



ООО «Среда Разработки»
ИНН/КПП 7731343398/ 772801001
ОГРН 1177746046895

117437, г. Москва, вн.тер.г. Муниципальный округ Коньково,
ул. Академика Арцимовича, д. 17, помещ. 3/1

**РУКОВОДСТВО АДМИНИСТРАТОРА
ПО ДЛЯ МАШИННОГО ОБУЧЕНИЯ НА БАЗЕ SredaFlowAi**

Листов 36



СОДЕРЖАНИЕ

1	Термины и сокращения	4
2	Общие сведения1	7
2.1	Назначение и краткая характеристика системы	7
2.2	Цели и задачи.....	8
2.3	Состав компонентов продукта.....	9
3	Технические требования системы.....	12
3.1	Требования к аппаратному обеспечению	12
3.2	Требования к установленному программному обеспечению.....	12
3.2.1	Профили использования системы	13
3.2.2	Требования к системе и к внешнему программному обеспечению.....	14
3.3	Требования к конфигурациям программного обеспечения.....	16
3.4	Требования к настройкам рабочих станций пользователей	18
4	Инструкция по установке продукта	19
4.1	Руководство по безопасности	19
4.1.1	Рекомендации по безопасности при развертывании и эксплуатации	19
4.1.2	Управление доступом	20
4.2	Порядок установки продукта	21
4.2.1	Предварительные требования.....	22
4.2.2	Установка продукта менеджером пакетов Helm.....	22
4.2.3	Установка с использованием Kustomize (альтернативный метод)	24
5	Конфигурирование продукта	26
5.1	Первоначальная настройка.....	26



5.2	Интеграции	28
5.2.1	Интеграция с системой аутентификации (OIDC / LDAP)	28
5.2.2	Интеграция с системами мониторинга	29
5.2.3	Интеграция с внешними хранилищами данных	30
6	Администрирование продукта.....	31
6.1	Управление пользователями и доступами.....	31
6.1.1	Создание профиля пользователя	31
6.1.2	Предоставление доступа к профилю другому пользователю	31
6.1.3	Управление квотами ресурсов.....	32
6.1.4	Удаление профиля пользователя.....	32
6.1.5	Управление доступом к компонентам через UI.....	33
6.2	Настройка справочников	33
6.2.1	Настройка доступных образов Notebook-серверов	33
6.2.2	Настройка PodDefaults	34
6.2.3	Настройка StorageClass для томов.....	34
6.3	Настройка рабочих областей	35
6.3.1	Создание и настройка пространств имен (Profiles)	35
6.3.2	Настройка квот и ограничений рабочих областей.....	35
6.3.3	Управление общими рабочими областями.....	35
6.3.4	Мониторинг рабочих областей	36



1 ТЕРМИНЫ И СОКРАЩЕНИЯ

В настоящем документе используются следующие термины и сокращения:

Термин / Сокращение	Расшифровка / Определение
AI	Artificial Intelligence - искусственный интеллект. Область компьютерных наук, занимающаяся созданием систем, способных выполнять задачи, требующие интеллекта.
API	Application Programming Interface - интерфейс программирования приложений. Набор протоколов и инструментов для взаимодействия программных компонентов.
AutoML	Automated Machine Learning - автоматизированное машинное обучение. Процесс автоматического подбора архитектуры и гиперпараметров модели.
CRD	Custom Resource Definition - определение пользовательского ресурса Kubernetes. Механизм расширения API Kubernetes для управления пользовательскими объектами.
DSL	Domain-Specific Language - предметно-ориентированный язык программирования.
GPU	Graphics Processing Unit - графический процессор, применяемый для ускорения вычислений при обучении моделей машинного обучения.
Helm	Менеджер пакетов для Kubernetes. Позволяет управлять установкой, обновлением и удалением приложений в кластере Kubernetes посредством чартов (charts).



Термин / Сокращение	Расшифровка / Определение
HTTPS	HyperText Transfer Protocol Secure - защищенный протокол передачи гипертекста с использованием TLS/SSL-шифрования.
IOPS	Input/Output Operations Per Second - количество операций ввода-вывода в секунду. Метрика производительности дискового хранилища.
Katib	Компонент SredaFlowAi, обеспечивающий автоматический подбор гиперпараметров (AutoML) и оптимизацию модели.
KServe	Компонент SredaFlowAi, предназначенный для публикации (сервинга) обученных ML-моделей в виде эндпоинтов.
Kubernetes (K8s)	Система оркестрации контейнеров с открытым исходным кодом, обеспечивающая автоматическое развертывание, масштабирование и управление контейнеризованными приложениями.
LLM	Large Language Model - большая языковая модель. Нейронная сеть с большим количеством параметров, обученная на обширных текстовых корпусах.
ML	Machine Learning - машинное обучение.
MLOps	Machine Learning Operations - практики и инструменты для организации полного жизненного цикла ML-моделей в производственной среде.
OIDC	OpenID Connect - протокол аутентификации, построенный поверх OAuth 2.0, обеспечивающий единый вход (SSO).
PVC	Persistent Volume Claim - запрос на выделение постоянного тома в Kubernetes.



