Приоритет 2030

Выездной семинар

“Модернизация образовательных программ по программной инженерии в условиях научно-технологического развития России”

SmartMLOps - Корпоративная информационная система поддержки моделей искусственного интеллекта НИУ ВШЭ

Приоритет 2030

ИИ-технологии для человека

SmartMLOps - Корпоративная информационная система поддержки моделей искусственного интеллекта НИУ ВШЭ

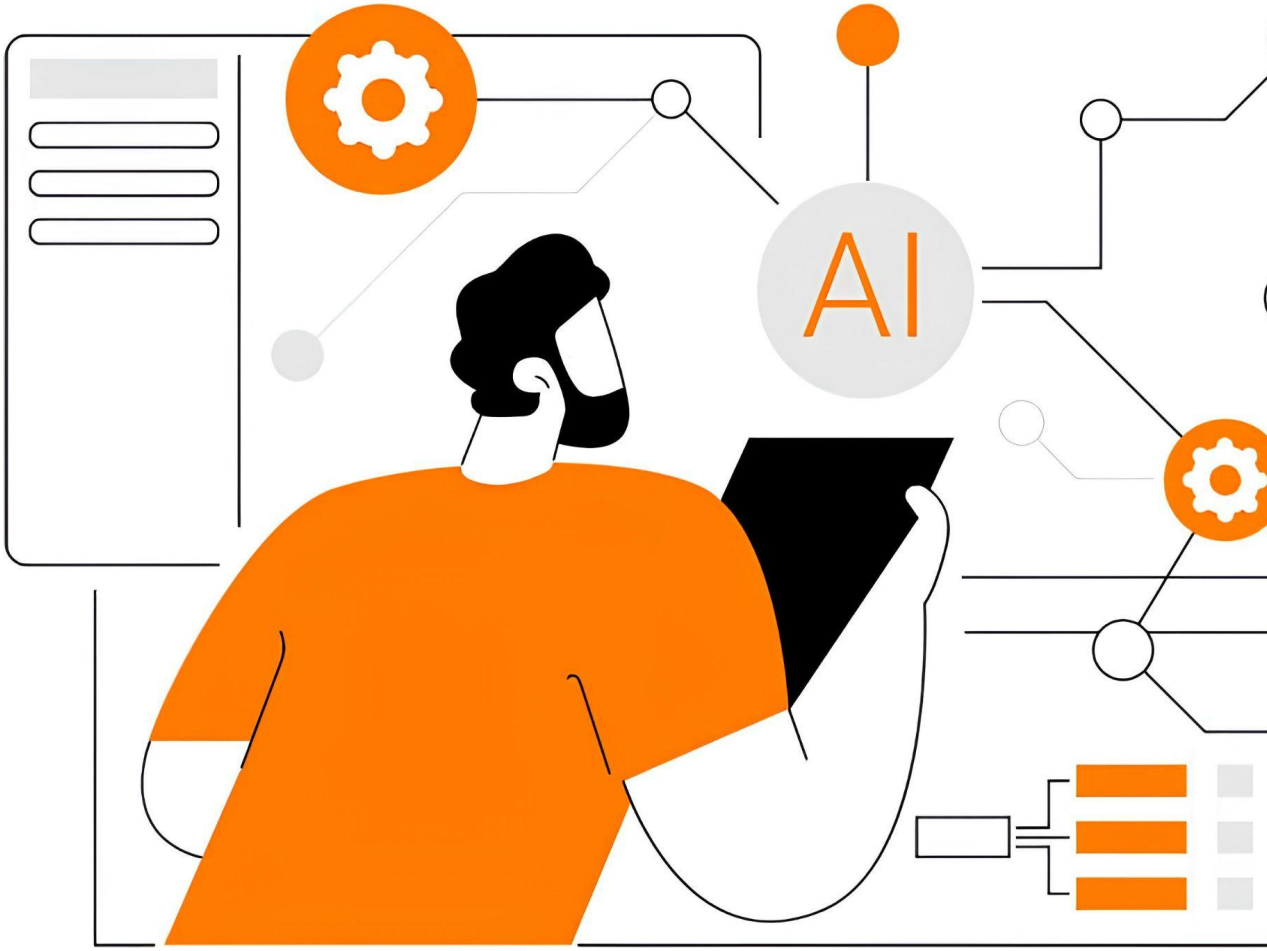
Салех Хади Мухаммед, Департамент программной инженерии
Факультета компьютерных наук НИУ ВШЭ г. Москва
hsalekh@hse.ru

Москва 2025

Целевая аудитория

MLOPS платформа нацелена на разработчиков ИИ и ML моделей желающих масштабировать свои интеллектуальные сервисы без необходимости решать инфраструктурные задачи и задачи безопасности.

- Лаборатории ВШЭ
- Проекты СП4,СП5
- Подразделения ВШЭ
- Студенты, аспиранты, ППС и сотрудники ВШЭ
- Внешние исследователи и партнеры

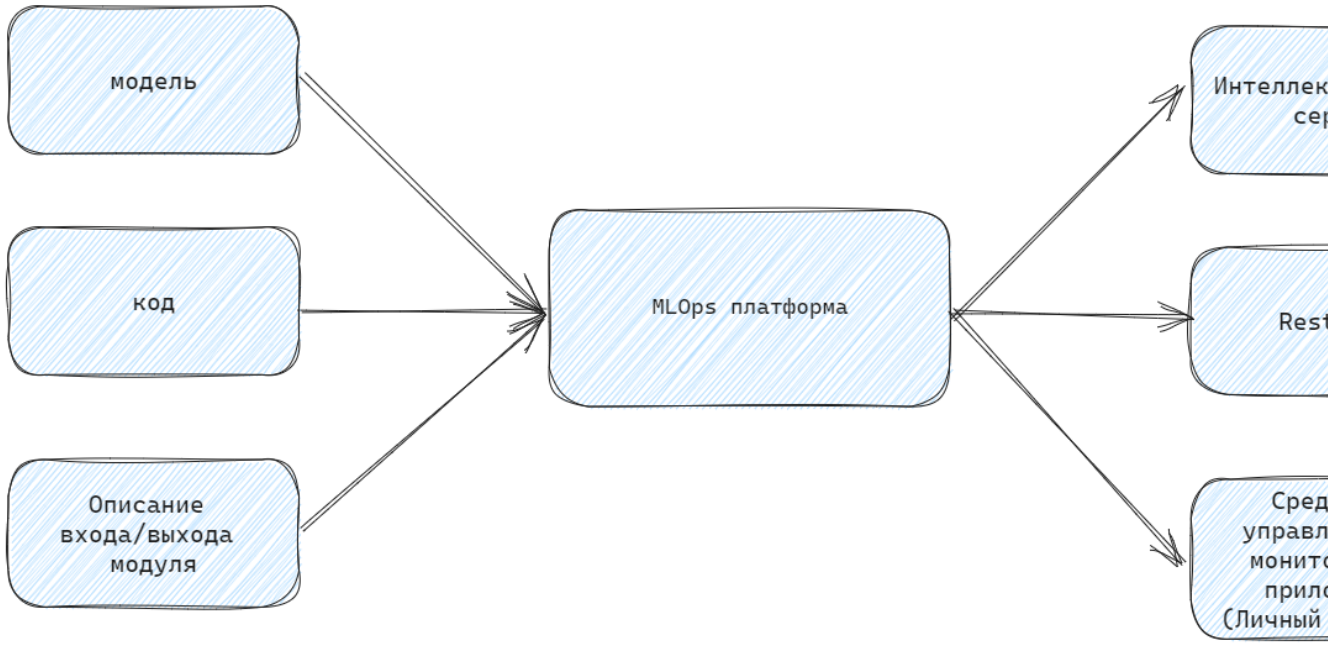


MLOps платформа

Что такое MLOps платформа:

- вычислительные ресурсы
- хранилища данных
- система безопасности
- платформа для пользователей
- Доступ к суперкомпьютеру cHARISMa
- Мониторинг ресурсов и работы моделей
- Управление доступом и лицензиями к модулям
- Консультации и сопровождение от команды техподдержки

