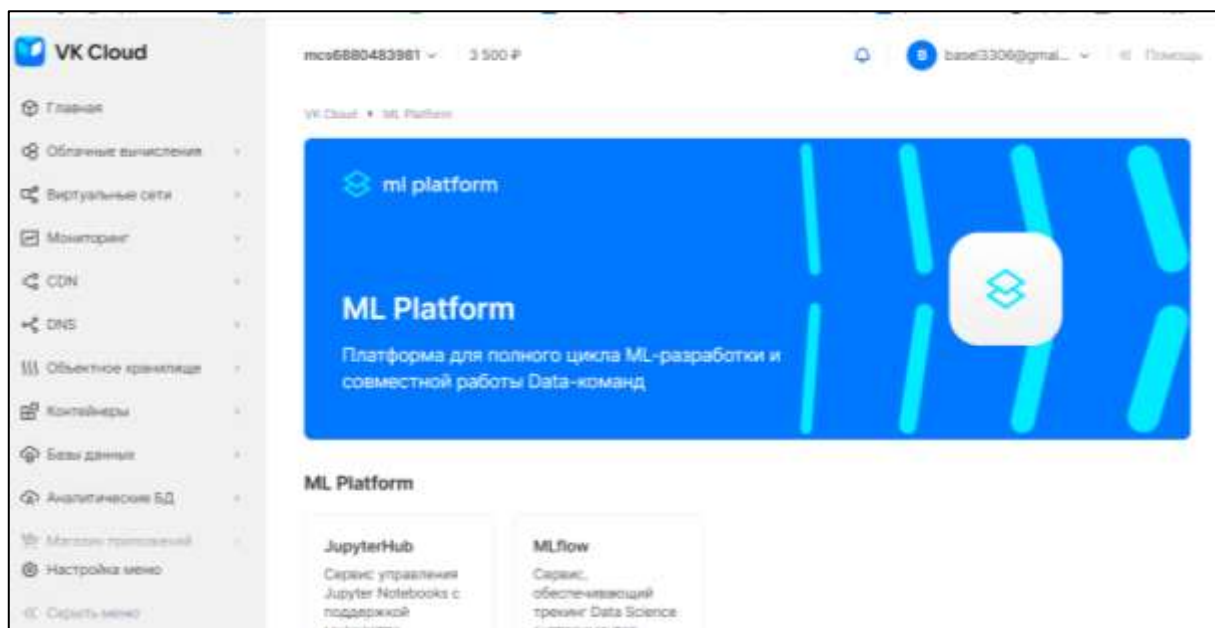


Рис. 9. Главная страница ML Platform

ML Platform – Создание инстанса

Для дальнейшей работы, необходимо создать инстанс. Для этого достаточно кликнуть на «Создать инстанс». После небольшой загрузки откроется меню создания нового инстанса с различными параметрами (Рис. 10).

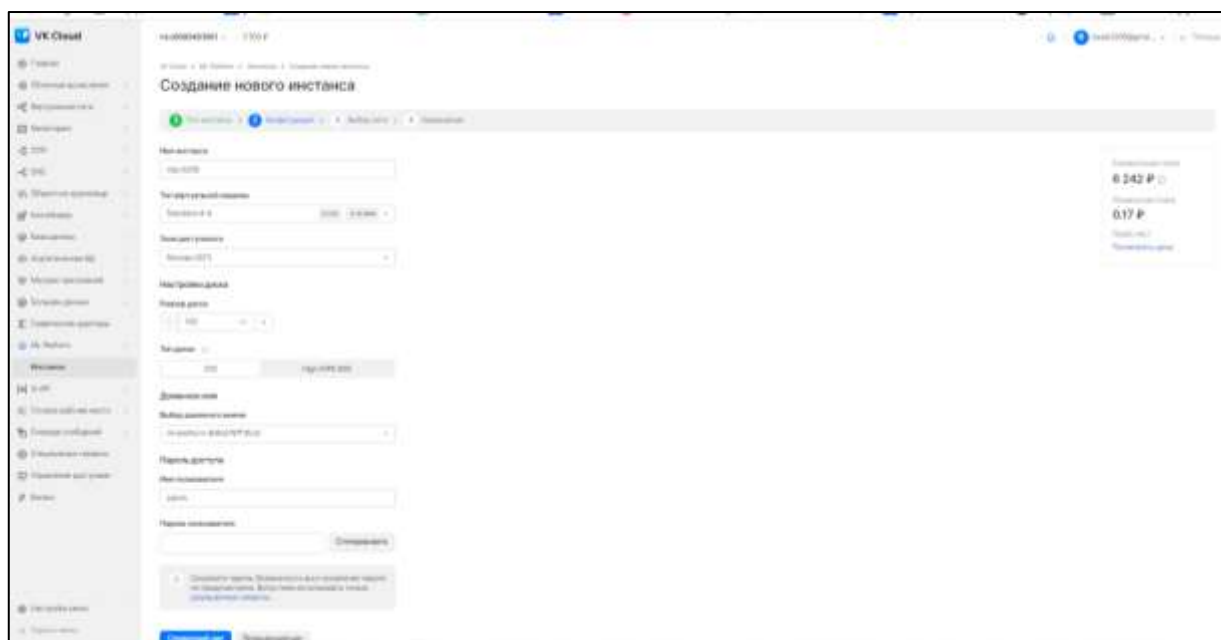


Рис. 10. Меню создание инстанса

В нем, Вы можете изменить созданное по умолчанию имя инстанса или же оставить автоматически созданное самим сервисом, как в данном случае - mlp-5078.

