

# Руководство администратора

Всего листов: 129

## Аннотация

Данный документ является частью эксплуатационной документации информационной системы «Сфера: Каталог сервисов», а также модуля «Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам» (ПВСУ), разработанных в рамках семейства продуктов Производственной платформы «Сфера».

Руководство содержит описание действий администратора автоматизированной системы «Сфера: Каталог сервисов». Описаны действия, необходимые для ввода в эксплуатацию, запуска и остановки системы, а также модуля «Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам», возможности администратора по проверке работоспособности и восстановлению системы после сбоев и состав дистрибутива поставки.

# Содержание

Аннотация.....	2
1 Термины и сокращения. ....	7
2 Общие сведения. ....	11
2.1 Назначение и краткая характеристика системы. ....	11
2.2 Цели и задачи. ....	12
2.3 Состав компонентов продукта.....	13
3 Технические требования системы. ....	15
3.1 Требования к аппаратному обеспечению. ....	15
3.2 Требования к установленному программному обеспечению. ....	15
3.2.1 Профили использования системы.....	15
3.2.2 Требования к системе и ко внешнему программному обеспечению. ....	15
3.2.3 Матрица совместимости с интегрируемым продуктам Сферы.....	17
3.3 Требования к конфигурациям программного обеспечения.....	18
3.3.1 Требования к базе данных.....	18
3.3.2 Требования к Kafka. ....	19
3.3.3 Требования к S3. ....	19
3.3.4 Требования к учётным записям (УЗ). ....	19
3.4 Требования к настройкам рабочих станций пользователей. ....	20
3.4.1 Требования к рабочим станциям. ....	20
3.4.2 Требования к настройкам веб-браузера. ....	21
4 Инструкция по установке продукта.....	22
4.1 Руководство по безопасности. ....	22
4.1.1 Рекомендации по безопасности при развёртывании и эксплуатации. ....	22
4.1.1.1 Использование защищённого соединения (HTTPS, TLS/mTLS). ....	22
4.1.1.2 Создание базы данных PostgreSQL. ....	22
4.1.1.3 Подготовка и настройка кластера Kubernetes.....	23
4.1.1.4 Подготовка и настройка балансировщика Load Balancer. ....	23
4.1.1.5 Работа с CRON. ....	23
4.1.2 Управление доступом. ....	25
4.2 Порядок установки продукта.....	25
4.2.1 Установка версий продукта менеджером пакетов Helm. ....	25
4.2.2 Установка версий продукта без использования менеджера пакетов Helm.....	27
4.2.2.1 Установить версии ПО. ....	27
4.2.2.1.1 «Сфера. Каталог сервисов» - добавить сервис catalog. ....	27
4.2.2.1.2 «Сфера. Каталог сервисов» - добавить сервис catalog-ui. ....	29

4.2.2.1.3	«Сфера. Каталог сервисов» - добавить сервис catalog-gateway. ....	30
4.2.2.1.4	«Сфера. Каталог сервисов» - добавить сервис data-consumer-service (опционально).....	32
4.2.3	Действия по установке продукта после наката версий.....	33
4.2.3.1.....	Перевод взаимодействия с «Сфера. Доступы и Роли» в локальный режим. .....	33
4.2.3.2	Обновление конфигурации системы для «Сфера. Каталог сервисов». ....	33
4.2.3.2.1	Создание новой сессии. ....	33
4.2.3.2.2	Загрузка конфигураций. ....	34
4.2.3.3	Настройка процессов и переменных в справочниках системы для «Сфера. Каталог сервисов». 34	
4.2.3.3.1	Настройка справочника «Переменные среды» (environmentVariable) для «Сфера. Каталог сервисов». ....	34
4.2.3.3.2	Настройка справочника «Пользовательские роли» (securityRole) для «Сфера. Каталог сервисов». ....	38
4.2.3.3.3	Настройка справочника «Глобальные переменные» (globalVariables) для «Сфера. Каталог сервисов». ....	40
4.2.3.3.4	Настройка интеграций для «Сфера. Каталог сервисов».....	43
4.2.3.3.5	Настройка шаблонов выгрузки паспортов объектов для «Сфера. Каталог сервисов». 51	
4.2.3.3.6	Настройка производственного календаря для «Сфера. Каталог сервисов». ....	53
4.2.3.4	Обновление ролей для «Сфера. Каталог сервисов».....	54
4.2.3.5	Обновление политик безопасности в системе ДиР.....	55
4.2.3.6	Проверка корректности обновления внутри сессии для «Сфера. Каталог сервисов». 55	
4.2.3.7.....	Перевод взаимодействия с «Сфера. Доступы и Роли» в эксплуатационный режим. .....	56
4.2.3.8	Смоук-тест в рамках сессии для «Сфера. Каталог сервисов». ....	56
4.2.3.9	Применение сессии для «Сфера. Каталог сервисов». ....	57
4.2.3.10	Рестарт поды. ....	57
4.2.4	Переменные окружения «Сфера. Каталог сервисов». ....	57
4.2.4.1	Переменные сервиса catalog.....	57
4.2.4.2	Переменные сервиса data-consumer-service. ....	64
4.2.4.3	Переменные сервиса catalog-gateway. ....	65
4.2.4.4	Переменные сервиса catalog-ui.....	65
4.3	Проверка корректности установки ..... 66	
5	Конфигурирование продукта..... 68	
5.1	Первоначальная настройка. .... 68	
5.1.1	Настройка политик безопасности. .... 68	

5.1.1.1	Обновление политик в ДиР через импорт готовых файлов json. ....	69
5.1.1.2	Обновление политик в ДиР через импорт файлов json, выгруженных вручную. ....	70
5.1.1.3	Каталог сервисов. Обновление политик в ДиР автоматически из UI. ....	71
5.1.1.3.1	Настройка автоматического обновления политик безопасности через UI. ....	72
5.2	Интеграции. ....	73
5.2.1	Общие настройки интеграций. ....	74
5.2.1.1	Настройка нового подключения к Kafka. ....	78
5.2.1.2	Запуск настроенной интеграции на чтение данных из топиков Kafka. ....	79
5.2.1.3	Запуск настроенной интеграции по первичной загрузке данных по REST. ....	79
5.2.2	Интеграция с «Сфера. Ядро». ....	80
5.2.2.1	Встраивание в Главную страницу платформы Сфера. ....	80
5.2.3	Журналирование. ....	80
5.2.3.1	Уровни логирования. ....	81
5.2.3.2	Рекомендации. ....	81
5.2.4	Аудит. ....	81
5.2.4.1	Хранение событий аудита. ....	81
5.2.4.2	Список событий аудита. ....	82
5.2.4.3	Структура поля content с деталями события. ....	82
5.2.4.4	Настройка аудита с http-вызовом. ....	83
5.2.5	Мониторинг. ....	84
5.2.5.1	Мониторинг сервисов. ....	85
5.2.6	Трассировка. ....	96
5.3	Миграция данных посредством импорта .xlsx – «Сфера. Каталог сервисов». ....	96
5.3.1	Миграция справочников. ....	96
5.3.2	Миграция сервисов и цепочек. ....	97
5.3.3	Миграция заявок сервисов и цепочек. ....	100
5.3.4	Миграция связей. ....	104
5.3.5	Миграция заявок связей. ....	106
6	Администрирование продукта. ....	111
6.1	Управление пользователями и доступами. ....	111
6.2	Настройка справочников. ....	111
6.3	Настройка рабочих областей. ....	112
6.4	Управление конфигурациями для «Сфера. Каталог сервисов». ....	112
6.4.1	Создание сессии. ....	113
6.4.2	Применение сессии. ....	114
6.4.3	Отчёт об ошибках. ....	115

6.4.4	Экспорт конфигурации. ....	116
6.4.4.1	Экспорт конфигураций из сессии. ....	116
6.4.4.2	Экспорт конфигураций из тенанта. ....	118
6.4.5	Импорт конфигурации в сессию. ....	119
6.4.6	Полное восстановление конфигурации системы из снапшота. ....	122
6.4.7	Отмена изменений в рамках сессии. ....	123
7	Рекомендации по обеспечению надёжности и восстановлению работы продукта. ....	125
7.1	Диагностика продукта. ....	125
7.2	Резервное копирование. ....	126
7.3	Восстановление из резервной копии. ....	126
7.4	Устранение проблем. ....	126
7.4.1	Критерии аварийного события. ....	126
7.4.2	Порядок действий при выявлении аварийного события. ....	127
7.4.3	Типовые аварийные ситуации и пути их решения. ....	128
7.4.3.1	Аварийные ситуации системы оркестровки контейнеризированных приложений. 128	
7.4.3.2	Аварийные ситуации СУБД. ....	128
7.4.3.3	Аварийные ситуации сервера непрерывной интеграции. ....	129

# 1 Термины и сокращения.

Таблица 1-1 — Перечень сокращений

Сокращение/аббревиатура	Значение
API	Application Programming Interface - набор способов и правил, по которым различные программы общаются между собой и обмениваются данными
CDP	Continuous Deployment – Непрерывное развёртывание
FQDN	Fully Qualified Domain Name – Полностью определённое имя домена (имя домена, не имеющее неоднозначности в определении)
ITSM	IT Service Management – внедрение и управление качественными ИТ-услугами, соответствующими потребностям бизнеса за счёт оптимального сочетания процессов, людей и информационных технологий
S3	Открытый протокол от компании Amazon, ставший фактически мировым стандартом хранения больших объёмов данных
S3-бакет	Организационная единица, позволяющая группировать логически связанные данные в S3-совместимом хранилище. Фактически представляет собой "каталог" верхнего уровня с фиксированным именем, в который складывается все файлы конкретного проекта
SAP УВХД	Автоматизированная система, поставляемая компанией SAP, и предназначенная для управления внутрихозяйственной деятельностью организации (планирование и контроль бюджета, управление финансами, распределение расходов и т.п.)
SLA	Service Level Agreement (соглашение об уровне сервиса). Соглашение о том, какие услуги и в какой срок будут предоставляться
SSE	Server-Sent Events (события, посылаемые сервером). Технология отправки уведомлений от сервера к веб-браузеру
UI	User interface – совокупность методов и средств, обеспечивающих передачу информации между пользователем-человеком и программным обеспечением. Чаще всего, говоря о UI, подразумевают визуальную составляющую интерфейса системы (экраны, формы, кнопки и их расположение)
URL	Uniform Resource Locator – Единообразный адрес ресурса
ИС	Информационная система – набор компонентов системного и прикладного ПО и данных, развёрнутый на независимых вычислительных мощностях, со своим независимым жизненным циклом, используемый определённой группой работников в процессе своей профессиональной деятельности и обеспечивающий реализацию бизнес-процесса или его части с использованием средств ИТ
КС	Каталог сервисов

Сокращение/аббревиатура	Значение
Модуль ПВСУ	Модуль «Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам»
ОС	Операционная система
РМ	Расчётный модуль
ТП БР	Технологический процесс Банка России
ТУЗ	Технологическая учётная запись

**Таблица 1-2** — Перечень терминов и определений

Термин	Определение
Kubernetes, K8s	Распространяемое под лицензией Apache 2.0 программное решение с открытым исходным кодом для автоматического распределения произвольного количества контейнеров по произвольному набору серверов или виртуальных машин. Для загрузки образов запускаемых контейнеров Kubernetes использует совместимый с ним реестр образов
Бизнес-процесс	Долгосрочная повторяемая последовательность связанных активностей (шагов), исполняемых сотрудниками или системами, имеющая внутреннее устройство и самостоятельный (неделимый на рассматриваемом уровне представления) результат, представляющий ценность для бизнеса
Жизненный цикл	1). Последовательность стадий и фаз, определяющих динамику реализации и развития процесса или объекта. 2) В системе – самостоятельный объект, описывающий возможную последовательность стадий развития другого объекта.
Заявка	Запрос на изменение объекта (данные полей, связанные атрибуты), редактируемая версия объекта, имеющая собственную статусную модель и жизненный цикл. В ходе согласования и публикации новой заявки основной объект может быть отредактирован
Инцидент	Любое событие, не являющееся частью стандартных операций по предоставлению Сервиса, которое привело к нарушению или снижению качества этого Сервиса
Контейнер	Программный процесс, изолированный от других программных процессов средствами контейнеризации
Нормативный срок	Регламентированный период, установленный в паспорте Сервиса, в течение которого Исполнитель обязан предоставить Клиенту информацию по Обращению или устранить нарушения в работе Сервиса
Образ контейнера	Архив специального формата, содержащий исполняемую программу, а также полный набор зависимостей (операционную систему, системные утилиты, библиотеки и другие данные), необходимых для корректной работы



Термин	Определение
Переход жизненного цикла	Описание правил и ограничений смены статуса объекта с одного на другой; описание совокупности изменений, происходящих с объектом во время этой смены.
Под (Pod)	Набор из одного или нескольких контейнеров, имеющих одинаковый сетевой адрес и запускаемых как единое целое в кластере Kubernetes
Пререквизиты	Предпосылки, необходимые начальные условия
Резиденты сессии	<p>Это пользователи, имеющие доступ к конфигурациям в рамках выбранной сессии.</p> <p>Если пользователь обладает ролью администратора, то он может выполнять пользовательские сценарии, а также сценарии администрирования (сценарии конфигурирования системы) в рамках выбранной сессии.</p> <p>Если пользователь не обладает правами администратора, то он может выполнять только пользовательские сценарии.</p>
Реестр образов (registry)	Специальное централизованное хранилище образов контейнеров, имеющее программный интерфейс (API) и позволяющее загружать, хранить и скачивать образы пользователю и другим программам
Роль	Набор системных и объектных прав пользователя, которые могут быть выданы и отозваны как единое целое, и после добавления этой Роли, могут быть временно активированы и деактивированы во время существования сессии
Сервис	<p>1) Возможная к выявлению и поддающаяся последующей декомпозиции логическая последовательность действий участников процесса, осуществляемая в автоматизированном или автоматическом виде, или по своей сущности требующая автоматизации, и имеющая целью обслуживание клиента для достижения результата, непосредственно связанного с бизнес-потребностью клиента.</p> <p>2) Независимый функциональный модуль, описываемый спецификацией (паспорт сервиса), который реализует от одной до нескольких отдельных, логически завершённых бизнес-операций, сгруппированных в сервис на основании общего функционального контекста</p>
Сессия конфигурирования	Изолированная административная среда, создаваемая для безопасного тестирования и настройки системных конфигураций. В рамках такой сессии администраторы могут вносить изменения в настройки системы и тестировать их работу без влияния на основную функциональную сессию.
Статус жизненного цикла	Стадия развития объекта в жизненном цикле и совокупность описывающих её характеристик
«Сфера. Каталог сервисов», ИС "Каталог сервисов", Каталог сервисов, КС	Система для хранения и поддержания в актуальном состоянии информации о Сервисах и Цепочках сервисов (их атрибутах, характеристиках и связях), предоставляемых заказчиком Системы. Предназначена для работы с Паспортами сервисов и цепочек сервисов и настройки процессов их сопровождения

Термин	Определение
«Сфера. Расчётный модуль SLA», Расчётный модуль, РМ	Программный продукт Платформы Сфера, фактически являющийся одним из модулей Каталога сервисов. Предназначен для калькуляции Параметров SLA
Тенант	Основная рабочая среда, в которой все пользователи продукта выполняют свою функциональную работу. Это стабильная производственная среда, где происходит вся операционная деятельность пользователей без каких-либо административных изменений конфигураций.
Технологический процесс	Регуляторная цепочка сервисов, характеризующая набор взаимосвязанных операций, созданная как отдельный объект. В рамках модуля ПВСУ объект используется для ведения внутри системы Технологических процессов Банка России (ТП БР), их изменений и влияния на сервисы заказчика через создание связей.
Функциональная сессия (слепок системы)	Применённая сессия конфигурирования.
Цепочка сервисов	Последовательность из нескольких сервисов, функционально связанных между собой для решения конкретной бизнес-задачи. Цепочка начинается с сервиса-инициатора и заканчивается завершающим сервисом. Принципы работы в Каталоге сервисов с объектами типа "Сервис" и "Цепочка сервисов" аналогичны

## 2 Общие сведения.

### 2.1 Назначение и краткая характеристика системы.

Продукт «Сфера: Каталог сервисов» является централизованным хранилищем паспортов сервисов и предназначен для обеспечения жизненного цикла сервисов, а именно для:

1. поддержки версионности сервисов;
2. хранения истории изменений сервисов;
3. предоставления метрик качества, относящихся к определённому сервису, другим ИС.

Продукт предназначен для следующих ключевых групп пользователей:

1. Менеджеры процессов и руководители подразделений, участвующие в планировании релизов ИС.
2. Администраторы ИС, участвующие в разработке ПО и обеспечивающие стабильность ИТ-услуг, которым требуется получение данных о сервисах в своей работе.
3. Специалисты, ответственные за ведение справочника сервисов и поддержку паспортов сервисов.

Продукт «Сфера: Каталог сервисов» реализован в виде веб-приложения, работа с которым осуществляется через браузер.

Продукт «Сфера: Каталог сервисов» разработан в рамках семейства продуктов Производственной платформы «Сфера».

Продукт может поставляться в следующих комплектациях:

**Таблица 2-1 – Комплектация**

№	Комплектация	Назначение и функции
1.	Сфера: Каталог сервисов	Базовая функциональность – централизованное хранилище паспортов сервисов для обеспечения жизненного цикла сервисов
2.	* Модуль «Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам»	Дополнительная функциональность автоматизации обмена данными между системой «Сфера: Каталог Сервисов» и внешними системами учёта, обеспечивающего формирование отчётности по предоставляемым ИТ-услугам

\* - разработан и введён в эксплуатацию в виде отдельного нематериального актива

Продукт «Сфера: Каталог сервисов» рекомендовано внедрять в комплекте с модулем «Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам», это позволит обеспечить двустороннюю интеграцию с системами учёта, передачу ключевых параметров сервисов и технологических процессов, а также автоматическое формирование аналитической информации для управленческих и регуляторных целей.

#### **Основные функции системы «Сфера: Каталог сервисов»:**

- внесение, хранение и модификация информации о предоставляемых сервисах;
- классификация и группировка информации о сервисах исходя из запросов отчётности;

- хранение и предоставление другим системам информации о Параметрах качества сервисов;
- настройка автоматизированных процессов работы с жизненным циклом сервиса;
- управление процессами согласования сохраняемых данных о сервисах;
- получение и отправка уведомлений об изменениях в сервисах;
- построение цепочек сервисов для управления комплексными процессами;
- фиксация взаимосвязей между сервисами и визуализация их в виде блок-схем;
- хранение версий сервиса и истории изменений Параметров сервиса;
- администрирование и настройка локальных справочных данных;
- администрирование и настройка файлов пользовательских выгрузок;
- администрирование и настройка пользовательских представлений для работы с каталогом сервисов;
- администрирование и настройка форм объектов каталога сервисов.

**Основные функции модуля «Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам» (ПВСУ):**

- Автоматический обмен данными между «Сфера: Каталог Сервисов» и системами учёта (SAP УВХД, «Сфера: Архитектура», «Сфера: Документы»);
- Получение и обработка данных о договорах из SAP УВХД с последующей автоматической привязкой к соответствующим сервисам;
- Отображение в карточке сервиса статуса проверки чек-листов из «Сфера: Документы» (синий, жёлтый, красный индикаторы);
- Ведение атрибутивного состава и жизненного цикла согласования технологических процессов Банка России (ТП БР);
- Классификация сервисов и их параметров в соответствии с нормативными требованиями и внутренними регламентами;
- Передача информации по ТП БР в систему «Сфера: Архитектура»;
- Логирование изменений конфигурационных единиц (КЕ) с признаком «КА» и хранение истории изменений;
- Обеспечение формирования сводной управленческой отчётности по предоставляемым услугам на основе данных учётных систем.

## 2.2 Цели и задачи.

Главной целью продукта «Сфера: Каталог сервисов» является автоматизация работы с информацией о сервисах, используемых в компании.

Использование ИС «Сфера. Каталог сервисов» позволяет решить следующие задачи:

- Каталогизированный учёт, хранение и управление данными об актуальных сервисах и их параметрах (атрибутах, характеристиках, связях).

- Фиксация изменений в сервисах на всём протяжении их жизненного цикла – от создания до вывода из эксплуатации.
- Проведение коллективного согласования внесённых изменений в данные о сервисах.
- Настройка процессов сопровождения сервисов.

**Целью создания модуля «Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам»** является повышение точности и полноты данных в отчётности, сокращение времени на сбор информации и обеспечение консистентности данных между корпоративными системами. Модуль обеспечивает построение сквозного процесса взаимодействия между «Сфера: Каталог Сервисов» и внешними системами, такими как SAP УВХД, «Сфера: Архитектура», «Сфера: Документы» и др. Основное назначение — консолидация данных об услугах, договорах и процессах для последующего построения управленческой и регуляторной отчётности.

## 2.3 Состав компонентов продукта.

В данном разделе представлены состав компонентов системы «Сфера: Каталог сервисов». Функционал модуля «Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам» к программе для ЭВМ «Сфера: Каталог сервисов» добавляется не установкой дополнительных компонентов, а средствами конфигурирования и настройки интеграций. Процесс описан в разделах **4.2.3.2.2. Загрузка конфигураций.** (см. **Таблица 4-3**) и **4.2.3.3.4. Настройка интеграций для «Сфера. Каталог сервисов».** (см. **Таблица 4-9**).

**Таблица 2-2**— Перечень сервисов поставки системы "Сфера: Каталог сервисов"

№	Наименование компонента	Версия	Описание	Ссылка на дистрибутив
1.	catalog	catalog:1704-release-1.45.0	Серверная часть Каталога сервисов. Модуль бизнес-функциональности информационной системы	<a href="https://docker-ppdl-dist.sfera-t1.ru/ppcs-docker-int/catalog:1704-release-1.45.0">https://docker-ppdl-dist.sfera-t1.ru/ppcs-docker-int/catalog:1704-release-1.45.0</a>
2.	catalog-ui	catalog-ui:527-release-1.45.0	Веб-клиент для Каталога сервисов	<a href="https://docker-ppdl-dist.sfera-t1.ru/ppcs-docker-int/catalog-ui:527-release-1.45.0">https://docker-ppdl-dist.sfera-t1.ru/ppcs-docker-int/catalog-ui:527-release-1.45.0</a>
3.	catalog-gateway	catalog-gateway:29-master	Модуль для маршрутизации запросов к информационной системе (Каталог сервисов), проверки токена и формирования заголовка (header) авторизации	<a href="https://docker-ppdl-dist.sfera-t1.ru/ppcs-docker-int/catalog-gateway:29-master">https://docker-ppdl-dist.sfera-t1.ru/ppcs-docker-int/catalog-gateway:29-master</a>
4.	data-consumer-service	data-consumer-service:36-master	Дополнительная точка интеграции с внешними системами/справочниками	<a href="https://docker-ppdl-dist.sfera-t1.ru/ppcs-docker-int/data-consumer-service:36-master">https://docker-ppdl-dist.sfera-t1.ru/ppcs-docker-int/data-consumer-service:36-master</a>

№	Наименование компонента	Версия	Описание	Ссылка на дистрибутив
5.	ppcs meta chart	ppcs-1.45.0-int.tgz	Мета-чарт для установки версии системы	<a href="https://dist.sferat1.ru/app/ppdl-new/api/repo/helm/ppcs-helm-int/ppcs/ppcs-1.45.0-int.tgz">https://dist.sferat1.ru/app/ppdl-new/api/repo/helm/ppcs-helm-int/ppcs/ppcs-1.45.0-int.tgz</a>

**Примечание:** data-consumer-service на текущий момент является не обязательным сервисом в составе системы, может не устанавливаться и не настраиваться. Сервис является легаси – будет изъят из комплектации сборки в течение нескольких релизов.

## 3 Технические требования системы.

В данном разделе представлены технические требования системы «Сфера: Каталог сервисов», а также модуля «Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам» к программе для ЭВМ «Сфера: Каталог сервисов».

### 3.1 Требования к аппаратному обеспечению.

*Таблица 3-1 – Требования к инфраструктуре*

Требования к инфраструктуре			
	CPU	RAM	HDD
Минимальные требования	x86_64, 1,2 GHz или выше, минимум 2 ядра	DDR3 1400 MHz или выше	HDD
Рекомендованные требования	x86_64, 3,6 GHz или выше, минимум 2 ядра	DDR4 3200 MHz или выше	SSD

### 3.2 Требования к установленному программному обеспечению.

#### 3.2.1 Профили использования системы.

*Таблица 3-2 - Профили использования системы*

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 создаваемых объектов в системе в день	<10	<200	<2 000
Количество импортируемых в систему накопленных исторических данных	<5 000	<100 000	<1 000 000
Интенсивность вызова публичного API	<1 RPS	<3 RPS	<10 RPS

#### 3.2.2 Требования к системе и ко внешнему программному обеспечению.

Раздел содержит рекомендации к системе, сервисам системы и ко внешнему программному обеспечению.

**Внимание!** Указанные параметры носят рекомендательный характер, т. к. не учитывают возможные особенности и требования к инфраструктуре Лицензиата. Индивидуальные параметры могут быть определены по итогам обследования инфраструктуры Лицензиата и проведения ИТ на 2025, ООО «ГК «Иннотех». Все права защищены.

профиле нагрузки Лицензиата, осуществляемые по отдельному договору по запросу Лицензиата.

Рекомендации даны исходя из профиля использования системы, без учёта отказоустойчивости и исходя из однонодовой конфигурации. При необходимости возможно работать с кластеризованными решениями. В этом случае требуется пересчёт инфраструктурных ресурсов.

Запрашиваемые ресурсы микросервисами (request) и ограничение этих ресурсов (limit) рекомендуется указывать одинаковым.