для вашего интеллектуального приложения

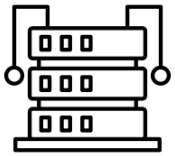


Что уже сделано 2024

Анализ конкурентных решений

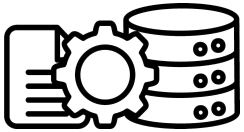
Развертывание фреймворка в облаке и настройка инфраструктуры

Развертывание ИИ моделей на платфор



Интеграция платформы с суперкомпьютером

Научные семинары



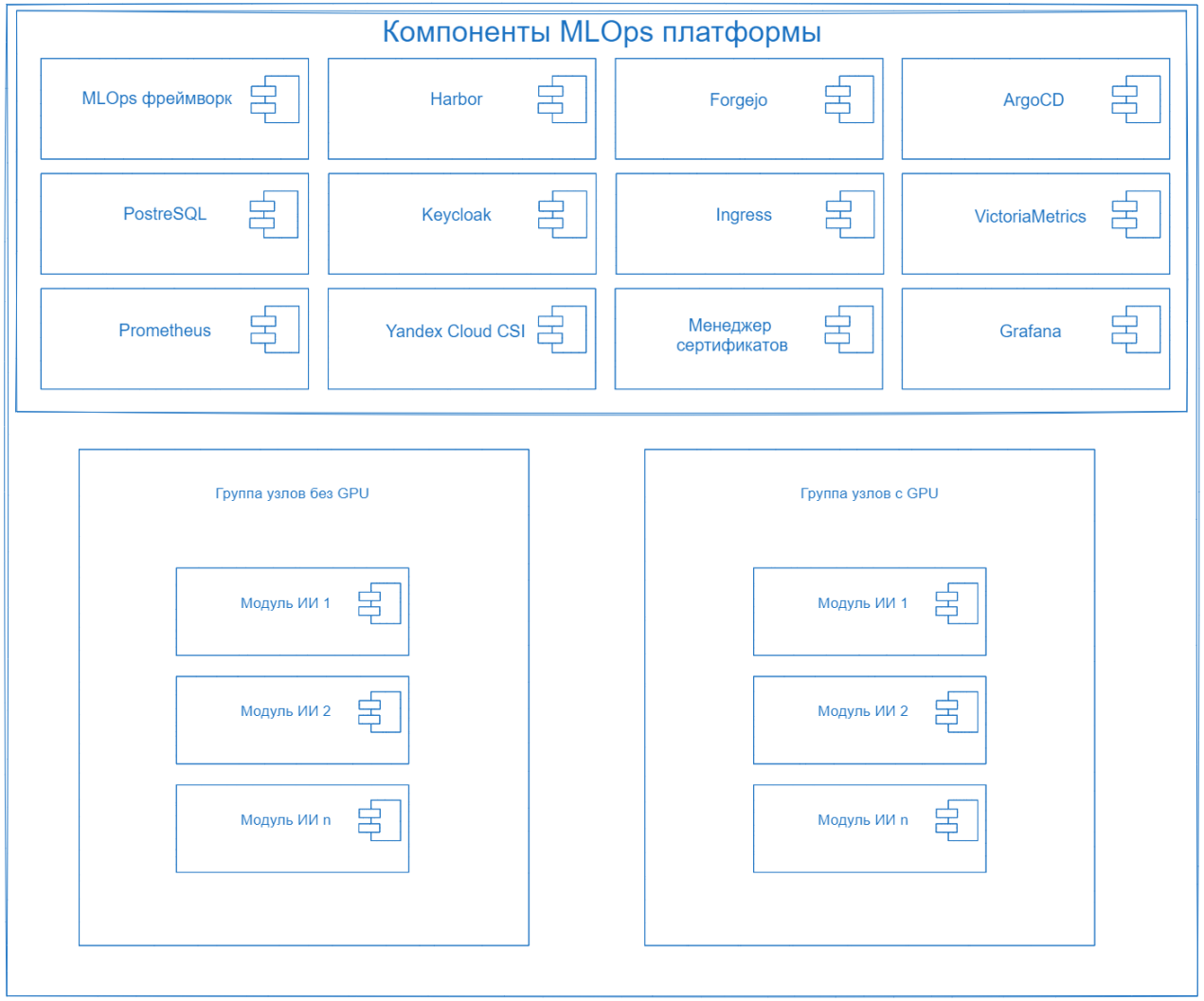
Интеграция КИС НИУ ВШЭ + ETL

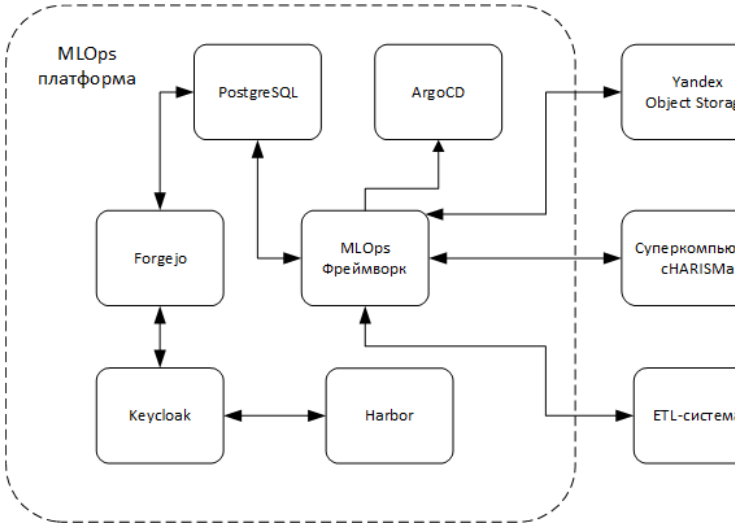
Реализация аналитических пан

РИД

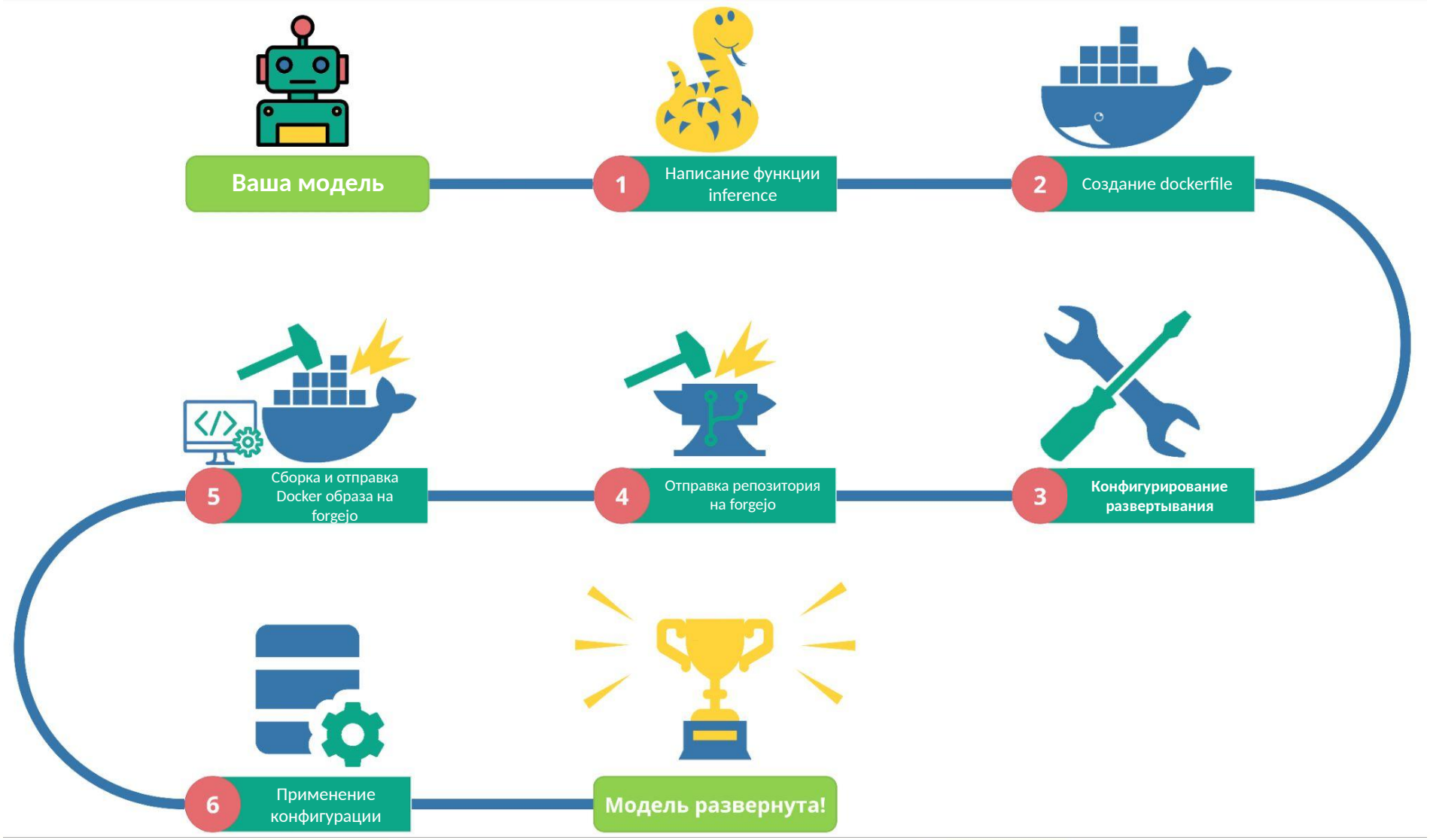
Безопасность