Тип виртуальной машины, Зону доступности, Размер диска и Тип диска рекомендуется не менять, поскольку стандартных значений будет достаточно для проведения необходимой работы в данном сервисе. Доменное имя также рекомендуется не изменять.

Теперь необходимо настроить пароль доступа. По умолчанию имя пользователя – admin, в целом его также можно изменить, если Вам это необходимо. Пароль же в свою очередь Вы создаете сами, или же позволяете VK Cloud сгенерировать его за Вас. В данном же руководстве, использовался пароль - Root1234! (Рис. 11).



Пароль доступа

Имя пользователя

admin

Пароль пользователя

Root1234!

Сгенерировать

Рис. 11. Пароль доступа

Также справа показана стоимость необходимой нам услуги за месяц, но деньги с нашего тестового баланса будут тратиться поминутно, непосредственно при работе с самим сервисом.

После заполнения всех полей, необходимо нажать на «Следующий шаг», после чего откроется новое окно, предлагающее настроить виртуальную машину, а именно сеть и ключ виртуальной машины. Здесь оставляем все по умолчанию и нажимаем «Создать инстанс» (Рис. 12).

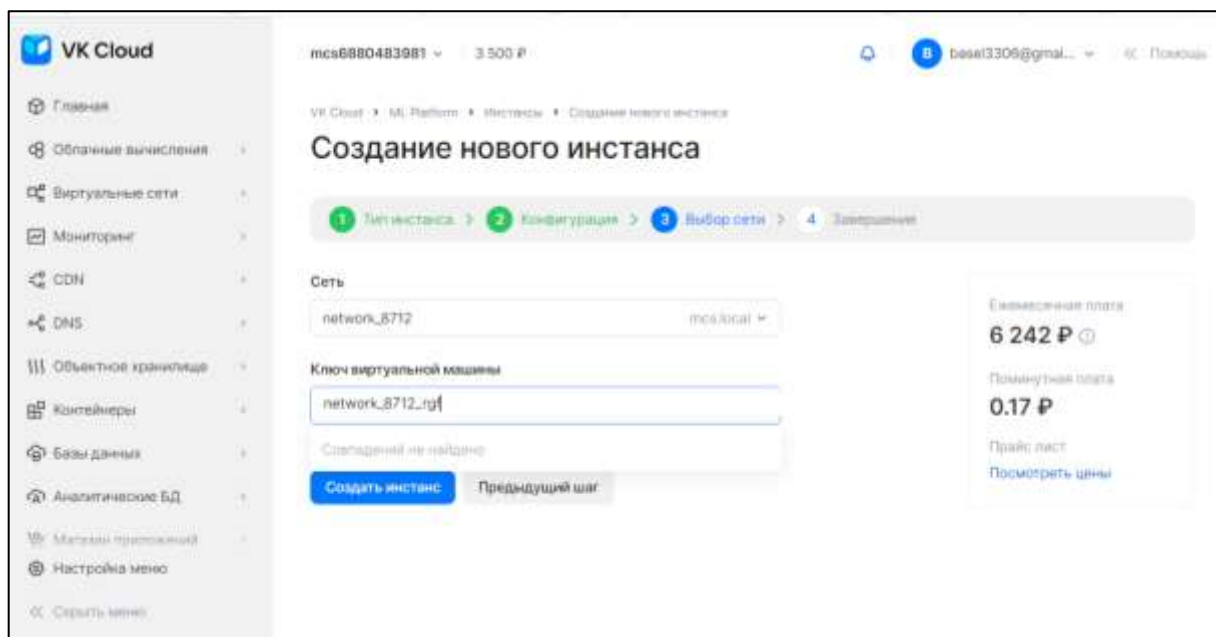


Рис. 12. Настройка ключа виртуальной машины

После чего придется немного подождать (около 5 минут), пока сервис VK Cloud создаст необходимый инстанс для работы с «ML Platform».