**Таблица 3-3** - Профиль №1 (минимальная инсталляция)

	CPU	RAM	Storage	Количество инстансов	Прочие требования
<b>Kubernetes (K8s)</b> полезная нагрузка (сумма по микросервисам)	4	8 Гб	0	1	
Catalog	2	4 Гб	0		
Catalog UI	1	2 Гб	0		
Catalog gateway	1	2 Гб	0		
<b>PostgreSQL [14.5]</b>	4	8 Гб	10 Гб	1	
<b>S3</b>	1	1 Гб	5 Гб		

**Таблица 3-4** - Профиль №2 (средняя инсталляция)

	CPU	RAM	Storage	Количество инстансов	Прочие требования
<b>Kubernetes (K8s)</b> полезная нагрузка (сумма по микросервисам)	12	24 Гб		1	
Catalog	4	8 Гб		2	
Catalog UI	2	4 Гб		1	
Catalog gateway	2	4 Гб		1	
<b>PostgreSQL [14.5]</b>	8	16 Гб	100 Гб	1	
<b>S3</b>	1	1 Гб	10 Гб		

**Таблица 3-5** - Профиль №3 (крупная инсталляция)

	CPU	RAM	Storage	Количество инстансов	Прочие требования
<b>Kubernetes (K8s)</b>	40	80 Гб		1	



	CPU	RAM	Storage	Количество инстансов	Прочие требования
полезная нагрузка (сумма по микросервисам)					
Catalog	8	16 Гб		4	
Catalog UI	4	8 Гб		1	
Catalog gateway	4	8 Гб		1	
<b>PostgreSQL [14.5]</b>	16	32 Гб	1000 Гб	2	
<b>S3</b>	1	1 Гб	10 Гб		

В системе используется Kafka продукта «Сфера. Ядро», предполагаемая нагрузка на которую составляет:

Размер сообщений: 12 - 15 кБ;

Интенсивность: равна количеству создаваемых объектов в системе, в зависимости от профиля использования.

### 3.2.3 Матрица совместимости с интегрируемым продуктам Сферы.

**Таблица 3-6** — Матрица совместимости с интегрируемым продуктам Сферы

Продукт Сферы	Версия	Обязательно/Не обязательно интегрироваться
«Сфера. Ядро»	2025.2 – 2025.3	Обязательно
«Сфера. Ядро» (pphm-production-calendar)	2025.2 – 2025.3	Обязательно
«Сфера. Главная страница»	2025.2 – 2025.3	Обязательно
«Сфера. Доступы и Роли»	2025.2 – 2025.3	Обязательно
«Сфера. ИИС»	2025.2 – 2025.3	Обязательно
«Сфера. Аудит»	2025.2 – 2025.3	Обязательно
«Сфера. Нотификации»	2025.2 – 2025.3	Обязательно
«Сфера. Согласования»	2025.2 – 2025.3	Обязательно
«Сфера. Обращения»	2025.2 – 2025.3	Обязательно
«Сфера. Расчётный модуль SLA»	2025.2 – 2025.3	Обязательно
«Сфера. Конфигурации»	2025.2 – 2025.3	Обязательно
«Сфера. Архитектура»	2025.2 – 2025.3	Не обязательно
«Сфера. Команды»	2025.2 – 2025.3	Не обязательно

## 3.3 Требования к конфигурациям программного обеспечения.

### 3.3.1 Требования к базе данных.

Для работы продукта необходима СУБД PostgreSQL. Установка СУБД может быть выполнена любым удобным способом.

Следующие микросервисы требуют доступа к БД:

Catalog (Сфера. Каталог сервисов)

Data Consumer (Сфера. Каталог сервисов)

Каждому из микросервисов необходимо по две отдельные схемы, которые представляют из себя пару:

- схема данных;
- схема инструментов миграции.

Для инициализации базы данных необходимо запустить сессию с правами суперпользователя и выполнить следующий SQL скрипт:

В случае использования в продукте нескольких пар схем данных разными микросервисами, требуется изменить скрипт для создания нескольких схем миграций.

Пример скрипта:

```
--  
-- Инициализация схемы для инструментов миграции  
--  
-- Создание роли администратора  
CREATE ROLE <DB_ADMIN_USER> WITH LOGIN PASSWORD '<DB_ADMIN_PASSWORD>';  
-- Создание базы данных  
CREATE DATABASE <DB_NAME> OWNER <DB_ADMIN_USER> ENCODING 'UTF8' LC_COLLATE  
='ru_RU.UTF-8' LC_CTYPE='ru_RU.UTF-8' TEMPLATE='template0' CONNECTION LIMIT 300;  
-- Создание схемы для инструментов миграции  
CREATE SCHEMA <SCHEMA_MIGRATION> AUTHORIZATION <DB_ADMIN_USER>;  
ALTER USER <DB_ADMIN_USER> SET search_path to '<SCHEMA_MIGRATION>';  
--  
-- Инициализация основной схемы  
--  
-- Создание роли пользователя  
CREATE ROLE <DB_USER> WITH LOGIN PASSWORD '<DB_PASSWORD>';  
--  
-- Установка расширений
```

--

```
CREATE EXTENSION pg_trgm WITH SCHEMA pg_catalog;
```

```
CREATE EXTENSION ltree WITH SCHEMA pg_catalog;
```

### 3.3.2 Требования к Kafka.

Для настройки приложения вам потребуется Kafka для обработки следующих случаев:

- Публикация изменений данных;
- Обмен событиями о процессе согласования со «Сфера. Согласования»;
- Интеграция с другими системами «Сфера» на получение и отправку данных (см.

#### 4.2.3.3.4 Настройка интеграций для «Сфера. Каталог сервисов». и 5.2.1 Общие настройки интеграций.)

Сообщения по каждому кейсу рекомендуется публиковать в отдельные топики, рекомендует завести их заранее.

Указать топики для публикации тех или иных сообщений можно через настройки переменных окружения (см. **4.2.4 Переменные окружения «Сфера. Каталог сервисов».**) и через настройки интеграций в инструментах администрирования «Сфера. Каталог сервисов» (см. **4.2.3.3.3 Настройка справочника «Глобальные переменные» (globalVariables) для «Сфера. Каталог сервисов».**)

В случае запрета на создание топиков по запросу, необходимо предсоздать их и добавить разрешения на чтение/запись.

### 3.3.3 Требования к S3.

Для работы сервисов подойдёт любое S3-совместимое хранилище.

Установка и настройка S3-Like Object Storage производится в соответствии с документацией для выбранного объектного хранилища (как правило, это Sph или MinIO). После установки необходимо создать аккаунт для подключения к S3.

Должны быть созданы secret key, access key, bucket и проставлены в переменные Catalog (см.

#### 4.2.4 Переменные окружения «Сфера. Каталог сервисов».)

### 3.3.4 Требования к учётным записям (УЗ).

Для «Сфера. Каталог сервисов» необходимо завести ТУЗ ppcs\_tua (название по умолчанию) с доступом ко всем системам, с которыми происходит интеграция через REST и Каталог является потребителем. Доступы выдаются через админ-панель системы «Сфера. Доступы и роли».

Подробнее о доступах технической учётной см. **Таблица 3-7.**

Также название ТУЗ необходимо не забыть внести в справочник «Глобальные переменные» (см.

#### 4.2.3.3.3 Настройка справочника «Глобальные переменные» (globalVariables) для «Сфера. Каталог сервисов».) и в переменные системы (см. 4.2.4.1 Переменные сервиса catalog.)

**Таблица 3-7** — Доступы для ТУЗ «Сфера. Каталог сервисов»

№	Название роли	Объекты доступа
1.	Администратор главной страницы (Канал)	Админки - PPHM_ADM_PROD_CAL – Производственные календари
2.	Прикладной Администратор (Calculation module)	N/A
3.	Редактор назначений ДиР (Канал)	Система – PPCS-Catalog PPRQ-Service Desk
4.	Редактор объектов ДиР (Канал)	Система – PPCS-Catalog
5.	Редактор правил ДиР (Канал)	Система – PPCH-Канал
6.	Редактор типов объектов ДиР (Канал)	Система – PPCS-Catalog
7.	Роль для редактирования Каталога Сервисов (Catalog)	N/A
8.	Читатель пользователей ДиР (Канал)	Система – PPCH-Канал
9.	Читатель профилей пользователей (Канал)	N/A
10.	ТУЗ (Service Desk)	N/A

Конфигурирование продукта в «Сфера. Каталог сервисов» проводится с помощью учётной записи с ролью «Администратор» (см. **6.1. Управление пользователями и доступами.**).

Подготовка ролей для системы проводится в соответствии с регламентом ДиР (см. **5.1.1. Настройка политик безопасности.**). Список ролей может быть расширен относительно дефолтного, предоставляемого разработчиком.

## 3.4 Требования к настройкам рабочих станций пользователей.

### 3.4.1 Требования к рабочим станциям.

Для работы одного клиентского места пользователя рекомендуется использовать персональный компьютер, оснащённый техническими средствами со следующими минимальными характеристиками:

- процессор IntelCore 2-ядерный с тактовой частотой 3 ГГц или аналогичный;
- оперативная память объёмом 4 Гб;
- цветной монитор, позволяющий использовать для отображения веб-приложения рабочую область экрана размером 1680x1050 px (но не менее 1024x768 px);

#### Подсказка.

Для корректного отображения элементов веб-приложения «Каталог сервисов» рекомендуется

использовать 100% масштаб дисплея в настройках операционной системы ПК.

- устройства ввода (клавиатура, «мышь»);
- пропускная способность канала связи между рабочей станцией и веб-сервером – 2 Мбит/с.
- версия операционной системы должна поддерживать последние версии web-браузеров Google Chrome, Mozilla Firefox, Yandex Browser.

### 3.4.2 Требования к настройкам веб-браузера.

Работа с приложением «Каталог сервисов» выполняется с помощью веб-браузеров.

В качестве веб-браузера могут быть использованы Google Chrome, Mozilla Firefox, Yandex Browser последних устойчивых версий. Требования к браузерам являются общими для всех сервисов платформы Сфера, конкретно см. в документах по ОПС, вне рамок настоящего документа.

Основные требования по настройке веб-браузеров:

- убедитесь, что установлены последние обновления браузеров;
- отключите все сторонние дополнения и расширения (Addons), которые могут блокировать веб-элементы приложения «Каталог сервисов»;
- используйте 100% масштаб браузера.

#### Подсказка

Если персональные настройки ОС Windows на ПК пользователя приводят к некорректному отображению веб-форм (например, нестандартный масштаб шрифтов и пр.), то для устранения проблемы выполните сброс настроек темы ОС.

## 4 Инструкция по установке продукта.

В данном разделе представлены инструкции по установке системы «Сфера: Каталог сервисов», а также модуля «Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам».

Обращаем ваше внимание, что развёртывание и установка, а также настройка рекомендуемого к поставке вместе с системой «Сфера: Каталог сервисов» модуля «Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам» осуществляется системными администраторами платформы «Сфера» у Заказчика наряду с необходимым предустановленным ПО, и не требует специальных действий при развёртывании и установке, а также настройке системы «Сфера: Каталог сервисов».

### 4.1 Руководство по безопасности.

Подраздел содержит рекомендации по безопасности при развёртывании и эксплуатации, а также управлению доступами.

#### 4.1.1 Рекомендации по безопасности при развёртывании и эксплуатации.

##### 4.1.1.1 Использование защищённого соединения (HTTPS, TLS/mTLS).

- Все входные точки приложения (например, Ingress в Kubernetes) должны работать **только через HTTPS**.
- Внутрисервисное взаимодействие (между микросервисами, с Kafka, с PostgreSQL) должно быть по **TLS или mTLS**.
- Использовать только **современные версии TLS (1.2/1.3)**, отключить устаревшие шифры и протоколы.

##### Работа с keystore и truststore:

- Для настройки HTTPS и mTLS в Spring Boot используются хранилища:
- keystore — содержит закрытый ключ и сертификат приложения.
- truststore — содержит цепочку доверенных сертификатов (например, CA, self-signed).

##### 4.1.1.2 Создание базы данных PostgreSQL.

1. Заказать кластер(ы) базы данных PostgreSQL Patroni.
2. Создать БД.
3. Для созданной БД создать пользователя <db\_name>\_svc с ролью <db\_name>\_user.

*Примечание: скрипт должен выполняться через psql, иначе не работает команда \c.*

-- Создание ролей

```
CREATE ROLE <db_name>_user;
```

```
CREATE ROLE <db_name>_reader;
```

-- Создание TVЗ

```
CREATE USER <db_name>_admin PASSWORD '<CHANGEME>';
```

```
CREATE USER <db_name>_svc PASSWORD '<CHANGEME>';
-- Присвоение ролей TV3
GRANT <db_name>_user TO <db_name>_svc;
-- Создание БД
CREATE DATABASE <db_name> OWNER <db_name>_admin ENCODING 'UTF8' LC_COLLATE
='ru_RU.UTF-8' LC_CTYPE='ru_RU.UTF-8' TEMPLATE='template0' CONNECTION LIMIT 300;
\c <db_name>
-- Создание схемы для инструментов миграции
CREATE SCHEMA migration AUTHORIZATION <db_name>_admin;
-- Выдача прав для роли <db_name>_user
ALTER DEFAULT PRIVILEGES FOR ROLE <db_name>_admin GRANT SELECT, INSERT, UPDATE,
DELETE ON TABLES TO <db_name>_user;
ALTER DEFAULT PRIVILEGES FOR ROLE <db_name>_admin GRANT USAGE ON SCHEMAS TO
<db_name>_user;
-- Выдача прав для роли <db_name>_reader
ALTER DEFAULT PRIVILEGES FOR ROLE <db_name>_admin GRANT SELECT ON TABLES TO
<db_name>_reader;
ALTER DEFAULT PRIVILEGES FOR ROLE <db_name>_admin GRANT USAGE ON SCHEMAS TO
<db_name>_reader;
```

#### 4.1.1.3 Подготовка и настройка кластера Kubernetes.

1. Создать кластер k8s.
2. Создать namespace (проект) в созданном кластере.

#### 4.1.1.4 Подготовка и настройка балансировщика Load Balancer.

1. Создать балансировщик для Kubernetes.
2. Создать публикацию серверной части системы (backend).
3. Создать публикацию интерфейсной части системы (frontend).

#### 4.1.1.5 Работа с CRON.

Некоторые сервисы имеют настройки служб, выполняемых по расписанию, которое задаётся в формате CRON.

```
_____ секунда (0-59)
| _____ минута (0 - 59)
| | _____ час (0 - 23)
| | | _____ день месяца (1 - 31)
| | | | _____ месяц (1 - 12) (or JAN-DEC)
| | | | | _____ день недели (0 - 7)
| | | | | _____ (0 or 7 is Sunday, or MON-SUN)
| | | | |
* * * * *
```

Правила заполнения:

- Поле может быть помечено звёздочкой (\*), которая всегда означает «от первого до последнего». В полях «День месяца» или «число недели» вместо звёздочки может

использоваться вопросительный знак (?).

- Запятые (,) используются для разделения элементов списка.
- Два числа, разделённые дефисом (-), обозначают диапазон чисел. Указанный диапазон является включающим.
- После диапазона (или \*) с / указывается интервал между значениями числа в диапазоне.
- Для полей месяц и день недели также можно использовать названия на английском языке. Используйте первые три буквы определённого дня или месяца (регистр не имеет значения).
- Поля день месяца и день недели могут содержать букву L, которая имеет другое значение.
- В поле «день месяца» L обозначает последний день месяца. Если за ним следует отрицательное значение (то есть L-n), это означает, что число от n до последнего дня месяца.
- В поле «день недели» буква «L» обозначает последний день недели. Если перед ней стоит цифра или название из трёх букв (dL или DDDL), это означает последний день недели (d или DDDD) в месяце.
- Полем «день месяца» может быть nW, что означает ближайший день недели к n-му числу месяца. Если n приходится на субботу, это означает, что пятница предшествует этому дню. Если n выпадает на воскресенье, это приводит к следующему понедельнику, что также происходит, если n равно 1 и приходится на субботу (то есть: 1W означает первый рабочий день месяца).
- Если в поле «день месяца» указано значение LW, это означает последний рабочий день месяца.
- В поле «день недели» может быть указано значение d#n (или DDD#n), что соответствует n-му дню недели d (или DDD) в месяце.

Примеры заполнения:

**Таблица 4-1** — Примеры заполнения

Выражение	Расписание
* * * * *	каждую секунду (несколько раз в секунду)
0 * * * * *	раз в секунду
0 0 * * * *	в начале каждого часа каждого дня
*/10 * * * * *	каждые десять секунд
* */1 * * * *	каждую минуту
0 0 8-10 * * *	в 8, 9 и 10 часов каждого дня
0 0 6,19 * * *	в 6:00 утра и 7:00 вечера каждый день
0 0/30 8-10 * * *	в 8:00, 8:30, 9:00, 9:30, 10:00 и 10:30 каждый день



Выражение	Расписание
0 0 9-17 * * MON-FRI	с девяти до пяти по будням
0 0 0 1 JAN ?	каждый Новый год в полночь
0 0 0 L * *	последний день месяца в полночь
0 0 0 L-3 * *	предпоследний день месяца в полночь
0 0 0 * * 5L	последняя пятница месяца в полночь
0 0 0 * * THU L	последний четверг месяца в полночь
0 0 0 1W * *	первый рабочий день месяца в полночь
0 0 0 LW * *	в последний рабочий день месяца в полночь
0 0 0 ? * 5#2	во вторую пятницу месяца в полночь
0 0 0 ? * MON#1	первый понедельник месяца в полночь

#### 4.1.2 Управление доступом.

Валидация токена и получение разрешений проводится средствами сервисов ОПС («Сфера: Ядро», «Сфера: Аутентификация», «Сфера: Доступы и Роли»).

Для управления доступом в модуле «Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам» используется стандартная ролевая модель продукта «Сфера: Каталог сервисов»: «Администратор», «Редактор», «Читатель».

## 4.2 Порядок установки продукта.

Установка продукта можно произвести при помощи менеджера пакетов Helm для Kubernetes.

Альтернативно каждый микросервис может быть установлен и настроен и без использования Helm.

Вне зависимости от того, как будет выполнена установка версий продукта, последующие шаги настройки нужно будет выполнить для обоих способов (см. **4.2.3 Действия по установке продукта после наката версий.**).

### 4.2.1 Установка версий продукта менеджером пакетов Helm.

Перед установкой продукта убедитесь, что менеджер пакетов Helm установлен и настроен на нужный кластер Kubernetes.

1. Выполните вход в репозиторий продукта Сферы "Дистрибутивы и Лицензии".

Для этого выполните следующую команду Helm:

```
helm repo add ppcs https://dist.sfera-t1.ru/app/ppdl-
new/api/repo/helm/ppcs-helm-int/ --username <login> --
password '<password>'
```

где:

<login> - имя учётной записи для доступа к <https://dist.sfera-t1.ru>;

<password> - пароль от учётной записи для доступа к ресурсу.

Пример вывода команды:

```
"ppcs" has been added to your repositories
```

## 2. Проверьте подключение к репозиторию "Дистрибутивов и Лицензий".

Для вывода последней версии опубликованного чарта выполните следующую команду:

```
helm search repo ppcs --devel
```

В случае успешного выполнения команды, будет выведена последняя версия опубликованного чарта.

Для вывода всех доступных чартов добавьте в команду параметр "-l":

```
helm search repo ppcs --devel -l
```

## 3. Переопределите переменные, зависящие от контура.

Если в продукте отсутствуют контурозависимые переменные, данный пункт не требует выполнения.

Подготовьте файл с переменными, зависящими от контура. Для этого скачайте архив с чартами расположенный по указанному пути:

```
https://dist.sfera-tl.ru/app/ppdl-new/api/repo/helm/ppcs-helm-int  
/<version>.tar.gz
```

где:

<version> - версия продукта.

Ссылку на мета-чарт поставки см. **Таблица 2-2**.

Извлеките values.yaml файл, расположенный в корне архива. Откройте его и отредактируйте, выставив нужные параметры переменным, требующим замены. Например, строку подключения к базе данных.

Полный список переменных и их описание приведены в разделе **4.2.4 Переменные окружения «Сфера. Каталог сервисов»**.

### Внимание!

Не используйте целиком extra-values.yaml от предыдущей установки продукта, так как в этом обновлении могли появиться новые переменные или поменяться значения существующих.

Переименуйте файл в extra-values.yaml

## 4. Установите продукт используя meta-chart.

Для установки продукта выполните следующую команду:

```
helm -n <namespace> upgrade -i ppcs -f extra-values.yaml <chart_name> --  
devel --version <version>
```

где:

<namespace> - имя namespace в Kubernetes для установки;

<chart\_name> - имя чарта (колонка "NAME" вывода команды search, смотрите пункт 2 выше);

<version> - версия чарта (колонка "CHART VERSION" вывода команды search, смотрите пункт 2

выше).

Пример вывода команды:

```
Release "name" does not exist. Installing it now.  
NAME: ppcs  
LAST DEPLOYED: <date>  
NAMESPACE: <namespace>  
STATUS: deployed
```

5. Сохраните extra-values.yaml файл для упрощения дальнейших обновлений системы.

## 4.2.2 Установка версий продукта без использования менеджера пакетов Helm.

Раздел содержит шаги по альтернативной установке сервисов без использования meta-charts.

**Примечание:** при использовании нескольких под для бэкенда необходимо перед обновлением оставить активной только одну поду.

После завершения обновления продукта можно включить необходимое количество под для бэкенда.

### 4.2.2.1 Установить версии ПО.

Установить новые версии системы в соответствии с данными **Таблица 2-2** для «Сфера. Каталог сервисов».

#### 4.2.2.1.1 «Сфера. Каталог сервисов» - добавить сервис catalog.

- Добавить Deployment в кластер Kubernetes (пример ниже);
- Добавить Service-манифест для доступа к приложению по имени внутри кластера (пример ниже);
- Добавить переменные окружения — см. раздел **4.2.4.1 Переменные сервиса catalog.**

```
kind: Service  
metadata:  
  name: catalog  
  namespace: <namespace_name>  
spec:  
  selector:  
    app: catalog  
  ports:  
    - protocol: TCP  
      port: 80  
      targetPort: 8000  
kind: Deployment  
apiVersion: apps/v1  
metadata:  
  name: catalog
```

```

namespace: <namespace_name>
labels:
  app: catalog
  version: <image_tag>
spec:
  replicas: 1
  selector:
    matchLabels:
      app: catalog
  template:
    metadata:
      labels:
        app: catalog
        version: <image_tag>
    spec:
      restartPolicy: Always
      containers:
        - name: catalog
          image: >-
            nexus-ci.XXX/ppcs-docker-snapshot/catalog:<image_tag>
          ports:
            - name: http
              containerPort: 8000
              protocol: TCP
          resources:
            limits:
              cpu: «1»
              memory: 2Gi
            requests:
              cpu: 500m
              memory: 512Mi
          env:
            - name: <обязательные переменные окружения>
              value: <обязательные переменные окружения>
      strategy:
        type: RollingUpdate
        rollingUpdate:
          maxUnavailable: 25%
          maxSurge: 25%

```

#### 4.2.2.1.2 «Сфера. Каталог сервисов» - добавить сервис catalog-ui.

- Добавить Deployment в кластер Kubernetes (пример ниже);
- Добавить Service-манифест для доступа к приложению по имени внутри кластера (пример ниже);
- Добавить переменные окружения — см. раздел **4.2.4.4 Переменные сервиса catalog-ui.**

```
kind: Service
metadata:
  name: catalog-ui
  namespace: <namespace_name>
spec:
  selector:
    app: catalog-ui
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8000
kind: Deployment
apiVersion: apps/v1
metadata:
  name: catalog-ui
  namespace: <namespace_name>
  labels:
    app: catalog-ui
    version: <image_tag>
spec:
  replicas: 1
  selector:
    matchLabels:
      app: catalog-ui
  template:
    metadata:
      labels:
        app: catalog-ui
        version: <image_tag>
    spec:
      restartPolicy: Always
      containers:
        - name: catalog-ui
```

```

image: >-
  nexus-ci.XXX/ppcs-docker-snapshot/catalog-ui:<image_tag>
ports:
  - name: http
    containerPort: 8000
    protocol: TCP
resources:
  limits:
    cpu: «1»
    memory: 2Gi
  requests:
    cpu: 500m
    memory: 512Mi
env:
  - name: <обязательные переменные окружения>
    value: <обязательные переменные окружения>
strategy:
  type: RollingUpdate
  rollingUpdate:
    maxUnavailable: 25%
    maxSurge: 25%

```

#### 4.2.2.1.3 «Сфера. Каталог сервисов» - добавить сервис catalog-gateway.

- Добавить Deployment в кластер Kubernetes (пример ниже);
- Добавить Service-манифест для доступа к приложению по имени внутри кластера (пример ниже);
- Добавить переменные окружения — см. раздел **4.2.4.3 Переменные сервиса catalog-gateway.**

```

kind: Service
metadata:
  name: catalog-gateway
  namespace: <namespace_name>
spec:
  selector:
    app: catalog-gateway
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8000
kind: Deployment

```

```

apiVersion: apps/v1
metadata:
  name: catalog-gateway
  namespace: <namespace_name>
  labels:
    app: catalog-gateway
    version: <image_tag>
spec:
  replicas: 1
  selector:
    matchLabels:
      app: catalog-gateway
  template:
    metadata:
      labels:
        app: catalog-gateway
        version: <image_tag>
    spec:
      restartPolicy: Always
      containers:
        - name: catalog-gateway
          image: >-
            nexus-ci.XXX/ppcs-docker-snapshot/catalog-
gateway:<image_tag>
          ports:
            - name: http
              containerPort: 8000
              protocol: TCP
      resources:
        limits:
          cpu: «1»
          memory: 2Gi
        requests:
          cpu: 500m
          memory: 512Mi
      env:
        - name: <обязательные переменные окружения>
          value: <обязательные переменные окружения>
    strategy:
      type: RollingUpdate

```

```
rollingUpdate:
  maxUnavailable: 25%
  maxSurge: 25%
```

#### 4.2.2.1.4 «Сфера. Каталог сервисов» - добавить сервис data-consumer-service (опционально).

- Добавить Deployment в кластер Kubernetes (пример ниже);
- Добавить Service-манифест для доступа к приложению по имени внутри кластера (пример ниже);
- Добавить переменные окружения — см. раздел **4.2.4.2 Переменные сервиса data-consumer-service**.

```
kind: Service
metadata:
  name: data-consumer-service
  namespace: <namespace_name>
spec:
  selector:
    app: data-consumer-service
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8000
kind: Deployment
apiVersion: apps/v1
metadata:
  name: data-consumer-service
  namespace: <namespace_name>
  labels:
    app: data-consumer-service
    version: <image_tag>
spec:
  replicas: 1
  selector:
    matchLabels:
      app: data-consumer-service
  template:
    metadata:
      labels:
        app: data-consumer-service
        version: <image_tag>
```



```
spec:
  restartPolicy: Always
  containers:
    - name: data-consumer-service
      image: >-
        nexus-ci.XXX/ppcs-docker-snapshot/data-consumer-
service:<image_tag>
      ports:
        - name: http
          containerPort: 8000
          protocol: TCP
      resources:
        limits:
          cpu: «1»
          memory: 2Gi
        requests:
          cpu: 500m
          memory: 512Mi
      env:
        - name: <обязательные переменные окружения>
          value: <обязательные переменные окружения>
  strategy:
    type: RollingUpdate
    rollingUpdate:
      maxUnavailable: 25%
      maxSurge: 25%
```

### 4.2.3 Действия по установке продукта после наката версий.

Эти шаги нужно проделать после того, как были установлены версии продукта – независимо от того, шла установка с использованием helm-chart или без него.