Инфраструктура платформы

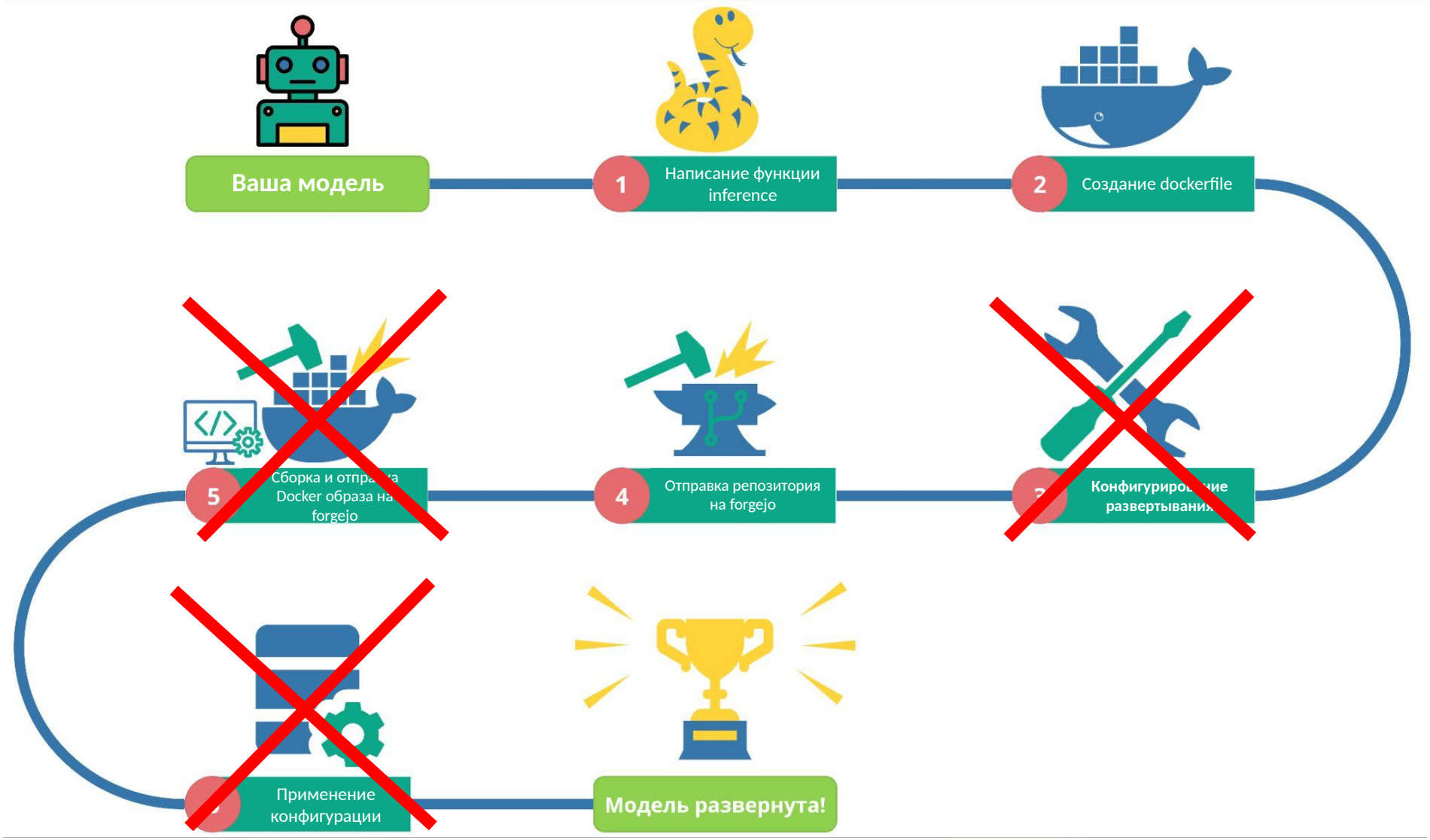




Развертывание моделей на MLOPS платформе в 2024



Развертывание моделей на MLOPS платформе в 2025

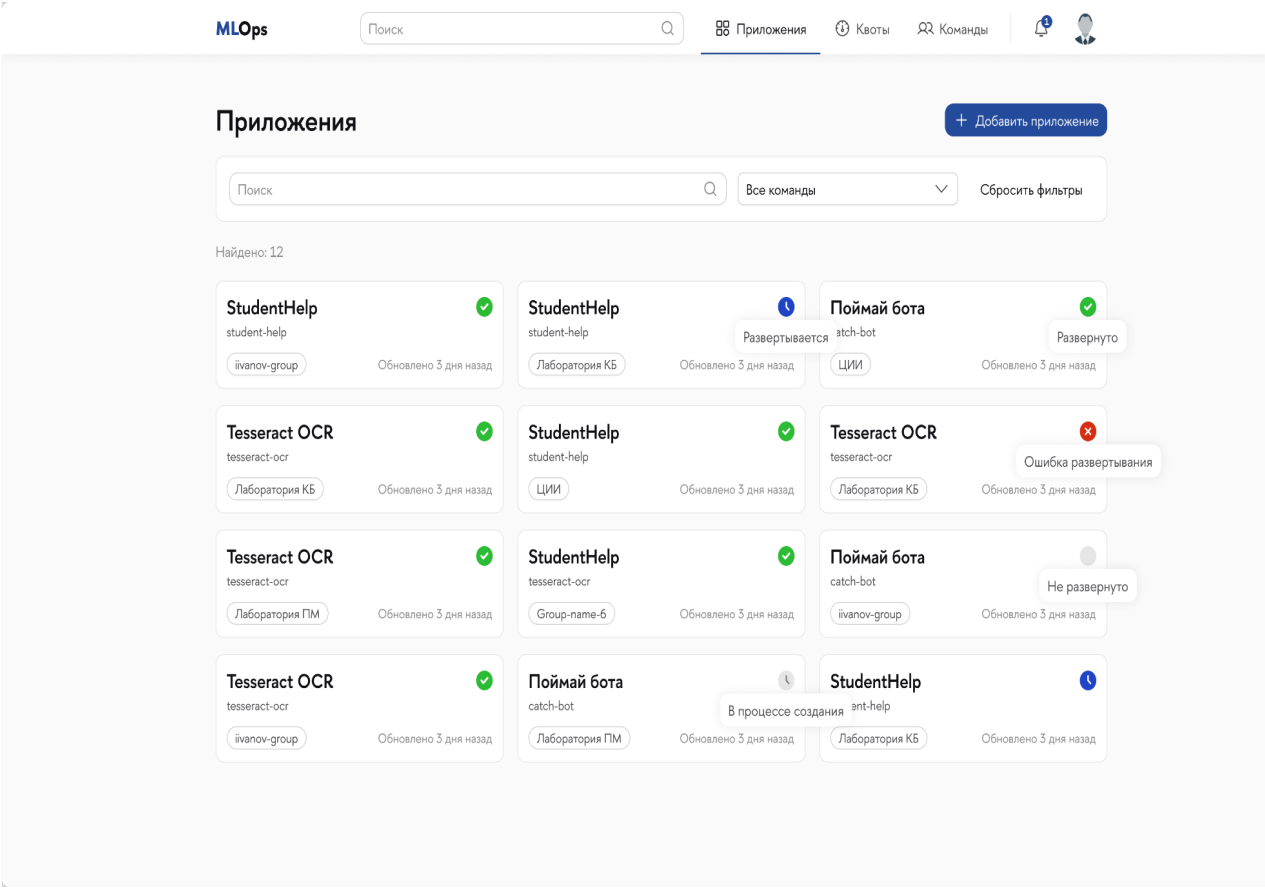


Личный кабинет разработчика

Функционал ЛК:

- Упрощение процесса создания и управления приложением:
  - Генерация и шаблонное заполнение манифестов
  - Логи развертывания
  - Управление компонентами
  - Статусы
- Мониторинг
  - Дашборды разработчика
- Разграничение доступа
  - OIDC

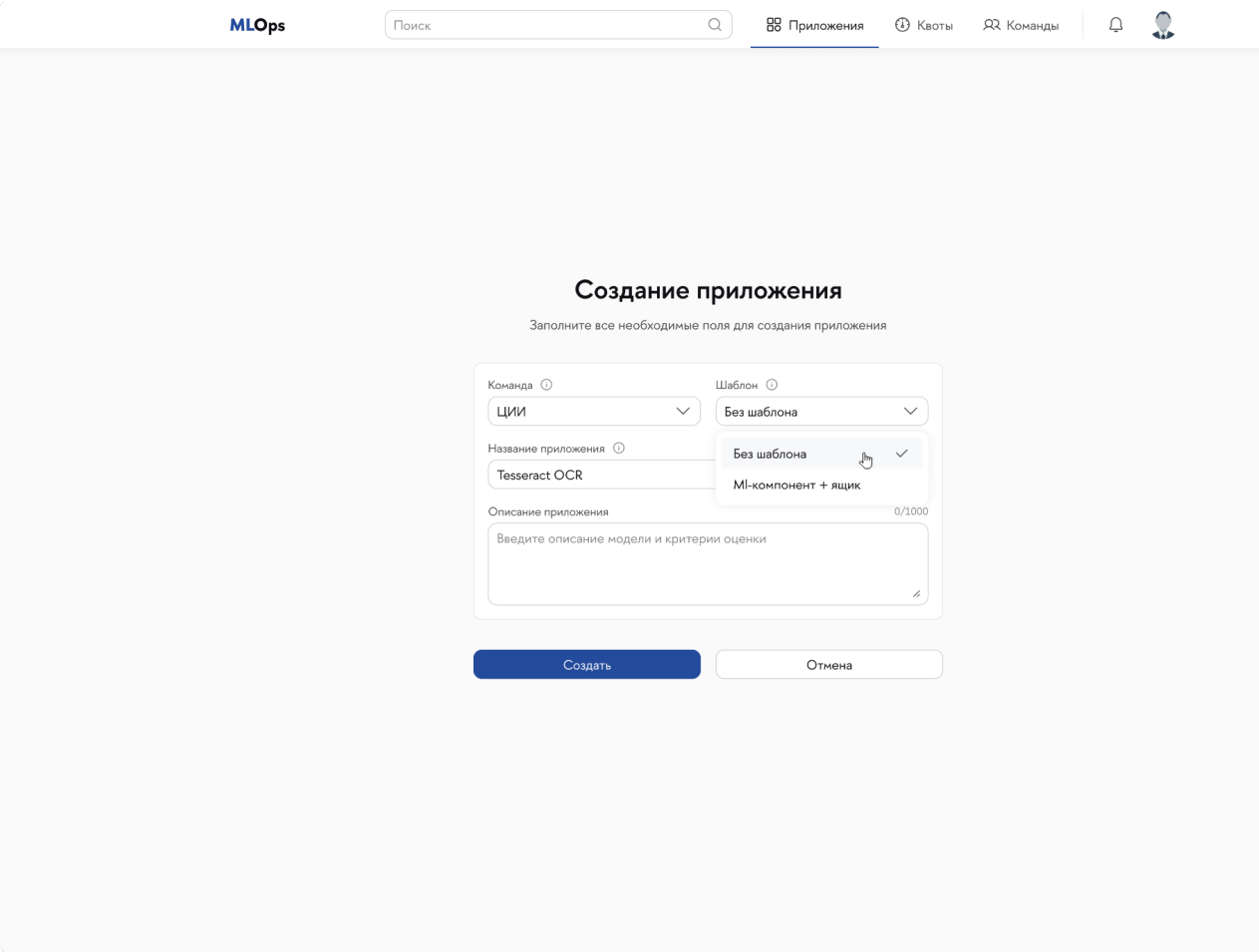