Термин / Сокращение	Расшифровка / Определение
RBAC	Role-Based Access Control - управление доступом на основе ролей. Модель разграничения прав, где разрешения назначаются ролям, а роли - пользователям.
RPS	Requests Per Second - количество запросов в секунду. Метрика нагрузки на систему.
SDK	Software Development Kit - комплект средств разработки. Набор инструментов, библиотек и документации для создания приложений.
SredaFlowAi	Платформа для управления жизненным циклом AI/ML-разработок, развертываемая на Kubernetes.
SSL / TLS	Secure Sockets Layer / Transport Layer Security - криптографические протоколы для защиты данных при передаче по сети.
SSO	Single Sign-On - технология единого входа, позволяющая пользователю проходить аутентификацию один раз для доступа ко всем подключенным системам.



2 ОБЩИЕ СВЕДЕНИЯ

2.1 Назначение и краткая характеристика системы

SredaFlowAi - программная платформа для обучения искусственного интеллекта, предназначенная для разработки, обучения, развертывания и сопровождения моделей машинного обучения, а также автоматизации полного жизненного цикла их эксплуатации в корпоративной ИТ-инфраструктуре организаций.

Система предоставляет единую среду, охватывающую все этапы разработки и эксплуатации моделей машинного обучения: от интерактивного исследования данных и написания кода до масштабируемого обучения моделей, автоматической оптимизации гиперпараметров, управления пайплайнами и публикации готовых моделей в виде сервисов.

Система предназначена для:

- специалистов по данным (Data Scientists) - разработка, тестирование и запуск ML-экспериментов;
- ML-инженеров - постройка автоматизированных пайплайнов обработки данных и обучение моделей;
- администраторов платформы - управление ресурсами кластера, пользователями, пространствами имен и настройками безопасности;
- DevOps/MLOps-инженеров - интеграции платформы в корпоративные CI/CD-процессы и инфраструктуру.



Основные функции системы SredaFlowAi:

1. Предоставление интерактивной среды для разработки AI/ML-решений на базе Jupyter Notebook и аналогичных инструментов;
2. Управление распределенным обучением моделей с поддержкой фреймворков PyTorch, JAX, XGBoost и других;
3. Автоматический подбор гиперпараметров и оптимизация архитектуры моделей (AutoML);
4. Построение и оркестрация ML-пайплайнов с поддержкой Python SDK и REST API;
5. Регистрация, версионирование и управление артефактами ML-моделей;
6. Публикация обученных моделей в виде масштабируемых HTTP-эндпоинтов для онлайн- и пакетного инференса;
7. Мониторинг экспериментов, запусков пайплайнов и метрик обучения через единый веб-интерфейс;
8. Многопользовательская изоляция рабочих пространств на уровне пространств имен Kubernetes;
9. Управление постоянными томами (Persistent Volumes) для хранения данных и артефактов;
10. Визуализация метрик обучения с помощью TensorBoard.

2.2 Цели и задачи

Цель системы - обеспечить организациям единую, масштабируемую и воспроизводимую среду для разработки, обучения, оптимизации и эксплуатации AI/ML-моделей на базе инфраструктуры Kubernetes.



Задачи системы:

- Унификация инструментального стека ML-разработки в единой платформе, доступной через веб-браузер;
- Сокращение времени вывода ML-моделей в производственную эксплуатацию за счет автоматизации повторяющихся этапов (обучение, оценка, публикация);
- Обеспечение воспроизводимости экспериментов посредством версионирования конфигураций, кода и артефактов;
- Эффективное использование вычислительных ресурсов кластера (в том числе GPU) за счет автоматического планирования заданий;
- Соблюдение принципов многопользовательской изоляции и ролевого разграничения доступа;
- Интеграция с корпоративными системами аутентификации, мониторинга и хранения данных

2.3 Состав компонентов продукта

SredaFlowAi включает следующие функциональные компоненты, отображаемые в навигационном меню платформы:

Компонент	Назначение
Главный дашборд	Центральная веб-панель управления. Обеспечивает единую точку доступа ко всем компонентам платформы, отображает состояние пространства имен пользователя и навигационное меню.
Блокноты (Notebooks)	Управление серверами Jupyter Notebook и аналогичными IDE. Позволяет создавать, запускать и



