Руководство по развертыванию DUO TECH IMA MBAL Composition

Оглавление

1. Общие сведения....3
2. Список аббревиатур....4
3. Установка и конфигурация DUO TECH IMA....5
3.1. Требования к программному обеспечению и оборудованию....5
3.2. Установка компонентов системы....6
3.2.1. Подготовка файлов значений для helm values....6
3.2.2. Создание Namespace в Kubernetes....19
3.2.3. Авторизация в registry ext-repo.ois-bright.ru....19
3.2.4. Экспорт конфигурации Kubernetes для helm....20
3.2.5. Скачивание helm-charts из гит репозитория....20
3.2.6. Перенос дополнительных helm-charts.....20
3.2.7. Изменение значения в чарте ois-platform для redirectUris....20
3.2.8. Установку kafka, postgresql и redis....21
3.2.9. Установка сервисов Системы DUO TECH IMA....21
3.3. Настройка Web-core-back....22
3.4. Настройка ПО из GUI....22
3.4.1. Добавление приложений в Реестр Приложений....24
3.4.2. Настройка меню Приложений....28
3.4.3. Настройка ролей для пользователей.....33

1. Общие сведения

Настоящее руководство содержит описание развертывания DUO TECH IMA и описывает возможности по администрированию системы.

Модуль **Администрирование** объединяет в себе приложения, которые позволяют выполнить настройку параметров приложений, учётных записей и ролей пользователей, а также получить информацию об использовании системы.

Модуль включает в себя следующие приложения:

- **Реестр приложений** — настройка перечня и параметров подключения приложений. Подробнее см. в разделе Реестр приложений.
- **Приложения** — настройка меню DUO TECH IMA и базовых параметров приложений. Подробнее см. в разделе Приложения.
- **Пользователи** — просмотр учётных записей пользователей. Подробнее см. в разделе Пользователи.
- **Управление ролями** — настройка параметров доступа пользовательских ролей. Подробнее см. в разделе Управление ролями.

Подробнее о взаимодействии с интерфейсом системы DUO TECH IMA, см. раздел Навигация в DUO TECH IMA.

2. Список аббревиатур

БД	БАЗА ДАННЫХ
ИБ	Информационная безопасность
ИМА	Интегрированное моделирование актива
ОС	Операционная система
ПО	Программное обеспечение
СРК	Система резервного копирования
СУБД	Систему управления базами данных
СУИД	Система управления идентификационными данными
УЗ	Учетная запись
БД	База данных

3. Установка и конфигурация DUO TECH IMA

Данный раздел содержит описание и инструкции по установке и настройке DUO TECH IMA в ручном режиме (без использования программ-инсталляторов).

В разделе описаны следующие аспекты настройки и конфигурации системы:

- Требования к программному обеспечению и оборудованию
- Установка компонентов системы
- Настройка HUB
- Обновление БД системы
- Параметры ролевого доступа

3.1. Требования к программному обеспечению и оборудованию

Для корректной работы DUO TECH IMA программное обеспечение и оборудование на серверной и пользовательской сторонах системной инфраструктуры должно соответствовать системным требованиям, описанным в этом разделе.

Требования к аппаратной конфигурации системы DUO TECH IMA

В данном пункте приведены требования к аппаратной платформе, которые должны быть выполнены до установки компонентов Системы DUO TECH IMA. В системе реализованы алгоритмы, для выполнения которых требуются значительные процессорные мощности и большой объем оперативной памяти для хранения буферизованных данных, необходимых для выполнения расчетов. Далее приведены в таблице с требованиями к оборудованию в зависимости от количества пользователей, которые работают в Системе.

Сервер	Процессор	Память, Гб	СХД, IOPS	Дисковое пространство, Гб	Сетевое подключение	Примечание
Сервер БД	16 ядер	64	1000 и более	500	1Гб+	Максимум 150 пользователей

Требования к системному программному обеспечению

Система поставляется в виде docker образов. На серверах должно быть развернуто ПО Kubernetes версии не ниже v1.22 (рекомендуемая v1.24), пакетный менеджер helm версии не ниже v3.8.0

Система в качестве брокера сообщений использует Apache Kafka. Для корректной работы Системы требуется версия Kafka не ниже 3.4.0.

Система в качестве СУБД использует БД PostgreSQL. Для корректной работы Системы требуется версия PostgreSQL или Postgres Pro Standard/Enterprise не ниже 14.

Дополнительно используется сервер баз данных типа ключ-значение Redis не ниже 18.2.0.

Apache Kafka, PostgreSQL и Redis в рамках данной инструкции устанавливаются в контейнерную среду Kubernetes в отдельный Namespace (common-applications).

Требования к рабочим станциям пользователей Системы

В данном параграфе приведены рекомендуемые характеристики персональных компьютеров пользователей Системы, при соблюдении которых можно гарантировать плавную работу интерфейса Системы.

Компонент	Минимальные требования	Рекомендуемые разработчиком
Процессор	Intel Core i3 или аналогичный с частотой 2,0 ГГц или более производительный (4 физ. ядра)	Intel Core i5 или аналогичный с частотой 2,1 ГГц или более производительный (4 и более физ. ядер)
Оперативная память	4 ГБ	8 ГБ или более
Дисковый накопитель	50* ГБ свободного места	100* ГБ свободного места
Браузер	Браузеры, основанные на базе проекта Chromium версии не ниже 86.x	Браузеры, основанные на базе проекта Chromium версии не ниже 86.x
Монитор	Монитор с диагональю 23" с разрешением 1920×1080	Монитор с диагональю 23" (или более) с разрешением 1920×1080 (или более)
Дополнительное ПО (необязательное)		Р7-Офис Профессиональный (десктопная версия)

3.2. Установка компонентов системы

3.2.1. Подготовка файлов значений для helm values

Скопируйте файлы конфигурации (чарты конфигурации) в целевую директорию: /opt/configs

После подготовьте значения для helm-values.

<server_hostname> - заменить на FQDN/IP адрес сервера.

Подготовка файла значений kafka.yaml

```
kafka:
  global:
    imageRegistry: "ext-repo.ois-bright.ru"
    imagePullSecrets:
      - ext-repo.ois-bright.ru
  replicaCount: 1
  zookeeper:
    enabled: true
    replicaCount: 1

kafka-ui:
  image:
    registry: ext-repo.ois-bright.ru
  imagePullSecrets:
    - name : "ext-repo.ois-bright.ru"
  service:
    type: NodePort
```

Подготовка файла значений postgresql.yaml

```
global:
  imageRegistry: "ext-repo.ois-bright.ru"
  imagePullSecrets:
    - ext-repo.ois-bright.ru
postgresql:
  image:
    tag: 14.8.0-debian-11-r4
  auth:
    postgresPassword: "secretpwd"
    username: "ipm"
    password: "ipm"
    database: "ipm"
  primary:
    persistence:
      storageClass: "local-path"
      size: 10Gi
    service:
      type: NodePort
      port: 30543
```

Подготовка файла значений ois-redis.yaml

```
# Default values for redis.
redis:
```

```
    global:
      imageRegistry: "ext-repo.ois-bright.ru"
      imagePullSecrets:
        - name: ext-repo.ois-bright.ru
    envPlain:
      ALLOW_EMPTY_PASSWORD: "no"
      REDIS_TLS_ENABLED: "no"
      REDIS_PORT: "6379"
    auth:
      password: "BP38oC3GXT"
    master:
      resources:
        limits:
          cpu: '1'
          memory: 1Gi
        requests:
          cpu: 100m
          memory: 128Mi
```

Подготовка файла значений ois-platform.yaml

```
  global:
    repositoryHost: ext-repo.ois-bright.ru
    imagePullSecrets:
      - name: ext-repo.ois-bright.ru
    routingDomainName: ""
    logstashAddress: null
    logstashAddressSyslog: null
    apiGatewayHost: <server_hostname>
    oauth2Provider:
      clientId: ois-web-core
      clientName: DuoTech
      clientSecret: 5d1ca098-ab6e-4007-9b8d-7b572671e240
    appSettingsTitle: DUO TECH IMA
    mainDataSource: "Host=postgresql.common-
applications;Port=5432;Database=ipm;Username=ipm;Password=ipm;Pooling=true;Appli
cation Name=OIS IPM;"

  ois-platform-api-gateway:
    enabled: true
    core:
      envPlain:
        OPENIDC_INTERNAL_URL: "http://ois-platform-hub.{{ .Release.Namespace
}}.svc.cluster.local"
        OPENIDC_DISCOVERY: "http://{{ .Values.global.apiGatewayHost
}}:31443/hub/.well-known/openid-configuration"
      containerSecurityContext:
```

```
          enabled: false
          readOnlyRootFilesystem: false
        service:
          type: NodePort
          nodePort: 31443
        ingress:
          enabled: false

    ois-platform-web-core:
      enabled: true
      back:
        envPlain:
          ASPNETCORE_ENVIRONMENT: Postgres
          ConnectionStrings__Entities: "{{ .Values.global.mainDataSource }}"
          AppSettings__MetaModelDirectory: BasePg.Base
          AppSettings__appShortName: DUO TECH
          AppSettings__title: "{{ .Values.global.appSettingsTitle }}"
          AppSettings__LogoText: "© «OIS-BRIGHT»"
          AppSettings__authFooText: "© «OIS-BRIGHT»"
          AppSettings__logoSolution: <svg width="24" height="24" viewBox="0 0
24 24" fill="none" xmlns="http://www.w3.org/2000/svg"><path d="M20.5 3L20.34
3.03L15 5.1L9 3L3.36 4.9C3.15 4.97 3 5.15 3 5.38V20.5C3 20.6326 3.05268 20.7598
3.14645 20.8536C3.24021 20.9473 3.36739 21 3.5 21L3.66 20.97L9 18.9L15 21L20.64
19.1C20.85 19.03 21 18.85 21 18.62V3.5C21 3.36739 20.9473 3.24021 20.8536
3.14645C20.7598 3.05268 20.6326 3 20.5 3ZM10 5.47L14 6.87V18.53L10 17.13V5.47ZM5
6.46L8 5.45V17.15L5 18.31V6.46ZM19 17.54L16 18.55V6.86L19 5.7V17.54Z"
fill="#6A6A6A"/></svg>
          AppSettings__logoCustomer: <svg xmlns="http://www.w3.org/2000/svg"
viewBox="-5 -1.5 24 24" width="24" height="24" fill="none"><path fill-
rule="ovenodd" fill="#6A6A6A" d="M7 .565c4.667 6.09 7 10.423 7 13a7 7 0 1 1-14
0c0-2.577 2.333-6.91 7-13z"/></svg>
          AppSettings__HiddenModules: "empty-module"
          AppSettings__HelpRemoteUrl: "http://{{ .Values.global.cdnReleaseName
}}-help-guide.{{ .Release.Namespace }}.svc.cluster.local/"
          AppSettings__homeAppId: 30004
          StartInitialization__Enabled: true
          StartInitialization__SingleRun: true
          Security__OpenIdConnect__Authority: "http://{{
.Values.global.apiGatewayHost }}:31443/hub"
        containerSecurityContext:
          enabled: true
          readOnlyRootFilesystem: true
        ingress:
          enabled: false

    ois-platform-convert:
      enabled: true
```

```
      ois-convert:
        containerSecurityContext:
          enabled: true
          readOnlyRootFilesystem: true
        envPlain:
          JAVA_OPTS: -Xms256m -Xmx2048m -Dreport.server.logging.level=INFO  -
Dreport.server.logging.namespace={{ .Release.Namespace }} -Dserver.tomcat.max-
http-post-size=52428800 -Djava.io.tmpdir=/preserved-data/
        pvc:
          enabled: true

    ois-platform-report:
      enabled: true
      ois-report:
        envPlain:
          JAVA_OPTS: -Xms256m -Xmx2048m -Dreport.server.logging.level=INFO  -
Dreport.server.logging.namespace={{ .Release.Namespace }} -Dserver.tomcat.max-
http-post-size=52428800 -Djava.io.tmpdir=/preserved-data/
        containerSecurityContext:
          enabled: true
          readOnlyRootFilesystem: true
        pvc:
          enabled: true

    ois-platform-hub:
      enabled: true
      hub:
        envPlain:
          oishub_database: "Postgres"
          oishub_database_url: "jdbc:postgresql://postgresql.common-
applications:5432/ipm"
          oishub_application_address: "http://{{ .Values.global.apiGatewayHost
}}:31443/hub"
          oishub_audit_cef_enabled: true
          HUB_OPTS: "-Duser.country=RU -Duser.language=ru -Dserver.max-http-
header-size=8000 -Djava.io.tmpdir=/preserved-data/"
        notOverrideEnvSecrets:
          oishub_database_username: ipm
          oishub_database_password: ipm
        volumeSecret:
          enabled: false
        containerSecurityContext:
          enabled: true
          readOnlyRootFilesystem: true
        pvc:
          enabled: true
        ingress:
```

```
      enabled: false

ois-platform-data-discovery:
  enabled: false

ois-platform-scheduler:
  enabled: false
```