#### 4.2.3.1 Перевод взаимодействия с «Сфера. Доступы и Роли» в локальный режим.

Опционально перевести в локальный режим интеграцию с системой Доступы и Роли (опционально - чтобы было меньше проблем при накате новых представлений). Для этого в сервисе Catalog установить значение переменной SECURITY\_VALIDATOR\_TYPE= local.

#### 4.2.3.2 Обновление конфигурации системы для «Сфера. Каталог сервисов».

##### 4.2.3.2.1 Создание новой сессии.

Создать сессию и войти в неё.

Подробнее о процедуре см. **6.4 Управление конфигурациями для «Сфера. Каталог**

сервисов».

#### 4.2.3.2.2 Загрузка конфигураций.



Загрузить конфигурационные файлы из комплекта – подробнее о процедуре см. **6.4 Управление конфигурациями для «Сфера. Каталог сервисов».**

Для установки всех сценариев базовой версии функциональности должны быть загружено всё содержимое файла (см. **Таблица 4-2**).



Если проходит установка системы с модулем "Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам", необходимо установить конфигурацию для модуля.

Сначала устанавливается файл dove-scm\*.zip, потом catalog-configuration\*.zip.

**Таблица 4-2** – Файлы для загрузки конфигурации – базовая версия функциональности

№	Маска имени	Экземпляр	Описание
1.	dove-scm*.zip	 dove-scm_05.11.2025_1.45.0_base.zip	Конфигурационный файл служебных сервисов. Включается в поставку опционально
2.	catalog-configuration_*.zip	 catalog-configuration_05.11.2025_1.45.0_base.zip	Конфигурационный файл Каталога сервисов

**Таблица 4-3** – Файлы для загрузки конфигурации – версия для модуля "Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам"

№	Маска имени	Экземпляр	Описание
3.	dove-scm*.zip	 dove-scm_05.11.2025_1.45.0_addon.zip	Конфигурационный файл служебных сервисов. Включается в поставку опционально
4.	catalog-configuration_*.zip	 catalog-configuration_05.11.2025_1.45.0_addon.zip	Конфигурационный файл Каталога сервисов

#### 4.2.3.3 Настройка процессов и переменных в справочниках системы для «Сфера. Каталог сервисов».

При установке конфигурации необходимо помнить о параметрах, заполняемых в соответствии с настройками окружения, – их требуется заполнить актуальными для стенда разворота данными.

##### 4.2.3.3.1 Настройка справочника «Переменные среды» (environmentVariable) для «Сфера. Каталог сервисов».

Через пользовательский интерфейс каталога сервисов необходимо войти в справочник системы «Переменные среды» (environmentVariable) (подраздел "Конфигурация справочников" раздела "Администрирование"), изучить записи справочника, при необходимости изменить их значения

согласно бизнес-логике процесса (см. **Таблица 4-4**.)

**Таблица 4-4** – Описание справочника «Переменные среды»

№	Идентификатор записи	Имя записи	Значение (по умолчанию)	Описание
1.	backReferenceDefaultDeep	Глубина связей по умолчанию	3	<p>Значение уровней в структуре зависимостей, которое будет по умолчанию загружаться при открытии структуры.</p> <p><b>См. 5.9.1 Структура зависимостей.</b> в Руководстве пользователя системы.</p> <p>Переменная относится к системным для инструментов администрирования DOVE.</p>
2.	disHost	Значение url используемое в appealControlContainer для формирования ссылки в ДИЗ	Change it for current environment (ACRQ url)	<p>Значение url используемое в appealControlContainer для формирования ссылки в ДИЗ</p> <p><b>Важно!</b></p> <p>Переменная является кастомной – не используется повсеместно.</p> <p>Переменная используется только в системе «Сфера. Каталог сервисов».</p>
3.	npsWidgetEnabled	Разрешение на включение виджета опроса	false	<p>Разрешение на включение виджета опроса</p> <p>Переменная используется только в системе «Сфера. Каталог сервисов».</p>
4.	offerChainChangeRequestDataConfigId	Идентификатор представления заявки цепочки сервисов	ChainOfferChangeRequestObjectView	<p>Идентификатор представления заявки цепочки сервисов</p> <p>Переменная используется только в системе «Сфера. Каталог сервисов».</p>
5.	offerChainCompositionPageId	Идентификатор страницы composition для ChainOffer	form	<p>Идентификатор страницы composition для ChainOffer</p>

№	Идентификатор записи	Имя записи	Значение (по умолчанию)	Описание
				Переменная используется только в системе «Сфера. Каталог сервисов».
6.	offerManualChangeRequestDataConfigId	Идентификатор представления заявки ручного сервиса	ManualOfferChangeRequestObjectView	Идентификатор представления заявки ручного сервиса Переменная используется только в системе «Сфера. Каталог сервисов».
7.	offerManualCompositionPageId	Идентификатор страницы composition для ManualOffer	form	Идентификатор страницы composition для ManualOffer  Переменная используется только в системе «Сфера. Каталог сервисов».
8.	offerSourceChangeRequestDataConfigId	Идентификатор представления заявки сервиса	SourceOfferChangeRequestObjectView	Идентификатор представления заявки сервиса  Переменная используется только в системе «Сфера. Каталог сервисов».
9.	offerSourceCompositionPageId	Идентификатор страницы composition для SourceOffer	form	Идентификатор страницы composition для SourceOffer  Переменная используется только в системе «Сфера. Каталог сервисов».
10.	sessionsLimit	Максимальное количество сессий	7	Максимальное количество сессий.  <b>См. 5.9.2 Управление конфигурациями.</b> в Руководстве пользователя системы.  Переменная относится к системным для инструментов администрирования DOVE.
11.	softDeleteReferenceEnabled	Переменная определяющая правила удаления справочников	false	Переменная определяющая правила удаления справочников. При установке значения false, либо любого отличного от true,

№	Идентификатор записи	Имя записи	Значение (по умолчанию)	Описание
				<p>выполняется обычный механизм удаления. При установке значения true выполняется механизм мягкого удаления.</p> <p>См. <b>5.9.3.8 Удаление записи в справочнике (мягкое удаление включено)</b>. в Руководстве пользователя системы.</p> <p>Переменная относится к системным для инструментов администрирования DOVE.</p>
12.	softDeleteTimeScheduleEnabled	Переменная определяющая правила удаления RBP	true	<p>Переменная определяющая правила удаления RBP. При установке значения false, либо любого отличного от true, выполняется обычный механизм удаления. При установке значения true выполняется механизм мягкого удаления.</p> <p>См. <b>5.9.7.6 Удаление расписания (мягкое удаление включено)</b>. в Руководстве пользователя системы.</p> <p>Переменная относится к системным для инструментов администрирования DOVE.</p>
13.	ssePath	Путь подключения к sse	/api/api/v1/sse	<p>Путь подключения к sse.</p> <p>Переменная относится к системным для инструментов администрирования DOVE.</p>
14.	sseReconnectTimeMs	Временной интервал переподключения к sse при обрыве (в мс)	20000	<p>Временной интервал переподключения к sse при обрыве (в мс).</p> <p>Переменная относится к системным для инструментов администрирования DOVE.</p>

№	Идентификатор записи	Имя записи	Значение (по умолчанию)	Описание
15.	updateDirEnableAutoRetry	Флаг для обновления политик безопасности в ДИР	false	Флаг для обновления политик безопасности в ДИР.  Переменная относится к системным для инструментов администрирования DOVE.
16.	updateDirSystemCode	Код текущей системы при обновлении в ДИР	PPCS	Код текущей системы при обновлении в ДИР.  Переменная относится к системным для инструментов администрирования DOVE.

#### 4.2.3.3.2 Настройка справочника «Пользовательские роли» (securityRole) для «Сфера. Каталог сервисов».

Через пользовательский интерфейс каталога сервисов необходимо войти в справочник системы Пользовательские роли (securityRole) (подраздел "Конфигурация справочников" раздела "Администрирование"), создать и заполнить записи справочника согласно таблице (см. **Таблица 4-5**).

**Таблица 4-5** – Описание справочника «Пользовательские роли»

№	Идентификатор записи	Имя записи
1.	ppcs_catalog_corp_architect	Роль Корпоративного архитектора
2.	ppcs_catalog_editor	Роль для редактирования Каталога Сервисов
3.	ppcs_catalog_viewer	Роль для просмотра Каталога Сервисов
4.	ppcs_api_client	Роль для доступа к API Каталога Сервисов
5.	ppcs_catalog_auditor	Роль для Аудит Каталога Сервисов
6.	ppcs_catalog_administrator	Роль Администратора Каталога Сервисов

Роли 1, 4, 5 являются дополнительными и используются только в дополнительных сценариях системы. Если планируется только использование базовых сценариев, эти роли можно не добавлять. Со списком дополнительных и основных сценариев можно подробно ознакомиться в Руководстве пользователя «Сфера. Каталог сервисов».

Роль «Роль Администратора» (dove\_administrator) может быть удалена из справочника, так как используется как универсальная и не относится к ролевой модели конкретной системы. Также роль можно не удалять, но в этом случае требуется не использовать её при настройке разрешений конфигураций.

Поле "Дополнительные детали записи" заполняется в следующем формате:

```
{
  "roleId": " UUID роли"
}
```

где UUID роли - id из «Сфера. Доступы и Роли».

Для получения id роли необходимо зайти в админ-панель «Сфера. Доступы и Роли», вкладка "Роли". Найти данные в ответе на запрос roles в панели разработчика (для поиска нужных ролей применить поиск по значению ppcs\_catalog).

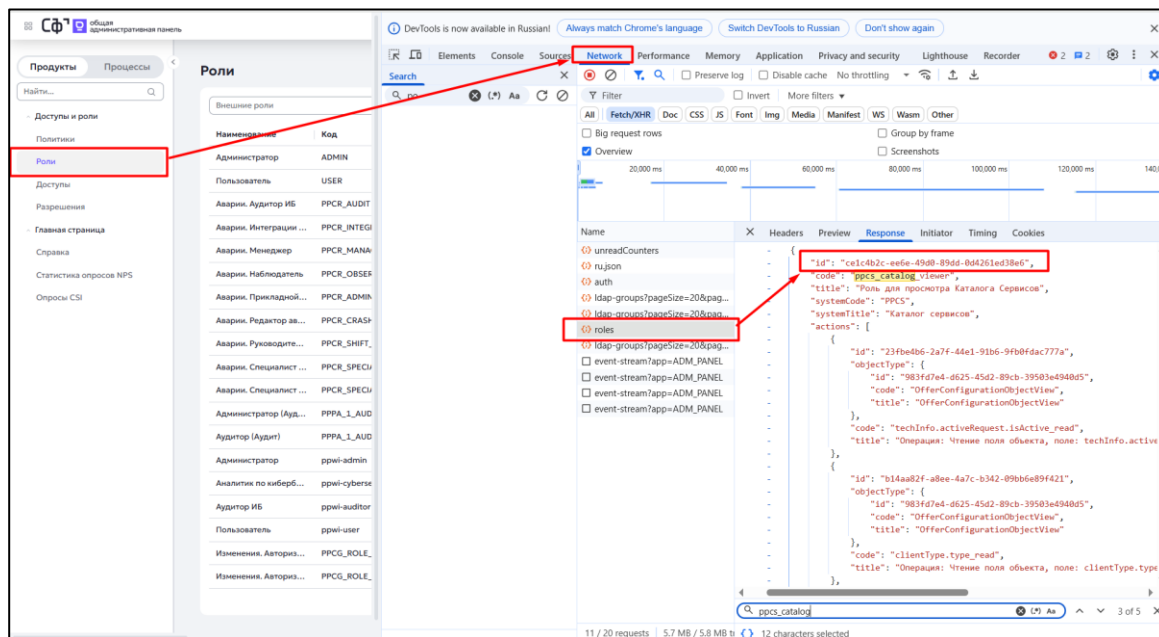


Рис. 4-1 – Данные админ-панели «Сфера. Доступы и Роли».

В справочнике "Пользовательские роли" (securityRole) Каталога сервисов для роли «Роль Администратора Каталога Сервисов» (ppcs\_catalog\_administrator) в дополнительные детали записи добавлен флаг

```
{ "isSystemAdmin": true}.
```

В справочнике на уровне системы должна быть строго одна запись с таким флагом.

Флаг используется для сопоставления разрешений по роли «Роль Администратора» (dove\_administrator) с локальной ролью администратора в системе.

Когда производится импорт dove-scm в систему, разрешения в представлениях автоматически меняются на ту роль, у которой в справочнике присутствует флаг { "isSystemAdmin": true}.

Пример дополнительных деталей записи «Роль Администратора Каталога Сервисов» (ppcs\_catalog\_administrator) после заполнения:

```
{
  "roleId": "d9a24a76-3523-439e-8857-6829092b7d4e",
  "isSystemAdmin": true
}
```

Также при заполнении справочника можно воспользоваться инструментом миграции - **5.3.1 Миграция справочников.**

#### 4.2.3.3.3 Настройка справочника «Глобальные переменные» (globalVariables) для «Сфера. Каталог сервисов».

Через пользовательский интерфейс каталога сервисов необходимо войти в справочник системы «Глобальные переменные» (globalVariables) (подраздел "Конфигурация справочников" раздела "Администрирование"), создать и настроить новые переменные согласно **Таблица 4-6**. При создании записи следует в точности воспроизвести «Идентификатор записи» и «Имя записи» из таблицы, в «Дополнительных деталях записи» заполнить value значением, релевантным для среды установки.

Также при заполнении справочника можно воспользоваться инструментом миграции - **5.3.1 Миграция справочников.**

Подробнее о принципе работе глобальных переменных можно прочитать в Руководстве пользователя «Сфера. Каталог сервисов», раздел **5.9.3.10.3 Глобальные переменные.**

**Таблица 4-6 – Записи справочника «Глобальные переменные»**

№	Идентификатор записи	Имя записи	Пример атрибута value	Место использования
1.	env.url	URL	https://<имя хоста>	<p>URL для системы «Сфера» на окружении</p> <p>Жизненный цикл:</p> <ul style="list-style-type: none"> <li>ЖЦ для объекта ChainOfferChangeRequest</li> <li>ЖЦ для объекта ManualOfferChangeRequest</li> <li>ЖЦ для объекта SourceOfferChangeRequest</li> </ul> <p>переход DESIGN_COMPLETED -&gt;PUBLISHED, действие "Установка в поле объекта значения"</p> <p>Интеграция:</p> <ul style="list-style-type: none"> <li>Получение функциональных позиций по REST</li> <li>Получение стримов по REST</li> <li>Получение сотрудников по REST</li> </ul>



№	Идентификатор записи	Имя записи	Пример атрибута value	Место использования
				<ul style="list-style-type: none"> <li>Получение сервисной структуры из «Сфера. Команда»</li> <li>Получение пользователей по REST</li> <li>Получение кодов календарей</li> <li>Отклонение согласования от ТУЗ</li> <li>Выгрузка календаря</li> </ul>
2.	technicalAccount	ТУЗ	ppcs_tua	<p>Название технической учётной записи «Сфера. Каталог сервисов».</p> <p>Преобразования данных:</p> <ul style="list-style-type: none"> <li>Создание конфигурационной единицы: переходы target-&gt;ТУЗ-obj, ТУЗ-obj-&gt;ТУЗ-ref</li> <li>Создание связи между КЕ: переходы target-&gt;ТУЗ-obj, ТУЗ-obj-&gt;ТУЗ-ref</li> <li>Обновление конфигурационной единицы: переходы target-&gt;ТУЗ-obj, ТУЗ-obj-&gt;ТУЗ-ref</li> <li>Обновление связи между КЕ: переходы target-&gt;ТУЗ-obj, ТУЗ-obj-&gt;ТУЗ-ref</li> </ul>
3.	kafka.ppcm-ke-output	Топик Сфера.Конфигурации для передачи КЕ в другие системы	2078.ppcm-dataEvent-ke-output-1.0	<p>Интеграции:</p> <ul style="list-style-type: none"> <li>Получение информации по КЕ из Kafka</li> <li>Получение ИС</li> <li>Получение Подсистем ИС</li> </ul>
4.	kafka.ppcm-ke-input	Топик Сфера.Конфигурации для создания КЕ и связей	2078.ppcm-dataEvent-ext-ke-input-1.0	<p>Интеграции:</p> <ul style="list-style-type: none"> <li>Создание и обновление связей между КЕ по Kafka</li> </ul>

№	Идентификатор записи	Имя записи	Пример атрибута value	Место использования
				<ul style="list-style-type: none"> <li>Создание и обновление KE по Kafka</li> </ul>
5.	kafka.ppcs-service	Топик Сфера.Каталог сервисов для передачи сервисов в другие системы	2062.ppcs-dataEvent-Service-1.0	Интеграции: <ul style="list-style-type: none"> <li>Отправка цепочек сервисов в Kafka</li> <li>Отправка сервисов в Kafka</li> <li>Отправка ручных сервисов в Kafka</li> </ul>
6.	kafka.appealControlContainerImport	Топик для передаче ДИЗ статуса импорта контейнера обращения и матрицы	2062.AppealControlContainerImportStatus-v1	Интеграции: <ul style="list-style-type: none"> <li>Отправка статуса по импорту контейнера шаблона и матрицы нормативных сроков</li> </ul>
7.	kafka.notification-create-request	Топик Сфера.Нотификации для отправки нотификаций	sfera_notifications.create_request	Интеграции: <ul style="list-style-type: none"> <li>Отправка сообщений в нотификации</li> </ul>
8.	kafka.config-unit-update	Топик для передачи связки идентификаторов сервисов и KE	config-unit-update	Интеграции: <ul style="list-style-type: none"> <li>Отправка ИД сервиса и KE в Kafka</li> </ul>
9.	kafka.approvals	Топик Сфера.Согласования для передачи статусов согласования	2069_APPROVALS_EVENT_FOR_CATALOG_SERVICES	Интеграции: <ul style="list-style-type: none"> <li>Сообщения от Согласований</li> </ul>
10.	kafka.team-employee	Топик Сфера.Команда для передачи информации по сотрудникам	TEAM-EMPLOYEE	Интеграции: <ul style="list-style-type: none"> <li>Получение сотрудников по Kafka</li> </ul>
11.	kafka.ppan-ppcs-system-relation	Топик Сфера.Архитектура для передачи информации по связям сервисов с ИС	ppan_ppcs_bank_srv_system_srv_relations	Интеграции: <ul style="list-style-type: none"> <li>Получение связей с ИС из «Сфера. Архитектура»</li> </ul>
12.	kafka.ppan-ppcs-service-relation	Топик Сфера.Архитектура для передачи информации по связям между сервисами	ppan_ppcs_bank_srv_bank_srv_relations	Интеграции: <ul style="list-style-type: none"> <li>Получение связей между сервисами из «Сфера. Архитектура»</li> </ul>
13.	kafka.ppan-ppcs-chain-relation	Топик Сфера.Архитектура для передачи информации по связям цепочек и сервисов	ppan_ppcs_service_chain_info	Интеграции: <ul style="list-style-type: none"> <li>Получение связей Цепочка-сервис из «Сфера. Архитектура»</li> </ul>

№	Идентификатор записи	Имя записи	Пример атрибута value	Место использования
14.	kafka.profile	Топик Сфера.Профили для передачи информации по пользователям	profile_replication	Интеграции: <ul style="list-style-type: none"> <li>Получение пользователей из «Сфера. Профили»</li> </ul>
15.	kafka.ppsl	Топик Сфера.Расчетный модуль для получения конфигурации матриц нормативного срока	ppsl-topic	Интеграции: <ul style="list-style-type: none"> <li>Получение конфигурации по матрице нормативных сроков из Расчётного модуля</li> </ul>
16.	kafka.pprq-ppcs	Топик Сфера.Обращения для получения информации по шаблону	pprq-master	Интеграции: <ul style="list-style-type: none"> <li>Получение информации по шаблону из Модуля Обращения</li> </ul>
17.	kafka.team-funcStruct	Топик Сфера.Команда для получения функциональной структуры	TEAM-FUNC_STRUCT	Интеграции: <ul style="list-style-type: none"> <li>Обновление информации по стримам из «Сфера. Команды»</li> </ul>
18.	kafka.appealControlContainerImport-input	Топик для импорта контейнера обращения и матрицы из ДИЗ	2062.AppealControlContainerImport-v1	Интеграции: <ul style="list-style-type: none"> <li>Импорт контейнера шаблонов и матрицы нормативных сроков</li> </ul>

#### 4.2.3.3.4 Настройка интеграций для «Сфера. Каталог сервисов».

Через пользовательский интерфейс каталога сервисов необходимо войти в настройки интеграций системы (подраздел "Интеграции" раздела "Администрирование") - проверить активность нужной интеграции - в колонке "Активна" тогл должен находиться во включённом состоянии.

Настройки активных интеграции для полей URL (для rest) и Имя топика (для kafka) заполняются с помощью глобальных переменных, заполненных на предыдущем шаге.

Эти поля должны быть заполнены в соответствии с настройками окружения, на котором производится разворот системы.

Полное описание полей для настройки интеграций и сценарии настройки можно найти в Руководстве пользователя «Сфера. Каталог сервисов», раздел **5.9.10 Интеграции** и раздел **6.5 Настройки интеграций**.

Для базовой версии системы требуется установка и настройка всех интеграций, приведённых в **Таблица 4-7**.

Если проходит установка кастомной версии для внешнего заказчика, необходимо вдобавок к

интеграциям базовой версии настроить интеграции кастомной версии (см. **Таблица 4-8**)

Если проходит установка системы с модулем "Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам", необходимо вдобавок к интеграциям базовой версии настроить интеграции для модуля (см. **Таблица 4-9**).

**Таблица 4-7** — Базовые настройки интеграций Каталога сервисов

№	Код (Название)	Тип интеграции	Интегрируемая система	Примечание
Базовые настройки (нужны всегда)				
1.	ApprovalContainer (Сообщения от Согласований)	Чтение из Kafka	«Сфера. Согласования»	
2.	RejectApproval (Отклонение согласования от ТУЗ)	REST	«Сфера. Согласования»	Для корректной работы интеграции необходимо запросить у системы «Сфера. Согласования» доступ к требуемому запросу (POST /public/api/v1/approval/set-status) для используемой в интеграции ТУЗ.
3.	ApprovalsContainerManagement (Интеграция с Сфера.Согласования (Шаблоны обращений))	Чтение из Kafka	«Сфера. Согласования»	
4.	InNotificationSendingMessages (Отправка сообщений в нотификации)	Запись в Kafka	«Сфера. Нотификации»	
5.	CalculationConfigDynamics (Получение конфигурации по матрице нормативных сроков из Расчетного модуля)	Чтение из Kafka	«Сфера. Расчётный модуль SLA»	Должна быть выключена переходная интеграция «Получение конфигурации по матрице нормативных сроков из Расчетного модуля» (CalculationConfig).
6.	TemplateInfoDynamics (Получение информации по шаблону из Модуля Обращения)	Чтение из Kafka	«Сфера. Обращения»	Должна быть выключена переходная интеграция «Получение информации по шаблону из Модуля Обращения» (TemplateInfo).