Компонент	Назначение
	останавливать интерактивные среды разработки с настраиваемыми вычислительными ресурсами.
Tensorboards	Запуск и управление экземплярами TensorBoard для визуализации метрик обучения моделей.
Томы (Volumes)	Управление постоянными томами (PVC - Persistent Volume Claims) для хранения данных, датасетов и артефактов моделей.
Эксперименты Katib	AutoML-компонент для автоматического подбора гиперпараметров и оптимизации архитектуры нейронных сетей. Поддерживает алгоритмы байесовской оптимизации, случайного поиска, нейронного архитектурного поиска (NAS).
Эндпоинты KServe	Компонент для публикации (сервинга) обученных ML-моделей в виде масштабируемых REST-эндпоинтов. Поддерживает онлайн- и пакетный режимы инференса.
Пайплайны - Пайплайны	Определения автоматизированных ML-пайплайнов, созданных с помощью Python SDK.
Пайплайны - Эксперименты	Логические группы запусков пайплайнов для организации и сравнения экспериментов.
Пайплайны - Запуски	Экземпляры однократного выполнения пайплайна с фиксацией входных параметров и результатов.
Пайплайны - Повторяющиеся запуски	Расписания периодического автоматического запуска пайплайнов.
Пайплайны - Артефакты	Реестр артефактов (модели, датасеты, метрики), созданных в ходе выполнения пайплайнов.



ООО «Среда Разработки»
ИНН/КПП 7731343398/ 772801001
ОГРН 1177746046895

117437, г. Москва, вн.тер.г. Муниципальный округ Коньково,
ул. Академика Арцимовича, д. 17, помещ. 3/1

Компонент	Назначение
Пайплайны - Выполнения	Детальная история выполнения отдельных шагов пайплайнов с логами и метаданными.



3 ТЕХНИЧЕСКИЕ ТРЕБОВАНИЯ СИСТЕМЫ

3.1 Требования к аппаратному обеспечению

SredaFlowAi разворачивается на кластере Kubernetes и не имеет жестко фиксированных требований к конкретной аппаратной конфигурации - потребности в ресурсах определяются профилем нагрузки (см. раздел 3.2.1). При планировании инфраструктуры необходимо учитывать следующее:

- Для работы компонентов платформы (управляющие плоскости, операторы, сервисы) резервируется базовый объем ресурсов кластера;
- Для задач обучения моделей (Trainer) и нейронного архитектурного поиска (Katib) рекомендуется наличие узлов с GPU-ускорителями (NVIDIA CUDA-совместимые);
- Для хранения артефактов, датасетов и метаданных необходимо подключение к системе хранения, поддерживающей PVC (блочное или файловое хранилище).

Рекомендуемые требования к оборудованию приведены в таблицах раздела 3.2.2 с разбивкой по профилям нагрузки. Приведенные ресурсы предназначены только для компонентов SredaFlowAi и не включают ресурсы, необходимые для инфраструктурных сервисов Kubernetes (etcd, kube-apiserver, CoreDNS и др.), а также не учитывают нефункциональные требования (надежность, отказоустойчивость).

3.2 Требования к установленному программному обеспечению

В настоящем подразделе приведены требования к программной среде, профилям использования системы и параметрам внешнего программного обеспечения, необходимым для разворачивания и эксплуатации SredaFlowAi.



3.2.1 Профили использования системы

Для целей планирования ресурсов определены три профиля нагрузки:

Таблица 3.2.1 - Профили использования системы SredaFlowAi

Нефункциональные требования	Профиль №1 (минимальная инсталляция)	Профиль №2 (средняя инсталляция)	Профиль №3 (крупная инсталляция)
Количество зарегистрированных пользователей в системе	< 50	< 1 000	< 10 000
Количество одновременно работающих пользователей	< 5	< 100	< 1 000
Количество одновременно запущенных Notebook-серверов	< 5	< 50	< 500
Количество запускаемых пайплайнов в день	< 10	< 200	< 2 000
Количество зарегистрированных экспериментов Katib	< 10	< 100	< 1 000
Количество опубликованных эндпоинтов KServe	< 5	< 50	< 500
Объем накопленных артефактов (датасеты, модели)	< 100 ГБ	< 5 ТБ	< 50 ТБ
Интенсивность вызова публичного API	< 1 RPS	< 10 RPS	< 100 RPS

3.2.2 Требования к системе и к внешнему программному обеспечению

Ниже приведен перечень программных компонентов и рекомендуемых версий, необходимых для функционирования системы независимо от выбранного профиля внедрения.