Подготовка файла значений ois-cdn.yaml

```
global:
  repositoryHost: ext-repo.ois-bright.ru
  imagePullSecrets:
    - name : "ext-repo.ois-bright.ru"

schemes-mx:
  enabled: false
  containerSecurityContext:
    enabled: true
    readOnlyRootFilesystem: true
  envPlain:
    NGINX_ENVSUBST_OUTPUT_DIR: /opt/nginx/pvc/conf.d
  pvc:
    enabled: true
    mountPath: /opt/nginx/pvc/
    storage: 100Mi
  ingress:
    enabled: false

help-guide:
  enabled: true
  containerSecurityContext:
    enabled: true
    readOnlyRootFilesystem: true
  envPlain:
    NGINX_ENVSUBST_OUTPUT_DIR: /opt/nginx/pvc/conf.d
  pvc:
    enabled: true
    mountPath: /opt/nginx/pvc/
    storage: 100Mi
  ingress:
    enabled: false
```

Подготовка файла значений ois-ipm.yaml

```
global:
  repositoryHost: ext-repo.ois-bright.ru
  imagePullSecrets:
    - name: ext-repo.ois-bright.ru
  routingDomainName: ""
```

```
      apiGatewayHost: <server_hostname>
      oauth2Provider:
        clientId: ois-web-core
      kafkaTopicSuffix: "ipm"
      kafkaHost: "kafka.common-applications:9092"
      redisPassword: "BP38oC3GXT"
      redisConf: "ois-redis-master.common-
applications.svc.cluster.local:6379,ConnectTimeout=20000,SyncTimeout=20000,Async
Timeout=30000,allowAdmin=true"
      mainDataSource: "Host=postgresql.common-
applications;Port=5432;Database=ipm;Username=ipm;Password=ipm;Pooling=true;Appli
cation Name=OIS IPM;"
      mainDataSourceWE: "Host=postgresql.common-
applications;Port=5432;Database=ipm;Username=ipm;Password=ipm;Pooling=false;Appl
ication Name=OIS IPM (well-editor);"
      mainDataSourceWC: "Host=postgresql.common-
applications;Port=5432;Database=ipm;Username=ipm;Password=ipm;Pooling=false;Time
out=300;CommandTimeout=300;Application Name=OIS IPM (well-calc);"
      mainDataSourcePVT: "Host=postgresql.common-
applications;Port=5432;Database=ipm;Username=ipm;Password=ipm;Pooling=false;Appl
ication Name=OIS IPM (pvt-calculator);"

    ipm-gateway:
      enabled: true
      containerSecurityContext:
        enabled: true
        readOnlyRootFilesystem: true
      envPlain:
        KAFKA_TOPIC_SUFFIX: "{{ .Values.global.kafkaTopicSuffix }}"
        Caching__Redis__Password: "{{ .Values.global.redisPassword }}"
        Caching__Redis__Configuration: "{{ .Values.global.redisConf }}"
        Connections__ConnectionStrings__IPM: "{{ .Values.global.mainDataSource
}}"
        Security__OpenIdConnect__Audiences: "ois-web-core;ima-console-
app;fastapi-ext-app"
        Security__OpenIdConnect__Authority: "http://{{
.Values.global.apiGatewayHost }}:31443/hub"
        Licensing__IsConcurrent: true
        Licensing__LicenseServerUrl: "http://ois-platform-hub"
        NLOG_CONSOLE_LEVEL: Debug
      pvc:
        enabled: true
        storage: 100Mi
        mountPath: /tmp/
      resources:
        limits:
          cpu: '1'
```

```
          memory: 1Gi

    messenger:
      enabled: true
      containerSecurityContext:
        enabled: true
        readOnlyRootFilesystem: true
      envPlain:
        Kafka__Host: "{{ .Values.global.kafkaHost }}"
        KAFKA_TOPIC_SUFFIX: "{{ .Values.global.kafkaTopicSuffix }}"
      resources:
        limits:
          cpu: '1'
          memory: 1Gi

    projects:
      enabled: true
      containerSecurityContext:
        enabled: true
        readOnlyRootFilesystem: true
      envPlain:
        Kafka__Host: "{{ .Values.global.kafkaHost }}"
        KAFKA_TOPIC_SUFFIX: "{{ .Values.global.kafkaTopicSuffix }}"
        Connections__ConnectionStrings__IPM: "{{ .Values.global.mainDataSource
}}"
      pvc:
        enabled: true
        storage: 8Gi
        mountPath: /tmp/
      resources:
        limits:
          cpu: '3'
          memory: 3Gi

    data-transfer:
      enabled: true
      containerSecurityContext:
        enabled: true
        readOnlyRootFilesystem: true
      envPlain:
        MaxFileSize: 70971520000
        enabled: true
        Kafka__Host: "kafka.common-applications:9092"
        KAFKA_TOPIC_SUFFIX: "ipm"
        Connections__ConnectionStrings__IPM: "Username=ipm;Password=ipm;{{
.Values.global.mainDataSource }}"
        NLOG_CONSOLE_LEVEL: Debug
```

```
      pvc:
        enabled: true
        storage: 8Gi
        mountPath: /tmp/
      resources:
        limits:
          cpu: '3'
          memory: 6Gi

    reservoirs:
      enabled: true
      containerSecurityContext:
        enabled: true
        readOnlyRootFilesystem: true
      envPlain:
        Kafka__Host: "{{ .Values.global.kafkaHost }}"
        KAFKA_TOPIC_SUFFIX: "{{ .Values.global.kafkaTopicSuffix }}"
        Connections__ConnectionStrings__IPM: "{{ .Values.global.mainDataSource
}}"
      pvc:
        enabled: true
        storage: 100Mi
        mountPath: /tmp/
      resources:
        limits:
          cpu: '1'
          memory: 1Gi

    security:
      enabled: false

    fluidmodels:
      enabled: true
      containerSecurityContext:
        enabled: true
        readOnlyRootFilesystem: true
      envPlain:
        Kafka__Host: "{{ .Values.global.kafkaHost }}"
        KAFKA_TOPIC_SUFFIX: "{{ .Values.global.kafkaTopicSuffix }}"
        Connections__ConnectionStrings__IPM: "{{ .Values.global.mainDataSource
}}"
      resources:
        limits:
          cpu: '1'
          memory: 1Gi

    pvt-calculator:
```

```
      enabled: true
      containerSecurityContext:
        enabled: true
        readOnlyRootFilesystem: true
      envPlain:
        Kafka__Host: "{{ .Values.global.kafkaHost }}"
        KAFKA_TOPIC_SUFFIX: "{{ .Values.global.kafkaTopicSuffix }}"
        Connections__ConnectionStrings__IPM: "{{ .Values.global.mainDataSource
}}"
      resources:
        limits:
          cpu: '1'
          memory: 1Gi

    ipm-ui:
      enabled: true
      containerSecurityContext:
        enabled: true
        readOnlyRootFilesystem: true
      envPlain:
        AppSettings__HiddenSettings: "emptySetting"
        NGINX_ENVSUBST_OUTPUT_DIR: /opt/nginx/pvc/conf.d
      pvc:
        enabled: true
        mountPath: /opt/nginx/pvc/
        storage: 100Mi
      ingress:
        enabled: false
      resources:
        limits:
          cpu: '1'
          memory: 1Gi

    reservoirs-calc:
      enabled: true
      containerSecurityContext:
        enabled: true
        readOnlyRootFilesystem: true
      envPlain:
        IPM_DRIVER_TYPE: "PSQL"
        Kafka__Host: "{{ .Values.global.kafkaHost }}"
        KAFKA_TOPIC_SUFFIX: "{{ .Values.global.kafkaTopicSuffix }}"
        Caching__Redis__Password: "{{ .Values.global.redisPassword }}"
        Caching__Redis__Configuration: "{{ .Values.global.redisConf }}"
        Connections__ConnectionStrings__IPM: "{{ .Values.global.mainDataSource
}}"
      resources:
```

```
        limits:
          cpu: '2'
          memory: 2Gi

    pipe-editor:
      enabled: true
      containerSecurityContext:
        enabled: true
        readOnlyRootFilesystem: true
      envPlain:
        Kafka__Host: "{{ .Values.global.kafkaHost }}"
        KAFKA_TOPIC_SUFFIX: "{{ .Values.global.kafkaTopicSuffix }}"
        Connections__ConnectionStrings__IPM: "{{ .Values.global.mainDataSource
}}"
      pvc:
        enabled: true
        storage: 100Mi
        mountPath: /tmp/
      resources:
        limits:
          cpu: '2'
          memory: 4Gi

    im-calc:
      enabled: true
      containerSecurityContext:
        enabled: true
        readOnlyRootFilesystem: true
      envPlain:
        Kafka__Host: "{{ .Values.global.kafkaHost }}"
        KAFKA_TOPIC_SUFFIX: "{{ .Values.global.kafkaTopicSuffix }}"
        Caching__Redis__Password: "{{ .Values.global.redisPassword }}"
        Caching__Redis__Configuration: "{{ .Values.global.redisConf }}"
        Connections__ConnectionStrings__IPM: "{{ .Values.global.mainDataSource
}}"
        Licensing__LicenseServerUrl: "http://ois-platform-hub"
        OPENIDC_TOKEN_URL: "http://ois-platform-hub/oauth2/token"
        OPENIDC_CLIENT_ID: ois-web-core
        OPENIDC_CLIENT_SECRET: 5d1ca098-ab6e-4007-9b8d-7b572671e240
      resources:
        limits:
          cpu: '2'
          memory: 4Gi

    well-editor:
      enabled: true
      containerSecurityContext:
```

```
        enabled: true
        readOnlyRootFilesystem: true
      envPlain:
        Kafka__Host: "{{ .Values.global.kafkaHost }}"
        KAFKA_TOPIC_SUFFIX: "{{ .Values.global.kafkaTopicSuffix }}"
        Caching__Redis__Password: "{{ .Values.global.redisPassword }}"
        Caching__Redis__Configuration: "{{ .Values.global.redisConf }}"
        Connections__ConnectionStrings__IPM: "{{
.Values.global.mainDataSourceWE }}"
      pvc:
        enabled: true
        storage: 100Mi
        mountPath: /tmp/
      resources:
        limits:
          cpu: '1'
          memory: 1Gi

    well-calc:
      enabled: true
      containerSecurityContext:
        enabled: true
        readOnlyRootFilesystem: true
      envPlain:
        Kafka__Host: "{{ .Values.global.kafkaHost }}"
        KAFKA_TOPIC_SUFFIX: "{{ .Values.global.kafkaTopicSuffix }}"
        Caching__Redis__Password: "{{ .Values.global.redisPassword }}"
        Caching__Redis__Configuration: "{{ .Values.global.redisConf }}"
        Connections__ConnectionStrings__IPM: " {{
.Values.global.mainDataSourceWC }}"
      containerHealthCheck:
        isGrpc: true
        livenessProbe:
          timeoutSeconds: 10
          failureThreshold: 10
        readinessProbe:
          timeoutSeconds: 10
          failureThreshold: 10
      resources:
        limits:
          cpu: '4'
          memory: 4Gi

    equipment:
      enabled: true
      containerSecurityContext:
        enabled: true
```

```
        readOnlyRootFilesystem: true
      envPlain:
        Kafka__Host: "{{ .Values.global.kafkaHost }}"
        KAFKA_TOPIC_SUFFIX: "{{ .Values.global.kafkaTopicSuffix }}"
        Connections__ConnectionStrings__IPM: "{{ .Values.global.mainDataSource
}}"
      resources:
        limits:
          cpu: '1'
          memory: 1Gi

    pyservice:
      enabled: true
      containerSecurityContext:
        enabled: true
        readOnlyRootFilesystem: true
      pvc:
        enabled: true
        storage: 1Gi
      envPlain:
        COMPlus_EnableDiagnostics: 0
        TMPDIR: "/preserved-data"
        XDG_CACHE_HOME: "/preserved-data"
        GUNICORN_TEMP_DIR: "/preserved-data"
        Kafka__Host: "{{ .Values.global.kafkaHost }}"
        Caching__Redis__Password: "{{ .Values.global.redisPassword }}"
        Caching__Redis__Configuration: "{{ .Values.global.redisConf }}"
        IPMPY_CONNECTION_STRING: "{{ .Values.global.mainDataSource }}"
        Licensing__LicenseServerUrl: "http://ois-platform-hub"
        OPENIDC_TOKEN_URL: "http://ois-platform-hub/oauth2/token"
        OPENIDC_CLIENT_ID: ois-web-core
        OPENIDC_CLIENT_SECRET: 5d1ca098-ab6e-4007-9b8d-7b572671e240
      containerHealthCheck:
        livenessProbe:
          timeoutSeconds: 5
          failureThreshold: 10
        readinessProbe:
          timeoutSeconds: 5
          failureThreshold: 10
      resources:
        limits:
          cpu: '4'
          memory: 4Gi

    pyoptimizeservice:
      enabled: true
      containerSecurityContext:
```

```
      enabled: true
      readOnlyRootFilesystem: true
    pvc:
      enabled: true
      storage: 100Mi
    envPlain:
      TMPDIR: "/preserved-data"
      XDG_CACHE_HOME: "/preserved-data"
      GUNICORN_TEMP_DIR: "/preserved-data"
      Kafka__Host: "{{ .Values.global.kafkaHost }}"
      Caching__Redis__Password: "{{ .Values.global.redisPassword }}"
      Caching__Redis__Configuration: "{{ .Values.global.redisConf }}"
      IPMPY_CONNECTION_STRING: "{{ .Values.global.mainDataSource }}"
      Licensing__LicenseServerUrl: "http://ois-platform-hub"
      OPENIDC_TOKEN_URL: "http://ois-platform-hub/oauth2/token"
      OPENIDC_CLIENT_ID: ois-web-core
      OPENIDC_CLIENT_SECRET: 5d1ca098-ab6e-4007-9b8d-7b572671e240
    resources:
      limits:
        cpu: '4'
        memory: 4Gi
      requests:
        cpu: 500m
        memory: 512Mi
```