№	Код (Название)	Тип интеграции	Интегрируемая система	Примечание
7.	CloneMatrix (Клонирование матрицы нормативных сроков)	REST	«Сфера. Расчётный модуль SLA»	
8.	CloneTemplate (Клонирование шаблона обращения)	REST	«Сфера. Обращения»	
9.	CreateTemplate (Создание шаблона)	REST	«Сфера. Обращения»	
10.	CreateMatrixSLM (Создание матрицы нормативных сроков)	REST	«Сфера. Расчётный модуль SLA»	
11.	CreateNewVersionMatrix (Создания новой версии матрицы нормативных сроков)	REST	«Сфера. Расчётный модуль SLA»	
12.	CreateNewVersionTemplate (Создания новой версии шаблона обращения)	REST	«Сфера. Обращения»	
13.	UpdateStatusTemplate (Обновление статуса шаблона)	REST	«Сфера. Обращения»	
14.	UpdateStatusMatrix (Обновление статуса матрицы)	REST	«Сфера. Расчётный модуль SLA»	
15.				
16.	ISConsumerPPCM (Получение ИС)	Чтение из Kafka	«Сфера. Конфигурации»	
17.	SubISConsumerPPCM (Получение Подсистем ИС)	Чтение из Kafka	«Сфера. Конфигурации»	

№	Код (Название)	Тип интеграции	Интегрируемая система	Примечание
18.	KEConsumer (Получение информации по KE из Kafka)	Чтение из Kafka	«Сфера. Конфигурации»	
19.	KEProducer (Создание и обновление KE по Kafka)	Запись в Kafka	«Сфера. Конфигурации»	<p>В преобразовании данных «Создание конфигурационной единицы» в переходе из контейнера «target» в контейнер «Конфигурационная единица» в поле "Ключ подстановки" указать значение ТУЗ, используемой для интеграции со «Сфера. Конфигурации» для атрибута «default» у поля «createdBy». ТУЗ, отправляемый в этом поле в систему «Сфера. Конфигурации» должен быть зарегистрирован на их стороне для корректности получаемых данных.</p> <p>В преобразовании данных «Обновление конфигурационной единицы» в переходе из контейнера «target» в контейнер «Конфигурационная единица» в поле "Ключ подстановки" указать значение ТУЗ, используемой для интеграции со «Сфера. Конфигурации» для атрибута «default» у поля «updatedBy». ТУЗ, отправляемый в этом поле в систему «Сфера. Конфигурации» должен быть зарегистрирован на их стороне для корректности получаемых данных.</p>
20.	KERelationProducer (Создание и обновление связей между KE по Kafka)	Запись в Kafka	«Сфера. Конфигурации»	
21.	SourceOfferKafka Producer	Запись в Kafka	«Сфера. Каталог сервисов»	

№	Код (Название)	Тип интеграции	Интегрируемая система	Примечание
	(Отправка сервисов в Kafka)			
22.	ManualOfferKafkaProducer (Отправка ручных сервисов в Kafka)	Запись в Kafka	«Сфера. Каталог сервисов»	
23.	ConfigUnitIdProducer (Отправка ИД сервиса и КЕ в Kafka)	Запись в Kafka	«Сфера. Каталог сервисов»	
24.	ChainOfferKafkaProducer (Отправка цепочек сервисов в Kafka)	Запись в Kafka	«Сфера. Каталог сервисов»	
25.	UserConsumerPPCH (Получение пользователей из «Сфера. Профили»)	Чтение из Kafka	«Сфера. Профили»	
Настройки опциональных интеграций (в зависимости от комплекта используемых продуктов Сфера)				
26.	EmployeesConsumerTeams (Получение сотрудников по Kafka)	Чтение из Kafka	«Сфера. Команда»	
27.	StreamConsumerTeam (Обновление информации по стримам)	Чтение из Kafka	«Сфера. Команда»	
28.	RelationServiceConsumerArchitecture (Получение связей между сервисами из «Сфера. Архитектура»)	Чтение из Kafka	«Сфера. Архитектура»	
29.	RelationIsConsumerArchitecture (Получение связей с ИС из «Сфера. Архитектура»)	Чтение из Kafka	«Сфера. Архитектура»	

№	Код (Название)	Тип интеграции	Интегрируемая система	Примечание
30.	RelationChainOfferConsumerArchitecture (Получение связей Цепочка-сервис из Сфера.Архитектура)	Чтение из Kafka	«Сфера. Архитектура»	
Настройки для разовой загрузки данных (могут быть отключены после применения)				
31.	GetServiceStructure (Получение сервисной структуры из «Сфера. Команда»)	Первичная загрузка из REST	«Сфера. Команда»	<p>Для корректной работы интеграции необходимо запросить у системы «Сфера. Команды» доступ к требуемому запросу (GET api/v1/service-structure) для используемой в интеграции ТУЗ.</p> <p>В преобразовании данных "Преобразование сервиса в набор назначений" в переходах из контейнера "Связи" в контейнеры: "ФП - manager", "ФП - supportManager", "ФП - owner", "ФП - coordinator", "ФП - chainManager", "ФП - chainOrganized" в поле "Ключ подстановки" указать код сервисной роли, используемый в «Сфера. Команда» на используемом стенде (коды могут отличаться на разных стендах).</p>
32.	GetEmployees (Получение сотрудников по REST)	Первичная загрузка из REST	«Сфера. Команда»	Для корректной работы интеграции необходимо запросить у системы «Сфера. Команды» доступ к требуемому запросу (GET api/v1/employees) для используемой в интеграции ТУЗ.
33.	GetStream (Получение стримов по REST)	Первичная загрузка из REST	«Сфера. Команда»	Для корректной работы интеграции необходимо запросить у системы «Сфера. Команды» доступ к требуемому запросу (GET /api/v1/teams) для



№	Код (Название)	Тип интеграции	Интегрируемая система	Примечание
				используемой в интеграции ТУЗ.
34.	GetFuncPosition (Получение функциональных позиций по REST)	Первичная загрузка из REST	«Сфера. Команда»	<p>Для корректной работы интеграции необходимо запросить у системы «Сфера. Команды» доступ к требуемому запросу (GET api/v1/func-positions) для используемой в интеграции ТУЗ.</p> <p>В преобразовании данных "Преобразование сервиса в набор назначений" в переходах из контейнера "Связи" в контейнеры: "ФП - manager", "ФП - supportManager", "ФП - owner", "ФП - coordinator", "ФП - chainManager", "ФП - chainOrganized" в поле "Ключ подстановки" указать код сервисной роли, используемый в «Сфера. Команда» на используемом стенде (коды могут отличаться на разных стендах).</p>
35.	GetUsers (Получение пользователей по REST)	Первичная загрузка из REST	«Сфера. Профили»	<p>Для корректной работы интеграции необходимо запросить у системы «Сфера. Доступы и Роли» следующие доступы для используемой в интеграции ТУЗ:</p> <ul style="list-style-type: none"> <li>• Редактор профилей пользователей (Канал)</li> <li>• Читатель пользователь ДиР <ul style="list-style-type: none"> <li>- PPCG</li> <li>- PPCN</li> </ul> </li> <li>• Читатель профилей пользователей (Канал).</li> </ul>
36.	GetCodeCalendar s (Получение кодов календарей)	Первичная загрузка из REST	«Сфера. Главная страница»	Для корректной работы интеграции необходимо запросить у системы «Сфера. Доступы и Роли» следующие

№	Код (Название)	Тип интеграции	Интегрируемая система	Примечание
				доступы для используемой в интеграции ТУЗ: Название роли – Администратор главной страницы канал. Объекты доступа - PPHM_ADM_PROD_CAL Производственные календари.
37.	prodCalendars (Выгрузка календаря)	REST	«Сфера.Главная страница»	

**Таблица 4-8 – Кастомные настройки интеграций Каталога сервисов**

№	Код (Название)	Тип интеграции	Интегрируемая система	Примечание
1.	TransferMatrixFor PPSL (Передача матрицы ДИЗ в расчетный модуль)	REST	«Сфера. Расчётный модуль SLA»	
2.	TransferContainer sForTemplate (Передача контейнера ДИЗ в обращения)	REST	«Сфера. Обращения»	
3.	AppealControlCon tainerImportAcq WrapperDinamic (Импорт контейнера шаблонов и матрицы нормативных сроков)	Чтение из Kafka	ДИЗ*	
4.	AppealControlCon tainerImportAcq WrapperStatusDi namic (Отправка статуса по импорту контейнера шаблона и матрицы	Запись из Kafka	ДИЗ*	

№	Код (Название)	Тип интеграции	Интегрируемая система	Примечание
	нормативных сроков)			

\*-система заказчика, осуществляющая внешнее создание шаблонов Обращений по арг.

**Таблица 4-9** – Настройки интеграций Каталога сервисов для модуля "Процесс взаимодействия с системами учёта для построения отчётности по предоставляемым услугам"

№	Код (Название)	Тип интеграции	Интегрируемая система	Примечание
1.	OfferCheckListConsumer (Получение статуса чек-листа по сервису)	Чтение из Kafka	«Сфера. Согласования»	Перед включением необходимо проверить и актуализировать значения полей "Kafka окружение (чтение)" и "Имя топика" (указать имя топика или глобальную переменную с названием топика в качестве значения, при выборе глобальной переменной её необходимо предварительно создать в справочнике "Глобальные переменные").
2.	updateStatusCheckList (Обновление статуса чек-листа)	Первичная загрузка из REST	«Сфера. Каталог сервисов»	Процесс "Обновление статуса чек-листа" (updateStatusCheckList) по крону запускает данную интеграцию для получения данных. Крон может быть перенастроен.
3.	TechProcessKafkaProducer (Отправка технологических процессов в Kafka)	Запись в Kafka	«Сфера. Каталог сервисов»	

Набор интеграций может быть изменён и дополнен в соответствии с требованиями бизнес-процесса.

При кастомизации интеграций также требуется обратить внимание на связанные конфигурируемые объекты системы и проверить необходимость изменения их параметров для сохранения работоспособности интеграции (см. **Таблица 5-3**).

Интеграции, которых нет в списке, но предоставлялись в более ранних версиях, - рекомендуется выключить.

#### 4.2.3.3.5 Настройка шаблонов выгрузки паспортов объектов для «Сфера. Каталог сервисов».

При необходимости использования выгрузки паспортов базовых объектов Каталога сервисов их требуется загрузить в систему.

Если используются базовые шаблоны выгрузки паспортов объектов, могут быть использованы файлы шаблонов, включённые в комплект поставки (см. **Таблица 4-10**).

Если используются кастомизированные шаблоны выгрузки, следует применить доработанные файлы шаблонов. В этом случае поставщик не отвечает за работоспособность данного функционала.

Если сценарий не используется потребителем, шаг может быть пропущен.

Шаг необходимо повторять при каждом обновлении конфигурации связанных представлений объектов. В противном случае после обновления конфигурации связь с шаблоном нарушится и выгрузка не будет осуществляться.

**Таблица 4-10** – Шаблоны паспортов объектов

№	Файл	Описание
	ServicePassportTemplateTmp.xlsx  ServicePassportTemplateTmp.xlsx	Шаблон паспорта сервиса
	ManualServicePassportTemplateTmp.xlsx  ManualServicePassportTemplateTmp.xlsx	Шаблон паспорта ручного сервиса
	ChainPassportTemplateTmp.xlsx  ChainPassportTemplateTmp.xlsx	Шаблон паспорта цепочки сервисов

#### **Активация шаблона выгрузки паспорта объекта – пошаговая инструкция.**

Настройку требуется провести, если используется сценарий выгрузки паспорта объектов «Сервис», «Ручной сервис», «Цепочка сервисов» (базовый сценарий).

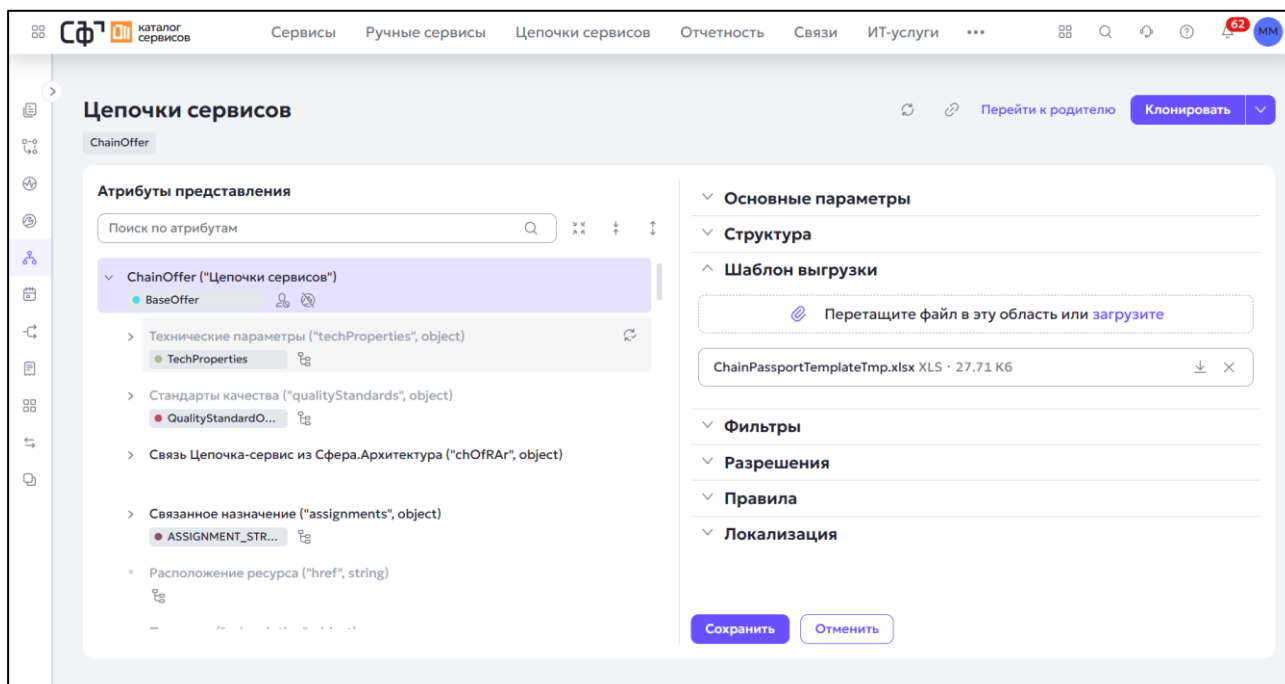
Настройка проводится стандартно для каждого объекта.

Порядок действий для администратора:

1. Зайти в раздел «Администрирование» в подраздел «Управление конфигурациями».
2. Создать новую сессию.
3. Войти в новую сессию.
4. Зайти в подраздел «Представление объектов» раздела «Администрирование».
5. Найти в списке нужной для настройки представление объектов. Для сервиса – «Сервисы» (SourceOffer), для ручного сервиса – «Ручной сервис» (ManualOffer), для цепочки сервисов – «Цепочки сервисов» (ChainOffer).
6. Открыть представление двойным кликом левой кнопки мыши.
7. Выделить родительскую запись представления (верхняя в списке) одинарным кликом левой

кнопки мыши.

8. Выбрать блок «Шаблон выгрузки» в правой части и раскрыть его.



**Рис. 4-2** – Блок «Шаблон выгрузки» в родительской записи представления объекта

6. Переместить файл шаблона выгрузки в область загрузки или прикрепить его как вложение по кнопке «Загрузить».
7. Убедиться, что название файла появилось в блоке вложенных объектов.
8. Нажать «Сохранить».
9. Перейти в детальное представление сессии.
10. Применить сессию.

#### **⚠ Предупреждение.**

Загрузку следует выполнять даже в том случае, если в поле ниже отображается информация о файле, загруженном ранее. Старый файл можно удалить кнопкой (X), а можно этого и не делать – новый файл при загрузке затрёт старый.

В дальнейшем ранее загруженный файл при необходимости можно скачать (кнопкой со стрелкой), отредактировать и снова загрузить.

Ограничение: размер файла шаблона не должен превышать 50 кб.

#### **4.2.3.3.6 Настройка производственного календаря для «Сфера. Каталог сервисов».**

Производственный календарь необходим для работы систем и используется:

- В «Сфера. Каталог сервисов» – для расчёта графиков Регламентного времени работы (РВР) сервисов.

### Загрузка производственного календаря для «Сфера. Каталог сервисов».

Загрузка производственных календарей осуществляется посредством интеграции с «Сфера. Главная страница».

Порядок действий можно выполнить как в рамках тенанта, так и в рамках сессии.

Если требуется настройка интеграции, то необходимо предварительно создать сессию и войти в неё. Перед загрузкой производственных календарей необходимо убедиться, что в интеграциях GetCodeCalendars (Получение кодов календарей) и prodCalendar (Выгрузка календаря) корректно заполнены стенодозависимые атрибуты (URL и Тип авторизации), проверить, что интеграции включены. Для корректной работы этих интеграций необходимо запросить у системы «Сфера. Доступы и Роли» следующие доступы для используемой в интеграции ТУЗ:

- Название роли – Администратор главной страницы канал.
- Объекты доступа - PPHM\_ADM\_PROD\_CAL Производственные календари.

Порядок действия для администратора:

1. В разделе «Администрирование» перейти в подраздел «Интеграции».
2. В интеграции GetCodeCalendars перевести тогл в активное состояние. Данная интеграция получает справочные данные о календарях и записывает в базу данных в таблицу references. Так же просмотреть информацию по календарям можно в справочнике «Коды календарей» (codeCalendar).
3. В интеграции prodCalendar перевести тогл в активное состояние. Данная интеграция получает производственные календари и записывает в базу данных в таблицу production\_calendars.
4. В разделе «Администрирование» перейти в подраздел «Процессы».
5. Найти процесс «Загрузка календаря» (CalendarDownload) и перейти в его карточку.
6. Выполнить ручной запуск процесса. При запуске заполнить входные данные значением: {"year": "<год, за который необходимо выгрузить календари>"}

**Например:**

```
{  
  "year": "2025"  
}
```

7. Дождаться успешного выполнения процесса.
8. Проверить наличие записей в справочнике «Коды календарей» (codeCalendar).
9. Проверить наличие записей в базе данных в таблице production\_calendars.

#### 4.2.3.4 Обновление ролей для «Сфера. Каталог сервисов».

Необходимо зайти в подраздел «Представление объектов» раздела «Администрирование» и нажать кнопку «Обновить»-«Роли» (находится под мультикнопкой).

Данное действие перезаполнит универсальную «Роль Администратора» (dove\_administrator)

данными по администратору системы, заполненному на шаге **4.2.3.3.2 Настройка справочника «Пользовательские роли» (securityRole) для «Сфера. Каталог сервисов».**

По итогу операции, если всё сработало корректно, в представлениях не должно остаться разрешений с ролью «Роль Администратора» (dove\_administrator).

#### 4.2.3.5 Обновление политик безопасности в системе ДиР.

Обновить политики безопасности в системе «Сфера. Доступы и Роли» одним из способов, соответствующих текущей настройке системы и степени кастомизации конфигурации:

1. Если настроено автоматическое обновление политик безопасности ДиР в разделе «Администрирование» системы нажать на кнопку «Обновить политики безопасности». Политики обновятся в соответствии с составом конфигурации, установленной в системе. Способ применим только для «Сфера. Каталог сервисов».
2. Если автоматическое обновление политик безопасности не настроено, получить данные для импорта в ДиР с помощью REST-запросов. Полученные данные перенести в файл с разрешением \*.json. Файлы импортировать в систему ДиР через админ-панель. Политики обновятся в соответствии с составом конфигурации, установленной в системе.
3. Допустимым также является способ прямой загрузки предварительно подготовленных файлов в систему ДиР через админ-панель, если на среде разворачивания системы используются те же настройки, что и на среде-источнике. Файлы готовятся по методике, описанной в п. 2.

Все способы обновления политик подробно описаны в разделе **5.1.1 Настройка политик безопасности.**

##### Особенности обновления политик безопасности из сессии.

При обновлении политик важно помнить, что в ДиР отправятся политики из сессии, активной для пользователя, запустившего обновление. Эти политики применятся также и к арендатору среды, так как ДиР пока не поддерживает сессионность продуктов.

#### 4.2.3.6 Проверка корректности обновления внутри сессии для «Сфера. Каталог сервисов».

Необходимо проверить появление в системе "Сфера. Каталог сервисов" изменений и новых объектов по системным и бизнесовым справочникам (дельту изменений можно увидеть при сравнении предыдущей версии конфигурации с новой в разделе "Управление конфигурациями" - для этого нужно сравнить арендатор с новой сессией после импорта конфигураций, подробнее о процедуре см. **6.4 Управление конфигурациями для «Сфера. Каталог сервисов».**).

Убедиться в доступности ключевых функций в рамках сессии. Открыть UI системы, проверить доступность разделов. Убедиться, что при открытии разделов видно их содержимое, не выводятся сообщения об ошибках.

#### 4.2.3.7 Перевод взаимодействия с «Сфера. Доступы и Роли» в эксплуатационный режим.

Если система находилась в локальном режиме интеграции с системой Доступы и Роли, необходимо вернуть её в эксплуатационный режим. Для этого в сервисе Catalog установить значение переменной SECURITY\_VALIDATOR\_TYPE= external.

Если в рамках тестирования не планируется проверять роли пользователей, этот шаг можно опустить или провести позже (когда потребность в функционировании ролей есть).

#### 4.2.3.8 Смоук-тест в рамках сессии для «Сфера. Каталог сервисов».

Тестирование проводится внутри сессии. Для этого нужно войти в эту сессию под всеми перечисленными ролями. Подробнее о процедуре

Если взаимодействие с «Сфера. Доступы и Роли» не было переведено в эксплуатационный режим, то все пользователи будут действовать с правами роли «Администратор».

Под ролью «Администратор»:

1. Залогиниться в систему.
2. Открыть раздел «Администрирование».
3. Открыть каждый подраздел «Администрирования».
4. Во всех подразделах доступны интерфейсы, не выходят ошибки 403, 500, 503, 404, 409, доступны кнопки создания и редактирования объектов (подробнее – см. Руководство пользователя «Сфера. Каталог сервисов», раздел 5.9).
5. Открыть последовательно разделы «Сервисы», «Ручные сервисы», «Цепочки сервисов», «Отчётность», «Функциональные области», «ИТ-услуги», «Продукты».
6. Во всех разделах доступен интерфейс, доступны кнопки создания и редактирования объектов, не выходят ошибки 403, 500, 503, 404, 409.

Под ролью «Редактор»:

1. Залогиниться в систему.
2. Открыть последовательно разделы «Сервисы», «Ручные сервисы», «Цепочки сервисов», «Отчётность».
3. Во всех разделах доступен интерфейс, доступны кнопки создания и редактирования объектов, не выходят ошибки 403, 500, 503, 404, 409.

Под ролью «Просмотр»:

1. Залогиниться в систему.
2. Открыть последовательно разделы «Сервисы», «Ручные сервисы», «Цепочки сервисов», «Отчётность».
3. Во всех разделах доступен интерфейс, не выходят ошибки 403, 500, 503, 404, 409.



#### 4.2.3.9 Применение сессии для «Сфера. Каталог сервисов».

После выполнения всех предыдущих шагов требуется применить сессию, чтобы изменения стали доступны для всех пользователей. Подробнее о процедуре см. **6.4 Управление конфигурациями для «Сфера. Каталог сервисов».**

#### 4.2.3.10 Рестарт поды.

Выполнить рестарт поды с сервисом Catalog («Сфера. Каталог сервисов»).

#### 4.2.4 Переменные окружения «Сфера. Каталог сервисов».

##### 4.2.4.1 Переменные сервиса catalog.

**Таблица 4-11** — Основные переменные окружения сервиса catalog, «Сфера. Каталог сервисов»

№	Параметр	Назначение	Добавлен в версии	Пример	Значение по умолчанию
<b>Параметры БД Catalog</b>					
1.	CATALOG_DB_URL	URL базы данных Postgres			jdbc:postgresql://<БД хост>:<порт>/ppcs_catalog
2.	CATALOG_DB_USERNAME	Имя пользователя с правами на создание схемы			postgres
3.	CATALOG_DB_PASSWORD	Пароль пользователя с правами на создание схемы			postgres
4.	CATALOG_DEFAULT_SCHEMA	Схема			ppcs_catalog
5.	MIGRATION_ENABLED	Флаг включения миграций (в положении true новые таблицы создаются, запускаются скрипты из ликвибэйз при установке релиза)			true
<b>Параметры подключения к общеканальному брокеру сообщений Kafka</b>					
6.	CATALOG_SERVICE_KAFKA	Сервера Kafka			http://localhost:9092
7.	CATALOG_KAFKA_SECURITY_PROTOCOL	Протокол обмена данными			PLAINTEXT

№	Параметр	Назначение	Добавлен в версии	Пример	Значение по умолчанию
8.	CATALOG_TRUSTSTORE_PATH	Путь к хранилищу открытых ключей			
9.	CATALOG_TRUSTSTORE_PASSWORD	Пароль к хранилищу открытых ключей			
10.	CATALOG_TRUSTSTORE_TYPE	Тип хранилища открытых ключей			
11.	CATALOG_KEYSTORE_PATH	Путь к хранилищу частных ключей			
12.	CATALOG_KEYSTORE_PASSWORD	Пароль к хранилищу частных ключей			
13.	CATALOG_KEYSTORE_TYPE	Тип хранилища частных ключей			
<b>Настройки SSE</b>					
14.	DOMAIN_ADDRESS	Полный адрес расположения каталога			https://<адрес системы>
15.	DOMAIN_ENABLE	Флаг включения того, учитывать ли адрес при формировании сообщений в сервис нотификаций			true
16.	SSE_ENABLED	Флаг включения формирования сообщений в SSE			true
17.	SSE_ENDPOINT	Точка подключения к SSE			/api/v1/sse
18.	SSE_TIMEOUT	Время пересоздания подключения SSE в случае долгого непоступления сообщения (пинга), в минутах			5 #minutes

№	Параметр	Назначение	Добавлен в версии	Пример	Значение по умолчанию
19.	SSE_PING	Частота, с которой запрашивается статус SSE-подключения, в секундах			*/15 * * * * *
<b>Настройки хранилища S3</b>					
20.	BUCKET	Бакет			d5-ppcs-common-bucket
21.	STORAGE_URL	URL S3 файлового хранилища			http://127.0.0.1:19000
22.	STORAGE_ACCESS_KEY	Ключ доступа к хранилищу			
23.	STORAGE_SECRET_KEY	Секретный ключ доступа к хранилищу			
24.	S3_DEFAULT_REGION	Регион подключения к сервису S3 по умолчанию (окружение может не использовать для подключения этот атрибут – в этом случае установка и смена значения не будет ни на что влиять)	1.40.0	us-east-1	us-east-1
<b>Настройки аутентификации Catalog</b>					
25.	AUTHENTICATION_TYPE	Тип аутентификации			user_password_client_id_client_secret
26.	AUTHENTICATION_URL	Адрес сервера аутентификации			https://<адрес хоста сервера аутентификации>
27.	AUTHENTICATION_USERNAME	Пользователь ТУЗ			ppcs_tua
28.	AUTHENTICATION_PASSWORD	Пароль ТУЗ			
29.	AUTHENTICATION_CLIENT_ID	Идентификатор (мнемокод) каталога сервисов			ppcs