Таблица 3.2.2.1 - Требования к программному обеспечению (общие для всех профилей)

Компонент	Версия	Назначение
Kubernetes	≥ 1.27	Платформа оркестрации контейнеров
kubectl	Соответствует версии кластера	CLI для управления кластером Kubernetes
Helm	≥ 3.10	Менеджер пакетов для Kubernetes
Container Runtime	containerd ≥ 1.7 или CRI-O ≥ 1.27	Среда выполнения контейнеров
Cert-Manager	≥ 1.12	Управление TLS-сертификатами
Istio (Service Mesh)	≥ 1.17	Сервисная сетка, маршрутизация трафика, mTLS
Dex (OIDC-провайдер)	≥ 2.36	Аутентификация и SSO
Object Storage (S3-совместимое)	MinIO $\geq$ RELEASE.2024 или AWS S3 / аналог	Хранение артефактов пайплайнов и моделей
Kustomize	≥ 5.0	Управление манифестами (при установке без Helm)
NVIDIA GPU Operator	≥ 23.9 (при наличии GPU)	Управление GPU-ресурсами в кластере

Минимальные требования к вычислительным ресурсам и инфраструктуре для развертывания системы приведены в таблице ниже.

Таблица 3.2.2.2 - Профиль №1 (минимальная инсталляция)

Компонент	CPU	RAM	Storage	Количество узлов	Прочие требования
Управляющая плоскость Kubernetes (control plane)	4 vCPU	8 ГБ	50 ГБ SSD	1	IOPS $\geq$ 3 000
Рабочие узлы (worker nodes)	8 vCPU	16 ГБ	100 ГБ SSD	2	Пропускная способность сети: $\geq$ 1 Гбит/с
Хранилище артефактов (S3/MinIO)	2 vCPU	4 ГБ	500 ГБ	1	-
Итого (минимум)	14 vCPU	28 ГБ	650 ГБ	4	-

Для эксплуатации системы в условиях средней нагрузки рекомендуются следующие параметры инфраструктуры и вычислительных ресурсов.

Таблица 3.2.2.3 - Профиль №2 (средняя инсталляция)

Компонент	CPU	RAM	Storage	Количество узлов	Прочие требования
Управляющая плоскость Kubernetes	8 vCPU	16 ГБ	100 ГБ SSD	3 (HA)	IOPS $\geq$ 5 000
Рабочие узлы (CPU)	16 vCPU	64 ГБ	200 ГБ SSD	4	Пропускная способность сети: $\geq$ 10 Гбит/с
Рабочие узлы (GPU)	8 vCPU	32 ГБ	100 ГБ SSD	2	2× NVIDIA A10G или аналог
Хранилище артефактов	4 vCPU	8 ГБ	5 ТБ	1 (или HA-кластер)	-

Компонент	CPU	RAM	Storage	Количество узлов	Прочие требования
Итого	60+ vCPU	200+ ГБ	5+ ТБ	10+	-

Для эксплуатации системы в условиях высокой нагрузки и повышенных требований к масштабируемости рекомендуется следующая конфигурация инфраструктуры.

Таблица 3.2.2.4 - Профиль №3 (крупная инсталляция)

Компонент	CPU	RAM	Storage	Количество узлов	Прочие требования
Управляющая плоскость Kubernetes	16 vCPU	32 ГБ	200 ГБ NVMe	3+ (HA)	IOPS $\geq$ 10 000
Рабочие узлы (CPU)	32 vCPU	128 ГБ	500 ГБ SSD	8+	Пропускная способность сети: $\geq$ 25 Гбит/с
Рабочие узлы (GPU)	16 vCPU	64 ГБ	200 ГБ SSD	4+	4–8× NVIDIA A100/H100 или аналог
Хранилище артефактов	8 vCPU	16 ГБ	50 ТБ	HA-кластер	Выделенная сеть хранения
Итого	200+ vCPU	800+ ГБ	50+ ТБ	20+	-

3.3 Требования к конфигурациям программного обеспечения

Kubernetes:

- Должна быть включена поддержка RBAC (--authorization-mode=RBAC);
- Должна быть доступна поддержка CRD (Custom Resource Definitions);
- Рекомендуется включить Pod Security Admission (PSA) в режиме Restricted или Baseline для системных пространств имен;



- Рекомендуется настроить сетевую политику (NetworkPolicy) для изоляции трафика между пространствами имен пользователей.

Istio:

- Должен быть установлен и настроен Istio Service Mesh для обеспечения маршрутизации запросов через центральный шлюз (Istio Ingress Gateway);
- Необходимо настроить mTLS для межсервисного взаимодействия компонентов платформы.

Dex (OIDC):

- Должен быть настроен как внешний OIDC-провайдер или подключен к корпоративной системе аутентификации (Active Directory, LDAP, Google Workspace и др.);
- Должны быть сконфигурированы клиентские приложения для компонентов платформы.

Object Storage (MinIO / S3):

- Должны быть созданы бакеты для хранения артефактов пайплайнов (mlpipeline), метаданных и моделей;
- Доступы (Access Key, Secret Key) должны быть переданы в виде Kubernetes Secrets.