3.2.2. Создание Namespace в Kubernetes

```
    sudo kubectl create namespace common-applications

    sudo kubectl create secret docker-registry ext-repo.ois-bright.ru --
docker-server=ext-repo.ois-bright.ru --docker-username=XXXXX --docker-
password='XXXXX' -n common-applications

    sudo kubectl create namespace ois-ipm

    sudo kubectl create secret docker-registry ext-repo.ois-bright.ru --
docker-server=ext-repo.ois-bright.ru --docker-username=XXXXX --docker-
password='XXXXX' -n ois-ipm
```

3.2.3. Авторизация в registry ext-repo.ois-bright.ru

```
    helm registry login ext-repo.ois-bright.ru
    Username=XXXXX
    Password=XXXXX
```

3.2.4. Экспорт конфигурации Kubernetes для helm

В случае использования k3s выполнить экспорт конфигурации для использования helm.

```
export KUBECONFIG=/etc/rancher/k3s/k3s.yaml
sudo chmod 755 /etc/rancher/k3s/k3s.yaml
```

3.2.5. Скачивание helm-charts из гит репозитория

Начать клонирование гит репозитория.

```
sudo mkdir -p /opt/GIT/GIT-DATE
cd /opt/GIT/GIT-DATE
sudo git clone https://ext-git.ois-bright.ru/GPN/imaimport.git
```

3.2.6. Перенос дополнительных helm-charts.

Получить локальные helm-charts и перенести их в /opt/configs

```
sudo mkdir -p /opt/configs
cp -r <path_to_source>/* /opt/configs/
```

3.2.7. Изменение значения в чарте ois-platform для redirectUris

Изменить в `/opt/GIT/GIT-DATE/imaimport/redos/ois-platform`

Значение в параметре "redirectUris":

```
  "redirectUris": [
    "http:// <server_hostname>:31443/redirect_uri"
```

3.2.8. Установку kafka, postgresql и redis

```
    helm upgrade --install --values kafka.yaml kafka /opt/configs/helm-charts/kafka -n common-applications

    helm upgrade --install --values postgresql.yaml postgresql /opt/configs/helm-charts/postgresql -n common-applications

    helm upgrade --install --values ois-redis.yaml ois-redis /opt/GIT/GIT-DATE/imaimport/ois-redis -n common-applications
```

Проверить, что поды запустились без ошибок:

```
    kubectl get pod -n common-applications
```

Результат должен быть аналогичным:

```
NAME                                READY   STATUS
kafka-0                             1/1     Running
kafka-kafka-ui-7c874b8c4d-mpx8m     1/1     Running
kafka-zookeeper-0                   1/1     Running
ois-redis-master-0                  1/1     Running
postgresql-0                        1/1     Running
```

3.2.9. Установка сервисов Системы DUO TECH IMA

```
    helm upgrade --install --values /opt/configs/helm-values/ois-platform.yaml ois-platform /opt/GIT/GIT-DATE/imaimport/redos/ois-platform -n ois-ipm

    helm upgrade --install --values /opt/configs/helm-values/ois-ipm.yaml ois-ipm /opt/GIT/GIT-DATE/imaimport/redos/ois-ipm -n ois-ipm

    helm upgrade --install --values /opt/configs/helm-values/ois-cdn.yaml ois-cdn /opt/GIT/GIT-DATE/imaimport/redos/ois-cdn -n ois-ipm
```

Запуск подов может занять до 10 минут. Отследить запуск можно командой:

```
    kubectl get pod -n ois-ipm
```

Результат должен быть аналогичный:

NAME	READY	STATUS
ois-cdn-help-guide-7cb5dd7747-tv2h9	1/1	Running
ois-ipm-data-transfer-759fb88589-nhx5f	1/1	Running
ois-ipm-equipment-56578cd8d-8zws2	1/1	Running
ois-ipm-fluidmodels-78dc747dbd-srpqq	1/1	Running
ois-ipm-im-calc-78585b494-rgd7p	1/1	Running
ois-ipm-ipm-gateway-d64d75b6c-99j9f	1/1	Running
ois-ipm-ipm-ui-5db964c4bd-5gtlv	1/1	Running
ois-ipm-messenger-849d7846c4-txd8n	1/1	Running
ois-ipm-pipe-editor-54d9bfb669-27bz9	1/1	Running
ois-ipm-projects-788dff5cb5-72klv	1/1	Running
ois-ipm-pvt-calculator-d495db858-5zhph	1/1	Running
ois-ipm-pyoptimizeservice-587bf4d686-tb52b	1/1	Running
ois-ipm-pyservice-748ff4c9bf-c5rqh	1/1	Running
ois-ipm-reservoirs-5c584d5fff-gc4ck	1/1	Running
ois-ipm-reservoirs-calc-5bbfc5b498-2w8mc	1/1	Running
ois-ipm-well-calc-84849ddc68-zgkm8	1/1	Running
ois-ipm-well-editor-d8bc94d68-q9nkm	1/1	Running
ois-platform-api-gateway-core-58bcf998c6-b2xjg	1/1	Running
ois-platform-convert-59d8cb7567-b5lxv	1/1	Running
ois-platform-hub-75684f947d-lb5v9	1/1	Running
ois-platform-report-fcb5cdf67-7v7jh	1/1	Running
ois-platform-web-core-back-69fd8bd96b-r4v2b	1/1	Running

3.3. Настройка Web-core-back

Для выполнения настройки web-core-back провести инициализацию БД. Для этого надо зайти в новосозданный ПОД ois-platform-web-core-back и выполнить команду.

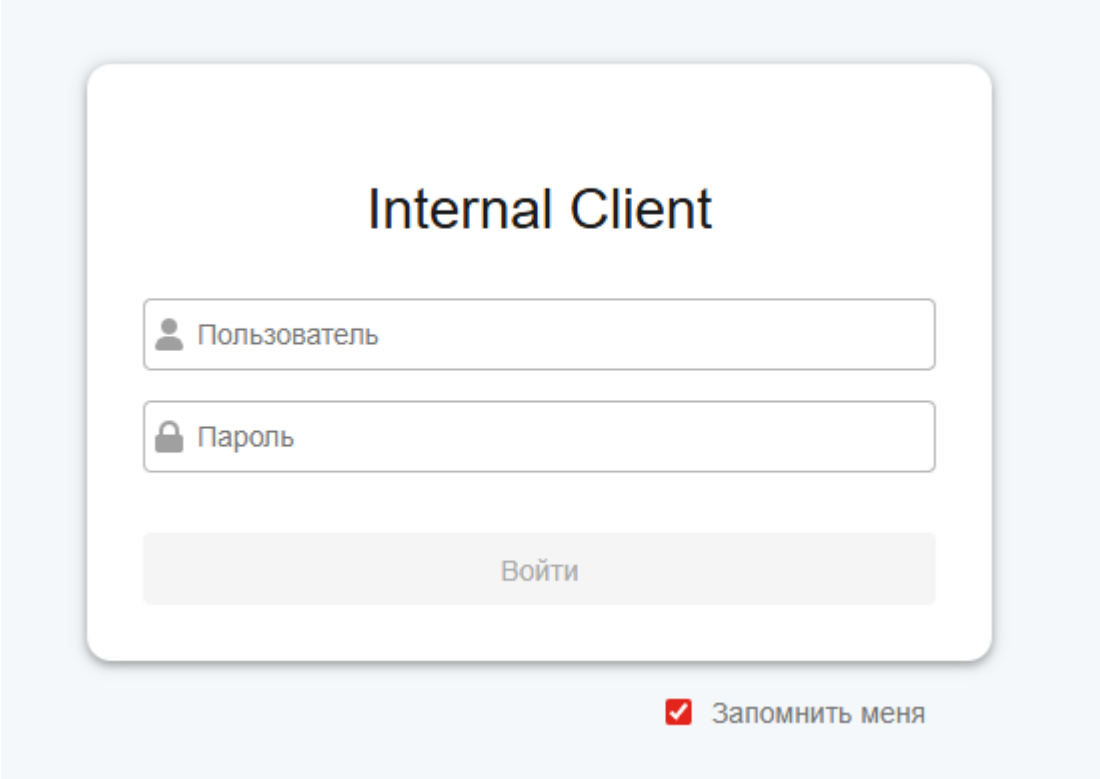
```
kubectl exec -it <ИМЯ_ПОДА_OIS_PLATFORM_WEB_CORE> -n ois-ipm -- curl -i
http://localhost:8080/updatesilent
```

3.4. Настройка ПО из GUI

Выполнить авторизацию из браузера по адресу.