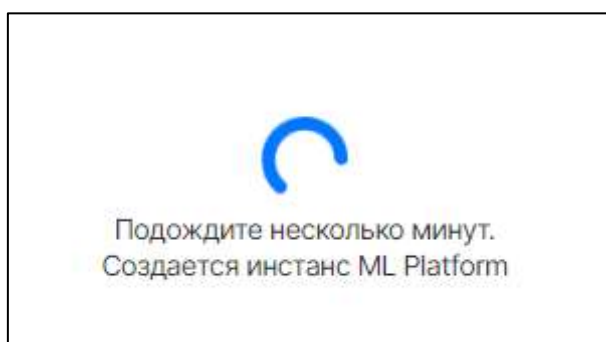
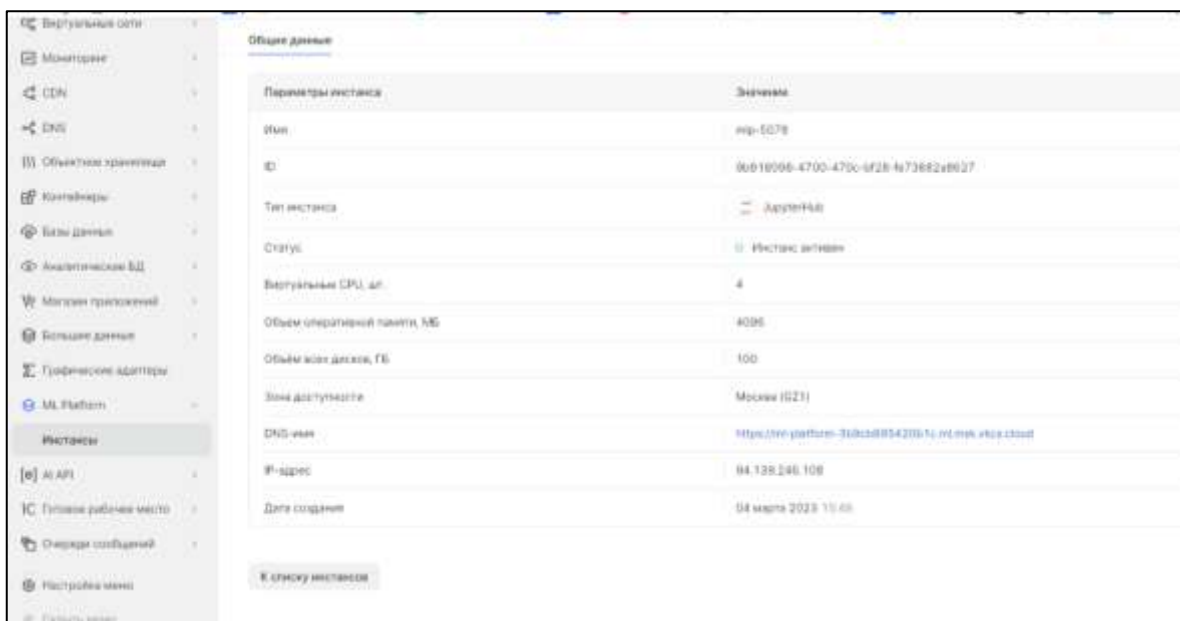


Рис. 13. Создание инстанса

После успешного создания инстанса Вы увидите соответствующее сообщение и страницу с его данными (Рис. 14).



Параметры инстанса	Значение
Имя	mlp-5078
ID	96b18066-4700-4700-af28-4e73682a8627
Тип инстанса	JupyterHub
Статус	Инстанс активен
Виртуальные CPU, шт.	4
Объем оперативной памяти, МБ	4096
Объем жесткого диска, Гб	100
Зона доступности	Москва (SO1)
DNS-имя	https://ml-platform-318ac5d854205-1j-ml-npk.vkcs.cloud
IP-адрес	84.138.246.108
Дата создания	04 марта 2023 15:48

К списку инстансов

Рис. 14. Общие данные о созданном инстансе

ML Platform – JupyterHub : конфигурация

Итак, следующая часть работы с ML Platform заключается в использовании JupyterHub непосредственно на созданном Вами инстансе, размещенном на облачном сервере VK Cloud. Для дальнейшей работы, Вам необходимо нажать на ссылку, расположенную на одном уровне с DNS-имя. После чего откроется окно авторизации jupyterhub, в котором Вам необходимо авторизоваться, используя имя пользователя и пароль, написанные при создании инстанса, поле с двухфакторной аутентификацией «2FA code» оставьте пустым. (Рис. 15).

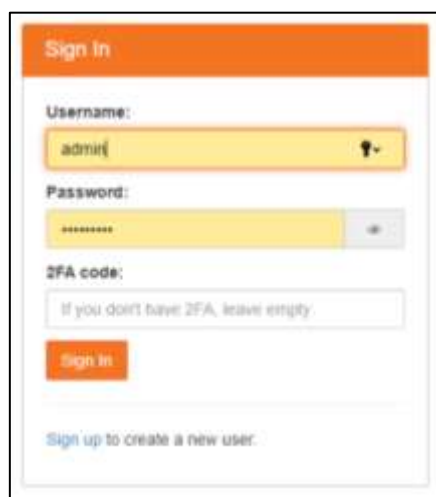


Рис. 15. Окно авторизации jupyterhub

После успешной авторизации, Вы получите доступ к JupyterHub на облачном сервере VK Cloud и увидите следующее окно (Рис. 16).