№	Параметр	Назначение	Добавлен в версии	Пример	Значение по умолчанию
30.	AUTHENTICATION_CLIENT_SECRET	Пароль доступа КС к серверу аутентификации			
31.	AUTHENTICATION_GRANT_TYPE	Способ доступа КС к серверу аутентификации			password
32.	AUTHENTICATION_TOKEN_URL	Адрес сервера получения технического токена			/api/auth/openid-connect/token
33.	CATALOG_TOKEN_CACHE_ENABLE	Флаг включения кэширования токена			true
34.	CATALOG_TOKEN_CACHE_EXPIRE_MINUTES	Период, через который сбрасывается кэш токена (в минутах)			15
<b>Настройки интеграции с «Сфера. Аудит»</b>					
35.	SFERA_AUDIT_ENABLED	Флаг включения аудита			true
36.	SFERA_AUDIT_ADAPTER_URL	URL аудит-адаптера			http://localhost:8002/api
37.	SFERA_AUDIT_INITIATOR_SUB	Инициатор процесса аудита			ADMIN
38.	SFERA_AUDIT_INITIATOR_SERVICE	Сервис-инициатор процесса аудита			Service
39.	SFERA_AUDIT_SOURCE	Источник данных для аудита			Sfera.Catalog
<b>Настройки интеграции с «Сфера. Доступы и Роли»</b>					
40.	DIR_URL	URL сервиса ДиР			https://<адрес хоста сервиса>/app/ppam
41.	SECURITY_VALIDATOR_TYPE	Настройка, чтобы управлять ориентацией на политики ДиР: <ul style="list-style-type: none"> <li>local – не учитывать ДиР (админ-мод);</li> <li>external – учитывать ДиР.</li> </ul>			local — при настройке. Предпочтительнее вести настройку первоначально с отключённым ДиР. external — при эксплуатации в рабочей среде. Режим

№	Параметр	Назначение	Добавлен в версии	Пример	Значение по умолчанию
					обеспечивает безопасность.
42.	DEFAULT_PERMISSIONS	Список операций, доступных по умолчанию при установке роли в блоке «Разрешения», уровень «Ролям».	1.37.0		create, get, update, delete, list, find, patch, deleteAll, findObject, bulkCreate, bulkUpdate
<b>Настройки интеграции с «Сфера. Обращения» и «Сфера. Расчётный модуль»</b>					
43.	PPSL_ADDRESS	URL Расчётного модуля			https://<адрес хоста расчётного модуля>/app/calculation
44.	APPEAL_ADDRESS	URL сервиса Обращений			https://<адрес хоста сервиса обращений>/app/sd/api/v0.1
45.	APPEAL_CLIENT_CONNECT_TIMEOUT	Таймаут соединения для клиента обращений			60 000 мс
46.	APPEAL_CLIENT_READ_TIMEOUT	Таймаут запроса для клиента обращений			60 000 мс
<b>Настройки интеграции с «Сфера. Нотификации»</b>					
47.	NOTIFICATION_KAFKA_BOOTSTRAP_SERVERS	Сервера Kafka			
48.	NOTIFICATION_KAFKA_TOPIC	Топик для отправки сообщений нотификаций			
49.	NOTIFICATION_KAFKA_ENABLE	Флаг включения отправки			false
50.	NOTIFICATION_KAFKA_KEY_PASSWORD	Пароль для хранилища ключей			
51.	NOTIFICATION_KAFKA_KEYSTORE_PATH	Путь к хранилищу приватных ключей			
52.	NOTIFICATION_KAFKA_KEYSTORE_PASSWORD	Пароль к хранилищу			

№	Параметр	Назначение	Добавлен в версии	Пример	Значение по умолчанию
		приватных ключей			
53.	NOTIFICATION_KAFKA_KEYSTORE_TYPE	Тип хранилища приватных ключей			JKS
54.	NOTIFICATION_KAFKA_TRUSTSTORE_PATH	Путь к хранилищу открытых ключей			
55.	NOTIFICATION_KAFKA_TRUSTSTORE_PASSWORD	Пароль к хранилищу открытых ключей			
56.	NOTIFICATION_KAFKA_TRUSTSTORE_TYPE	Тип хранилища открытых ключей			JKS
57.	NOTIFICATION_KAFKA_SECURITY_PROTOCOL	Протокол обмена данными			SSL
58.	NOTIFICATION_KAFKA_IDEMPOTENCE	Флаг включения идемпотентности Kafka			false
<b>Настройки интеграции с «Сфера. Согласования»</b>					
59.	APPROVAL_NEW_ROUTES	Флаг включения схемы выбора маршрутов согласования: по схеме с использованием правил (true) или без них (false)			true
60.	APPROVAL_KAFKA_BOOTSTRAP_SERVERS	Сервера Kafka			
61.	APPROVAL_KAFKA_TOPIC	Топик для отправки сообщений согласований			2069_APPROVALS_EVENT_FOR_CATALOG_SERVICES, 2069_APPROVALS_EVENT_FOR_CONTAINER_MANAGEMENT
62.	APPROVAL_KAFKA_GROUP_ID	Идентификатор группы в Kafka			
63.	APPROVAL_KAFKA_ENABLED	Флаг включения интеграции по Kafka			false
64.	APPROVAL_KAFKA_KEY_PASSWORD	Пароль для хранилища ключей			

№	Параметр	Назначение	Добавлен в версии	Пример	Значение по умолчанию
65.	APPROVAL_KAFKA_KEY_STORE_PATH	Путь к хранилищу приватных ключей			
66.	APPROVAL_KAFKA_KEY_STORE_PASSWORD	Пароль к хранилищу приватных ключей			
67.	APPROVAL_KAFKA_KEY_STORE_TYPE	Тип хранилища приватных ключей			JKS
68.	APPROVAL_KAFKA_TRUSTSTORE_PATH	Путь к хранилищу открытых ключей			
69.	APPROVAL_KAFKA_TRUSTSTORE_PASSWORD	Пароль к хранилищу открытых ключей			
70.	APPROVAL_KAFKA_TRUSTSTORE_TYPE	Тип хранилища открытых ключей			JKS
71.	APPROVAL_KAFKA_SECURITY_PROTOCOL	Протокол обмена данными			PLAINTEXT
72.	USE_STREAM_FROM_TEAMS	Флаг переключения на использование стримов из «Сфера. Команды» в согласованиях			false
<b>Настройки приложения «Сфера. Каталог сервисов»</b>					
73.	RETRY_PROCESS_INSTANCE_DELAY_MINUTES	Расписание повторных попыток для ошибочных процессов			0*/10****
74.	RETRY_PROCESS_INSTANCE_COUNT	Количество повторных попыток для ошибочных процессов			5
75.	CATALOG_TOPIC_NAME	Топик выгрузки Сервисов			service-update
76.	LINK_TOPIC_NAME	Топик выгрузки Связей			link-update

№	Параметр	Назначение	Добавлен в версии	Пример	Значение по умолчанию
77.	CONFIG_TOPIC_NAME	Топик выгрузки соответствия Сервиса ID внешней системы			config-unit-update
78.	KAFKA_UPDATE_CRON	Расписание проведения выгрузки списка Сервисов			0 0 0 * * *
79.	KAFKA_CONFIG_UPDATE_CRON	Расписание			0 0/10 * * * *
80.	KAFKA_UPDATE_ENABLE	Флаг включения выгрузки			true
81.	ADDITIONAL_NODE_QUANTITY	Количество используемых нод	1.43.0	1	1
82.	NODE_SYNC_TIMEOUT	Частота синхронизации между нодами в миллисекундах	1.43.0	16000	16000

#### 4.2.4.2 Переменные сервиса data-consumer-service.

**Таблица 4-12** — Основные переменные окружения сервиса data-consumer-service, «Сфера. Каталог сервисов»

№	Параметр	Назначение	Добавлен в версии	Пример	Значение по умолчанию
<b>Параметры БД Data Consumer</b>					
1.	DATA_CONSUMER_DB_URL	URL базы данных postgres			jdbc:postgresql://<БД хост>:<порт>/ppcs_data_consumer
2.	DATA_CONSUMER_DB_USERNAME	Имя пользователя			postgres
3.	DATA_CONSUMER_DB_PASSWORD	Пароль пользователя			postgres
4.	SERVICE_KAFKA	Сервера Кафка			kafka-kafka-bootstrap.shared:9092
<b>Параметры окружения Data Consumer</b>					
5.	DOMAIN_ADDRESS	Полный адрес расположения Каталога сервисов			https://<адрес системы>



№	Параметр	Назначение	Добавлен в версии	Пример	Значение по умолчанию
<b>Настройки аутентификации Data Consumer</b>					
6.	AUTHENTICATION_URL	URL сервера аутентификации			https://<адрес хоста сервера аутентификации>
7.	AUTHENTICATION_USERNAME	Пользователь ТУЗ аутентификации			ppcs_tua
8.	AUTHENTICATION_PASSWORD	Пароль ТУЗ аутентификации			

#### 4.2.4.3 Переменные сервиса catalog-gateway.

**Таблица 4-13** — Основные переменные окружения сервиса catalog-gateway, «Сфера. Каталог сервисов»

№	Параметр	Назначение	Добавлен в версии	Пример	Значение по умолчанию
1.	SFERA_SUBO_SERVICE_URL	URL backend			http://catalog:8080
2.	SFERA_SUBO_SERVICE_URL_DATA_CONSUMER	URL сервиса DataConsumer			http://data-consumer-service:8080
3.	SFERA_SUBO_DATA_CONSUMER_PATH	Префикс к пути к сервису DataConsumer			dc
4.	SFERA_SUBO_APPROVAL_URL	Адрес к сервису Согласования			https://<адрес хоста сервиса>

#### 4.2.4.4 Переменные сервиса catalog-ui.

**Таблица 4-14** — Основные переменные окружения сервиса catalog-ui, «Сфера. Каталог сервисов»

№	Параметр	Назначение	Добавлен в версии	Пример	Значение по умолчанию
1.	API_HOST	Название сервиса бэкенда			catalog
2.	API_PORT	Порт, на котором запущен бэкенд			8080
3.	API_PROTOCOL	Протокол			http

**Предупреждение.**

В штатном случае переменные веб-клиента, приведённые в таблице выше, менять не следует. Они используются разработчиками Системы только в случае необходимости специальной индивидуальной настройки.

## 4.3 Проверка корректности установки

**Таблица 4-15** — Проверка доступности сервисов

№	Проверка	Методика проверки
1.	Авторизация в консоли системы оркестровки контейнеризированных приложений	Вход в консоль системы оркестрации контейнеризированных приложений под учётной записью специалиста сопровождения
2.	Убедиться, что необходимые pod'ы запущены и находятся в статусе "Running"	Проверить наличие запущенных подов в соответствии с архитектурой решения. Проверить отсутствие циклических перезапусков подов, проверить консоль каждого пода на наличие множественных записей об ошибках.
3.	Убедиться в доступности ключевых функций	<p>Для системы «Сфера. Каталог сервисов»:</p> <p>Под ролью «Администратор»:</p> <ol style="list-style-type: none"> <li>1. Залогиниться в систему.</li> <li>2. Открыть раздел «Администрирование».</li> <li>3. Открыть каждый подраздел «Администрирования».</li> <li>4. Во всех подразделах доступны интерфейсы, не выходят ошибки 403, 500, 503, 404, 409, доступны кнопки создания и редактирования объектов (подробнее – см. Руководство пользователя «Сфера. Каталог сервисов», раздел 5.9).</li> <li>5. Открыть последовательно разделы «Сервисы», «Ручные сервисы», «Цепочки сервисов», «Отчётность», «Функциональные области», «ИТ-услуги», «Продукты».</li> <li>6. Во всех разделах доступен интерфейс, доступны кнопки создания и редактирования объектов, не выходят ошибки 403, 500, 503, 404, 409.</li> </ol> <p>Под ролью «Редактор»:</p> <ol style="list-style-type: none"> <li>1. Залогиниться в систему.</li> </ol>

№	Проверка	Методика проверки
		<ol style="list-style-type: none"> <li>Открыть последовательно разделы «Сервисы», «Ручные сервисы», «Цепочки сервисов», «Отчётность».</li> <li>Во всех разделах доступен интерфейс, доступны кнопки создания и редактирования объектов, не выходят ошибки 403, 500, 503, 404, 409.</li> </ol> <p>Под ролью «Просмотр»:</p> <ol style="list-style-type: none"> <li>Залогиниться в систему.</li> <li>Открыть последовательно разделы «Сервисы», «Ручные сервисы», «Цепочки сервисов», «Отчётность».</li> <li>Во всех разделах доступен интерфейс, не выходят ошибки 403, 500, 503, 404, 409.</li> </ol>

## 5 Конфигурирование продукта.

### 5.1 Первоначальная настройка.

Когда установка продукта окончена, предполагается, что настройка основных переменных и сценариев уже была произведена в ходе выполнения шагов из раздела **4 Инструкция по установке продукта**. Если не все шаги были выполнены, необходимо вернуться к ним и проверить.

После изменения значений переменных необходим ребут поды сервиса.

После изменения значений системных справочников и настроек интеграций необходимо не забыть применить сессию, также рекомендуется провести ребут поды.

#### 5.1.1 Настройка политик безопасности.

В рамках первоначальной настройки «Сфера. Каталог сервисов» необходимо провести выгрузку ролей и политик безопасности в общеплатформенный сервис «Сфера. Доступы и Роли» (ДиР). Это выполняется с помощью предварительно полученной технической учётной записи.

Ниже приведено описание возможных вариантов передачи обновления ролей в ДиР. Они приведены в порядке возрастания степени автоматизации:

1. Базовый вариант, без автоматизации. Пользователь берёт готовые файлы, подготовленные на другом окружении с использованием метода из пунктов 2 или 3 для основных ролей системы, и импортирует их в ДиР при помощи интерфейса администрирования системы ДиР (см. ниже, Раздел **5.1.1.1. Обновление политик в ДиР через импорт готовых файлов json**).

Плюсы – самый быстрый способ, который разблокирует базовую функциональность.

Минусы связаны с тем, что в самой системе конфигурируются доступы. При загрузке готовых файлов собственные настройки системы не будут переданы в ДиР. И наоборот, если собственные настройки системы уже были переданы в ДиР ранее (по варианту 2 или 3), то они будут заменены загрузкой готовых файлов. Кроме того, при использовании файла, подготовленного на другом окружении (например, тестовом), необходимо учитывать разницу объектов сред. Перед загрузкой файла ролей и доступов необходимо убедиться в сопоставимости объектов, ролей и доступов окружения-источника и окружения для импорта.

2. Пользователь самостоятельно формирует файлы ручной выгрузкой из системы (при помощи REST-запросов), затем импортирует эти файлы в ДиР (см. ниже, **5.1.1.2. Обновление политик в ДиР через импорт файлов json, выгруженных вручную**).

Плюсы – настройки, передаваемые в ДиР, уже будут отражать собственные настройки, сделанные в системе.

Минусы – при каждом обновлении настроек в системе необходимо повторять эту ручную операцию, что занимает много времени.

3. Пользователь настраивает процесс автоматического обновления ролей и политик безопасности в ДиР. После успешной настройки очередное обновление выполняется простым нажатием кнопки в интерфейсе Каталога сервисов (см. ниже, **5.1.1.3. Каталог сервисов. Обновление политик в ДиР автоматически из UI**).

Плюсы – самый удобный способ для регулярного обновления политик.

Минусы – требуется предварительная настройка.

#### 5.1.1.1 Обновление политик в ДиР через импорт готовых файлов json.



При установке системы с помощью готовых файлов пользователю надо взять файлы формата json, содержащие данные о базовых ролях и доступах, и импортировать их в систему ДиР при помощи пользовательского интерфейса администрирования ДиР:

1. Зайти в админ-панель «Доступы и роли».
2. Импортировать файл `import_ObjectsActions*.json` в разделе «Разрешения».
3. Импортировать файл `import_Roles*.json` в разделе «Роли».

Актуальные файлы json не входят в комплекты поставки Каталога сервисов и Расчётного модуля. Для самостоятельной подготовки файлов в процессе установки необходимо воспользоваться инструкций из п. **5.1.1.2 Обновление политик в ДиР через импорт файлов json, выгруженных вручную.**

В **Таблица 5-1** приведены примеры структуры файлов, которые можно использовать для проверки импорта ДиР. Файлы нельзя использовать в качестве рабочих и актуальных файлов для загрузки политик безопасности.

**Таблица 5-1** – Примеры готовых файлов для загрузки в ДиР

Система	Маска имени файла	Экземпляр файла
Каталог сервисов	<code>import_ObjectsActions_ppcs*.json</code>	 <code>import_ObjectsActions_ppcs_09.09.2025.json</code>
	<code>import_Roles_ppcs*.json</code>	 <code>import_Roles_ppcs_09.09.2025.json</code>

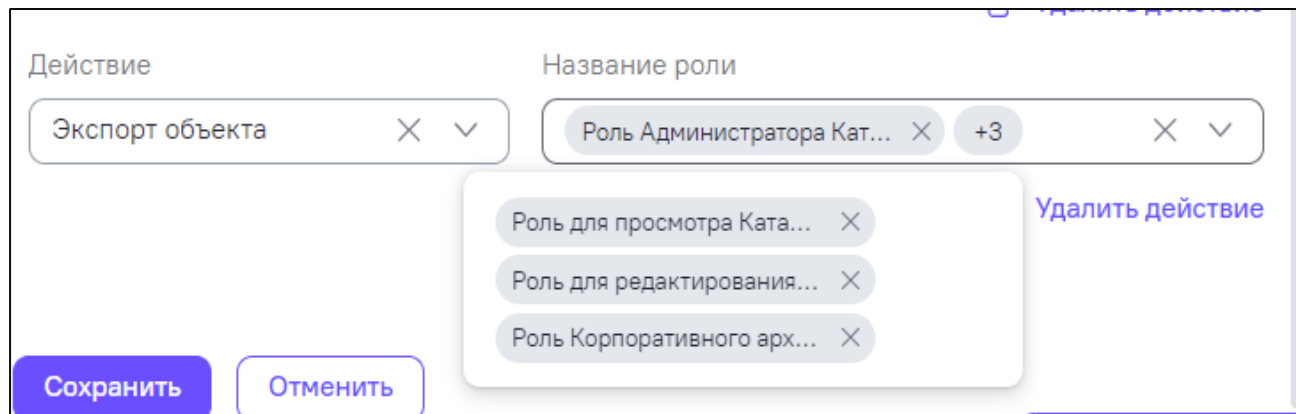
Для именования созданных файлов рекомендуется использовать приведённые в таблице маски, где:

- `<import или export>` – направление действия с файлом, относительно сервиса ДиР:  
`import` – файл, предназначенный для загрузки в ДиР;  
`export` – файл, выгружаемый из ДиР;
- `<Roles, ObjectsActions и т.п.>` – объекты (роли, операции и т.п.), в отношении которых транслируются политики безопасности;
- `<ppcs, ppsl и т.п.>` – код системы, политики безопасности которой транслируются в/из ДиР;
- `<*>` – детализированные реквизиты файла, например, дата формирования, порядковый номер и т.п.

### 5.1.1.2 Обновление политик в ДиР через импорт файлов json, выгруженных вручную.

Администратор Каталога сервисов может создавать/модифицировать новые представления объектов (OV, ObjectView) и новые поля в представлениях. При этом администратору нужно сконфигурировать преднастройки доступов ролей для объектов, чтобы создать файлы json для их импорта в ДиР.

Преднастройки представлений объектов типа «Сервис» и «Цепочка сервисов», чтобы были доступны конкретные действия при выгрузке в ДиР:



**Рис. 5-1** – Роли для действия "Экспорт объекта"

Порядок действий для администратора:

1. Создать новую сессию конфигурирования.
2. Войти в новую сессию конфигурирования.
3. Выполнить настройки в Каталоге сервисов в разделе "Администрирование", подраздел «Представление объекта», блок "Разрешения";
4. Инициировать обновление политик безопасности REST-запросом:
5. PUT <https://<catalogBackendIngress>/api/v1/security/updatePolicy>
6. В результате обновления меняются правила распределения доступа (access control) на бэкенде КС, генерируются новые политики (строки для файлов `import_ObjectsActions_ppcs*.json`, `import_Roles_ppcs*.json`).
7. Отправить в КС REST-запросы с методом GET:
8. GET <https://<catalogBackendIngress>/api/v1/security/action> — для получения общего списка действий;
9. GET <https://<catalogBackendIngress>/api/v1/security/actionPermission> — для получения списка разрешений.
10. Полученные ответы в формате json скопировать.
11. Скопированные ответы вставить в файлы, файлы загрузить в ДиР через процедуру импорта (процедура загрузки аналогична описанной в **5.1.1.1. Обновление политик в ДиР через импорт готовых файлов json**).
12. По завершению всех настроек применить сессию конфигурирования.

### ❗ Особенности обновления политик безопасности из сессии.

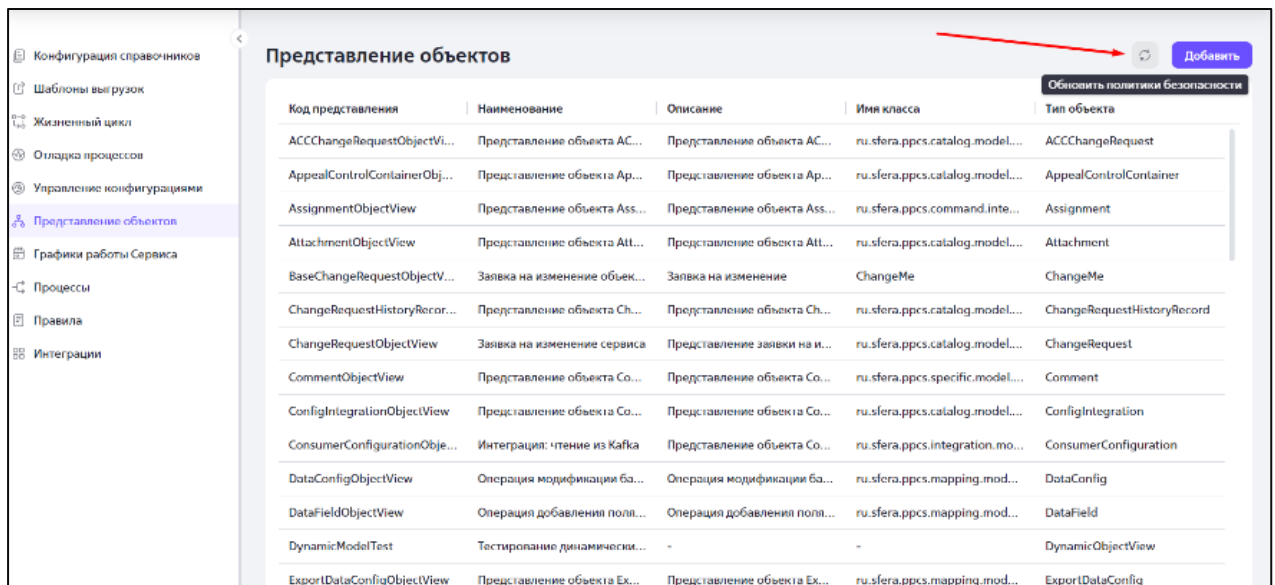
При обновлении политик важно помнить, что в ДиР отправятся политики из сессии, активной для пользователя, отправившего REST-запрос. Эти политики применяются также и к тенанту среды, так как ДиР пока не поддерживает сессионность продуктов.

#### 5.1.1.3 Каталог сервисов. Обновление политик в ДиР автоматически из UI.

Для реализации возможности автоматически обновлять роли и политики безопасности в ДиР необходимо предварительно выполнить настройки в Каталоге сервисов, или убедиться в их наличии, если эти настройки были выполнены ранее (см. **5.1.1.3.1 Настройка автоматического обновления политик безопасности через UI.**).

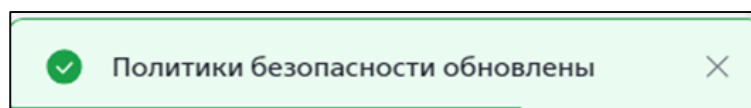
Порядок действий для администратора:

1. Создать новую сессию конфигурирования.
2. Войти в сессию конфигурирования.
3. Выполнить настройки в Каталоге сервисов в разделе "Администрирование", подраздел «Представление объекта», блок "Разрешения" в соответствии с требованиями бизнес-процессов.
4. Нажать кнопку «Обновить политики безопасности» в подразделе «Представление объектов» раздела Администрирования «Каталога сервисов».



**Рис. 5-2** – Кнопка обновления в списке представлений

5. В случае успеха процесса - политики обновятся, на экран выйдет уведомление-тостер:



**Рис. 5-3** – Уведомление

6. Применить сессию конфигурирования.

В разделе ДиР "Роли" следует проверить, что появились права доступа к новому объекту, флаги расставлены в соответствии с доступами в представлении объекта.

В случае неуспеха процесса можно обновить политики вручную – методом, описанным выше.

#### 5.1.1.3.1 Настройка автоматического обновления политик безопасности через UI.

Необходимые шаги:

1. Проверить наличие в справочнике (подраздел «Конфигурации справочников», раздел «Администрирование») "Переменные среды" (environmentVariable) 2 переменные: updateDirEnableAutoRetry (значение по умолчанию – false) и updateDirSystemCode (значение по умолчанию – PPCS).
2. Проверить наличие в справочнике (подраздел «Конфигурации справочников», раздел «Администрирование») "Глобальные переменные" (globalVariables) переменную: app.security.forceUpdate. Обновление политик через UI доступно если в дополнительных деталях записи указан json

```
{  
    "value": true
```

```
}
```

3. Проверить наличие процесса "Отправка объектов в ДиР и обновление используемых плиток" (dirUpdate) в списке процессов (подраздел «Процессы», раздел «Администрирование»).
4. Проверить настройки ТУЗ для обращения в ДиР.

AUTHENTICATION\_USERNAME - переменная в КС для задания имени пользователя для авторизации.

AUTHENTICATION\_USERNAME: ppcs\_tua - пример заполнения.