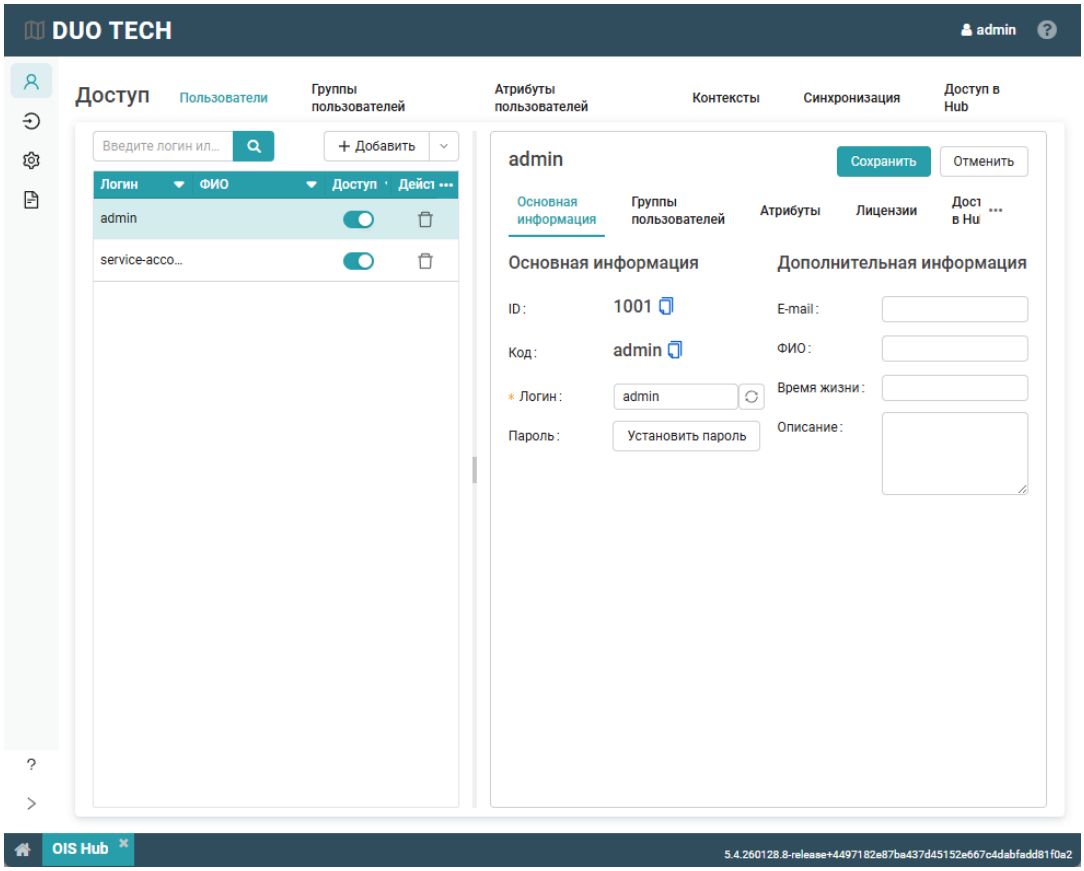
http:// **<server_hostname>:31443**

Для первоначальной настройки, доступ возможен по пользователю Admin с паролем admin.



Далее надо сменить пароль для пользователя admin.

Нажимаем Администрирование – OIS hub.



3.4.1. Добавление приложений в Реестр Приложений

Добавляем приложения для использования.

Администрирование – реестр приложений.

Нажимаем Добавить. И создаём приложение IPM

```
Тип приложения : Module Federation

Код WMF: ois_ipm_web

Название: IPM

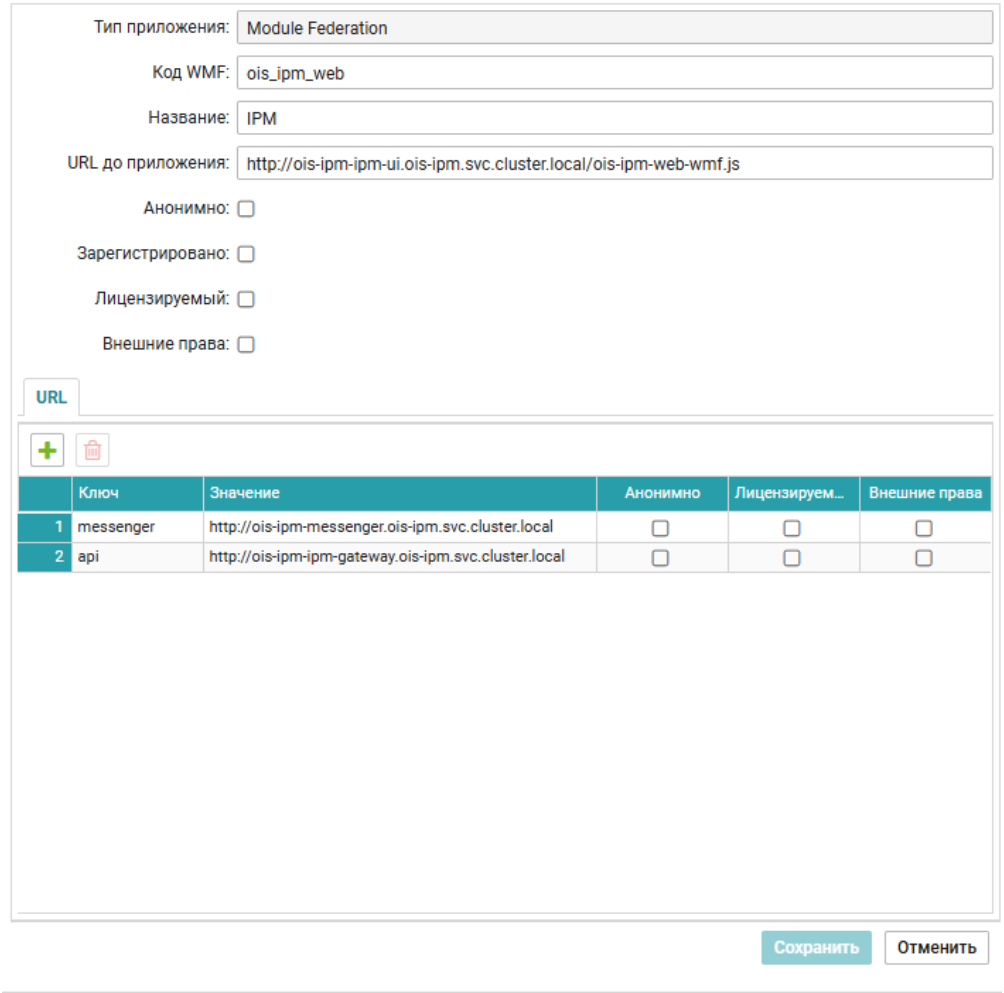
URL до приложения: http://ois-ipm-ipm-ui.ois-ipm.svc.cluster.local/ois-ipm-web-wmf.js

Ключ: messenger

Значение : http://ois-ipm-messenger.ois-ipm.svc.cluster.local

Ключ : api

Значение : http://ois-ipm-ipm-gateway.ois-ipm.svc.cluster.local
```