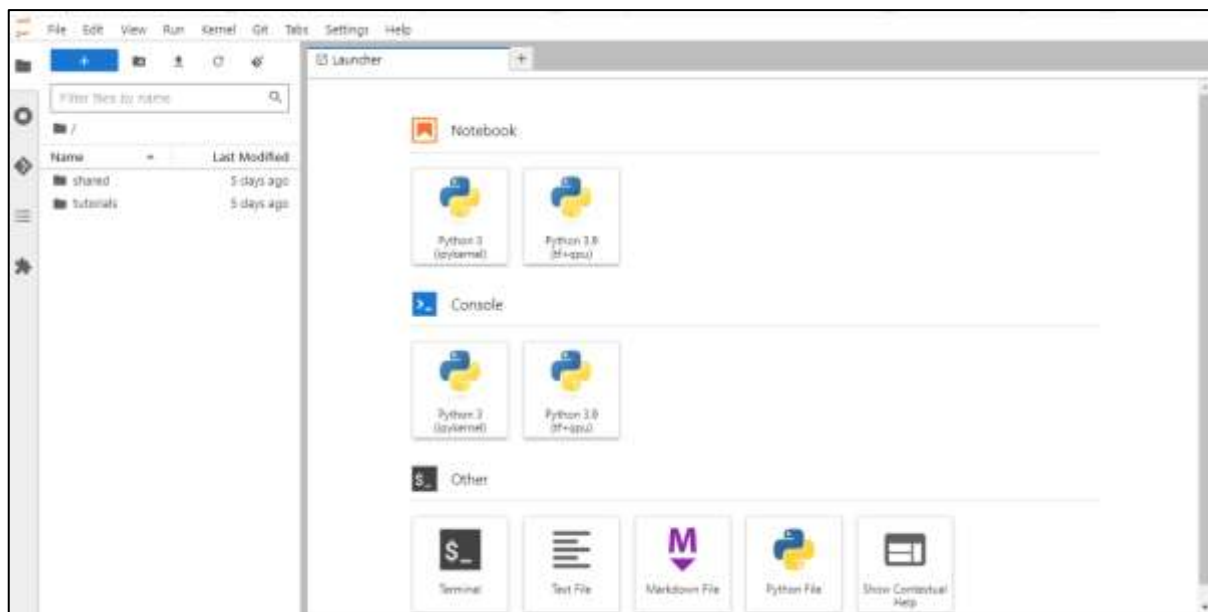


Рис. 16. Главное меню jupyterhub

В нем присутствуют различные обучающие примеры и инструкции по использованию. В левой части окна расположена панель – контент браузер или же местный проводник. В нем необходимо перейти в папку tutorials, а затем в vision.demo. После чего необходимо открыть единственный файл – Vision_Face_Recognition.ipynb (Рис. 17).

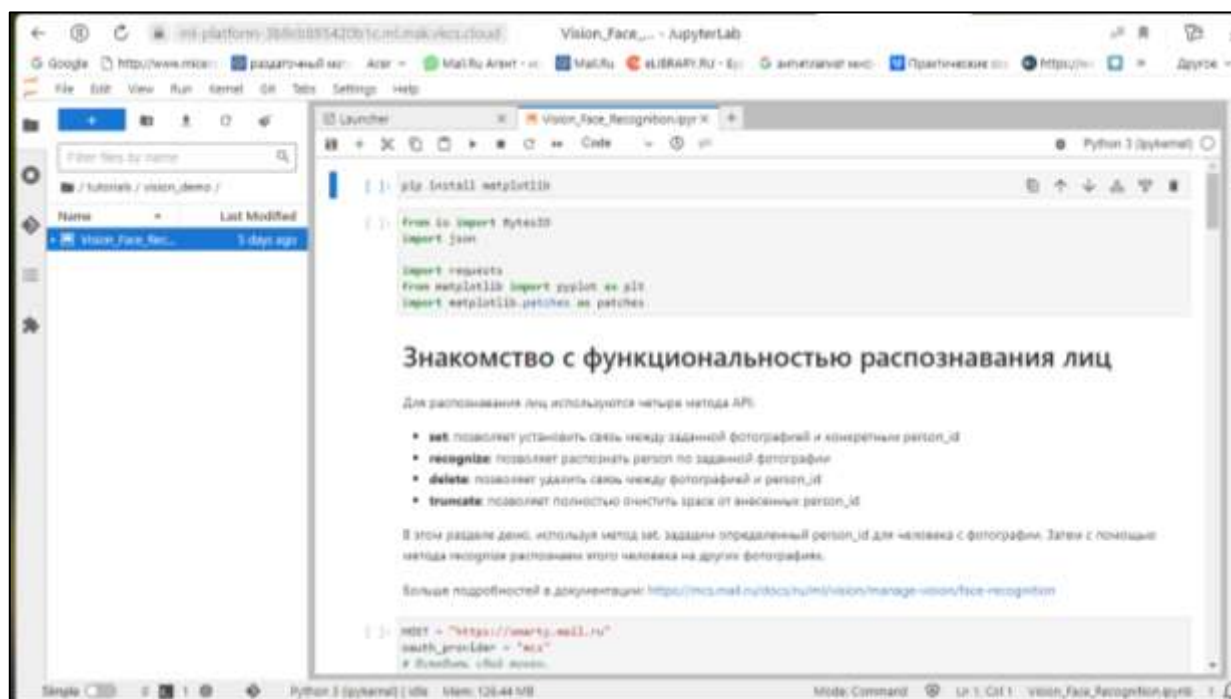


Рис. 17. Файл Vision_Face_Recognition.ipynb в jupyterhub

Данный файл является демонстрационным примером, хорошо показывающим возможности как облачного сервиса ML Platform, так и другого облачного решения в виде AI API – Vision API.