NVIDIA GPU Operator:

- При наличии GPU-узлов должен быть установлен GPU Operator с корректными версиями драйверов для узлов кластера;



- Узлы с GPU должны быть помечены соответствующими метками (nvidia.com/gpu: "true").

3.4 Требования к настройкам рабочих станций пользователей

К рабочим станциям пользователей, осуществляющих работу в SredaFlowAi, применяются следующие требования:

- Версия операционной системы должна поддерживать последние версии веб-браузеров (Windows 10/11, macOS 12+, Ubuntu 20.04+ или другой аналогичный Linux-дистрибутив);
- Веб-браузер: последние актуальные версии Google Chrome, Mozilla Firefox, Yandex Browser. Рекомендуется использовать Google Chrome версии 110 и выше;
- В браузере должны быть включены JavaScript и поддержка WebSocket-соединений;
- Разрешение экрана рекомендуется не ниже 1280×768 пикселей;
- Должен быть обеспечен сетевой доступ к HTTPS-эндпоинту (порт 443) или HTTP-эндпоинту (порт 80) кластера SredaFlowAi.

Для администраторов платформы дополнительно требуется:

- Установленный kubectl (версия, совместимая с кластером Kubernetes);
- Установленный helm (≥ 3.10) для выполнения операций установки и обновления;
- Настроенный kubeconfig с правами кластерного администратора (cluster-admin).



4 ИНСТРУКЦИЯ ПО УСТАНОВКЕ ПРОДУКТА

4.1 Руководство по безопасности

В настоящем подразделе приведены основные требования и рекомендации по обеспечению безопасности при развертывании и эксплуатации системы.

4.1.1 Рекомендации по безопасности при развертывании и эксплуатации

Управление секретами:

- Все чувствительные данные (пароли, ключи API, токены) должны храниться в виде Kubernetes Secrets, а не в открытом виде в конфигурационных файлах;
- Рекомендуется использовать внешние системы управления секретами в связке с оператором secrets-store-csi-driver;
- Регулярно ротируйте ключи доступа к Object Storage и сервисные учетные записи;
- Запрещайте монтирование секретов в неавторизованных пространствах имен с помощью Kubernetes RBAC.

Сетевая безопасность:

- Доступ к платформе должен предоставляться исключительно по протоколу HTTPS с валидным TLS-сертификатом (используйте Cert-Manager для автоматического получения и обновления сертификатов Let's Encrypt или корпоративного CA);



- Настройте Kubernetes NetworkPolicy для ограничения межнамспейсного трафика - пользовательские пространства имен не должны иметь прямого сетевого доступа друг к другу;
- Ограничьте внешний доступ к Kubernetes API Server с помощью IP-allowlist или VPN;
- Отключите неиспользуемые порты и сервисы.

Образы контейнеров:

- Используйте только образы из доверенных частных реестров;
- Регулярно сканируйте образы на уязвимости (Trivy, Clair);
- Запретите запуск контейнеров от имени root, используйте runAsNonRoot: true в Security Context;
- Включите readOnlyRootFilesystem: true там, где это не нарушает работу компонентов.

Аудит и мониторинг:

- Включите Kubernetes Audit Logging для отслеживания всех операций с API;
- Настройте алертинг на подозрительную активность (массовое создание Pod, доступ к секретам);
- Регулярно проверяйте права доступа сервисных аккаунтов (принцип минимальных привилегий).

4.1.2 Управление доступом

SredaFlowAi использует многоуровневую модель управления доступом. Аутентификация осуществляется через Dex - OIDC-провайдер,



интегрированный с Istio. Dex может быть подключен к следующим внешним системам: Active Directory / LDAP, корпоративный SAML-провайдер.

Авторизация реализована через:

- Kubernetes RBAC - управление доступом к ресурсам кластера (Pods, Secrets, CRD-объекты);
- Profile Controller - изолирует пространства имен пользователей, назначая роли edit (владелец профиля) или view (гость) для каждого профиля.

Роли в системе:

Роль	Описание
Администратор кластера (cluster-admin)	Полный доступ ко всем ресурсам кластера. Управление пользователями и профилями платформы.
Владелец профиля (Profile Owner)	Полный доступ к ресурсам своего пространства имен: создание блокнотов, запуск пайплайнов, публикация моделей.
Участник профиля (Contributor)	Доступ к пространству имен другого пользователя (права edit или view).

4.2 Порядок установки продукта

В настоящем подразделе описана последовательность действий, необходимых для подготовки инфраструктуры и выполнения установки продукта.