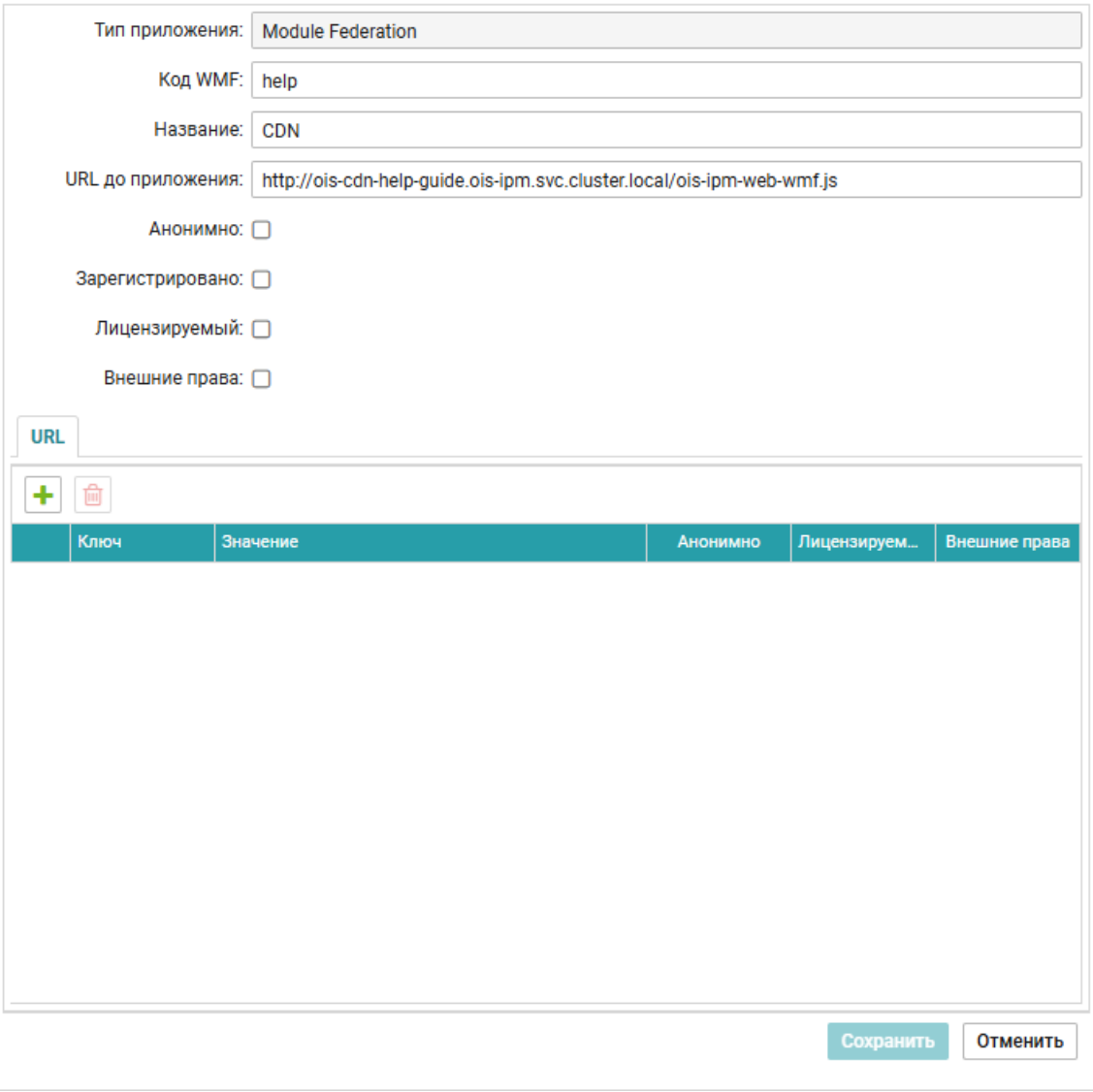
Нажимаем Добавить. И создаём приложение CDN

```
Тип приложения : Module Federation

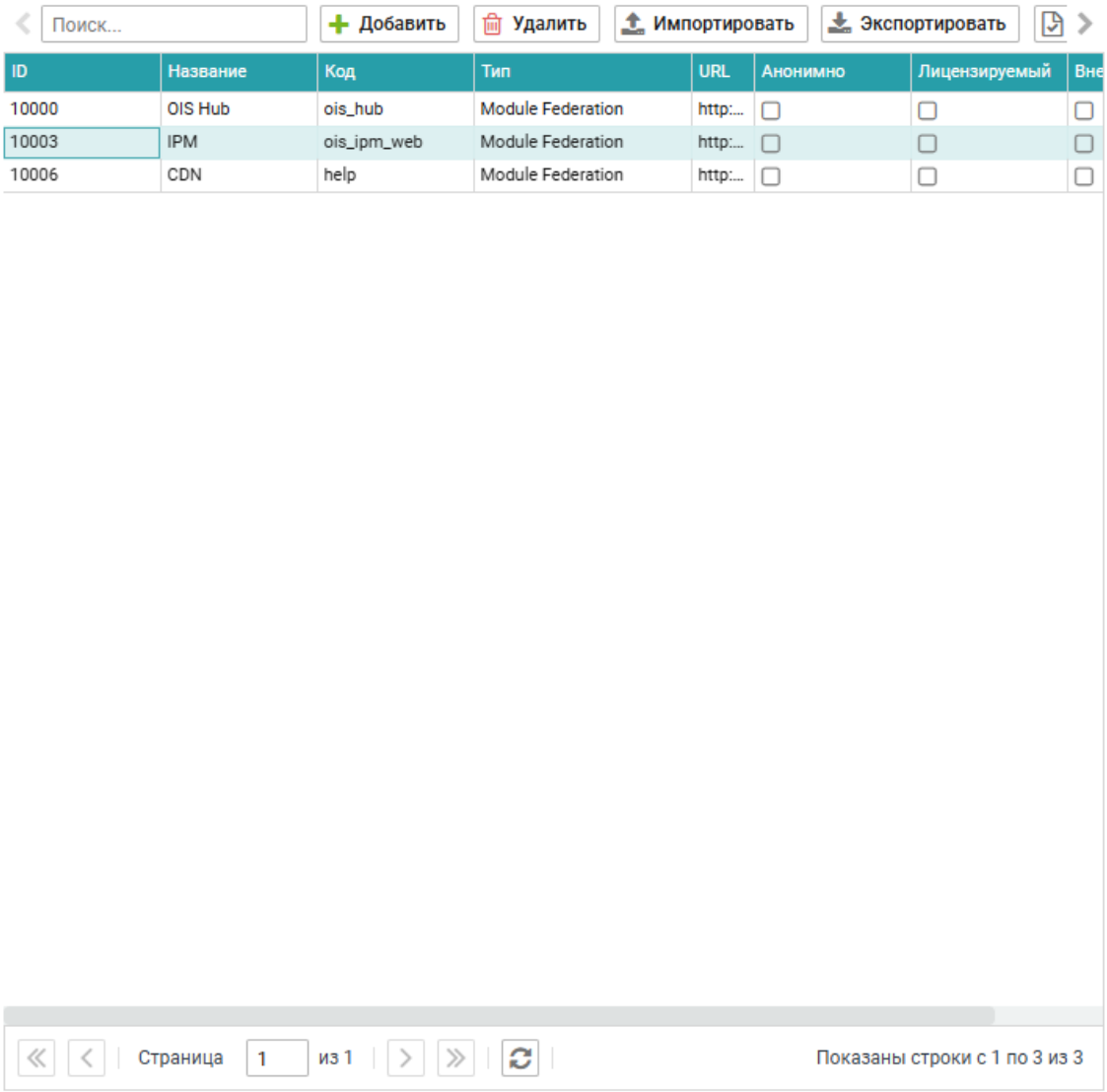
Код WMF: help

Название: CDN

URL до приложения: http://ois-cdn-help-guide.ois-ipm.svc.cluster.local/ois-ipm-web-wmf.js
```



После создания приложений запоминаем их ID, он потребуется для настройки меню приложений.


Поиск...
Добавить
Удалить
Импортировать
Экспортировать
ID
Название
Код
Тип
URL
Анонимно
Лицензируемый
Вне
10000
OIS Hub
ois_hub
Module Federation
http:...
10003
IPM
ois_ipm_web
Module Federation
http:...
10006
CDN
help
Module Federation
http:...
Страница
1
из 1
Показаны строки с 1 по 3 из 3

3.4.2. Настройка меню Приложений

Заходим в меню Администрирование – Приложения

Нажимаем + Главное меню. И добавляем расчетные сервисы DuoTechIma

Приложение из реестра – IPM

Наименование – DuoTechIma

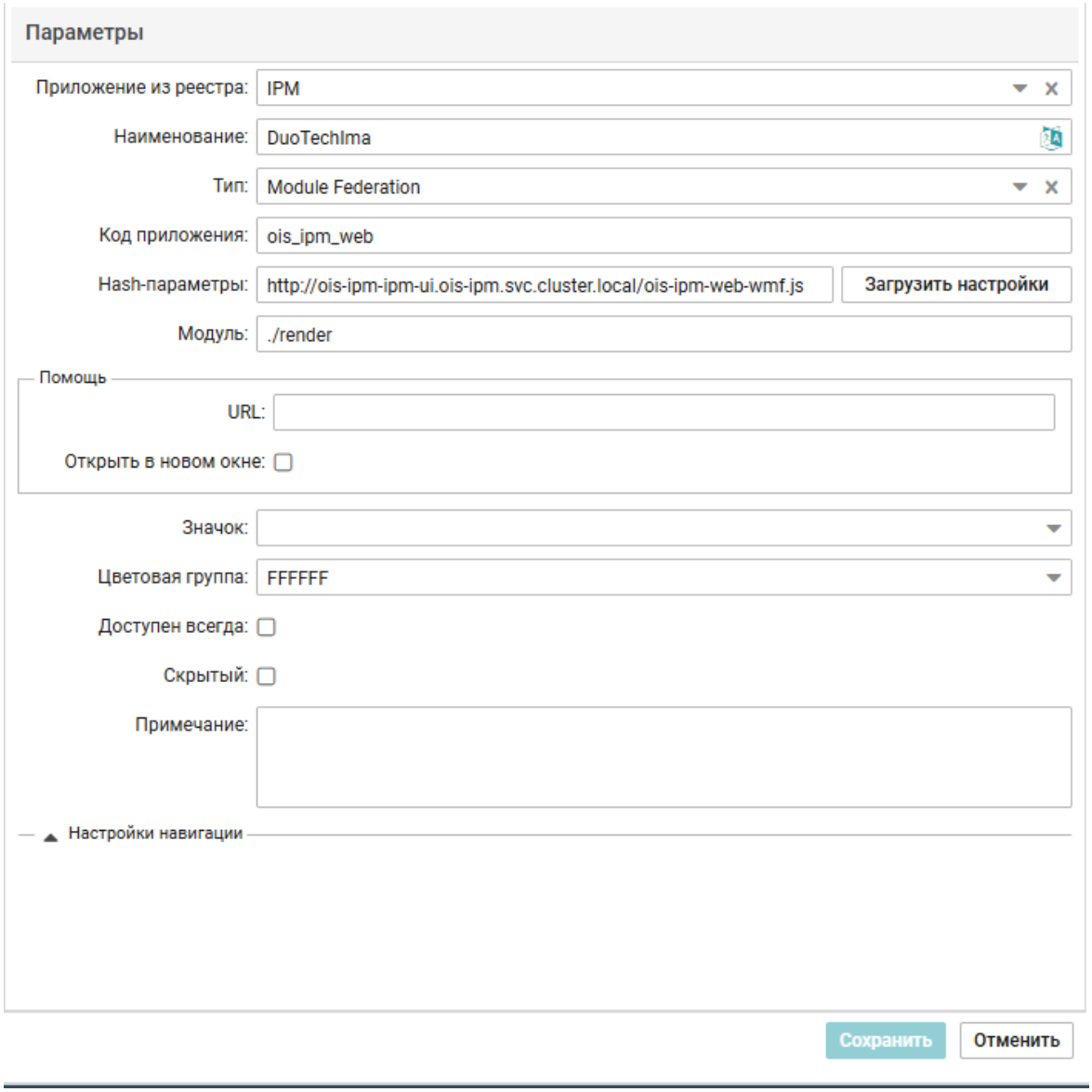
Тип: Module Federation

Код приложения: ois_ipm_web

Hash-параметры:http://ois-ipm-ipm-ui.ois-ipm.svc.cluster.local/ois-ipm-web-wmf.js