AI API – Vision API

Vision API - предоставляет методы и инструменты распознавания объектов, лиц, видео и других предметов.

Для дальнейшей работы с данным API внутри облачного сервиса ML Platform, необходимо получить сервисный токен. Для этого достаточно вернуться в главное меню VK Cloud и в панели слева найти раздел AI API, нажать на него и выбрать в нем Vision API или же посмотреть официальную инструкцию по получению токена <https://mcs.mail.ru/docs/ru/ml/vision/vision-start/auth-vision>.

По итогу Вы попадете на страницу Vision API (Рис. 18).

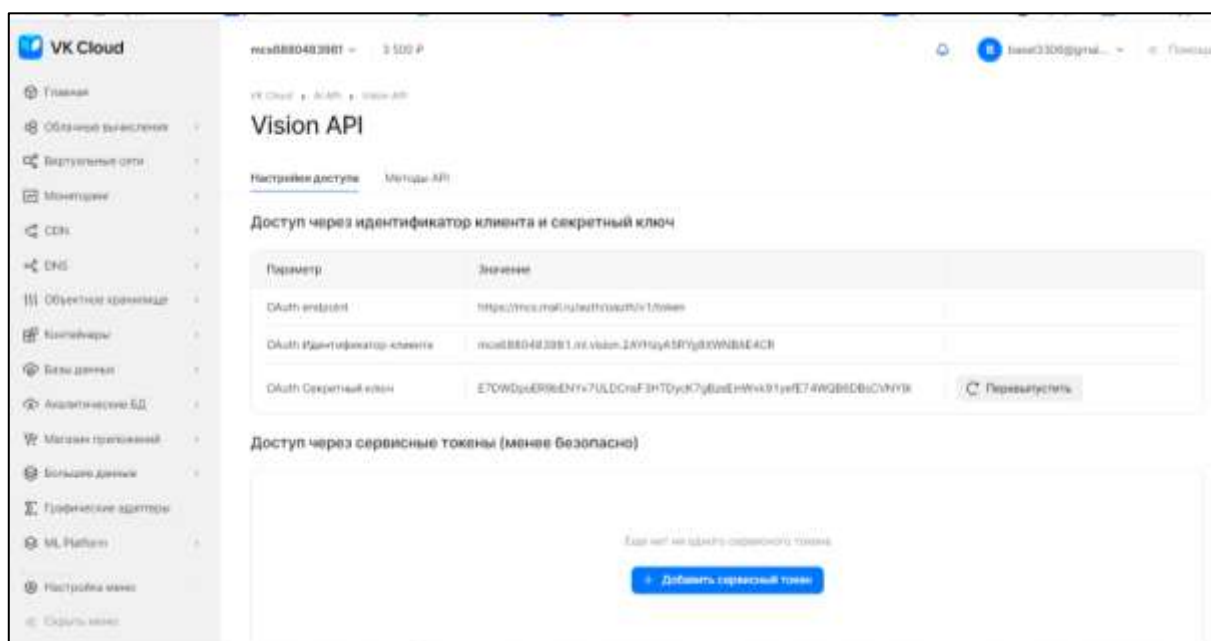


Рис. 18. Окно Vision API

Здесь, достаточно просто нажать на «Добавить сервисный токен», после чего в появившемся меню, проверить наличие всех галочек и нажать «Создать»

После чего в ранее пустом списке появится Ваш токен для работы с еще одним облачным сервисом AI API – Vision API (Рис 19).



Рис. 19. Созданный токен

ML Platform – JupyterHub : работа

Теперь, имея все необходимое, возвращаемся в JupyterHub в наш демонстрационный файл.