4.2.1 Предварительные требования

Перед началом установки убедитесь, что выполнены следующие условия:

1. Кластер Kubernetes версии ≥ 1.27 запущен и доступен;
2. Инструменты `kubectl` и `helm` установлены на рабочей станции администратора;
3. Файл `kubeconfig` настроен на целевой кластер с правами `cluster-admin`;
4. Object Storage (MinIO или S3-совместимый сервис) доступен из кластера;
5. Зарезервировано доменное имя для доступа к платформе;
6. (Опционально) настроен OIDC-провайдер или корпоративная система аутентификации.

Проверьте подключение к кластеру:

```
kubectl cluster-info  
kubectl get nodes
```

4.2.2 Установка продукта менеджером пакетов Helm

Установка `SredaFlowAi` производится с использованием менеджера пакетов `Helm`.

1. Добавление репозитория Helm

```
helm repo add sredaflowai <URL-репозитория-Helm-чартов>  
helm repo update
```

2. Создание пространства имен

```
kubectl create namespace sredaflowai
```



3. Создание файла конфигурации

Создайте файл values.yaml с параметрами установки:

values.yaml - пример конфигурации SredaFlowAi

global:

domain: sredaflowai.example.com # Доменное имя платформы
storageClass: standard # Класс хранилища Kubernetes

auth:

oidc:

issuerUrl: https://dex.example.com/dex
clientId: sredaflowai
clientSecret: <secret>

objectStorage:

endpoint: http://minio.minio:9000
accessKey: <access-key>
secretKey: <secret-key>
bucketName: mlpipeline

pipelines:

enabled: true

notebooks:

enabled: true

katib:

enabled: true

kserve:

enabled: true

tensorboard:

enabled: true

4. Установка платформы

```
helm install sredaflowai sredaflowai/sredaflowai \
--namespace sredaflowai \
```



```
--values values.yaml \  
--timeout 20m \  
--wait
```

5. Проверка установки

После завершения установки убедитесь, что все поды запущены:

```
kubectl get pods -n sredaflowai
```

Все поды должны находиться в состоянии Running или Completed.

Проверьте доступность центральной панели управления:

```
kubectl get ingress -n istio-system
```

Откройте браузер и перейдите по адресу `https://<domain>`, указанному в `values.yaml`.

4.2.3 Установка с использованием Kustomize (альтернативный метод)

При установке без Helm используются Kustomize-манифесты. Данный метод рекомендуется для опытных администраторов Kubernetes.

1. Клонирование репозитория манифестов

```
git clone https://example.com/sredaflowai/manifests.git  
cd manifests  
git checkout v1.11-branch
```

2. Применение манифестов

```
while ! kustomize build example | kubectl apply -f -; do  
  echo "Retrying to apply resources..."  
  sleep 20  
done
```

Повторный запуск команды может потребоваться из-за порядка применения CRD-ресурсов.



3. Проверка установки

```
kubectl get pods -n sredaflow  
kubectl get pods -n auth  
kubectl get pods -n cert-manager  
kubectl get pods -n istio-system  
kubectl get pods -n knative-eventing  
kubectl get pods -n knative-serving
```

4. Доступ к платформе (port-forward)

Для первоначального доступа без Ingress используйте port-forward:

```
kubectl port-forward svc/istio-ingressgateway -n istio-system 8080:80
```

Откройте браузер: <http://localhost:8080>



5 КОНФИГУРИРОВАНИЕ ПРОДУКТА

5.1 Первоначальная настройка

После успешной установки SredaFlowAi выполните следующие шаги первоначальной настройки:

1. Создание административного профиля пользователя

Первый пользователь, выполнивший вход в систему, должен получить права администратора. Для создания профиля администратора вручную примените следующий манифест:

```
apiVersion: sredaflow.org/v1
kind: Profile
metadata:
  name: admin-profile
spec:
  owner:
    kind: User
    name: admin@example.com
  resourceQuotaSpec:
    hard:
      cpu: "40"
      memory: 100Gi
      requests.nvidia.com/gpu: "4"
      persistentvolumeclaims: "20"
      requests.storage: "500Gi"
kubectl apply -f admin-profile.yaml
```

2. Настройка дефолтного образа Notebook-сервера

Отредактируйте ConfigMap с доступными образами блокнотов:

```
kubectl edit configmap jupyter-web-app-config -n sredaflow
```



Добавьте или измените список доступных образов в соответствии с корпоративным реестром контейнеров.