Модуль:./render

URL:http://<server_hostname>:31443/modules/<IDМодуляIPM>/ru/html/user/index.html



Нажимаем +Главное меню. И добавляем Каталог Оборудования

```
Приложение из реестра – IPM

Наименование – Каталог Оборудования

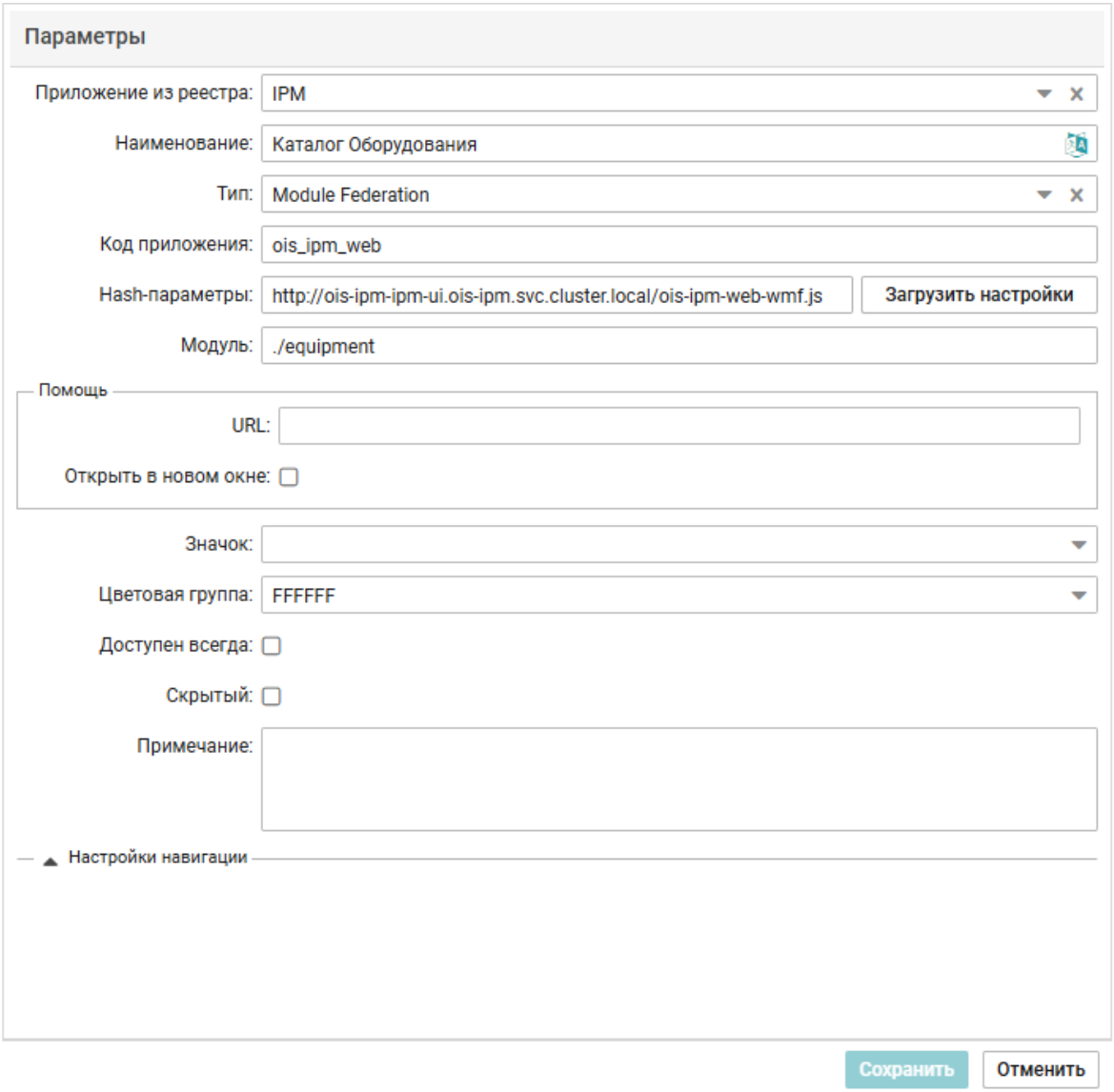
Тип: Module Federation

Код приложения: ois_ipm_web

Hash-параметры:http://ois-ipm-ipm-ui.ois-ipm.svc.cluster.local/ois-ipm-web-
wmf.js

Модуль:./equipment

URL:http://<server_hostname>:31443/modules/<IDМодуляIPM>/ru/html/user/index.html
```



Нажимаем +Главное меню. И добавляем Хранилище Моделей сервисы DuoTechIma

```
Приложение из реестра – IPM

Наименование – Хранилище Моделей

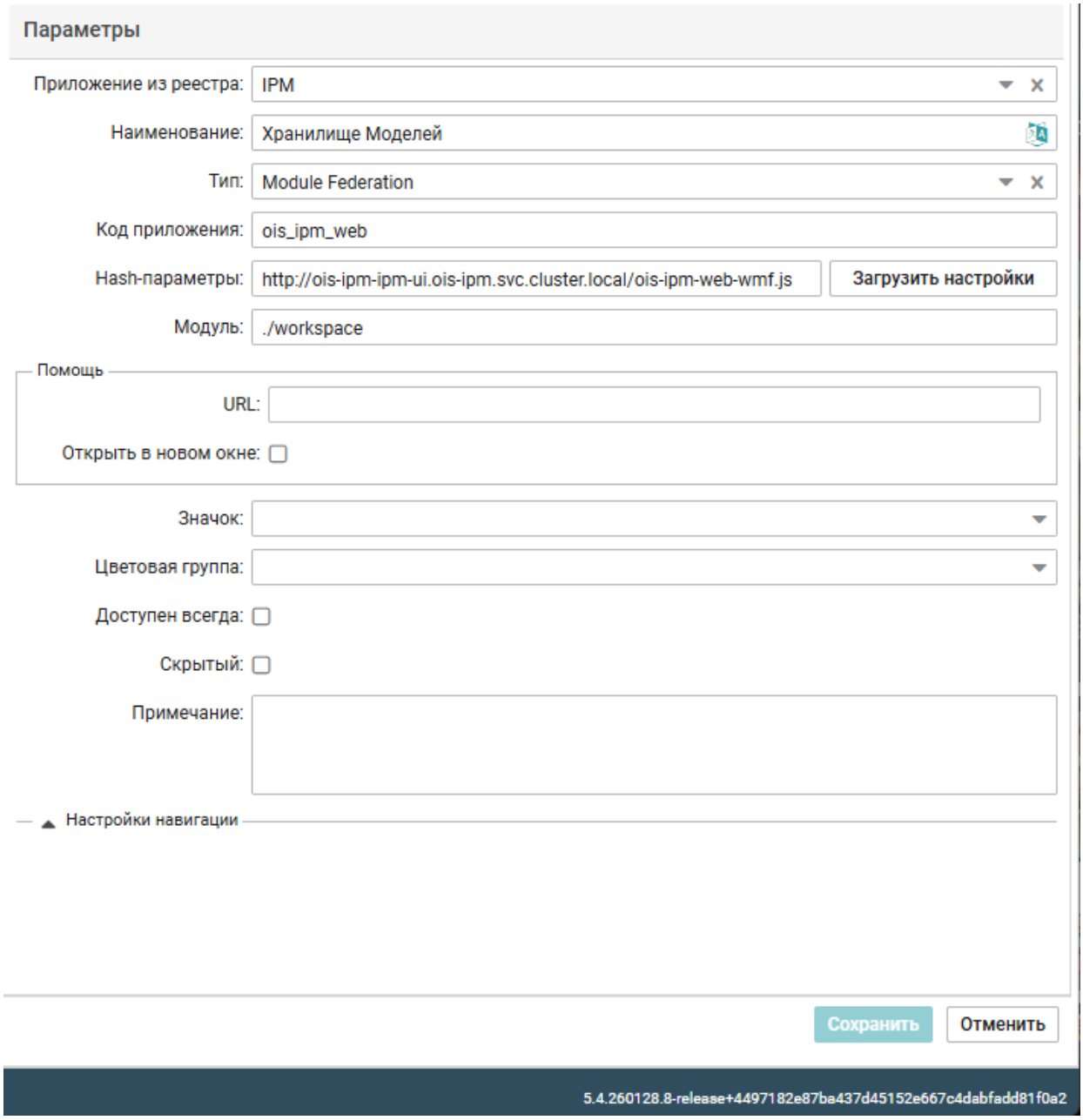
Тип: Module Federation

Код приложения: ois_ipm_web

Hash-параметры:http://ois-ipm-ipm-ui.ois-ipm.svc.cluster.local/ois-ipm-web-
wmf.js

Модуль:./workspace
```

```
URL:http://<server_hostname>:31443/modules/<IDМодуляIPM>/ru/html/user/index.html
```



Нажимаем +Главное меню. И добавляем Документация DUO Tech Server

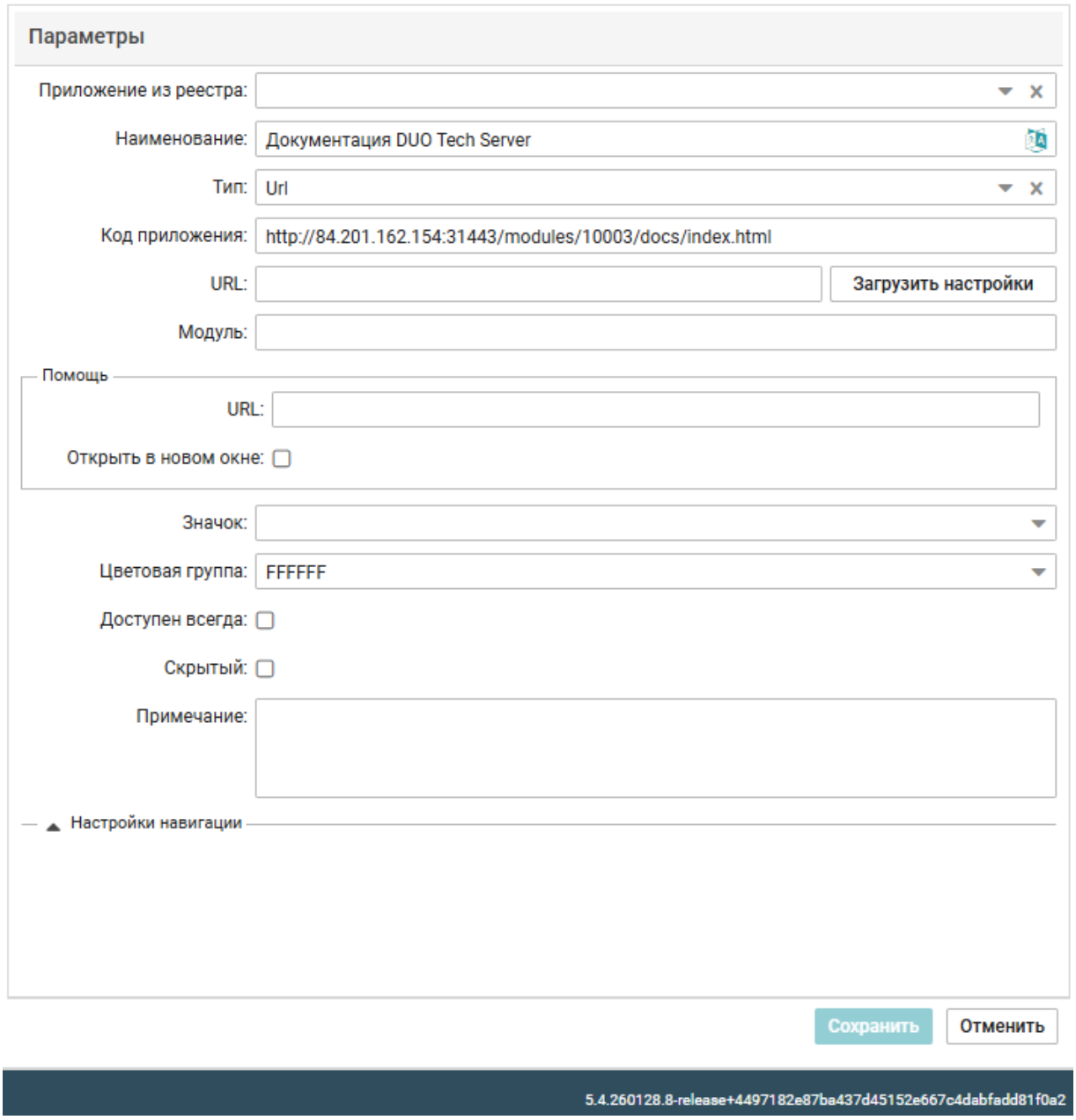
```
Приложение из реестра – IPM

Наименование – Документация DUO Tech Server

Тип: URL

Код приложения: ipm-server-doc

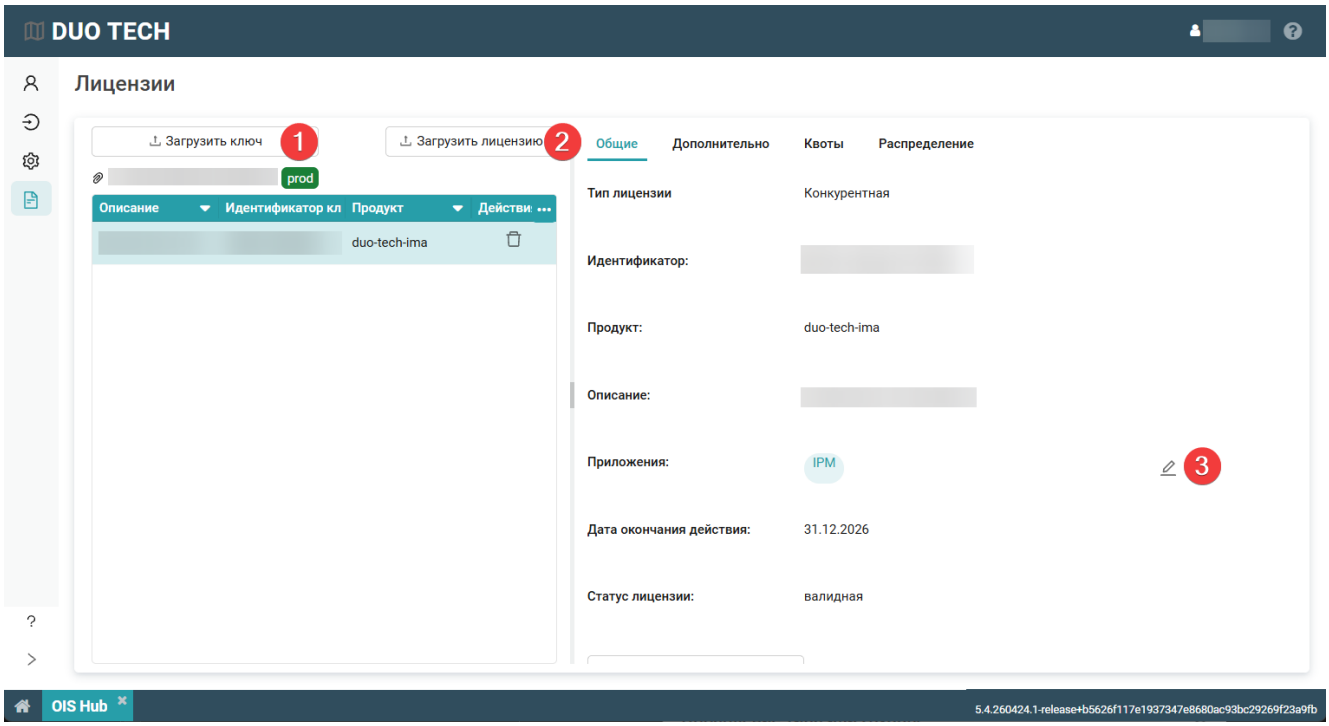
URL:http://<server_hostname>:31443/modules/<IDМодуляIPM>/docs/index.html
```