Этого пользователя нужно найти в админ-панель ДиР и задать следующие права в разделе «Доступы» для ТУЗ:

- Редактор назначений ДиР (Канальное приложение), объект доступа - "PPCS -Каталог сервисов".
  - Редактор типов объектов ДиР (Канальное приложение), объект доступа - "PPCS - Каталог сервисов".
5. В справочнике (подраздел «Конфигурации справочников», раздел «Администрирование») "Пользовательские роли" (securityRoles) настроить id в ДиР для отправки запроса.
    - Найти id ролей Каталога сервисов в ДиР (взять можно из консоли - у ДиР нет публичного API на предоставление этих данных).
    - Для этого открыть в админ-панели ДиР раздел «Роли», нажать F12, выбрать запрос roles во вкладке network, выбрать вкладку response. Найти все роли КС в списке (можно использовать поиск по сочетанию «ppcs»).



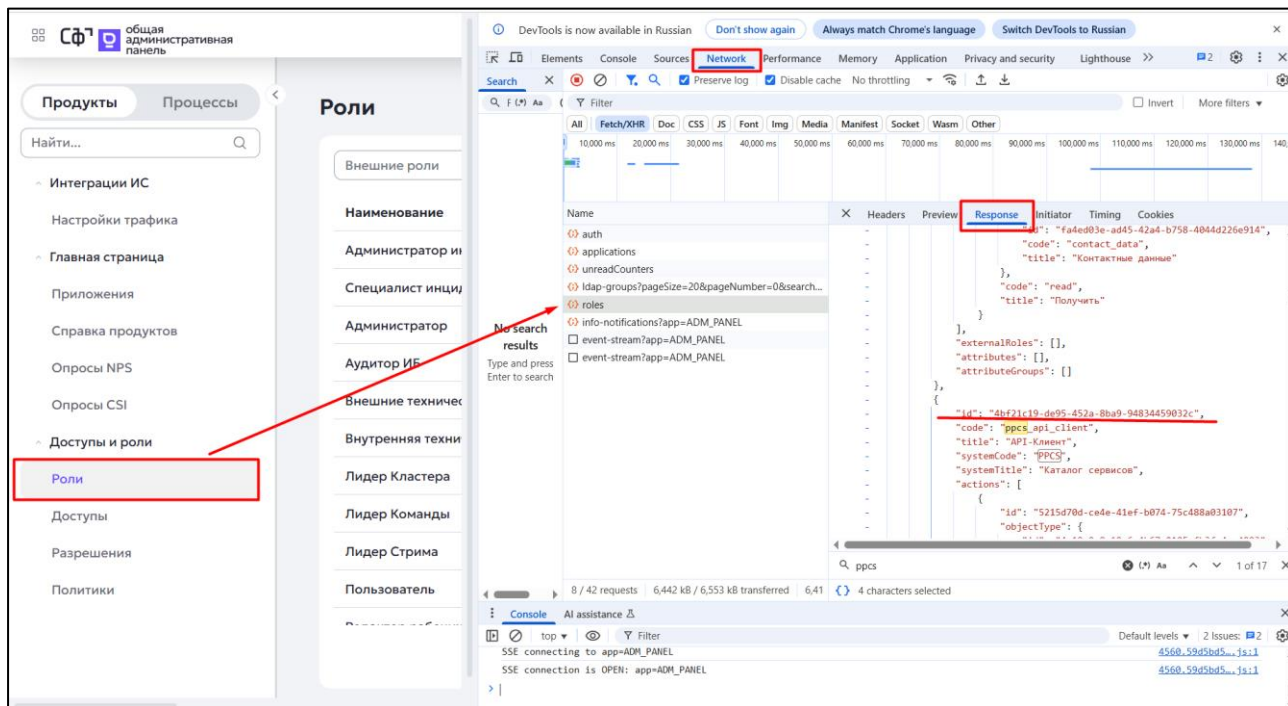


Рис. 5-4– Поиск id ролей системы в админ-панели ДиР

- В справочнике (подраздел «Конфигурации справочников», раздел «Администрирование») "Пользовательские роли" (securityRoles), в поле "Дополнительные детали справочника" добавить для каждой роли:

```
{
  "roleId": "UUID роли"
}
```

где UUID роли - id из п.5

- Проверить в ДиР у роли Администратора (в разделе «Роли» админ-панели ДиР) наличие галочки для действия «Управление созданием шаблонов процессов (process\_template)» -> «Выполнить процесс (execute)».

#### ❗ Особенности обновления политик безопасности из сессии.

При обновлении политик важно помнить, что в ДиР отправятся политики из сессии, активной для пользователя, запустившего обновление. Эти политики применяются также и к тенанту среды, так как ДиР пока не поддерживает сессионность продуктов.

## 5.2 Интеграции.

В «Сфера. Каталог сервисов» интеграции с системами настраиваются динамически – с помощью инструментов администрирования и конфигурации.

Предварительные настройки интеграций проводятся при установке системы – см. **4.2.3.3.4 Настройка интеграций для «Сфера. Каталог сервисов»**. Система также предоставляет возможность настройки кастомизированных интеграционных подключений и внесения изменений в сценарии текущих интеграций.

Полное описание полей для настройки интеграций и сценарии настройки можно найти в Руководстве пользователя «Сфера. Каталог сервисов», раздел **5.9.10 Интеграции** и раздел **6.5 Настройки интеграций**.

### 5.2.1 Общие настройки интеграций.

*Таблица 5-2 – Интеграции Каталога сервисов*

Система-партнёр по обмену данными	Позиция системы-партнёра в обмене данными	Способ интеграции	Передаваемые данные	Документация по системе-партнёру
<b>Необходимые интеграции</b>				
«Сфера. Аутентификация»	Источник	REST	Токен	<a href="#">PPAU</a>
«Сфера. Интеграции ИС»	Источник	REST	Настройка гэйтвэя для работы интеграций других систем	<a href="#">[ИИС] PPPA</a>
«Сфера. Главная страница»	Источник	REST	Главная страница и переход на неё	<a href="#">PPHM</a>
		REST	Список календарей, загруженных в ОПС	
		REST	Информация по запрашиваемому календарю (рабочие, выходные, праздничные дни)	
«Сфера. ДиР»	Получатель	REST	Данные по ролям (политики по ролям)	<a href="#">PPRM</a>
	Источник	REST	Данные по ролям (политики по ролям)	
«Сфера. Конфигурации»	Получатель	Kafka	Данные для создания КЕ, связей	<a href="#">PPCM</a>
	Источник	Kafka	CI (подтверждение создания КЕ, связи, ответы на запросы из КС на	

Система-партнёр по обмену данными	Позиция системы-партнёра в обмене данными	Способ интеграции	Передаваемые данные	Документация по системе-партнёру
			создание/модификацию)	
		Kafka	Список РИС (далее – список KE)	
«Сфера. Аудит»	Получатель	REST	Эвенты, информация о событиях в системе	<a href="#">[АУДИТ] PPPA</a>
«Сфера. Нотификации»	Получатель	Kafka	События о публикации Сервиса (может быть настроено)	<a href="#">PPMN</a>
<b>Интеграции второго порядка (в зависимости от сценариев заказчика)</b>				
«Сфера. Расчётный модуль»	Получатель	REST	Создание и модификация контейнера (смена статусов), модель контейнера	<a href="#">PPSL</a>
	Источник	Kafka	Информация о матрице (модель)	
«Сфера. Обращения»	Получатель	REST	Создание и модификация контейнера (смена статусов), модель контейнера	<a href="#">PPSD</a>
		Kafka	Список сервисов	
	Источник	Kafka	Информация о шаблоне (модель)	
«Сфера. Согласования»	Получатель	REST	Информация для инициализации контейнера согласования (выбор маршрута согласования, ID сервиса и др.) Информация о необходимости смены статуса согласования при получении связей	<a href="#">PPAP</a>

Система-партнёр по обмену данными	Позиция системы-партнёра в обмене данными	Способ интеграции	Передаваемые данные	Документация по системе-партнёру
			из «Сфера. Архитектура».	
	Источник	Kafka	События о смене статусов согласования	
«Сфера. Архитектура»	Получатель	Kafka	Список сервисов	<a href="#">PPAN</a>
«Сфера. Поиск»	Получатель	Kafka	Информация об объектах поиска	<a href="#">PPSE</a>
«Сфера. Команды»	Источник	Kafka	Информация о списке стримов	<a href="#">TEAM</a>
			Информация о списке сотрудников	
			Информация о списке назначений	
		REST	Информация о списке стримов (разовая загрузка)	
			Информация о списке сотрудников (разовая загрузка)	
			Информация о списке назначений (разовая загрузка)	

При изменении настроек интеграций в зависимости от требований процесса необходимо обратить внимание на связанные с интеграцией конфигурируемые объекты. Помимо внесения изменений в сами объекты интеграций, требуется проверить необходимость внесения изменений в конфигурируемые объекты для сохранения работоспособности интеграции при кастомизации.

**Таблица 5-3** – Связанные конфигурируемые объекты интеграций Каталога сервисов

№	Интеграция	Описание взаимодействия	Перечень связанных конфигурируемых объектов
1.	Сообщения от Согласований (ApprovalContainer)	При изменении статуса контейнера Согласований приходит обновление в общеплатформенную кафку от системы «Сфера. Согласования». При корректном получении обновлений по согласованиям виджет согласований корректно инициализируется и обновляет статус.	Представления объектов: <ul style="list-style-type: none"><li>Сообщение от Согласований (ApprovalStateChangeMessage)</li><li>Контейнер Согласования (ApprovalContainer)</li></ul> Преобразования данных: <ul style="list-style-type: none"><li>Преобразование сообщения от согласований в контейнер (approvalMessageToContainer)</li></ul>
2.	Отклонение согласования от ТУЗ (RejectApproval)	При получении связей Цепочка-Сервис, Сервис-Сервис из системы «Сфера.Архитектура» система архивирует текущие связи, архивирует заявки на изменение связей и отклоняет согласование заявок, у которых ApprovalContainer в статусе ON_APPROVE (На согласовании).	Представления объектов: <ul style="list-style-type: none"><li>RequestRejectApproval (RequestRejectApprovalObjectView)</li></ul>
3.	Получение кодов календарей (GetCodeCalendars)	Получение списка доступных календарей для загрузки из ОПС (коды календарей)	Представления объектов: <ul style="list-style-type: none"><li>Ответ на запрос списка календарей</li><li>Справочник</li></ul> Преобразование данных: <ul style="list-style-type: none"><li>Преобразование коллекции календарей</li><li>Загрузка календарей по их кодам из справочника</li></ul>

№	Интеграция	Описание взаимодействия	Перечень связанных конфигурируемых объектов
			<ul style="list-style-type: none"><li>Расчёт рабочих часов</li></ul>
4.	Выгрузка календаря (prodCalendar)	Получение календаря (рабочие, выходные, праздничные дни)	<p>Представления объектов:</p> <ul style="list-style-type: none"><li>Производственный календарь</li></ul> <p>Преобразование данных:</p> <ul style="list-style-type: none"><li>Расчёт рабочих часов</li><li>Преобразование месяца</li><li>Преобразование дня</li></ul>

#### 5.2.1.1 Настройка нового подключения к Kafka.

##### Порядок настройки подключения.

- В настройках deployment найти переменную JAVA\_OPTS
- В переменной JAVA\_OPTS указать параметры требуемые для настройки подключения. Формат указания переменных: -Dapp.kafka.<указать название подключения>.<указать название переменной>=<указать значение переменной>

Примечание:

- Название подключения должно быть уникальным. Название должно состоять из английских символов. У всех параметров, которые относятся к одному подключению, указывать одно и то же название подключения.
  - Название и значение переменной берём из документации к настройке подключения к конкретной Kafka.
- После добавления значения перезапустить бэкенд.
  - В «Сфера. Каталог сервисов» перейти в справочник Kafka (чтение).
  - Добавить новую запись в справочнике:
    - Перед добавление записи необходимо создать сессию и войти в нее.
    - После добавления записи необходимо применить сессию.
    - Идентификатор записи: указать значение в формате <название подключения указанное в JAVA\_OPT>ListenerFactory
    - Название записи: произвольное
  - В «Сфера . Каталог сервисов» перейти в справочник Kafka (запись).
  - Добавить новую запись в справочнике:

- Идентификатор записи: указать значение в формате <название подключения указанное в JAVA\_OPT>ProducerFactory
- Название записи: произвольное

#### **Примеры добавления параметров в JAVA\_OPT.**

-Dapp.kafka.approval.bootstrap-servers=10.228.64.22:30093,10.228.64.23:30093,10.228.64.25:30093

-Dapp.kafka.approval.security.protocol=PLAINTEXT

### **5.2.1.2 Запуск настроенной интеграции на чтение данных из топигов Kafka.**

Необходимые шаги:

1. Проверить необходимые изменения в интеграции, сохранить конфигурацию, перезапустить поду Catalog.
2. Запустить интеграцию.

Для включения интеграции необходимо найти требуемую интеграцию в табличном представлении в подразделе "Интеграции" раздела "Администрирование" и изменить состояние тогла с "Не активен" на "Активен". Интеграция будет запущена – данные начнут обновляться.

3. Проверить поступление данных.

Если есть доступ в Kafka системы-источника – можно предварительно проверить наличие новых сообщений в ней.

Открыть в базе данных таблицу, созданную на основе представления объекта, выбранного в поле «Представление интеграционного ответа» или «Выход». В таблице должны появиться данные, полученные из системы-источника.

### **5.2.1.3 Запуск настроенной интеграции по первичной загрузке данных по REST.**

Необходимые шаги:

1. В табличном списке интеграций перевести интеграцию в состояние «Активна» (переместить тогл на значение «Включено»).
2. В табличном списке процессов найти процесс для ручного запуска интеграции.
3. В карточке процесса для запуска интеграции нажать кнопку «Запустить».
4. В открывшейся форме на ввод входных данных оставить в качестве значения пустой json и нажать «Запустить». Выйдет сообщение о старте процесса. В логах можно увидеть запрос и ответ на него. Если ошибок нет, на запрос придёт ответ с кодом 201. Если есть ошибки, на запрос придёт ответ с кодом ошибки, процесс будет остановлен.
5. Открыть в базе данных таблицу, созданную на основе представления объекта, выбранного в поле «Представление интеграционного ответа» или «Выход». В таблице должны появиться данные, полученные из системы-источника. Данные начинают появляться до окончания процесса полной загрузки. Когда процесс загрузки будет полностью завершён, выйдет уведомление об окончании процесса. Если данных нет (запрос вернул пустую страницу), система завершит запрос и отобразит уведомление об окончании процесса.

## 5.2.2 Интеграция с «Сфера. Ядро».

### 5.2.2.1 Встраивание в Главную страницу платформы Сфера.

Для встраивания продукта в Главную страницу платформы Сфера выполните следующие шаги:

1. Войдите на платформу Сфера с ролью Администратор.
2. В навигационном меню выберите «Общая Административная панель».
3. Откройте раздел «Главная страница», а затем подраздел «Приложения».
4. Нажмите на кнопку «Встроить приложение» и заполните поля, используя значения из **Таблица 5-4** для системы «Сфера. Каталог сервисов».
5. После добавления приложения необходимо в админ-панели «Сфера. Доступы и Роли» в Роли или через Доступы отметить объект «PPCS - Каталог сервисов» для роли «Пользователь главной страницы (Канал)».

**Таблица 5-4** — Параметры встройки в Главную страницу платформы Сфера для «Сфера. Каталог сервисов».

№	Параметр	Значение
1.	Тип приложения	APP
2.	Название группы приложения	Эксплуатация
3.	Название приложения	Каталог сервисов
4.	Код приложения	PPCS
5.	После указанной даты приложение не отображается как «новое»	Дата после которой приложение не будет отображаться как «новое» в навигационном меню. Указать актуальную дату
6.	Путь к странице приложения	catalog
7.	Описание приложения	Единый инструмент для всех процессов эксплуатации ПО: ИТ-активы, технологии и сервисы
8.	Публичное	Нет
9.	Общий путь	Нет
10.	Путь к иконке приложения	app/catalog/logo.svg
11.	Тип встройки	script
12.	Имя контейнера с микрофронтонтом	sfera_service_catalog_container
13.	Ссылка на файл с микрофронтомом	app/catalog/remoteEntry.js
14.	Точка входа в приложение в файле	microFrontend

### 5.2.3 Журналирование.

Сервисы логируют информацию в стандартный вывод. Формат используется стандартный для системы логирования Log4j2.



Все сервисы ИС для журналирования используют библиотеку Log4j2. Log4j2 — это мощная и гибкая библиотека для логирования в Java, которая является стандартным решением для Spring Boot. Log4j2 поддерживает различные способы записи логов, включая консоль, файлы и удалённые системы логирования.

### 5.2.3.1 Уровни логирования.

В Log4j2 используются следующие уровни логирования.

**Таблица 5-5** — Уровни логирования

Уровень	Описание
TRACE	Самый детальный уровень. Используется для отладки и отслеживания выполнения кода.
DEBUG	Менее детальный, чем TRACE, но также предназначен для отладки.
INFO	Общая информация о работе приложения.
WARN	Предупреждения о возможных проблемах.
ERROR	Ошибки, которые требуют внимания.

### 5.2.3.2 Рекомендации.

1. Логирование можно включать с помощью переменных среды, например: LOGGING\_LEVEL\_RU\_SFERA\_PPCS\_CATALOG: DEBUG (чтобы перевести наименование пакета, необходимо заменить. на \_, привести значение к верхнему регистру, добавить префикс LOGGING\_LEVEL\_)
8. Для включения логирования бизнес-логики всего приложения без общедоступных библиотек и фреймворков, можно настроить логирование для префикса пакета: ru.sfera. Все пакеты, содержащие этот префикс, будут логироваться с заданным уровнем логирования.
9. Установите уровень логирования INFO или WARN для большинства пакетов, чтобы минимизировать нагрузку.

### 5.2.4 Аудит.

Аудит осуществляется посредством библиотеки компонента системы общеплатформенных сервисов «Сфера. ЖАМТ» (Журналирование, Аудит, Мониторинг, Трассировка).

Системой предусмотрена возможность аудита событий.

Аудит позволяет фиксировать события вызова методов API в отдельную базу данных для хранения в возможной дальнейшей обработки. При вызове методов API формируется событие, которое посредством механизма transactional outbox попадает по http-запросу в адаптер, который фиксирует данные в базе данных.

#### 5.2.4.1 Хранение событий аудита.

Для хранения событий аудита используется таблица audit\_events.

Структура таблицы событий аудита:

**Таблица 5-6** — Структура таблицы событий аудита

Наименование	Описание
id	Идентификатор записи аудита. Первичный ключ.
code	Код события.
created_at	Дата – время создания записи аудита.
content	Описание события в форме json.

#### 5.2.4.2 Список событий аудита.

Ниже приведён список событий аудита, которые сохраняются в таблице audit\_events.

**Таблица 5-7** — Список событий аудита «Сфера. Каталог сервисов»

Действие	Описание
VIEW_SERVICE_RECORD	Просмотр карточки сервиса
UPDATE_SERVICE_RECORD	Редактирование карточки сервиса
EXPORT_CATALOG_DATA	Выгрузка данных из каталога сервисов
GENERAL_EVENT_PROPERTIES_LOG	Журналирование значений конфигураций сервиса

В системе «Сфера. Каталог сервисов» также добавлена таблица audit\_endpoints в качестве обёртки для настройки аудируемых событий разработчиком.

#### 5.2.4.3 Структура поля content с деталями события.

Ниже приведён список полей объекта аудита, которые сохраняются в поле audit\_events.content.

**Таблица 5-8** — Список полей объекта аудита

Код поля	Описание
method	http-метод запроса
localAddr	Локальный ip
remoteAddr	Вызывающий ip
forwardedAddr	Список переадресованных ip
userLogin	Логин УЗ или ТУЗ
requestUri	Путь запроса
serviceName	Наименование сервиса
securityType	Тип сообщения
status	Статус события – успех или ошибка
errorMessage	Сообщение об ошибке

Пример события разрешения доступа:

```
{
  "method": "GET",
  "status": "success",
  "localAddr": "1.0.6.82",
  "userLogin": "adminUser",
  "JSESSIONID": "AE60343F75A0F96DAF209594559CBE21",
  "remoteAddr": "1.0.7.196",
  "requestUri": "/api/v0.1/entities",
  "serviceName": "Service",
  "securityType": "OTHER",
  "forwardedAddr": "127.0.0.1"
}
```

Пример события запрещения доступа:

```
{
  "method": "GET",
  "status": "accessDenied",
  "localAddr": "1.0.6.82",
  "JSESSIONID": "284FF3CB3DB0B0EEDF723D7CD7AA7930",
  "remoteAddr": "1.0.6.192",
  "requestUri": "/api/v0.1/entities",
  "serviceName": "Service",
  "errorMessage": "No user with login 'unknownUser' exists",
  "forwardedAddr": "127.0.0.1,127.0.0.1"
}
```

#### 5.2.4.4 Настройка аудита с http-вызовом.

После настройки установки и настройки адаптера аудита необходимо указать следующие переменные окружения для сервиса (catalog – для «Сфера. Каталог сервисов»):

**Таблица 5-9** — Переменные окружения для подключения Аудита

Переменная окружения	Значение по умолчанию	Описание
SFERA_AUDIT_ENABLED	true	Флаг включения аудита
SFERA_AUDIT_ADAPTER_URL	http://localhost:8002/api	URL аудит-адаптера
SFERA_AUDIT_INITIATOR_SUB	ADMIN	Инициатор процесса аудита
SFERA_AUDIT_INITIATOR_SERVICE	Service	Сервис-инициатор процесса аудита (значение поля channel из

Переменная окружения	Значение по умолчанию	Описание
		преобразованного в читаемый вид jwt-токена на данном инстансе продукта)
SFERA_AUDIT_SOURCE	Sfera.Catalog	Источник данных для аудита. Значение должно совпадать с переменной окружения SFERA_AUDIT_ADAPTER_INFORMATION_SYSTEMS_i_KEY в адаптере аудита sfera-local-audit-adapter

### 5.2.5 Мониторинг.

Осуществляется посредством сервиса Мониторинг - компонента системы общеплатформенных сервисов «Сфера. ЖАМТ» (Журналирование, Аудит, Мониторинг, Трассировка).

Системой предусмотрена возможность мониторинга событий.

Сбор метрик должен выполняться через стандартное spring-boot приложение с подключённой spring-boot actuator.

Основные плановые мероприятия по мониторингу и поддержанию работоспособности системы подразделяются на следующие группы:

Мониторинг доступности — проверка доступности ИС "Каталог сервисов", "Расчётный модуль SLA" и составляющих их сервисов;

Мониторинг производительности — проверка допустимости нагрузки на сервисы и компоненты ИС;

Функциональный мониторинг — проверка правильности выполнения функций ИС.

**Таблица 5-10** — Процедуры проверки работоспособности системы и локализации неработающих компонентов

Содержание мероприятия	Периодичность	Ответственный
Мониторинг доступности		
Мониторинг логов на наличие сообщений, имеющих отношение к доступности. Обращать внимание на ошибки и предупреждения, связанные с высоким использованием и переполнением памяти.	Ежедневно	Прикладной администратор ИС
Мониторинг состояния веб-узла	Ежедневно	Прикладной администратор ИС
Проверка срока действия SSL сертификата	Ежедневно	Прикладной администратор ИС
Мониторинг производительности		

Содержание мероприятия	Периодичность	Ответственный
Проверка лога сервера на наличие сообщений, имеющих отношение к производительности	Ежедневно	Прикладной администратор ИС
Анализ таблиц, содержащих логи	Ежедневно	Прикладной администратор ИС
Функциональный мониторинг		
Проверка правильности выполнения функций системы исходя из данных Журналирования	Ежедневно	Прикладной администратор ИС

### 5.2.5.1 Мониторинг сервисов

Мониторинг системы статуса работы реализован с помощью стандартных средств spring-actuator.

MANAGEMENT\_ENDPOINT\_HEALTH\_PROBES\_ENABLED – Включает или выключает эндпоинты проверки состояния приложения.

MANAGEMENT\_ENDPOINTS\_WEB\_EXPOSURE\_INCLUDE – отвечает за включение и выключение эндпоинтов проверки состояния приложения. По умолчанию содержит health,info,prometheus,metrics,loggers.

На мониторинге находятся следующие сервисы:

- catalog.

Параметры мониторинга микросервисов:

- доступность: проверка доступности микросервиса для клиентов осуществляется с помощью мониторинга HTTP статусов ответов. Целевое значение доступности – не менее 99.99% времени.
- производительность: измерение времени отклика на запросы пользователей. Целевое значение времени отклика – менее 200 мс по 99 перцентиле.
- нагрузка: мониторинг нагрузки на микросервис для определения его способности обрабатывать запросы. Количество зависит от количества запросов коллег, которые интегрированы на проде.
- использование ресурсов: отслеживание использования CPU, памяти и сети микросервисом.
- логирование: сбор и анализ логов для выявления ошибок и проблем в работе микросервиса. Логи должны содержать информацию о запросах, ошибках и успешных операциях.
- мониторинг зависимостей: отслеживание взаимосвязей с другими микросервисами, такими как сервисы аутентификации, базы данных и кэширования. Цель – обнаружение проблем в связях между сервисами.

Эндпоинты для мониторинга:

- /actuator/metrics – для вывода общего списка метрик,
- /actuator/metrics/{requiredMetricName} – для вывода информации по конкретным метрикам с именем requiredMetricName,

- /actuator/prometheus – для вывода информации по всем метрикам в формате Prometheus;
- Healthcheck:
- /actuator/health/readiness – Readiness проба,
  - /actuator/health/liveness – Liveness проба.