3. Настройка лимитов ресурсов по умолчанию

Рекомендуется задать LimitRange для пользовательских пространств имен, чтобы предотвратить неконтролируемое потребление ресурсов:

```
apiVersion: v1
kind: LimitRange
metadata:
  name: default-limits
spec:
  limits:
  - default:
    cpu: "2"
    memory: "4Gi"
  defaultRequest:
    cpu: "500m"
    memory: "1Gi"
  type: Container
```

4. Настройка хранилища артефактов пайплайнов

Убедитесь, что Sredaflow Pipelines подключен к Object Storage. Проверьте Secret:

```
kubectrl get secret mlpipeline-minio-artifact -n sredaflow -o yaml
```

При необходимости создайте Secret заново:

```
kubectrl create secret generic mlpipeline-minio-artifact \
--from-literal=accesskey=<access-key> \
--from-literal=secretkey=<secret-key> \
-n sredaflow
```



5. Проверка работы компонентов через UI

Откройте браузер, войдите в систему под учетной записью администратора и убедитесь, что в навигационном меню отображаются все разделы: Блокноты, Tensorboards, Тома, Эксперименты Katib, Эндпоинты KServe, Пайплайны.

5.2 Интеграции

5.2.1 Интеграция с системой аутентификации (OIDC / LDAP)

SredaFlowAi использует Dex как универсальный OIDC-провайдер. Для подключения корпоративного каталога (Active Directory, LDAP) отредактируйте конфигурацию Dex:

```
kubectrl edit configmap dex -n auth
```

Пример конфигурации коннектора LDAP:

```
connectors:  
- type: ldap  
  name: Corporate LDAP  
  id: ldap  
  config:  
    host: ldap.example.com:636  
    insecureNoSSL: false  
    bindDN: cn=service-account,dc=example,dc=com  
    bindPW: <password>  
    userSearch:  
      baseDN: ou=users,dc=example,dc=com  
      filter: "(objectClass=person)"  
      username: sAMAccountName  
      idAttr: DN  
      emailAttr: mail  
      nameAttr: displayName  
    groupSearch:  
      baseDN: ou=groups,dc=example,dc=com
```



```
filter: "(objectClass=group)"  
userAttr: DN  
groupAttr: member  
nameAttr: cn
```

После редактирования перезапустите Dex:

```
kubectrl rollout restart deployment dex -n auth
```

5.2.2 Интеграция с системами мониторинга

Prometheus и Grafana:

Компоненты SredaFlowAi экспортируют метрики в формате Prometheus. Для подключения мониторинга добавьте ServiceMonitor-объекты, указывающие на сервисы компонентов платформы:

```
kubectrl apply -f monitoring/servicemonitors.yaml -n sredaflow
```

Рекомендуемые метрики для мониторинга:

- Количество активных Notebook-серверов;
- Статус выполнения пайплайнов (успех/ошибка);
- Загрузка CPU/GPU рабочих узлов;
- Объем использованного хранилища PVC.

Журналирование (Logging):

Для централизованного сбора логов подключите EFK-стек (Elasticsearch, Fluentd, Kibana) или аналогичное решение (Loki + Grafana). Настройте Fluentd/Fluent Bit для сбора логов из пространств имен sredaflow, istio-system, auth, cert-manager и пользовательских пространств имен.



5.2.3 Интеграция с внешними хранилищами данных

Для подключения внешних источников данных (корпоративные S3-хранилища, NFS, Ceph) создайте StorageClass и соответствующие PersistentVolume-объекты, после чего укажите их как доступные классы хранения в настройках платформы.

```
kubectl apply -f storage/storageclass-nfs.yaml
```



6 АДМИНИСТРИРОВАНИЕ ПРОДУКТА

6.1 Управление пользователями и доступами

6.1.1 Создание профиля пользователя

В SredaFlowAi каждый пользователь работает в изолированном пространстве имен Kubernetes, которое создается автоматически при создании объекта Profile. Администратор может создать профиль за пользователя:

```
apiVersion: sredaflow.org/v1
kind: Profile
metadata:
  name: user-john-doe
spec:
  owner:
    kind: User
    name: john.doe@example.com
  resourceQuotaSpec:
    hard:
      cpu: "8"
      memory: "32Gi"
      requests.nvidia.com/gpu: "1"
      persistentvolumeclaims: "5"
      requests.storage: "100Gi"
kubectl apply -f user-profile.yaml
```

После применения манифеста будет создано пространство имен user-john-doe с необходимыми ролями и квотами ресурсов.

6.1.2 Предоставление доступа к профилю другому пользователю

Для совместной работы пользователи могут делиться доступом к своему пространству имен. Администратор предоставляет доступ через объект RoleBinding:



```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: contributor-access
  namespace: user-john-doe
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: sredaflow-edit
subjects:
- apiGroup: rbac.authorization.k8s.io
  kind: User
  name: jane.smith@example.com
```

Доступные роли:

- sredaflow-edit - полный доступ к ресурсам пространства имен (создание, изменение, удаление).
- sredaflow-view - доступ только на чтение.

6.1.3 Управление квотами ресурсов

Для изменения квот ресурсов пользователя отредактируйте объект Profile:

```
kubectl edit profile user-john-doe
```

Измените секцию resourceQuotaSpec в соответствии с требуемыми значениями.