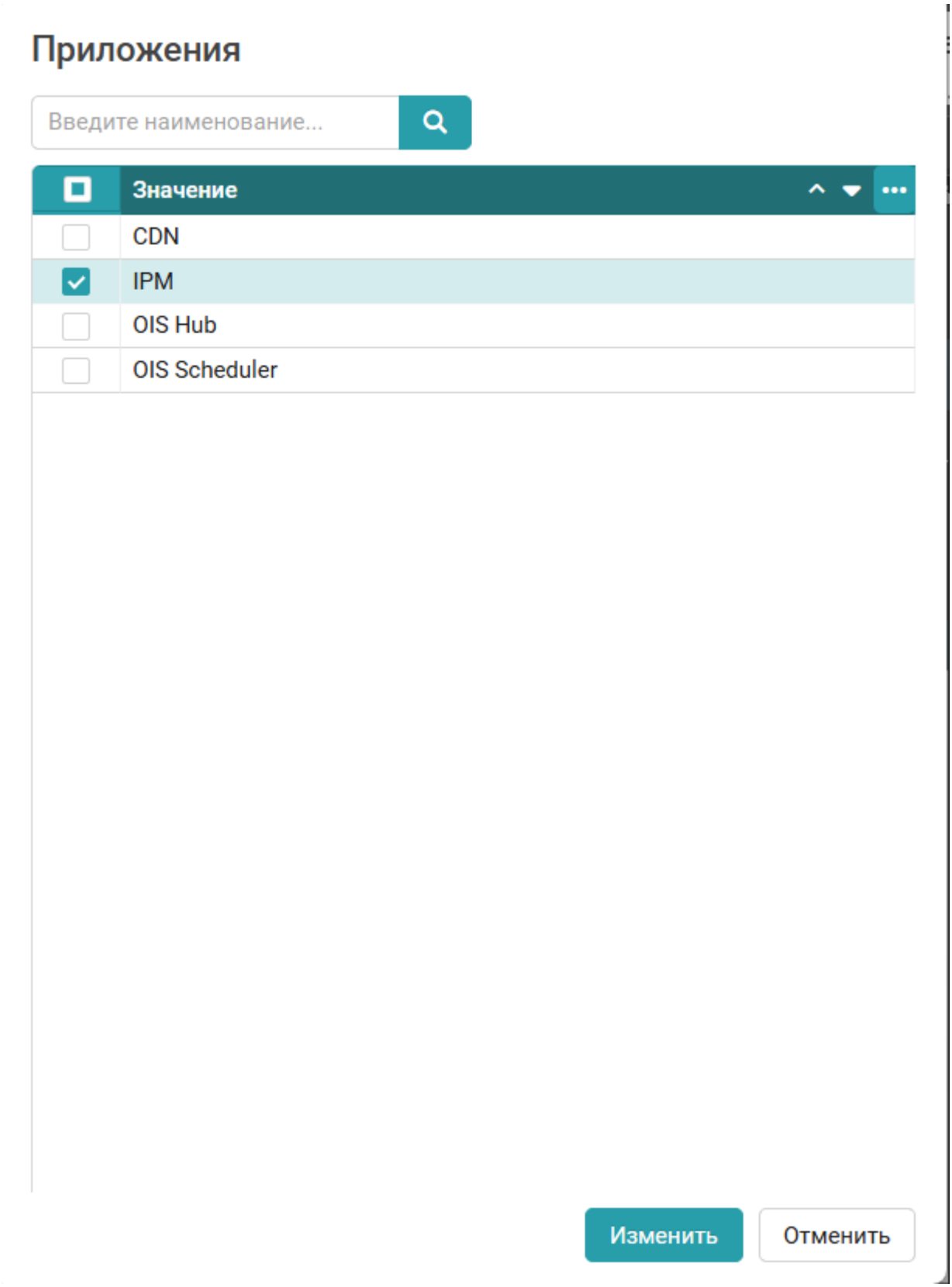

Параметры
Приложение из реестра:
Наименование:
Документация DUO Tech Server
Тип:
Url
Код приложения:
http://84.201.162.154:31443/modules/10003/docs/index.html
URL:
Загрузить настройки
Модуль:
Помощь
URL:
Открыть в новом окне:
Значок:
Цветовая группа:
FFFFFF
Доступен всегда:
Скрытый:
Примечание:
Настройки навигации
Сохранить
Отменить
5.4.260128.8-release+4497182e87ba437d45152e667c4dabfadd81f0a2

3.4.3. Настройка лицензирования.

Заходим в Администрирование – Ois Hub – Лицензии – Загружаем ключ лицензии(1) – Загружаем лицензию(2).

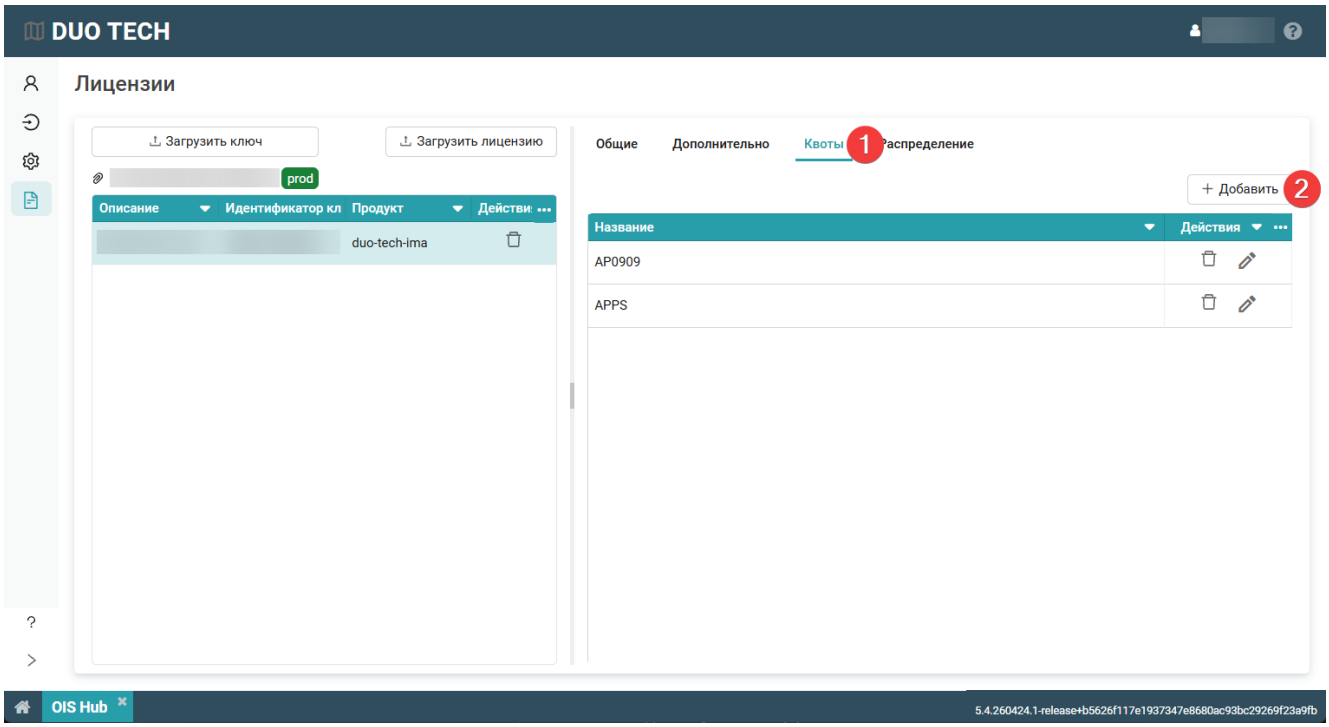
Выбираем приложение из списка Приложений(3) -- IPM



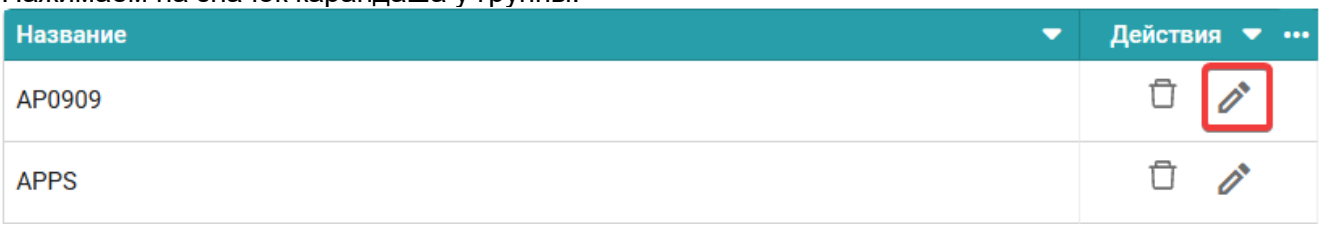


Заходим в пункт Квоты(1)