Работа над Личным кабинетом не окончена! Внешний вид интерфейса может быть изменен.



Создание приложения через Личный кабинет

Меню создания приложения:

- Имя приложения
- Выбор команды разработчиков поддерживающих приложение
- Выбор заранее созданного командой шаблона приложения
- Краткое описания для отображения в каталоге



Работа над Личным кабинетом не окончена! Внешний вид интерфейса может быть изменен.

Создание приложения через Личный кабинет

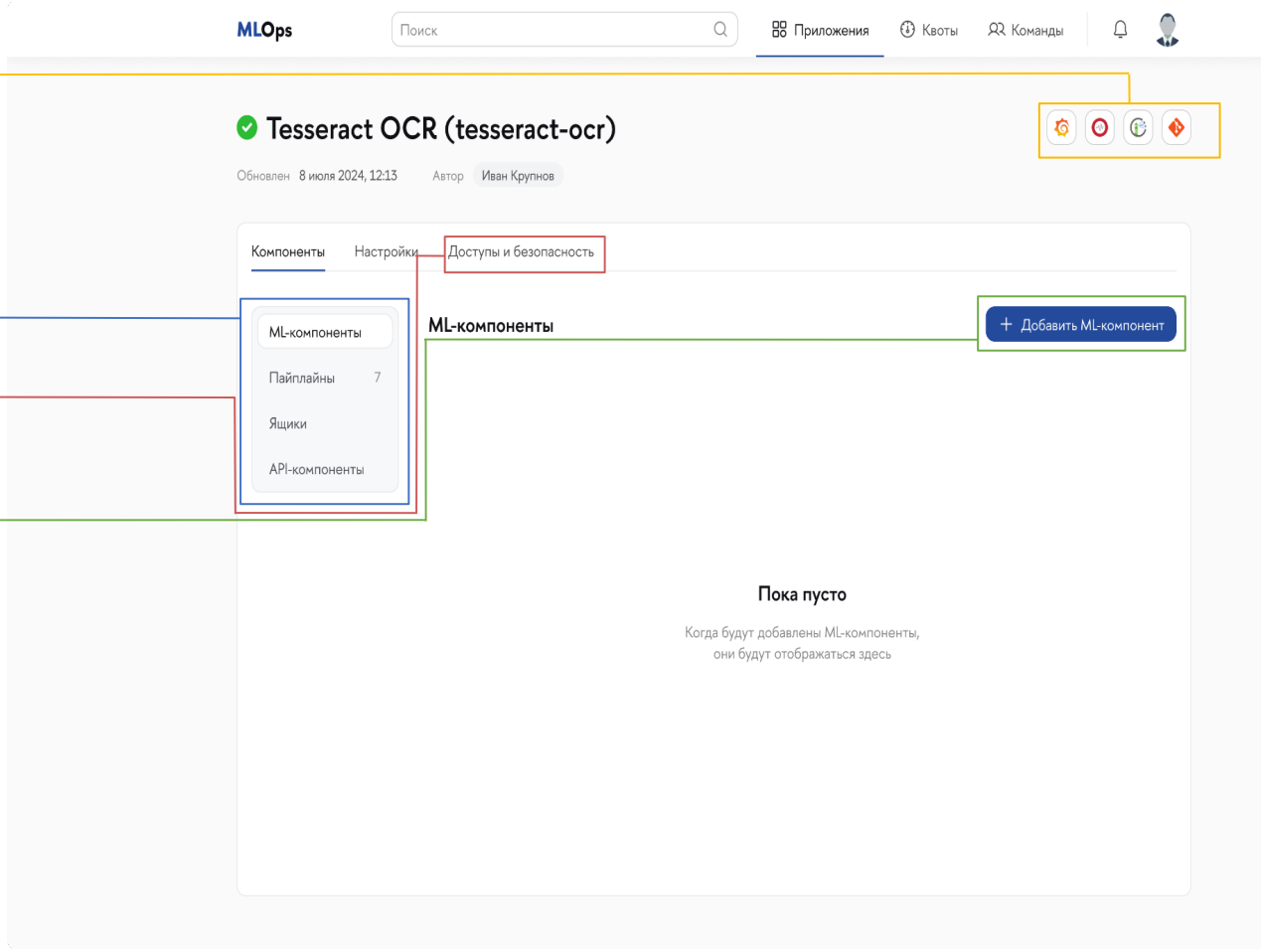
После создания разработчик получает:

- Организацию в Harbor
- Репозиторий приложения в Fogejo
- Мониторинг Grafana

Список компонент приложения

Контроль доступа к приложению

Возможность добавить новые компоненты



Работа над Личным кабинетом не окончена! Внешний вид интерфейса может быть изменен.

Появилась аутентификация на основе OIDC

Функция аутентификации и защиты доступа платформы предоставляет возможности:

- для разработчиков единая точка входа (Single Sign-On) во всех сервисах платформы;

- централизованное управление доступом разработчиков и конечных пользователей к сервисам платформы и сервисам размещенных на платформе приложений;

- разработчики сами могут управлять доступом к размещенным имм приложениям;

- подключение внешних служб идентичности и каталогов пользователей;

Появился Вэб портал MLOPS

Точка входа пользователя в систему MLOps:

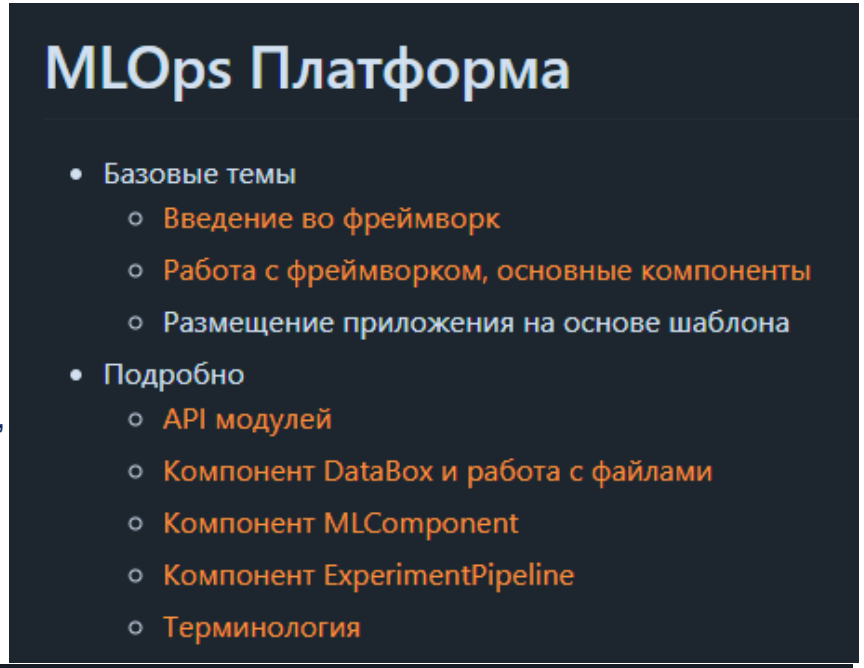
- Подача заявки для подключения
- Новости платформы
- Документация и гайды
- Обратная связь
- Тех. поддержка

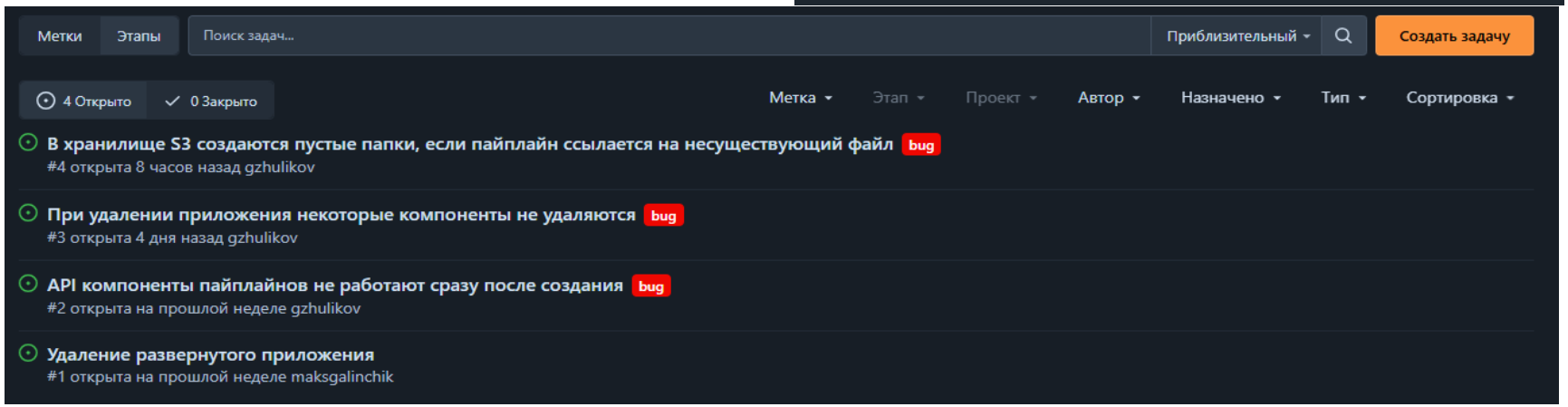
https://mlops.hse.ru/

Документация

Пользовательская документация:

- Размещена в общедоступном репозитории на forgejo
- Все еще в работе и постоянно дополняется
- Контент доступный на данный момент(На изображении справа):
- В том же репозитории можно сообщить о баге, предложить идею, задать вопрос о деталях работы платформы:





Репозиторий доступен
портала или по ссылк

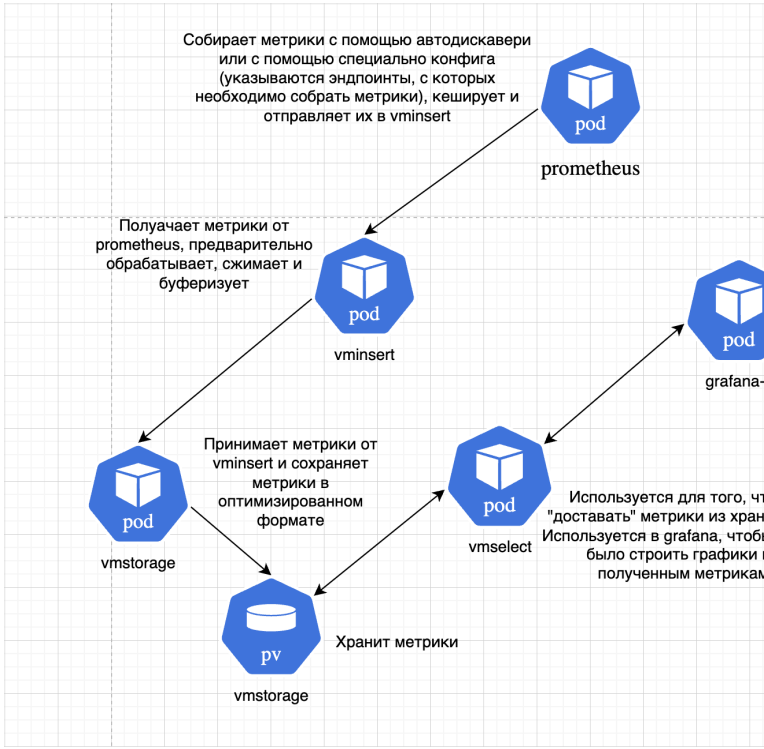
https://platform-forgejo.stratp
mlops_platform/documentatio

Система мониторинга платформы: сделано в 2024

Собраны базовые дашборды для администратора системы

Состав дашбордов:

- Состояние пространств имен Kubernetes
  - Просмотр состояния по нагрузке CPU и использованию памяти
- Состояние по нагрузке на Kube API server
- Мониторинг состояния Ingress контроллеров
- Мониторинг состояния СУБД
- Состояние системы бэкапов Velero
- Состояние системы хранения VictoriaMetrics
- Состояние выполнения задач на кластере cHARISMa



Система мониторинга платформы: план на 2025

Разработка персональных дашбордов для разработчиков

Цель: упрощение процесса мониторинга состояний приложений для разработчиков

Планируемый состав:

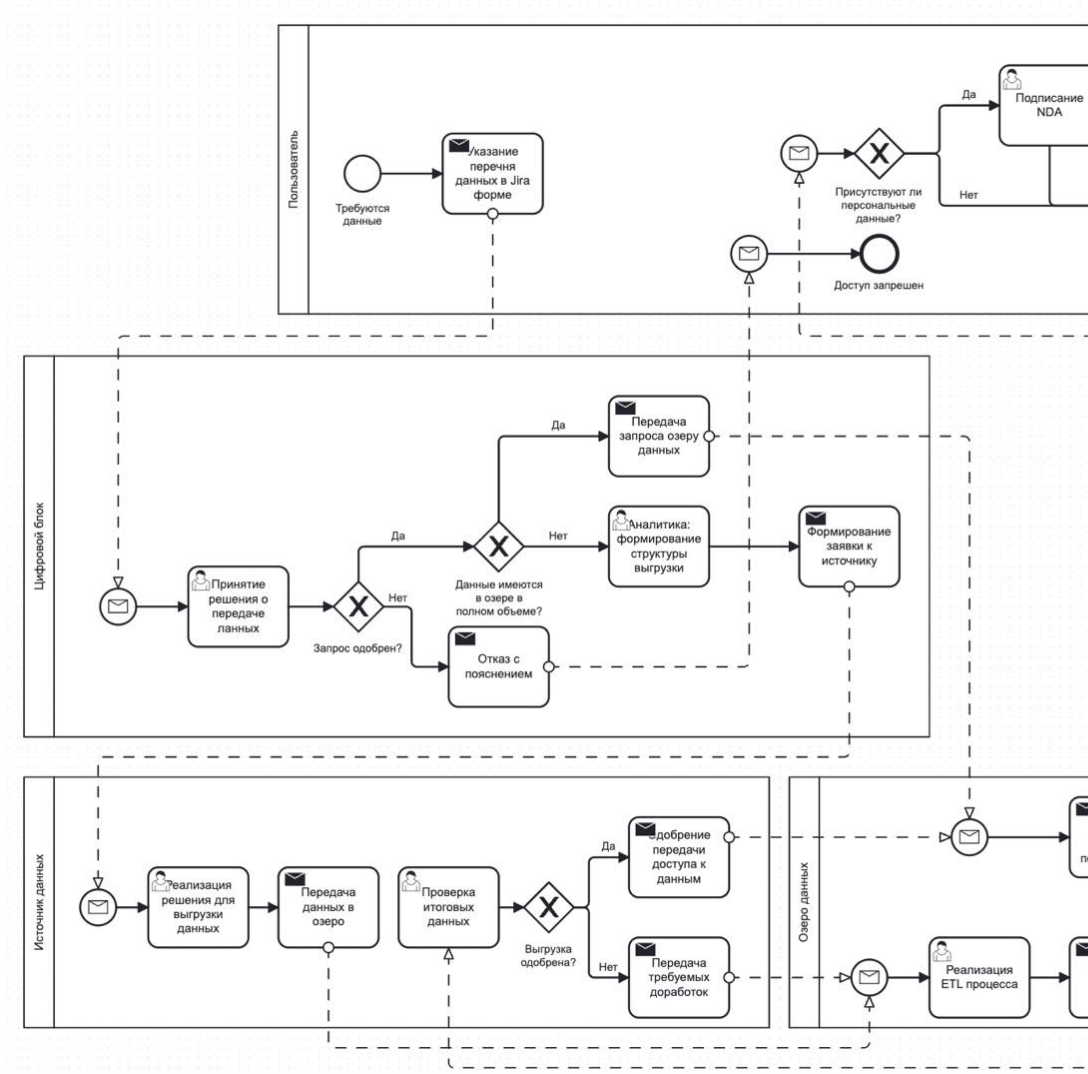
- Мониторинг ресурсов (vCPU, Ram, SSD) пространств имен приложение в Kubernetes
- Мониторинг журналов приложения
- Мониторинг журнала развертывания приложения
- Мониторинг сетевой нагрузки приложения
- Мониторинг журналов из Ingress связанные с приложением
- Формирование отчета синхронизации Argo CD для приложения
- Мониторинг состояния Harbor
- Мониторинг состояния Forgejo
- Мониторинг состояния S3

Расширение состава дашбордов для администратора системы

- Мониторинг состояния журналов событий всех простра всех пользователей
- Мониторинг журнала развертывания приложений
- Мониторинг состояния Harbor
- Мониторинг состояния Forgejo
- Мониторинг состояния S3
- Мониторинг текущих выставленных лимитов в приложе