**Таблица 5-11** — Стандартные метрики мониторинга

Наименование в actuator	Наименование в prometheus	Описание	Пример
<b>Приложение</b>			
application.ready.time	application_ready_time_seconds	Время перехода в состояние READY	{app_name="ppau-identity",app_version="@project.version@",main_application_class="ru.sfera.ppau.identity.Application"}, 36.312
application.started.time	application_started_time_seconds	Время перехода в состояние STARTED	{app_name="ppau-identity",app_version="@project.version@",main_application_class="ru.sfera.ppau.identity.Application"}, 35.804
<b>Логирование</b>			
log4j2.events	log4j2_events_total	Общее количество записанных лог-событий (по уровням: TRACE, DEBUG, INFO и т.д.)	{app_name="ppam-access",app_version="@project.version@",level="fatal"}, 0.0
<b>CPU</b>			
system.cpu.count	system_cpu_count	Общее количество доступных CPU-ядер	{app_name="ppam-access",app_version="@project.version@"}, 1.0
system.cpu.usage	system_cpu_usage	Текущая загрузка CPU в долях единицы (0–1)	{app_name="ppau-identity",app_version="@project.version@"}, 0.10843533104166667
system.load.average.1m	system_load_average_1m	Средняя нагрузка на систему за 1 минуту. Учитывает процессы в очереди	{app_name="ppau-identity",app_version="@project.version@"}, 0.2197265625
process.cpu.usage	process_cpu_usage	Загрузка CPU процессом с учётом всех ядер (диапазон [0.0–1.0], где 1.0 = 100%)	process_cpu_usage{app_name="ppam-access",app_version="@project.version@"}, 0.011428571428571429
process.files.max	process_files_max_files	Максимальное количество файловых дескрипторов, которое	{app_name="ppam-access",app_version="@project.version@"}, 65535.0

Наименование в actuator	Наименование в prometheus	Описание	Пример
		может открыть процесс (лимит ОС)	
process.files.open	process_files_open_files	Текущее количество открытых файловых дескрипторов	{app_name="ppam-access",app_version="@project.version@",} 173.0
process.start.time	process_start_time_seconds	Время старта процесса (Unix timestamp в секундах)	{app_name="ppam-access",app_version="@project.version@",} 1.745319303486E9
process.uptime	process_uptime_seconds	Время безотказной работы процесса (в секундах)	{app_name="ppam-access",app_version="@project.version@",} 8911.093
<b>Диск</b>			
disk.free	disk_free_bytes	Объем свободного места на диске (в байтах)	{app_name="ppau-identity",app_version="@project.version@",path="/opt/app/.",} 1.8106789888E10
disk.total	disk_total_bytes	Общий объем дискового пространства (в байтах)	{app_name="ppau-identity",app_version="@project.version@",path="/opt/app/.",} 8.7886127104E10
<b>БД (hikaricp)</b>			
hikaricp.connections	hikaricp_connections	Текущее общее количество соединений	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 10.0
hikaricp.connections.acquire	hikaricp_connections_acquire_seconds_max	Максимальное время ожидания соединения	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 6746.0
	hikaricp_connections_acquire_seconds_count	Общее количество запросов на получение соединения	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 0.151603137
	hikaricp_connections_acquire_seconds_sum	Суммарное время ожидания соединений (сек)	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 0.002227168
hikaricp.connections.active	hikaricp_connections_active	Активные (используемые) соединения	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 0.0
hikaricp.connections.creation	hikaricp_connections_creation_seconds_max	Максимальное время создания соединения	{app_name="ppau-identity",app_version="@project.version@",}

Наименование в actuator	Наименование в prometheus	Описание	Пример
			ect.version@",pool="HikariPool-2",} 100.0
	hikaricp_connections_creation_seconds_count	Количество созданных соединений	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 1.559
	hikaricp_connections_creation_seconds_sum	Суммарное время создания соединений	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 0.0
hikaricp.connections.idle	hikaricp_connections_idle	Неиспользуемые соединения	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 10.0
hikaricp.connections.max	hikaricp_connections_max	Максимальный размер пула	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 10.0
hikaricp.connections.min	hikaricp_connections_min	Минимальный размер пула	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 10.0
hikaricp.connections.pending	hikaricp_connections_pending	Запросы в очереди на соединение	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 0.0
hikaricp.connections.timeout	hikaricp_connections_timeout_total	Таймауты получения соединения	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 0.0
hikaricp.connections.usage	hikaricp_connections_usage_seconds_max	Максимальное время удержания соединения	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 6746.0
	hikaricp_connections_usage_seconds_count	Количество использований соединений	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 82.888
	hikaricp_connections_usage_seconds_sum	Суммарное время использования соединений	{app_name="ppau-identity",app_version="@project.version@",pool="HikariPool-2",} 0.0
<b>БД (jdbc)</b>			



Наименование в actuator	Наименование в prometheus	Описание	Пример
jdbc.connections.active	jdbc_connections_active	Текущее количество активных (используемых) jdbc соединений	{app_name="ppam-access",app_version="@project.version@",name="dataSource",} 0.0
jdbc.connections.idle	jdbc_connections_idle	Текущее количество свободных jdbc соединений	{app_name="ppam-access",app_version="@project.version@",name="dataSource",} 1.0
jdbc.connections.max	jdbc_connections_max	Максимальный размер пула для jdbc	{app_name="ppam-access",app_version="@project.version@",name="dataSource",} 1.0
jdbc.connections.min	jdbc_connections_min	Минимальный размер пула для jdbc	{app_name="ppam-access",app_version="@project.version@",name="dataSource",} 1.0
<b>БД (r2dbc)</b>			
r2dbc.pool.acquired	r2dbc_pool_acquired_connections	Количество r2dbc соединений, которые активно используются	{app_name="ppau-identity",app_version="@project.version@",name="connectionFactory",} 0.0
r2dbc.pool.allocated	r2dbc_pool_allocated_connections	Количество выделенных r2dbc соединений в пуле, которые активно используются или простаивают	{app_name="ppau-identity",app_version="@project.version@",name="connectionFactory",} 10.0
r2dbc.pool.idle	r2dbc_pool_idle_connections	Количество свободных r2dbc соединения	{app_name="ppau-identity",app_version="@project.version@",name="connectionFactory",} 10.0
r2dbc.pool.max.allocated	r2dbc_pool_max_allocated_connections	Максимальное количество возможных r2dbc соединений	{app_name="ppau-identity",app_version="@project.version@",name="connectionFactory",} 50.0
r2dbc.pool.max.pending	r2dbc_pool_max_pending_connections	Максимальное количество разрешённых ожидающих запросов по r2dbc	{app_name="ppau-identity",app_version="@project.version@",name="connectionFactory",} 2.147483647E9
r2dbc.pool.pending	r2dbc_pool_pending_connections	Текущее количество ожидающих запросов по r2dbc	{app_name="ppau-identity",app_version="@project.version@",name="connectionFactory",} 0.0

Наименование в actuator	Наименование в prometheus	Описание	Пример
<b>HTTP</b>			
http.client.requests	http_client_requests_seconds_max	Максимальное время выполнения клиентского запроса (сек)	{app_name="ppau-identity",app_version="@project.version@",error="none",http_method="POST",http_status_code="404",spring_cloud_gateway_route_id="auth",spring_cloud_gateway_route_uri="http://sfera-keycloak:8000/",} 0.0
	http_client_requests_seconds_count	Общее количество выполненных клиентских запросов	{app_name="ppau-identity",app_version="@project.version@",error="none",http_method="POST",http_status_code="404",spring_cloud_gateway_route_id="auth",spring_cloud_gateway_route_uri="http://sfera-keycloak:8000/",} 45.0
	http_client_requests_seconds_sum	Суммарное время выполнения клиентских запросов	{app_name="ppau-identity",app_version="@project.version@",error="none",http_method="POST",http_status_code="404",spring_cloud_gateway_route_id="auth",spring_cloud_gateway_route_uri="http://sfera-keycloak:8000/",} 0.347968093
http.client.requests.active	http_client_requests_active_seconds_max	Максимальное время выполнения активного клиентского запроса	{app_name="ppau-identity",app_version="@project.version@",http_method="POST",http_status_code="UNKNOWN",spring_cloud_gateway_route_id="auth",spring_cloud_gateway_route_uri="http://sfera-keycloak:8000/",} 0.0
	http_client_requests_active_seconds_active_count	Текущее количество активных клиентских запросов	{app_name="ppau-identity",app_version="@project.version@",http_method="POST",http_status_code="UNKNOWN",spring_cloud_gateway_route_id="auth",spring_cloud_gateway_route_uri="http://sfera-keycloak:8000/",} 0.0

Наименование в actuator	Наименование в prometheus	Описание	Пример
	http_client_requests_active_seconds_duration_sum	Суммарное время выполнения активных клиентских запросов	{app_name="ppau-identity",app_version="@project.version@",http_method="POST",http_status_code="UNKNOWN",spring_cloud_gateway_route_id="auth",spring_cloud_gateway_route_uri="http://sfera-keycloak:8000/",} 0.0
http.server.requests	http_server_requests_seconds_max	Максимальное время выполнения серверного запроса (сек)	{app_name="ppau-identity",app_version="@project.version@",error="none",exception="none",method="GET",outcome="SUCCESS",status="200",uri="/api/auth/login-formats",} 7.34238E-4
	http_server_requests_seconds_count	Общее количество обработанных серверных запросов	{app_name="ppau-identity",app_version="@project.version@",error="none",exception="none",method="GET",outcome="SUCCESS",status="200",uri="/api/adsync/admin/v1/sources",} 6.0
	http_server_requests_seconds_sum	Суммарное время обработки серверных запросов	{app_name="ppau-identity",app_version="@project.version@",error="none",exception="none",method="GET",outcome="SUCCESS",status="200",uri="/api/adsync/admin/v1/sources",} 0.877007972
http.server.requests.active	http_server_requests_active_seconds_max	Максимальное время выполнения активного серверного запроса	{app_name="ppau-identity",app_version="@project.version@",exception="none",method="GET",outcome="SUCCESS",status="200",uri="UNKNOWN",} 0.001276533
	http_server_requests_active_seconds_active_count	Текущее количество активных серверных запросов	{app_name="ppau-identity",app_version="@project.version@",exception="none",method="GET",outcome="SUCCESS",status="200",uri="UNKNOWN",} 1.0
	http_server_requests_active_seconds_duration_sum	Суммарное время выполнения активных серверных запросов	{app_name="ppau-identity",app_version="@project.version@",exception="none",method="GET",outcome="SUCCESS",status="200",uri="UNKNOWN",} 0.001275329

Наименование в actuator	Наименование в prometheus	Описание	Пример
<b>JVM</b>			
jvm.buffer.count	jvm_buffer_count_buffers	Количество буферов (прямых/непрямых)	{app_name="ppam-access",app_version="@project.version@",id="mapped",} 0.0
jvm.buffer.memory.used	jvm_buffer_memory_used_bytes	Память, занятая буферами (в байтах)	{app_name="ppam-access",app_version="@project.version@",id="mapped",} 0.0
jvm.buffer.total.capacity	jvm_buffer_total_capacity_bytes	Общая ёмкость буферов (в байтах)	{app_name="ppam-access",app_version="@project.version@",id="mapped",} 0.0
jvm.classes.loaded	jvm_classes_loaded_classes	Количество загруженных классов	{app_name="ppam-access",app_version="@project.version@",} 32902.0
jvm.classes.unloaded	jvm_classes_unloaded_classes_total	Общее количество выгруженных классов	{app_name="ppam-access",app_version="@project.version@",} 1114.0
jvm.compilation.time	jvm_compilation_time_ms_total	Общее время, затраченное на компиляцию (мс)	{app_name="ppam-access",app_version="@project.version@",compiler="HotSpot 64-Bit Tiered Compilers",} 170951.0
jvm.gc.live.data.size	jvm_gc_live_data_size_bytes	Размер долгоживущего пула памяти кучи после освобождения	{app_name="ppam-access",app_version="@project.version@",} 1.2201336E8
jvm.gc.max.data.size	jvm_gc_max_data_size_bytes	Максимальный размер долгоживущего пула памяти кучи	{app_name="ppam-access",app_version="@project.version@",} 1.78978816E8
jvm.gc.memory.allocated	jvm_gc_memory_allocated_bytes_total	Общий объём памяти, выделенной через сборщик мусора	{app_name="ppam-access",app_version="@project.version@",} 2.4033603E10
jvm.gc.memory.promoted	jvm_gc_memory_promoted_bytes_total	Общий объём памяти, перемещённой из младших поколений в старшее поколение кучи в процессе работы сборщика мусора	{app_name="ppam-access",app_version="@project.version@",} 8.8371704E8
jvm.gc.overhead	jvm_gc_overhead_percent	% времени CPU, затраченного на сборку мусора	{app_name="ppam-access",app_version="@project.version@",}

Наименование в actuator	Наименование в prometheus	Описание	Пример
			ct.version@",} 1.6333333333333334E-4
jvm.gc.pause	jvm_gc_pause_seconds_max	Максимальное время паузы при сборке мусора (сек)	{action="end of minor GC",app_name="ppam-access",app_version="@project.version@",cause="Allocation Failure",gc="Copy",} 0.007
	jvm_gc_pause_seconds_count	Количество сборок мусора	{action="end of minor GC",app_name="ppam-access",app_version="@project.version@",cause="Allocation Failure",gc="Copy",} 334.0
	jvm_gc_pause_seconds_sum	Суммарное время остановок в процессе сборки мусора	{action="end of minor GC",app_name="ppam-access",app_version="@project.version@",cause="Allocation Failure",gc="Copy",} 4.527
jvm.info	jvm_info	Информация о версии JVM	{app_name="ppam-access",app_version="@project.version@",runtime="OpenJDK Runtime Environment",vendor="Eclipse Adoptium",version="21.0.7+6-LTS",} 1.0
jvm.memory.committed	jvm_memory_committed_bytes	Объем выделенной памяти в байтах (heap/non-heap)	{app_name="ppam-access",app_version="@project.version@",area="nonheap",id="CodeHeap 'non-profiled nmethods'",} 2.7590656E7
jvm.memory.max	jvm_memory_max_bytes	Максимальный объем памяти в байтах, который может быть использован	{app_name="ppam-access",app_version="@project.version@",area="nonheap",id="CodeHeap 'non-profiled nmethods'",} 1.22916864E8
jvm.memory.usage.after.gc	jvm_memory_usage_after_gc_percent	Процент долгоживущего пула кучи, использованного после последнего события сборки мусора	{app_name="ppam-access",app_version="@project.version@",area="heap",pool="long-lived",} 0.68992924838658
jvm.memory.used	jvm_memory_used_bytes	Объем используемой памяти (heap/non-heap)	{app_name="ppam-access",app_version="@project.version@",area="heap",id="Eden Space",} 6.0792408E7

Наименование в actuator	Наименование в prometheus	Описание	Пример
jvm.threads.daemon	jvm_threads_daemon_threads	Текущее количество активных демон-потоков	{app_name="ppam-access",app_version="@project.version@",} 28.0
jvm.threads.live	jvm_threads_live_threads	Текущее количество всех потоков	{app_name="ppam-access",app_version="@project.version@",} 46.0
jvm.threads.peak	jvm_threads_peak_threads	Пиковое количество потоков с момента запуска	{app_name="ppam-access",app_version="@project.version@",} 54.0
jvm.threads.started	jvm_threads_started_threads_total	Общее количество созданных потоков	{app_name="ppam-access",app_version="@project.version@",} 232.0
jvm.threads.states	jvm_threads_states_threads	Текущее количество потоков по состояниям (RUNNABLE, BLOCKED и др.)	{app_name="ppam-access",app_version="@project.version@",state="timed-waiting",} 16.0
<b>Spring</b>			
spring.kafka.template	spring_kafka_template_seconds_max	Максимальное время отправки сообщения в kafka (сек)	{app_name="ppau-identity",app_version="@project.version@",exception="none",name="adReplicationKafkaTemplate",result="success",} 0.0
	spring_kafka_template_seconds_count	Количество отправленных сообщений в kafka	{app_name="ppau-identity",app_version="@project.version@",exception="none",name="adReplicationKafkaTemplate",result="success",} 6595.0
	spring_kafka_template_seconds_sum	Суммарное время отправки сообщений в kafka (сек)	{app_name="ppau-identity",app_version="@project.version@",exception="none",name="adReplicationKafkaTemplate",result="success",} 9667.713518759

**Таблица 5-12** — Использование обязательных метрик

Требуемая метрика	Используемая метрика
Состояния коннектов каждого пула (ConnectionPools)	<p>Список подходящих метрик:</p> <ul style="list-style-type: none"> <li>hikaricp_connections,</li> <li>hikaricp_connections_acquire_seconds_max,</li> <li>hikaricp_connections_acquire_seconds_count,</li> <li>hikaricp_connections_acquire_seconds_sum,</li> </ul>

Требуемая метрика	Используемая метрика
	<ul style="list-style-type: none"> <li>hikaricp_connections_active,</li> <li>hikaricp_connections_creation_seconds_max,</li> <li>hikaricp_connections_creation_seconds_count,</li> <li>hikaricp_connections_creation_seconds_sum,</li> <li>hikaricp_connections_idle,</li> <li>hikaricp_connections_max,</li> <li>hikaricp_connections_min,</li> <li>hikaricp_connections_pending,</li> <li>hikaricp_connections_timeout_total,</li> <li>hikaricp_connections_usage_seconds_max,</li> <li>hikaricp_connections_usage_seconds_count,</li> <li>hikaricp_connections_usage_seconds_sum,</li> <li>jdbc_connections_active,</li> <li>jdbc_connections_idle,</li> <li>jdbc_connections_max,</li> <li>jdbc_connections_min,</li> </ul>
Проверка работоспособности сервисов в целом(healthcheck)	<p>Реализуется через эндпоинты:</p> <ul style="list-style-type: none"> <li>/actuator/health/readiness,</li> <li>/actuator/health/liveness</li> </ul>
Нагрузка на сервис, количество запросов за одну секунду (RPS)	<p>Метрика http_server_requests_seconds_count. Собирается информация по каждому эндпоинту отдельно.</p> <p>Пример настройки в Grafana: sum(rate(http_server_requests_seconds_count[1m]))</p>
Общий объем памяти, используемый самой JVM	<p>Метрика jvm_memory_used_bytes.</p> <p>Пример настройки в Grafana: sum(jvm_memory_used_bytes)</p>
Количество тредов, находящихся в статусах: new, runnable, timed-waiting waiting, blocked	Метрика jvm_threads_states_threads
Общее использование heap memory самой JVM	<p>Метрика jvm_memory_used_bytes. Необходимо суммировать все значения с area="heap".</p> <p>Пример настройки в Grafana: sum(jvm_memory_used_bytes{area='heap'})</p>
Использование memory pools	<p>Список подходящих метрик:</p> <ul style="list-style-type: none"> <li>jvm_memory_used_bytes,</li> <li>jvm_memory_committed_bytes,</li> <li>jvm_memory_max_bytes,</li> </ul>

Требуемая метрика	Используемая метрика
	<ul style="list-style-type: none"> <li>jvm_memory_usage_after_gc_percent</li> </ul>
Значение garbage collection и время его выполнения	<p>Список подходящих метрик:</p> <ul style="list-style-type: none"> <li>jvm_gc_live_data_size_bytes,</li> <li>jvm_gc_max_data_size_bytes,</li> <li>jvm_gc_memory_allocated_bytes_total,</li> <li>jvm_gc_memory_promoted_bytes_total,</li> <li>jvm_gc_overhead_percent,</li> <li>jvm_gc_pause_seconds_max,</li> <li>jvm_gc_pause_seconds_count,</li> <li>jvm_gc_pause_seconds_sum</li> </ul>
Среднее время запроса (время ответа на запрос к api)	<p>Метрики http_server_requests_seconds_sum и http_server_requests_seconds_count.</p> <p>Пример настройки в Grafana: rate(http_server_requests_seconds_sum[1h]) / rate(http_server_requests_seconds_count[1h])</p>
Общее количество запросов к сервису	<p>Метрика http_server_requests_seconds_count.</p> <p>Пример настройки в Grafana: sum(rate(http_server_requests_seconds_count[1h]))</p>
Максимальное время запроса среди всех запросов в течение дня	<p>Метрика http_server_requests_seconds_max.</p> <p>Пример настройки в Grafana: max_over_time(http_server_requests_seconds_max[1d])</p>
Количество технических ошибок (количество 5xx ошибок при обращении к серверу)	<p>Метрика http_server_requests_seconds_count.</p> <p>Пример настройки в Grafana: sum(rate(http_server_requests_seconds_count{status="5.."}[1h]))</p>

### 5.2.6 Трассировка.

В данный момент трассировка отсутствует.

## 5.3 Миграция данных посредством импорта .xlsx – «Сфера. Каталог сервисов».

### 5.3.1 Миграция справочников.

В разделе описан процесс добавления новых справочников и наполнение их данными путём миграции данных через импорт файла формата «.xlsx».

Процесс применим для локальных справочников.

Для миграции данных в уже существующие справочники начать выполнение алгоритма с пункта 2.



Порядок действий администратора:

1. В разделе «Администрирование» на вкладке «Конфигурация справочников» добавить необходимый справочник.

Процесс добавления нового справочника описан в руководстве пользователя системы «Сфера. Каталог сервисов» в разделе **5.9.3.2. «Создание нового справочника»**.


2. Подготовить файл формата .xlsx:

- a. Название листа в файле должно соответствовать коду справочника. Коды справочников отображаются в табличном представлении в разделе «Администрирование» на вкладке «Конфигурация справочников».
- b. В ячейках первой строки листа указываются названия и json path атрибутов импортируемых записей в формате:  
`<название атрибута>#<json path>`
- c. Обязательные атрибуты: id, name, где id – идентификатор записи, name – имя записи.
- d. Все атрибуты, кроме id и name должны быть расположены внутри атрибута details. В первой строке столбца для таких атрибутов указывается значение следующего вида:  
`<название атрибута>#details.<json path атрибута>`
- e. Атрибуты key и type для каждой записи генерируются автоматически (указывать в файле не требуется):
  - i. Значение type берётся из названия листа;
  - ii. Значение key формируется по шаблону:  
`type_id`

3. Выполнить запрос POST api/v1/import/reference. В теле запроса передать сформированный файл.

- a. Проверить, что данные загружены. Для проверки можно перейти в раздел «Администрирование» на вкладку «Конфигурация справочников», найти нужный справочник, перейти в него, убедиться, что количество записей справочника совпадает с количеством записей в файле и структура записей справочника перенесена корректно.

**Таблица 5-13** – Пример файла импорта записей справочника

Файл	Описание
 ReferenceStandart.xls x	Пример файла импорта записей справочника.

### 5.3.2 Миграция сервисов и цепочек.

В разделе описан процесс миграции утверждённых сервисов / ручных сервисов / цепочек. Миграция черновиков сервисов / ручных сервисов / цепочек и неутверждённых версий этих объектов описана в следующем разделе.

Порядок действия для администратора:

### 1. Клонировать представления объектов:

- Для миграции сервисов требуется клонировать представление объекта «SourceOffer».
- Для миграции ручных сервисов требуется клонировать представление объекта «ManualOffer».
- Для миграции цепочек сервисов необходимо клонировать представление объекта «ChainOffer».

Процесс клонирования представлений объектов описан в руководстве пользователя в разделе **«5.9.6.3. Создание нового представления объекта клонированием»**.

### 2. В представлении объекта, полученном при клонировании, произвести следующие настройки:

- В настройках корневого атрибута в блоке «Структура» в поле «Тип дата сервиса для представления» выбрать значение «Базовый репозиторий».
- В настройках корневого атрибута в блоке «Разрешения» добавить действия: «Импорт объекта», «Импорт-патч объекта». Для каждого действия указать роль «Роль администратора Каталога сервисов».
- Обновить политики безопасности.

Процесс редактирования атрибутов представления объекта описан в руководстве пользователя в разделе **«5.9.6.5. Редактирование представления объекта – изменение параметров созданных полей»**.

### 3. Подготовить файл формата .xlsx. Для каждого клонированного представления объекта свой файл.

- Для каждого клонированного представления объекта свой файл.
- Первая строка в файле содержит названия атрибутов и их json path в формате:  
<название атрибута>#<json path атрибута>
- Атрибутный состав и тип данных для каждого атрибута можно посмотреть в представлении объекта, которое было получено путём клонирования.
- Если атрибут содержит в себе в качестве значения объект, то каждое поле такого объекта должно быть вынесено в отдельный столбец. В заголовке каждого столбца указывается значение в формате: <название атрибута>#<json path атрибута родителя>.<json path атрибута>
- Если атрибут содержит в себе справочное значение, то требуется указать атрибуты id, type, name:
  - В id указывается идентификатор записи справочника;
  - В name указывается значение записи;
  - В type указывается код справочника.
- Если атрибут содержит коллекцию значений, то необходимо добавить столбцы в количестве равном количеству значений атрибута. Если каждая запись в коллекции не является составным объектом, то в заголовке каждого столбца указывается значение в формате:

<название атрибута>#<json path атрибута>[номер элемента в коллекции (нумерация начинается с 0)]

Если записи коллекции представляют собой составные объекты, то в заголовке каждого столбца указывается значение в формате:

<название атрибута>#<json path объекта коллекции>[номер элемента в коллекции].<json path атрибута внутри объекта>

- Идентификаторы сервисов / ручных сервисов / цепочек, указанные в файле, должны быть уникальны. При импорте осуществляется проверка уникальности идентификаторов, если будут найдены дубли, то импорт остановится с ошибкой.
- Значение даты и времени в файле нужно указывать в формате timestamp.

**Таблица 5-14** – Обязательные поля

Поле	Json path	Описание
Идентификатор	id	Уникальный идентификатор сервиса / ручного сервиса / цепочки.
Тип объекта	model	Для сервиса – source. Для ручного сервиса – manual. Для цепочки – chain.
Наименование	name	Наименование сервиса / ручного сервиса / цепочки.
Статус	status	Статус сервиса / ручного сервиса / цепочки.
Код	code	Уникальный сервиса / ручного сервиса / цепочки.
Версия	version	Бизнесовая версия сервиса / ручного сервиса / цепочки.
Внутренняя версия	techInfo.internalVersion	Техническая версия записи.
Дата создания	techInfo.createdOn	Дата создания в формате timestamp.
Дата изменения	techInfo.updatedOn	Дата изменения в формате timestamp.
Представление объекта	techInfo.dataConfigId	Для сервиса – SourceOffer Для ручного сервиса – ManualOffer Для цепочки - ChainOffer




4. Выполнить запрос POST api/v1/object/<dataConfigId>/files/dataConfigBasedImport и передать в теле запроса сформированный файл.