Для просмотра текущего использования ресурсов:

```
kubectl describe resourcequota -n user-john-doe
```

6.1.4 Удаление профиля пользователя

Удаление профиля влечет за собой удаление всего пространства имен пользователя, включая блокноты, тома, пайплайны и другие ресурсы. Операция необратима.



```
kubectl delete profile user-john-doe
```

6.1.5 Управление доступом к компонентам через UI

Центральная панель управления SredaFlowAi позволяет пользователям самостоятельно управлять доступом к своему пространству имен через раздел Управление и Участники. Администратор платформы может управлять профилями всех пользователей через kubectl или через административный интерфейс.

6.2 Настройка справочников

6.2.1 Настройка доступных образов Notebook-серверов

Список образов, доступных пользователям при создании Notebook-сервера, настраивается в ConfigMap jupyter-web-app-config:

```
kubectl get configmap jupyter-web-app-config -n sredaflow -o yaml
```

Структура конфигурации:

```
spawnerFormDefaults:
```

```
  image:
```

```
    value: sredaflownotebookswg/jupyter-scipy:v1.8.0
```

```
  imageGroupOne:
```

```
    options:
```

```
      - displayName: "JupyterLab SciPy (CPU)"
```

```
        value: sredaflownotebookswg/jupyter-scipy:v1.8.0
```

```
      - displayName: "JupyterLab TensorFlow (GPU)"
```

```
        value: sredaflownotebookswg/jupyter-tensorflow-cuda:v1.8.0
```

```
      - displayName: "JupyterLab PyTorch (GPU)"
```

```
        value: sredaflownotebookswg/jupyter-pytorch-cuda:v1.8.0
```

```
  cpu:
```

```
    value: "1"
```

```
    readOnly: false
```

```
  memory:
```

```
    value: "2Gi"
```



`readOnly: false`

После редактирования ConfigMap перезапустите деплоймент Jupyter Web App:

```
kubectl rollout restart deployment jupyter-web-app-deployment -n sredaflow
```

6.2.2 Настройка PodDefaults

PodDefault - механизм автоматического добавления конфигурации в Pod'ы в определенном пространстве имен (например, монтирование секретов доступа к S3 во все Notebook-серверы):

```
apiVersion: sredaflow.org/v1alpha1
kind: PodDefault
metadata:
  name: access-minio
  namespace: user-john-doe
spec:
  selector:
    matchLabels:
      access-minio: "true"
  desc: "Доступ к MinIO/S3"
  volumeMounts:
    - mountPath: /var/run/secrets/minio
      name: minio-credentials
  volumes:
    - name: minio-credentials
      secret:
        secretName: minio-access-secret
```

6.2.3 Настройка StorageClass для томов

Для управления доступными классами хранения при создании PVC через интерфейс платформы убедитесь, что нужные StorageClass настроены в кластере:

```
kubectl get storageclass
```



Класс хранения с аннотацией `storageclass.kubernetes.io/is-default-class: "true"` будет предлагаться пользователям по умолчанию.

6.3 Настройка рабочих областей

6.3.1 Создание и настройка пространств имен (Profiles)

Рабочая область в SredaFlowAi - это Kubernetes-пространство имен (namespace), ассоциированное с объектом Profile. Каждая рабочая область изолирует ресурсы пользователя: блокноты, тома, пайплайны, модели.

Для создания новой рабочей области (профиля) используйте команды, описанные в разделе 6.1.1.

Для просмотра всех рабочих областей (профилей):

```
kubectl get profiles
```

6.3.2 Настройка квот и ограничений рабочих областей

Рекомендуется задавать квоты ресурсов и лимиты для каждой рабочей области во избежание монопольного захвата ресурсов кластера. Квоты задаются в поле `resourceQuotaSpec` объекта Profile (см. раздел 6.1.1).

Дополнительно можно применить `LimitRange` непосредственно к пространству имен профиля для ограничения ресурсов отдельных Pod'ов:

```
kubectl apply -f limit-range.yaml -n user-john-doe
```

6.3.3 Управление общими рабочими областями

Для создания общей рабочей области (доступной нескольким пользователям) создайте профиль с условным владельцем (сервисный аккаунт



или технический пользователь), а затем предоставьте доступ нужным пользователям через RoleBinding согласно разделу 6.1.2.

6.3.4 Мониторинг рабочих областей

Для получения сводки по всем рабочим областям и их ресурсному потреблению:

- Просмотр всех профилей

```
kubectl get profiles -o wide
```

- Просмотр потребления ресурсов по пространствам имен

```
kubectl get resourcequota --all-namespaces
```

- Просмотр активных Notebook-серверов

```
kubectl get notebooks --all-namespaces
```

- Просмотр активных запусков пайплайнов

```
kubectl get pipelinerun --all-namespaces
```