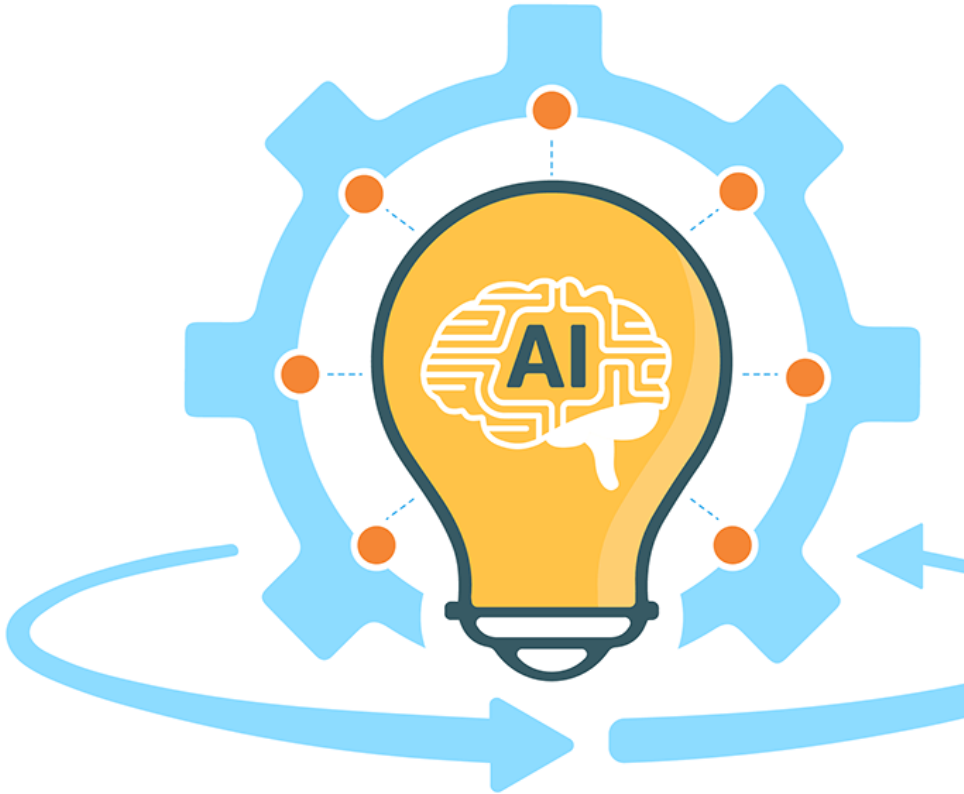
Озеро данных

- Необходимость централизованного хранения данных:
  - повторная реализация механизма выгрузки
  - Дублирование данных и неоднократные обращения к источникам
  - Сложности с единым управлением доступом к различным наборам данных
  - Необходимость обезличивания данных
- Появление Озера Данных даст:
  - Возможность переиспользования
  - Данные заранее очищены и структурированы
  - Сокращение трудоемкости процессов интеграции



План на 2025

- Личный кабинет разработчика ИИ моделей
- Конструктор пайплайнов в ЛК разработчика
- Мониторинг моделей для разработчика
- Мониторинг платформы для администраторов
- Биллинг
- Информационная безопасность
- Аутентификация пользователей
- Инструменты обезличивания данных КИС
- Опытная эксплуатация



Прогресс выполнения плана 2025

Личный кабинет разработчика ИИ моделей (Демо)

☐ Конструктор пайплайнов в ЛК разработчика

Мониторинг моделей для разработчика

Мониторинг платформы для администраторов

☐ Биллинг

☐ Информационная безопасность

Аутентификация пользователей (Демо)

☐ Инструменты обезличивания данных КИС

Опытная эксплуатация (Начало)

Спасибо за внимание!

Обратная связь:

Присоединяйтес
опытной эксплуа

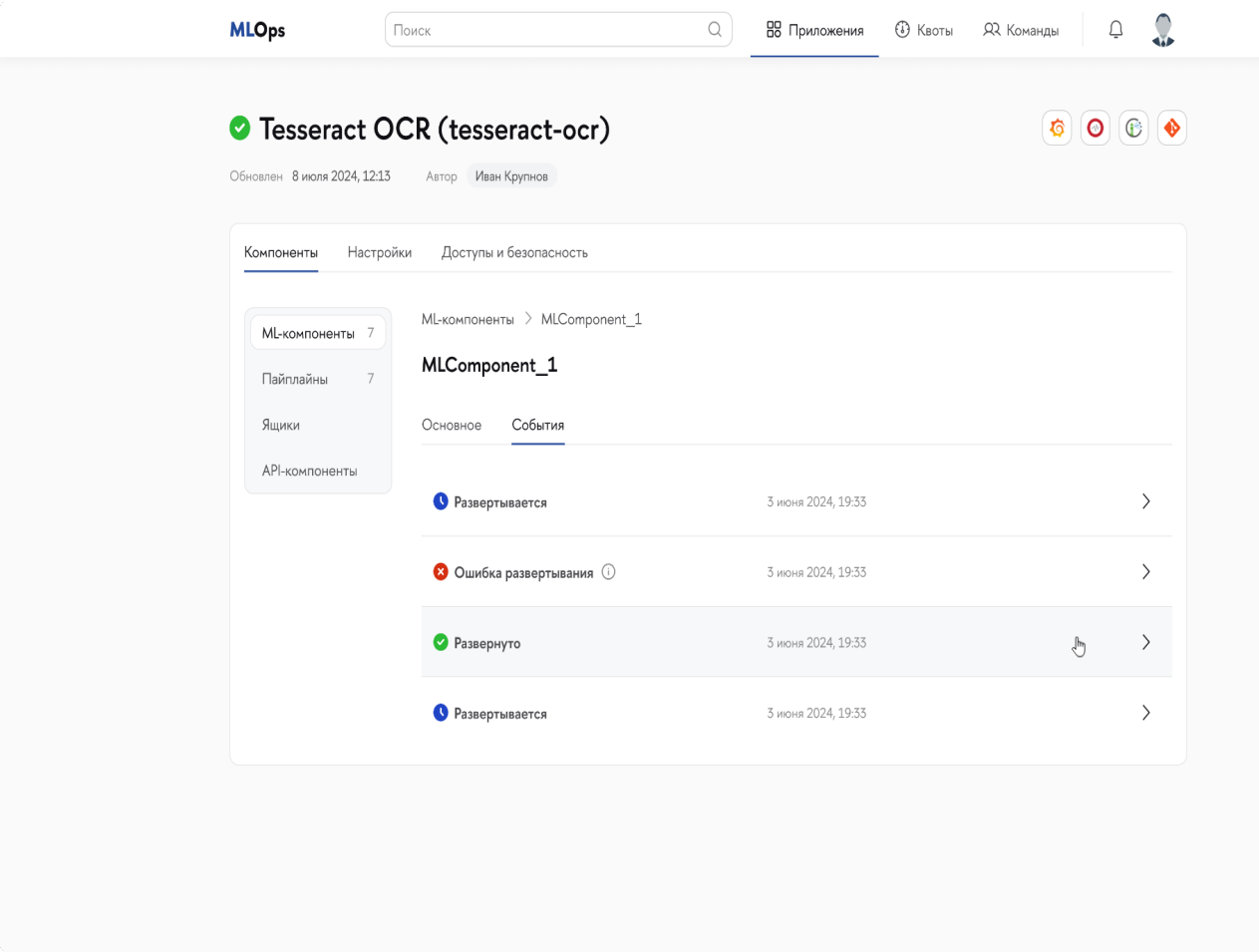
Салех Хади Мухаммед, Департамент программной инженерии
Факультета компьютерных наук НИУ ВШЭ г. Москва
hsalekh@hse.ru

Направление работы 2025

- Улучшение пользовательского опыта:
  - Упрощение процесса создание приложения
  - Личный кабинет
  - Мониторинг
- Развитие средств администрирования
  - Мониторинг администратора
  - Аутентификация пользователей
  - Биллинг
- Развитие платформы
  - Информационная безопасность
  - Инструменты обезличивания данных КИС
  - Конструктор пайплайнов в ЛК разработчика

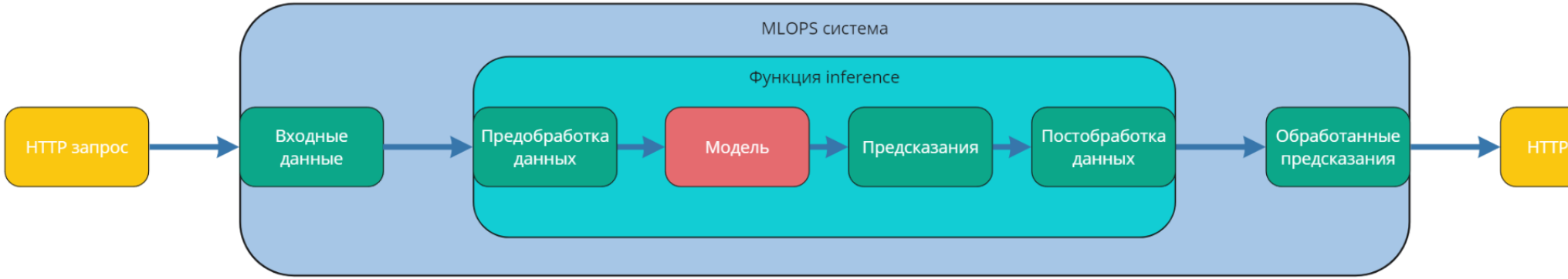
0 Создание приложения в ЛК

- Каждый компонент обладает экраном событий сигнализирующем о состоянии компонента.
- При переходе в статус можно будет увидеть логи указывающие на причину статуса



Работа над Личным кабинетом не окончена! Внешний вид интерфейса может быть изменен.