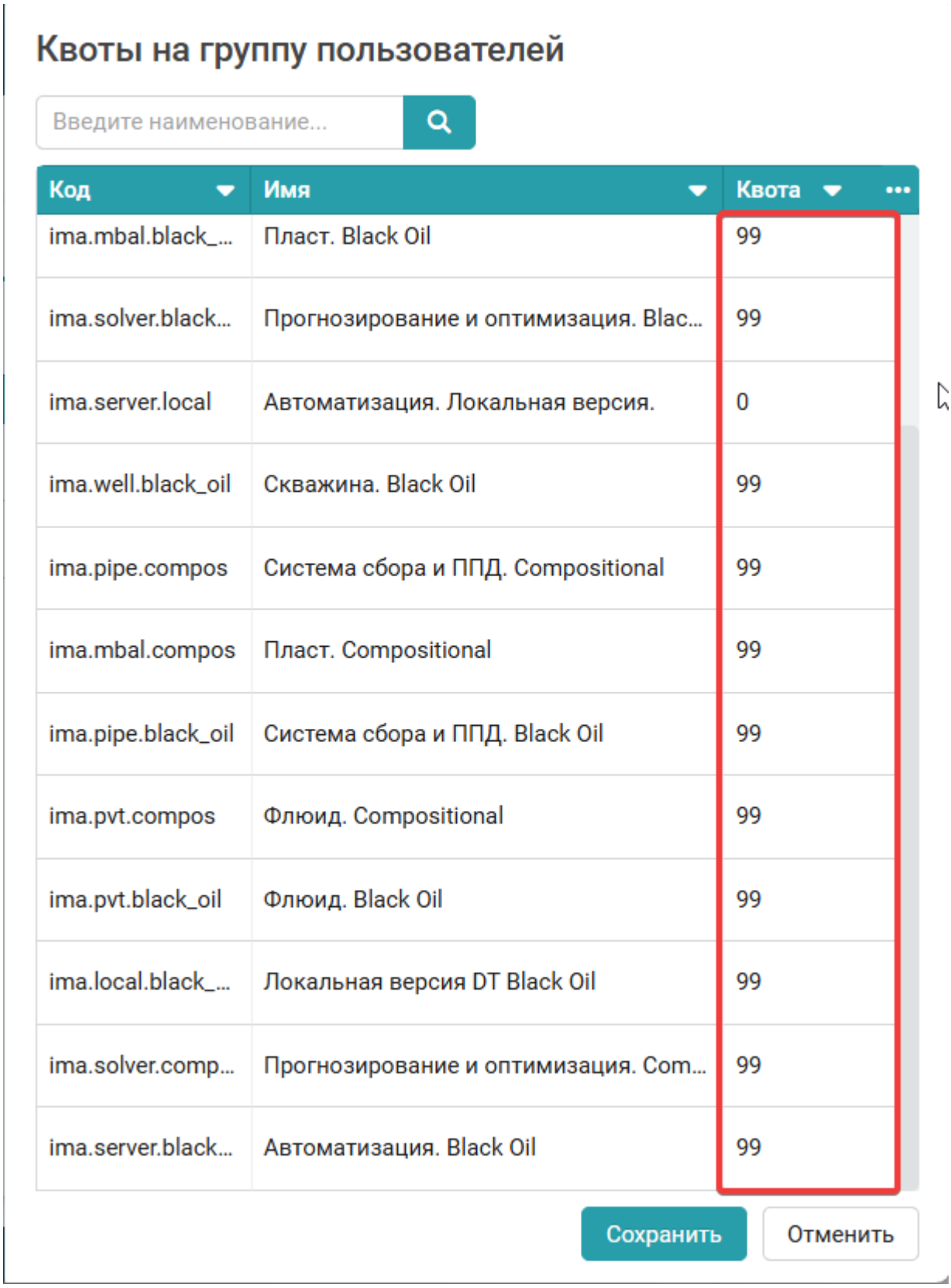
Добавляем группы пользователей(2).



Нажимаем на значок карандаша у группы.



Назначаем квоту на лицензии для группы. Нажимаем сохранить.

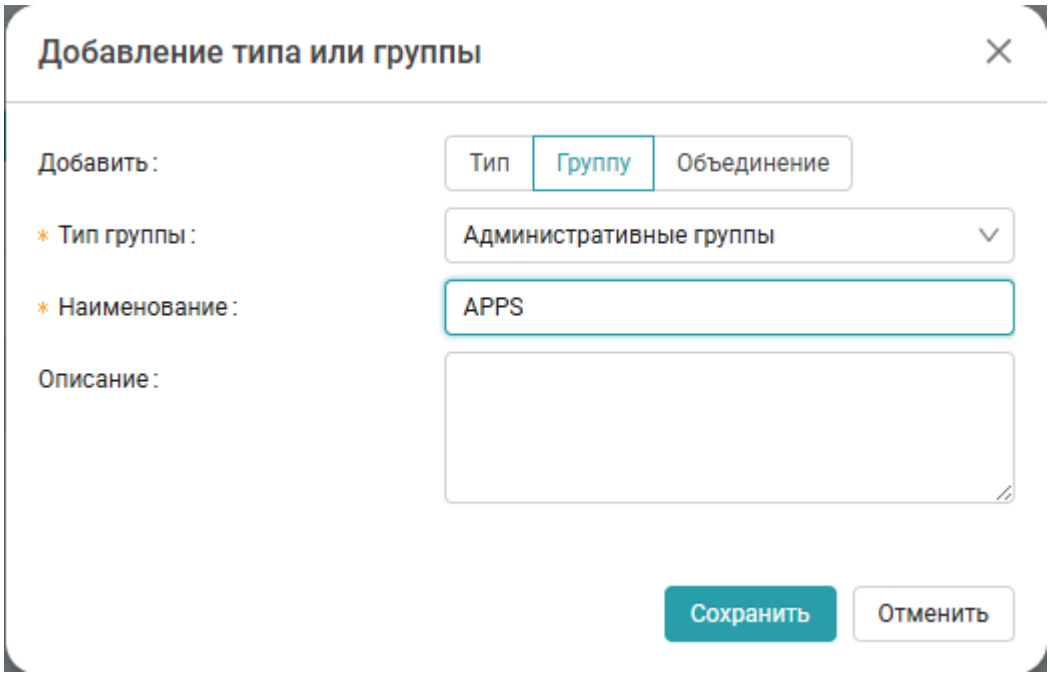


3.4.4. Настройка ролей для пользователей.

Заходим в Администрирование – Ois Hub – Доступ – Группы пользователей.

Нажимаем Добавить

Создаём группу для пользователей. APPS



Заходим в Администрирование – Роли сервисы DuoTechIma

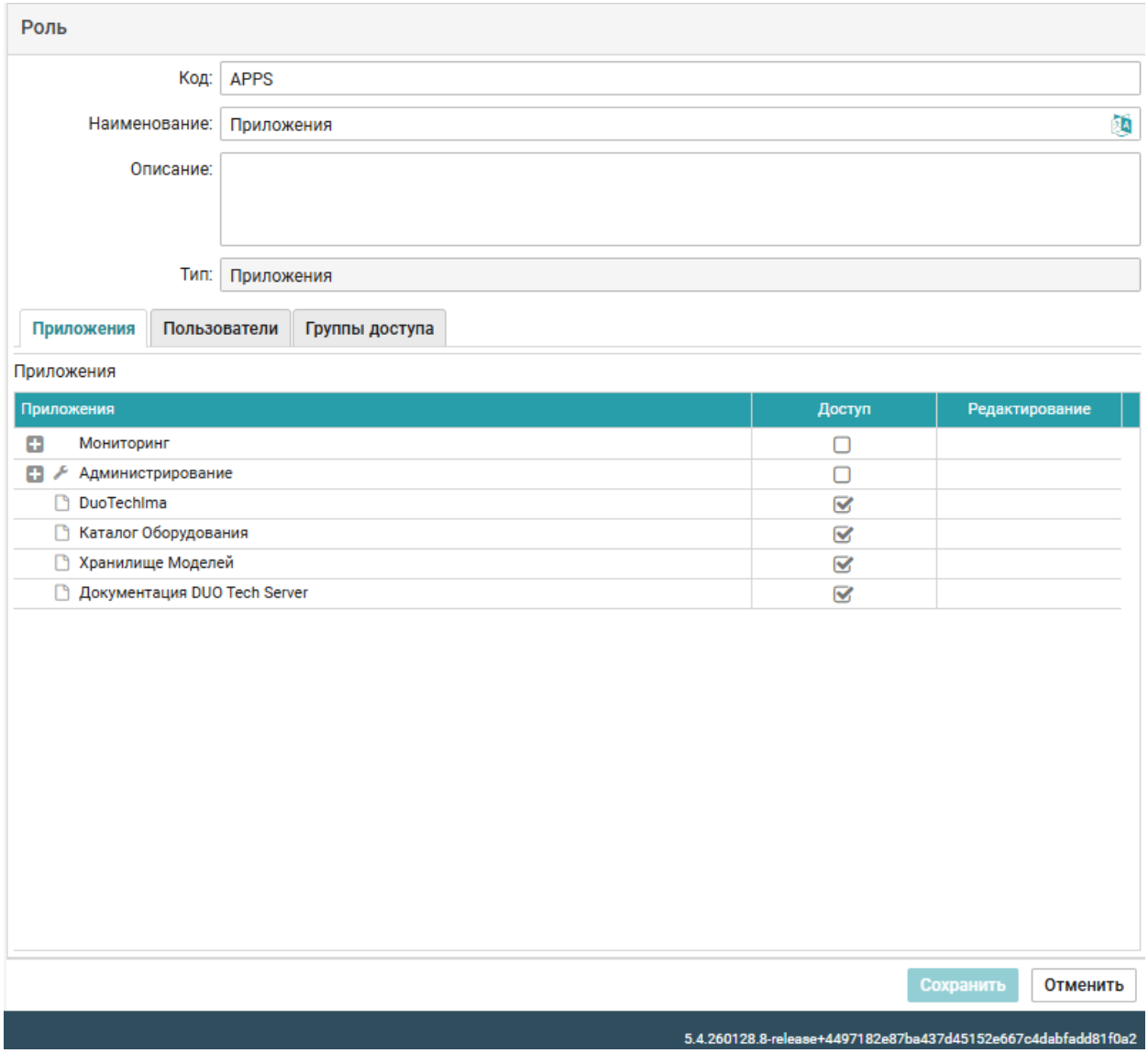
Нажимаем Добавить Роль

```
Код: APPS

Наименование: Приложения

Тип: Приложения
```

Выбираем те приложения которые будут видны этой роли.


Роль
Код:
APPS
Наименование:
Приложения
Описание:
Тип:
Приложения
Приложения
Пользователи
Группы доступа
Приложения
Приложения
Доступ
Редактирование
Мониторинг
Администрирование
DuoTechIma
Каталог Оборудования
Хранилище Моделей
Документация DUO Tech Server
Сохранить
Отменить
5.4.260128.8-release+4497182e87ba437d45152e667c4dabfadd81f0a2