Итак, для начала запускаем первые две секции(ячейки), для того, чтобы инициализировать и установить необходимые библиотеки для работы в облачном jupyter'е. Все действия по запуску выполняются путем нажатия на кнопку запуска на панели инструментов сверху.

```
pip install matplotlib
from io import BytesIO
import json

import requests
from matplotlib import pyplot as plt
import matplotlib.patches as patches
```

В секции 3, Вам необходимо ввести недавно полученный токен в соответствующую строчку, как показано в примере:

```
HOST = "https://smarty.mail.ru"
oauth_provider = "mcs"
# вставить свой токен.
# Инструкция для получения токена:
https://mcs.mail.ru/docs/ru/ml/vision/vision-start/auth-vision
```

```

oauth_token = "ВСТАВЬТЕ ВАШ ТОКЕН"

# параметры запроса для авторизации при обращении к API
authorization_params = {
    "oauth_provider": oauth_provider,
    "oauth_token": oauth_token
}
NAMESPACE_FOR_DEMO = "7"

```

Для распознавания лиц используются четыре метода API:

- **set:** позволяет установить связь между заданной фотографией и конкретным person_id
- **recognize:** позволяет распознать person по заданной фотографии
- **delete:** позволяет удалить связь между фотографией и person_id
- **truncate:** позволяет полностью очистить space от внесенных person_id

В этом разделе демо, используя метод set, зададим определенный person_id для человека с фотографии. Затем с помощью метода recognize распознаем этого человека на других фотографиях.

```

def plot_image(image, figsize=(15, 15)):
    """Функция для отрисовки изображений внутри ноутбука"""
    plt.figure(figsize=figsize)
    plt.imshow(image, interpolation="lanczos")
    plt.axis('off')

# Загрузка изображения, которое будет использовано для присвоения person_id
url = 'https://fr_demo_vision_vkcloud.hb.bizmrg.com/face_recognition/01_Set_Marshall.png'
r = requests.get(url, allow_redirects=True)

```

```

if r.status_code == 200:
    image_set_raw = r.content
    image_set = plt.imread(BytesIO(image_set_raw))

```

Далее, необходимо вызвать функцию для отрисовки тестовых изображений и загрузить само тестовое изображение.

Создание связи

Следующим шагом, необходимо установить связь между загруженной фотографией и конкретным `person_id`. Для этого создадим словарь с файлами для параметра `files` в `post` запросе и мета данные в `JSON` формате.

```

def request_set(URL: str, img: bytes, person_id: int, space: str="0"):
    """Функция-обёртка для обращения к методу set"""
    # создать словарь с файлами для параметра files в post запросе
    files = {}
    name = "file_0"
    files[name] = img

    # создать meta: Параметры запроса передаются в формате JSON в теле запроса
    с name="meta"
    meta = {
        "space": space,
        "images": [{"name": name, "person_id": person_id}]
    }
    data = {'meta': json.dumps(meta)}

    # отправка запроса
    response = requests.post(URL, params=authorization_params, data=data,
files=files)
    return response

```

Затем загружаем изображение, которое будет использовано для присвоения `person_id`, в результате чего Вы увидите тестовое изображение (Рис. 20).

```
# Загрузка изображения
url =
'https://fr_demo_vision_vkcloud.hb.bizmrg.com/face_recognition/01_Set_Marshall.
png'
r = requests.get(url, allow_redirects=True)

if r.status_code == 200:
    image_set_raw = r.content
    image_set = plt.imread(BytesIO(image_set_raw))
    plot_image(image_set, figsize=(6, 6))
else:
    print(f"Ошибка при получении файла. Код {r.status_code}")
```



Рис. 20. Тестовое изображение с человеком, которого необходимо будет распознать

Поставим в соответствие человеку с фотографии person_id=1 в неймспейсе 7.

```
response = request_set(URL_SET, img=image_set_raw, person_id=1,
space=NAMESPACE_FOR_DEMO)
response.json()
```

Распознавание заданного человека

Создадим аналогичные функции, но немного модифицируем метаданные для изображений, после чего добавим ряд новых функций для парсинга людей на изображении, после чего загрузим новое изображений, состоящее из нескольких людей и включающее нашего изначального человека, которого необходимо распознать (Рис. 21).

```
url =
"https://fr_demo_vision_vkcloud.hb.bizmrg.com/face_recognition/02_recognize_him
ym_all_1.png"
image_format = url.split(".")[1]
r = requests.get(url, allow_redirects=True)

if r.status_code == 200:
    image_recognize_raw = r.content
    image_recognize = plt.imread(BytesIO(image_recognize_raw),
format=image_format)
    plot_image(image_recognize, figsize=(7, 7))
else:
    print(f"Ошибка при получении файла. Код {r.status_code}")
```




Рис. 21. Тестовое изображение с несколькими людьми

После чего, задействовав функции Vision API, произведем распознавание, заданного ранее человека. В результате чего, можно увидеть, что нейронной сети удалось распознать нужного нам человека (Рис. 22).

```
response = request_recognize(URL_RECOGNIZE, img=image_recognize_raw,
create_new=False, space=NAMESPACE_FOR_DEMO)
response.json()
plot_image_with_bb(image_recognize, parse_persons_response(response))
```