1 Написание функции inference



Функция inference нужна для:

- Чтения входных данных из файла
- Предобработки входных данных
- Передачи входных данных в модель
- Получения результатов работы модели
- Обработки результатов работы модели, приведение данных в нужный формат
- Запись результатов в файл

Пример предобработки

Выхожу один я на дорогу;
Сквозь туман кремнистый путь блестит;
Ночь тиха. Пустыня внемлет богу,
И звезда с звездою говорит.
В небесах торжественно и чудно!

→

["Выхожу один я на дорогу",
"Сквозь туман кремнистый путь блестит",
"Ночь тиха", "Пустыня внемлет богу", "И звезда с звездою говорит",
"В небесах торжественно и чудно"]

Руководство разработчика

A

2 Создание dockerfile

- Можно взять шаблон и отредактировать
- Содержимое файла как правило не меняется от модели к модели
- На основе этого файла потом соберется docker образ
- В образ копируются все файлы, необходимые для работы модели (например файл, содержащий функцию inference)
- Сами файлы модели в образ не кладутся, они помещаются в S3 хранилище, но об этом позже

Пример dockerfile

```
FROM python:3.9-slim-bullseye

USER root
RUN useradd -ms /bin/bash app
WORKDIR /home/app

RUN pip3 install --upgrade pip wheel setuptool
COPY requirements.txt ./requirements.txt
RUN pip3 install -r ./requirements.txt

COPY --chown=app . ./
USER app

ENV PATH=$PATH:/home/app/.local/bin
ENTRYPOINT ["python3.9", "-m"]
```

3 Сборка и отправка Docker образа на forgejo

Сборка образа проходит в одну команду:

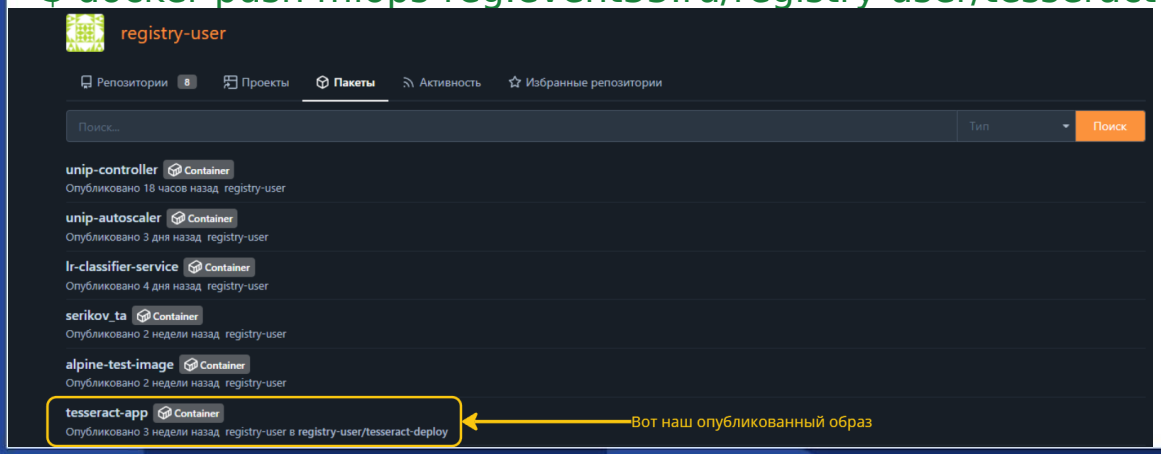
```
$ docker build -f service.dockerfile
```

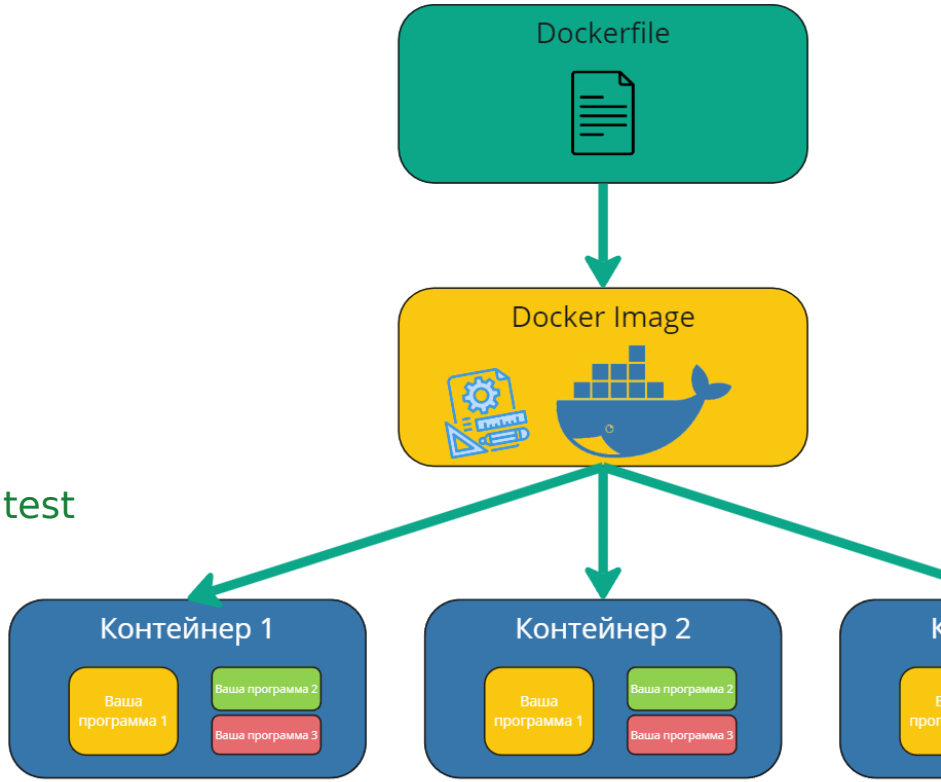
Войдем в докер-репозиторий:

```
$ docker login mlops-reg.event33.ru
```

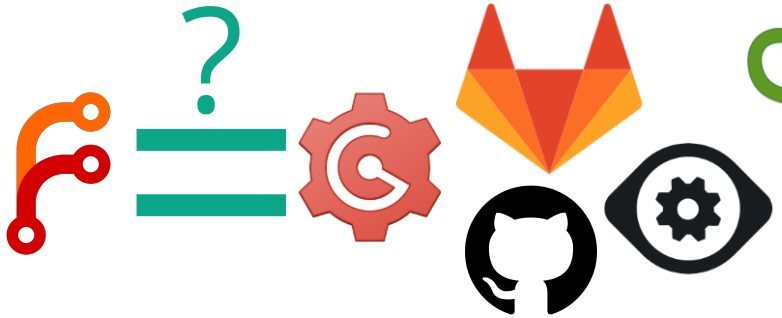
И запушим туда наш новый образ:

```
$ docker push mlops-reg.event33.ru/registry-user/tesseract-app:latest
```

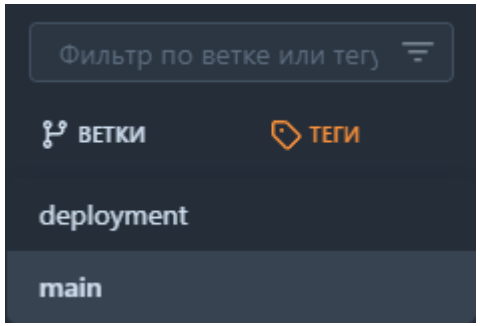




4 Отправка репозитория на forgejo



- Forgejo — это бесплатная и открытая платформа для хостинга репозиториев Git. Все знакомые нам git команды для работы с удаленными репозиториями будут работать с репозиториями на forgejo.
- Финальную версию проекта, подлежащего развертыванию нужно загрузить на ветку “deployment”.
- Сами файлы модели загружать на forgejo не нужно, их мы загрузим в S3 хранилище.



Структура git репозитория