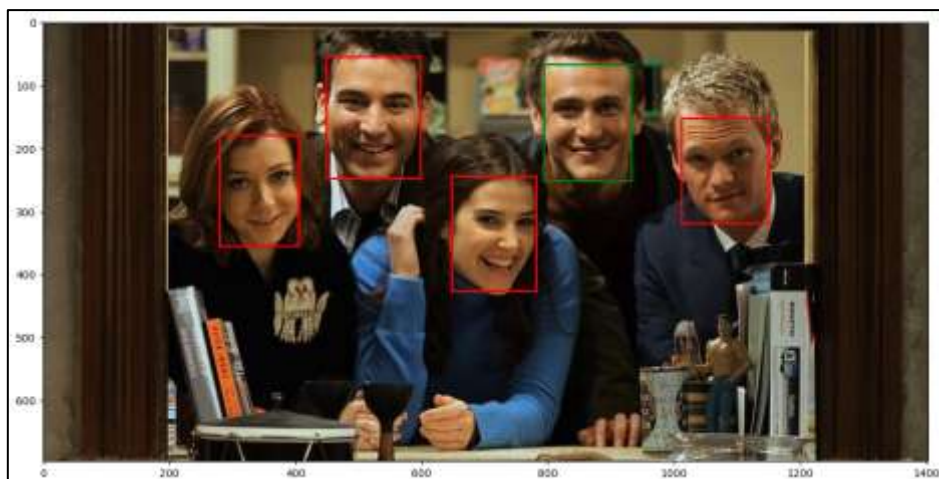


Рис. 22. Успешное распознавание заданного человека

Попробуем более сложную ситуацию, с большим количеством людей (Рис. 24).

```

url
"https://fr_demo_vision_vkcloud.hb.bizmrg.com/face_recognition/02_recognize_himy
m_all_2.jpeg"
image_format = url.split(".")[1]
r = requests.get(url, allow_redirects=True)

if r.status_code == 200:
    image_recognize_raw = r.content
    image_recognize = plt.imread(BytesIO(image_recognize_raw),
format=image_format)
    plot_image(image_recognize, figsize=(7, 7))
else:
    print(f"Ошибка при получении файла. Код {r.status_code}")
response = request_recognize(URL_RECOGNIZE, img=image_recognize_raw,
create_new=False, space=NAMESPACE_FOR_DEMO)
response.json()
plot_image_with_bb(image_recognize, parse_persons_response(response))

```



Рис. 23. Более сложное тестовое изображение с множеством людей

И, как видно из результатов, даже в таком случае нейронная сеть, запущенная в облачном сервисе VK Cloud, отлично справилась со своей задачей (Рис. 24).



Рис. 24. Успешное распознавание заданного человека на более сложном тестовом изображении

Пример работы системы распознавания с произвольными фотографиями

Теперь попробуем сделать то же самое, только уже со своими изображениями, для этого нам необходимо найти одну фотографию человека, которого необходимо распознать (Важно! Человек должен быть единственным на фото) и несколько других фотографий, где присутствует необходимый нам человек, но уже в компании людей.

Для этого, нам необходимо загрузить тестовые изображения (расширение не имеет значения). Находясь в той же папке по пути /tutorials/vision_demo/ нам необходимо перенести файлы с компьютера в контент браузер (Рис. 25).




Name	Last Modified
 1_me_2.JPG	6 minutes ago
 3_birthday.JPG	6 minutes ago
 Vision_Face_Rec...	in a few seconds

Рис. 25. Пример расположения файлов

После успешного добавления фото, необходимо написать их путь в коде, в следующем формате: `image_path_set = "./1_me_2.JPG"`, после чего отобразить загруженное изображение и присвоить ему `person_id` (Рис. 27).



Рис. 26. Тестовое изображение

После успешной загрузки фотографии с человеком, которого необходимо распознать, произведем практически аналогичные действия, но уже с фотографией, на которой необходимо найти нашего человека. В результате чего, увидим, что нейронная сеть отлично справилась с поставленной задачей (Рис. 27).

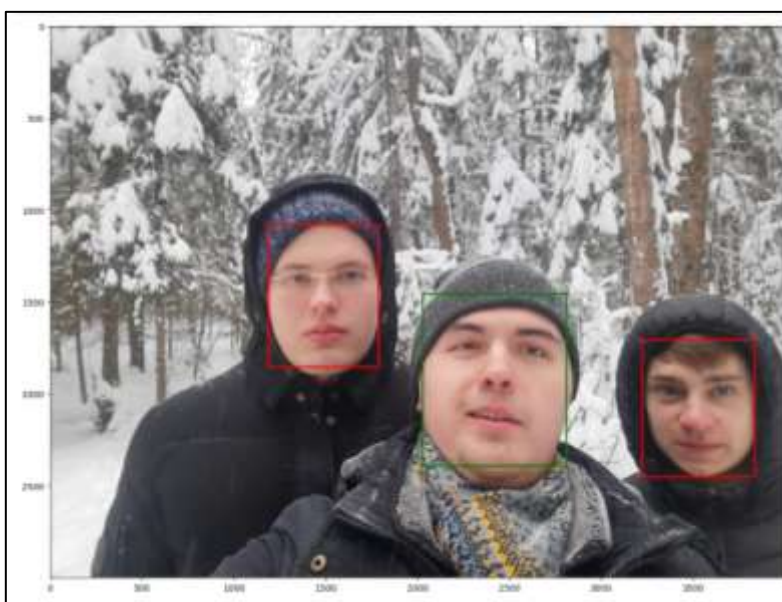


Рис. 27. Успешное распознавание заданного человека

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Теперь, попробуйте сделать то же самое, только уже со своими изображениями, для этого Вам необходимо найти одну фотографию человека, которого необходимо распознать (Важно! Человек должен быть единственным на фото) и несколько других фотографий, где присутствует необходимый Вам человек, но уже в компании людей.

ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ

Использовать сторонние классы и компоненты, реализующие функционал задания запрещено. По завершении готовится отчет.

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Что представляет собой MLOps Platform?
2. Что включают в себя MLOps задачи?
3. Опишите порядок регистрации в VK Cloud.
4. Охарактеризуйте сходство и различие сервисов MLflow и Cloud ML Platform.
5. Опишите этапы работы в ML Platform.
6. Опишите процесс работы с ML Platform – JupyterHub.
7. Перечислите методы API для распознавания лиц.

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 2 занятия (4 академических часа: 3 часа на выполнение и сдачу лабораторной работы и 1 час на подготовку отчета).

Номер варианта студенту выдается преподавателем.

Отчет на защиту предоставляется в печатном виде.

Структура отчета (на отдельном листе(-ах)): титульный лист, формулировка задания (вариант), листинг, результаты выполнения работы (графические изображения примеров работы приложения), выводы).

ОСНОВНАЯ ЛИТЕРАТУРА

1. Зиангирова, Л. Ф. Технологии облачных вычислений [Электронный ресурс]: учебное пособие / Л. Ф. Зиангирова. — Электрон, текстовые данные.— Саратов : Вузовское образование, 2016. — 300 с. — Режим доступа <http://www.iprbookshop.ru/41948.html>
2. Клементьев, И. П. Введение в облачные вычисления [Электронный ресурс]/ И. П. Клементьев, В. А. Устинов. — Электрон, текстовые данные. — Москва: Интернет- Университет Информационных Технологий (ИНТУИТ), 2016. — 298 с. — Режим доступа: <http://www.iprbookshop.ru/57372.html>
3. Савельев, А. О. Введение в облачные решения Microsoft [Электронный ресурс]/ А. О. Савельев. — Электрон, текстовые данные.— 2-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 230 с. — Режим доступа: <http://www.iprbookshop.ru/73665.html>
4. Рак, И.П. Технологии облачных вычислений [Электронный ресурс]: учебное пособие / И.П. Рак, А.В. Платёнкин, Э.В. Сысоев ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего образования «Тамбовский государственный технический университет». - Тамбов Издательство ФГБОУ ВПО «ТГТУ», 2017.— 82 с. : ил. -URL: <http://biblioclub.ru/index.php?page=book&id=499410>
5. Тропченко, А.А. Методы вторичной обработки и распознавания изображений [Электронный ресурс]: учебное пособие / А.А. Тропченко, А.Ю., Тропченко. — СПб.: Университет ИТМО, 2015. — 215 с. — Режим доступа: <http://www.iprbookshop.ru/67277.html>
6. Шапиро, Л. Компьютерное зрение [Электронный ресурс]: учебное пособие / Л. Шапиро, Д. Стокман. — М.: Издательство "Лаборатория знаний", 2015. — 763 с. — Режим доступа: <https://e.lanbook.com/book/84096>

ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

8. Костюк, А.И. Организация облачных и GRID-вычислений [Электронный ресурс]: учебное пособие / А.И. Костюк ; Министерство науки и высшего образования Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего образования «Южный федеральный университет», Инженерно-технологическая академия. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2018. - 122 с.: ил. -URL: <http://biblioclub.ru/index.php?page=book&id=561079>
9. Губарев, В. В. Введение в облачные вычисления и технологии [Электронный ресурс] : учебное пособие / В. В. Губарев, С. А. Савульчик, Н. А. Чистяков. — Электрон, текстовые данные.— Новосибирск : Новосибирский государственный технический университет, 2013.— 48 с. — Режим доступа: <http://vwww.iprbookshop.ru/44905.html>
10. Волкова, М.А. Методы обработки и распознавания изображений [Электронный ресурс]: учебно-методическое пособие по лабораторному практикуму / М.А. Волкова, В.Р. Луцив. — СПб.: Университет ИТМО, 2016. — 46 с. — Режим доступа: <http://www.iprbookshop.ru/67286.html>
11. Купельский, С. А. Использование облачных сервисов [Электронный ресурс]: учебно-методическое пособие / С. А. Купельский ; под редакцией Т. И. Алферова. — Электрон. текстовые данные.— Екатеринбург : Уральский .федеральный университет, ЭБС АСВ, 2016. – 136 с. — Режим доступа: — <http://www.iprbookshop.ru/69603.html>
12. Мирзоев, М.С. Основы математической обработки информации [Электронный ресурс]: учебное пособие / М.С. Мирзоев. — М.: Прометей, 2016. — 316 с. — 978-5-906879-01-1. — Режим доступа: <http://www.iprbookshop.ru/58165.html>