а. Вместо <dataConfigId> указать идентификатор представления объекта, который был

получен при клонировании.

- б. Проверить, что данные загружены. Для проверки необходимо открыть вкладку с табличным представлением импортированного объекта. Сравнить количество записей в файле и в табличном представлении. Количество должно совпадать.

**Таблица 5-15** – Примеры файлов импорта для сервисов, ручных сервисов и цепочек

Файл	Описание
 SourceOfferStandart. xlsx	Пример файла импорта сервиса
 ManualOfferStandart. xlsx	Пример файла импорта ручного сервиса
 ChainOfferStandart.xl sx	Пример файла импорта цепочки

### 5.3.3 Миграция заявок сервисов и цепочек.

В разделе описана миграция черновиков сервисов / ручных сервисов / цепочек, неподтверждённых версий этих объектов и архивных версий этих объектов. В терминах системы такие объекты называются «Заявка».

Порядок действий администратора:

1. Клонировать представления объектов:

- Для миграции заявок сервисов требуется клонировать представление объекта «SourceOfferChangeRequestObjectView».
- Для миграции заявок ручных сервисов требуется клонировать представление объекта «ManualOfferChangeRequestObjectView».
- Для миграции заявок цепочек сервисов необходимо клонировать представление объекта «ChainOfferChangeRequestObjectView».

Процесс клонирования представлений объектов описан в руководстве пользователя в разделе **«5.9.6.3. Создание нового представления объекта клонированием»**.

2. В представлении объекта, полученном при клонировании, произвести следующие настройки:

- В настройках корневого атрибута в блоке «Структура» в поле «Тип дата сервиса для представления» выбрать значение «Базовый репозиторий».
- В настройках корневого атрибута в блоке «Разрешения» добавить действия: «Импорт объекта», «Импорт-патч объекта». Для каждого действия указать роль «Роль администратора Каталога сервисов».
- Обновить политики безопасности.

Процесс редактирования атрибутов представления объекта описан в руководстве пользователя в

### разделе «5.9.6.5. Редактирование представления объекта – изменение параметров созданных полей».

3. Подготовить файл формата .xlsx. Для каждого клонированного представления объекта свой файл.

- Для каждого клонированного представления объекта свой отдельный файл.
- Файл с заявками содержит атрибуты самого объекта (см. раздел 5.3.2) и атрибуты заявки.
- Атрибутный состав и тип данных для каждого атрибута можно посмотреть в представлении объекта, которое было получено путём клонирования.
- Первая строка в файле содержит названия атрибутов и их json path в формате:  
`<название атрибута>#<json path атрибута>`
- Если атрибут содержит в себе в качестве значения объект, то каждое поле такого объекта должно быть вынесено в отдельный столбец. В заголовке каждого столбца указывается значение в формате: `<название атрибута>#<json path атрибута родителя>.<json path атрибута>`
- Если атрибут содержит в себе справочное значение, то требуется указать атрибуты id, type, name:
  - В id указывается идентификатор записи справочника;
  - В name указывается значение записи;
  - В type указывается код справочника.
- Если атрибут содержит коллекцию значений, то необходимо добавить столбцы в количестве равном количеству значений атрибута. Если каждая запись в коллекции не является составным объектом, то в заголовке каждого столбца указывается значение в формате:  
`<название атрибута>#<json path атрибута>[номер элемента в коллекции (нумерация начинается с 0)]`  
Если записи коллекции представляют собой составные объекты, то в заголовке каждого столбца указывается значение в формате:  
`<название атрибута>#<json path объекта коллекции>[номер элемента в коллекции].<json path атрибута внутри объекта>`
- Идентификаторы заявок, указанные в файле, должны быть уникальны. При импорте осуществляется проверка уникальности идентификаторов, если будут найдены дубли, то импорт остановится с ошибкой.
- Значение даты и времени в файле нужно указывать в формате timestamp.
- Не заполнять атрибут target.techInfo если импортируется черновик объекта (сервис / ручной сервис / цепочка в системе отсутствует).



**Таблица 5-16** – Обязательные поля

Поле	Json path	Описание
Идентификатора заявки	id	Уникальный идентификатор заявки.
Статус заявки	status	Статус заявки. Указать IN_DESIGN если заявка активная. Указать ARCHIVED если заявка архивная. Активная заявка у сервиса / ручного сервиса / цепочки может быть только одна.
Идентификатор объекта	targetId	Идентификатор сервиса / ручного сервиса / цепочки. Соответствует значению поля «target.id».
Представление объекта для заявки	techInfo.dataConfigId	Для сервиса – SourceOfferChangeRequestObjectView Для ручного сервиса – ManualOfferChangeRequestObjectView Для цепочки - ChainOfferChangeRequestObjectView
Идентификатор	target.id	Уникальный идентификатор сервиса / ручного сервиса / цепочки.
Тип объекта	target.model	Для сервиса – source. Для ручного сервиса – manual. Для цепочки – chain.
Наименование	target.name	Наименование сервиса / ручного сервиса / цепочки.
Статус сервиса	target.status	Статус сервиса / ручного сервиса / цепочки.
Код	target.code	Уникальный код сервиса / ручного сервиса / цепочки.
Версия	target.version	Бизнесовая версия сервиса / ручного сервиса / цепочки.
Внутренняя версия	target.techInfo.internalVersion	Техническая версия записи. Не заполнять если импортируется черновик объекта (сервис / ручной

Поле	Json path	Описание
		сервис / цепочка в системе отсутствует).
Дата создания	target.techInfo.createdOn	Дата создания в формате timestamp. Не заполнять если импортируется черновик объекта (сервис / ручной сервис / цепочка в системе отсутствует).
Дата изменения	target.techInfo.updatedOn	Дата изменения в формате timestamp. Не заполнять если импортируется черновик объекта (сервис / ручной сервис / цепочка в системе отсутствует).
Представление объекта	target.techInfo.dataConfigId	Для сервиса – SourceOffer Для ручного сервиса – ManualOffer Для цепочки – ChainOffer Не заполнять если импортируется черновик объекта (сервис / ручной сервис / цепочка в системе отсутствует).

4. Выполнить запрос POST api/v1/object/<dataConfigId>/files/dataConfigBasedImport и передать в теле запроса сформированный файл.
  - с. Вместо <dataConfigId> указать идентификатор представления объекта, который был получен при клонировании.
  - d. Проверить, что данные загружены. Для проверки необходимо открыть вкладку с табличным представлением импортированного объекта. Сравнить количество записей в файле и в табличном представлении. Количество должно совпадать.

**Таблица 5-17** – Примеры файлов импорта для сервисов, ручных сервисов и цепочек

Файл	Описание
 SourceOfferChangeRequestStandart.xlsx	Пример файла импорта заявки сервиса
 ManualOfferChangeRequestStandart.xlsx	Пример файла импорта заявки ручного сервиса

Файл	Описание
 ChainOfferChangeRe questStandart.xlsx	Пример файла импорта заявки цепочки

### 5.3.4 Миграция связей.

В разделе описан процесс миграции подтверждённых связей между объектами системы. Процесс миграции неподтверждённых связей между объектами и неподтверждённых версий этих объектов описан в следующем разделе.

Порядок действий администратора:

1. Клонировать представление объекта Relation.

Процесс клонирования представлений объектов описан в руководстве пользователя в разделе **«5.9.6.3. Создание нового представления объекта клонированием»**.

2. В представлении объекта, полученном при клонировании, произвести следующие настройки:
  - В настройках корневого атрибута в блоке «Структура» в поле «Тип дата сервиса для представления» выбрать значение «Базовый репозиторий».
  - В настройках корневого атрибута в блоке «Разрешения» добавить действия: «Импорт объекта», «Импорт-патч объекта». Для каждого действия указать роль «Роль администратора Каталога сервисов».
  - Обновить политики безопасности.

Процесс редактирования атрибутов представления объекта описан в руководстве пользователя в разделе **«5.9.6.5. Редактирование представления объекта – изменение параметров созданных полей»**.

3. Подготовить файл формата .xlsx.
  - Первая строка в файле содержит названия атрибутов и их json path в формате:  
<название атрибута>#<json path атрибута>
  - Атрибутный состав и тип данных для каждого атрибута можно посмотреть в представлении объекта, которое было получено путём клонирования.
  - Если атрибут содержит в себе в качестве значения объект, то каждое поле такого объекта должно быть вынесено в отдельный столбец. В заголовке каждого столбца указывается значение в формате: <название атрибута>#<json path атрибута родителя>.<json path атрибута>
  - Если атрибут содержит в себе справочное значение, то требуется указать атрибуты id, type, name:
    - В id указывается идентификатор записи справочника;
    - В name указывается значение записи;
    - В type указывается код справочника.



- Справочные значения указываются для атрибутов:
  - origin (источник связи) – справочник proxyBaseOffer;
  - destination (точка назначения связи):
    - Если точка назначения связи ИС или подсистема ИС, то справочник is;
    - Если точка назначения связи сервис или ручной сервис, то справочник proxyBaseOffer;
    - Если точка назначения связи Продукт, то справочник product;
    - Если точка назначения связи Функциональная область, то справочник function.
  - state (статус связи) – справочник relationState;
  - type (тип связи) – справочник relationType.
  - subType (подтип связи) – справочник relationSubType.
- Идентификатор связи, указанные в файле, должны быть уникальны. При импорте осуществляется проверка уникальности идентификаторов, если будут найдены дубли, то импорт остановится с ошибкой.
- Значение даты и времени в файле нужно указывать в формате timestamp.

**Таблица 5-18** – Обязательные поля при импорте связей

Поле	Json path	Описание
Идентификатор	id	Уникальный идентификатор связи.
Статус	status	Статус связи.
Источник связи	origin	Источник связи.
Точка назначения связи	destination	Точка назначения связи.
Тип связи	type	Тип связи.
Порядок сервисов в цепочке	order	Указывается только для связи цепочки с сервисом / ручным сервисом. Указывается целое число, соответствующее порядковому номеру сервиса в цепочке.
Подтип связи	subType	Подтип связи. Указывается только для связи цепочки с сервисом / ручным сервисом.
Представление объекта	techInfo.dataConfigId	Relation

4. Выполнить запрос POST `api/v1/object/<dataConfigId>/files/dataConfigBasedImport` и передать в теле запроса сформированный файл.

- a. Вместо <dataConfigId> указать идентификатор представления объекта, который был получен при клонировании.
- b. Проверить, что данные загружены. Найти связи можно в карточке объекта, с которым была установлена связь.

**Таблица 5-19** – Пример файла импорта для связи

Файл	Описание
 RelationStandart.xlsx	Пример файла импорта связи

### 5.3.5 Миграция заявок связей.

В разделе описаны шаги для миграции неподтверждённых связей и их изменений (в системных терминах – заявка связи). Миграция заявок доступна для следующих категорий связи:

1. Связь сервиса / ручного сервиса с ИС или Подсистемой ИС (все типы связи);
2. Связь сервиса / ручного сервиса / цепочки с Функциональной областью;
3. Связь сервиса / ручного сервиса / цепочки с Продуктом.

Порядок действий администратора:

1. Клонировать представление объекта:
  - a. Для заявки связи сервиса / ручного сервиса с основной ИС представление объекта – MainIsRelationObjectViewChangeRequest.
  - b. Для заявки связи сервиса / ручного сервиса с основной Подсистема ИС представление объекта – SubIsRelationObjectViewCR.
  - c. Для заявки связи сервиса / ручного сервиса с дополнительной ИС представление объекта – AdditionalIsRelationObjectViewCR.
  - d. Для заявки связи сервиса / ручного сервиса с дополнительной Подсистемой ИС представление объекта – AdditionalSubIsRelationObjectViewCR.
  - e. Для заявки связи сервиса / ручного сервиса / цепочки с Функциональной областью представление объекта – SourceOfferFunctionRelationChangeRequest.
  - f. Для заявки связи сервиса / ручного сервиса / цепочки с Продуктом представление объекта – SourceOfferProductRelationChangeRequest.

Процесс клонирования представлений объектов описан в руководстве пользователя в разделе «5.9.6.3. Создание нового представления объекта клонированием».

2. В представлении объекта, полученном при клонировании, произвести следующие настройки:
  - В настройках корневого атрибута в блоке «Структура» в поле «Тип дата сервиса для представления» выбрать значение «Базовый репозиторий».
  - В настройках корневого атрибута в блоке «Разрешения» добавить действия: «Импорт объекта», «Импорт-патч объекта». Для каждого действия указать роль «Роль

администратора Каталога сервисов».

- Обновить политики безопасности.

Процесс редактирования атрибутов представления объекта описан в руководстве пользователя в разделе **«5.9.6.5. Редактирование представления объекта – изменение параметров созданных полей»**.

### 3. Подготовить файл формата .xlsx.

- Для каждого клонированного представления объекта свой отдельный файл.
- Файл с заявками связи содержит атрибуты самой связи (см. раздел 5.3.3) и атрибуты заявки.
- Состав атрибутов и тип данных для каждого атрибута можно посмотреть в представлении объекта, которое было получено путём клонирования.
- Первая строка в файле содержит названия атрибутов и их json path в формате:  
<название атрибута>#<json path атрибута>
- Если атрибут содержит в себе в качестве значения объект, то каждое поле такого объекта должно быть вынесено в отдельный столбец. В заголовке каждого столбца указывается значение в формате: <название атрибута>#<json path атрибута родителя>.<json path атрибута>
- Если атрибут содержит в себе справочное значение, то требуется указать атрибуты id, type, name:
  - В id указывается идентификатор записи справочника;
  - В name указывается значение записи;
  - В type указывается код справочника.
- Справочные значения указываются для атрибутов:
  - status (статус заявки) – справочник changeRequestStatus;
  - target.origin (источник связи) – справочник proxyBaseOffer;
  - target.destination (точка назначения связи):
    - Если точка назначения связи ИС или подсистема ИС, то справочник is;
    - Если точка назначения связи сервис или ручной сервис, то справочник proxyBaseOffer;
    - Если точка назначения связи Продукт, то справочник product;
    - Если точка назначения связи Функциональная область, то справочник function.
  - target.state (статус связи) – справочник relationState;
  - target.type (тип связи) – справочник relationType.
  - target.subType (подтип связи) – справочник relationSubType.
- Идентификаторы заявок, указанные в файле, должны быть уникальны. При импорте осуществляется проверка уникальности идентификаторов, если будут найдены дубли, то

импорт остановится с ошибкой.

- Значение даты и времени в файле нужно указывать в формате timestamp.

**Таблица 5-20** – Обязательные поля при импорте заявок связей

Поле	Json path	Описание
Идентификатора заявки	id	Уникальный идентификатор заявки.
Статус заявки	status	Статус заявки. Указать DESIGN_COMPLETED если заявка активная. Указать ARCHIVED если заявка архивная. Активная заявка у связи может быть только одна.
Идентификатор связи	targetId	Идентификатор связи. Соответствует значению поля «target.id».
Представление объекта для заявки	techInfo.dataConfigId	Для заявки связи сервиса / ручного сервиса с основной ИС представление объекта – MainIsRelationObjectViewChangeRequest. Для заявки связи сервиса / ручного сервиса с основной Подсистема ИС представление объекта – SubIsRelationObjectViewCR. Для заявки связи сервиса / ручного сервиса с дополнительной ИС представление объекта – AdditionalIsRelationObjectViewCR. Для заявки связи сервиса / ручного сервиса с дополнительной Подсистемой ИС представление объекта – AdditionalSubIsRelationObjectViewCR. Для заявки связи сервиса / ручного сервиса / цепочки с Функциональной

Поле	Json path	Описание
		областью представление объекта – SourceOfferFunctionRelationChangeRequest. Для заявки связи сервиса / ручного сервиса / цепочки с Продуктом представление объекта – SourceOfferProductRelationChangeRequest.
Идентификатор связи	target.id	Уникальный идентификатор связи.
Статус связи	target.status	Статус связи.
Источник связи	target.origin	Источник связи.
Точка назначение связи	target.destination	Точка назначения связи.
Тип связи	target.type	Тип связи.
Порядок сервисов в цепочке	target.order	Указывается только для связи цепочки с сервисом / ручным сервисом. Указывается целое число, соответствующее порядковому номеру сервиса в цепочке.
Подтип связи	target.subType	Подтип связи. Указывается только для связи цепочки с сервисом / ручным сервисом.
Представление объекта	target.techInfo.dataConfigId	Relation

4. Выполнить запрос POST api/v1/object/<dataConfigId>/files/dataConfigBasedImport и передать в теле запроса сформированный файл.
  - a. Вместо <dataConfigId> указать идентификатор представления объекта, который был получен при клонировании.
  - b. Проверить, что данные загружены. Найти заявки связи можно в карточке объекта, с которым была установлена связь.

**Таблица 5-21** – Пример файла импорта для заявки связи

Файл	Описание
 RelationChangeRequ estStandart.xlsx	Пример файла импорта заявки связи

## 6 Администрирование продукта.

### 6.1 Управление пользователями и доступами.

В обычном случае ролевая модель настраивается так, как показано ниже, но заказчик может изменить настройки по своему усмотрению.

**Таблица 6-1** – Ролевая модель Каталога сервисов

Роль	Назначение
Читатель	Доступные функции: <ul style="list-style-type: none"><li>• просмотр Каталога сервисов</li><li>• выгрузка списка Сервисов в файл</li><li>• просмотр карточки Сервиса</li><li>• согласование Сервиса, если пользователь назначен согласующим</li></ul>
Редактор	Доступные функции: <ul style="list-style-type: none"><li>• просмотр Каталога сервисов</li><li>• выгрузка списка Сервисов в файл</li><li>• просмотр карточки Сервиса</li><li>• согласование Сервиса, если пользователь назначен согласующим</li><li>• создание Сервиса</li><li>• редактирование карточки Сервиса</li><li>• редактирование атрибутов Сервиса</li><li>• отправка Сервиса на согласование</li></ul>
Администратор	Доступные функции: <ul style="list-style-type: none"><li>• просмотр Каталога сервисов</li><li>• выгрузка списка Сервисов в файл</li><li>• просмотр карточки Сервиса</li><li>• согласование Сервиса, если пользователь назначен согласующим</li><li>• создание Сервиса</li><li>• редактирование карточки Сервиса</li><li>• редактирование атрибутов Сервиса</li><li>• отправка Сервиса на согласование</li><li>• импорт данных Сервисов в Систему</li><li>• загрузка/обновление справочников</li></ul>

Список ролей и доступов по умолчанию можно получить с помощью загрузки файла, полученного выгрузкой из системы, в «Сфера. Доступы и Роли».

Подробное описание процесса - **5.1.1 Настройка политик безопасности.**

### 6.2 Настройка справочников.

Настройка справочников в базовой комплектации осуществляется при помощи загрузки готовых файлов, входящих в комплект поставки. Эти действия выполняются в ходе первоначальной установки продукта (см. **4.2.3.2 Обновление конфигурации системы для «Сфера. Каталог сервисов».** и **6.4 Управление конфигурациями**).

Миграцию записей справочников можно провести с помощью операции, описанной в **5.3.1 Миграция справочников.**

Более подробно с ручным наполнением справочников можно ознакомиться в Руководстве пользователя «Сфера. Каталог сервисов», раздел **5.9.2 Конфигурация справочников**.

## 6.3 Настройка рабочих областей.

Не применимо.

## 6.4 Управление конфигурациями для «Сфера. Каталог сервисов».

Подраздел предназначен для управления работой с мультисессионностью, включающей в себя создание и управление сессиями конфигурирования, импорт и экспорт настроек, управление слепками и работу с интеграциями в различных средах. Конфигурация системы – совокупности настроек текущего экземпляра приложения, которые можно изменять в разделе «Администрирование». Функции импорта и экспорта позволяют быстро переносить настройки между экземплярами системы, а также управлять изменениями конфигурации, сохраняя целевой набор настроек в виде выгружаемого файла. Мультисессионность позволяет создавать изолированные административные среды для безопасного тестирования и настройки системных конфигураций без влияния на работу производственной среды пользователей (тенанта).

Каждая сессия представляет собой QuickSnapshot — быстрое временное копирование (слой) конфигурационных данных системы. Вся информация, связанная с рабочим процессом пользователей, хранится в базе данных на уровне тенанта, а не внутри отдельных сессий.

Управление слепками конфигурации системы также позволяет сравнивать загруженные в разное время наборы настроек, точно накатывать и восстанавливать необходимые изменения.

Объекты, попадающие в список конфигурации системы, могут быть описаны через основные подразделы администрирования системы: справочники и их записи, процессы, правила, интеграции, представления объектов, шаблоны визуальной композиции, жизненные циклы.

Настройки приложения, содержащиеся в файле `application.yaml` и заполняемые через переменные системы при установке, не относятся к списку настроек, которые могут быть перенесены с помощью подраздела «Управление конфигурациями» (см. Руководство администратора «Сфера. Каталог сервисов»).

Подраздел также не предназначен для работы с базой данной: сохранение и копирование объектов базы данных системы предполагается проводить с помощью штатных инструментов СУБД.



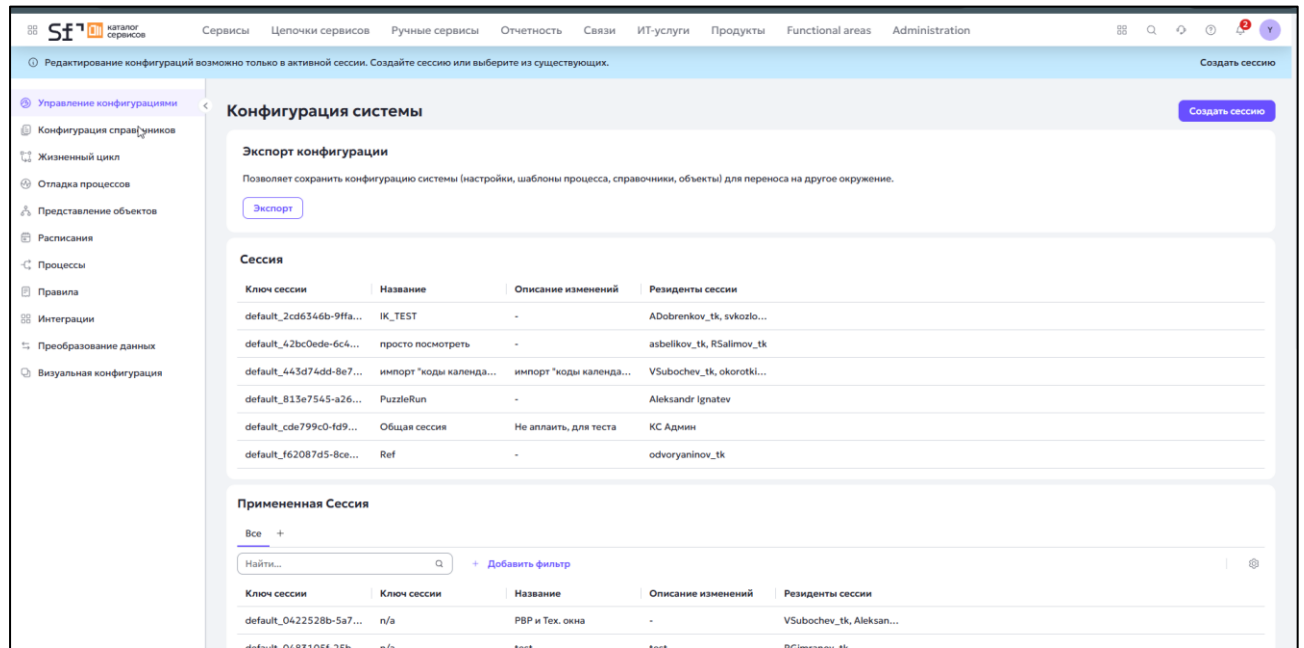


Рис. 6-1 – Общий вид подраздела «Управление конфигурациями»

#### 6.4.1 Создание сессии.

Для создания новой сессии конфигурирования пользователь с правами администратора должен выполнить следующие шаги:

1. Перейти в раздел «Администрирование», подраздел «Управление конфигурациями».
10. На вкладке «Сессии конфигураций» нажать кнопку «Создать сессию». Также создать сессию можно с помощью дисклеймера из любой вкладки.

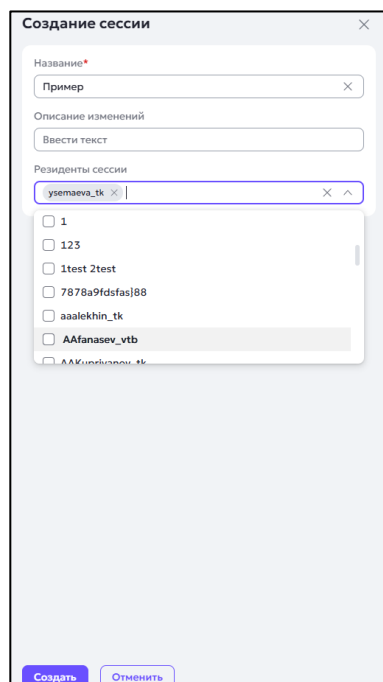


Рис. 6-2 – Окно создания сессии

11. В открывшейся форме ввести «Название сессии». Если необходимо добавить резидентов сессии, то в поле «Резиденты сессии» необходимо выбрать логины требуемых пользователей.

### ❗ Особенности назначения резидентов сессии.

Пользователь, назначенный резидентом, автоматически подключается к созданной сессии.

Если пользователь уже находился в другой сессии на момент назначения, он будет автоматически переведён в новую сессию.

Если создатель сессии указывает себя в списке резидентов, он автоматически входит в созданную сессию после её создания

12. Нажать кнопку «Создать»

13. Создаётся новая сессия идентичная рабочему тенанту (до момента внесения изменений в конфигурацию)

### ❗ Лимит сессий.

Количество одновременно существующих сессий ограничено системными ресурсами, при превышении лимита должна выводиться соответствующая ошибка: «Может быть создано не больше <кол-во сессий> сессий».

Ограничение лимита настраивается администратором. Создаётся переменная, в которой указывается лимит сессий. Лимит сессий может быть определён с помощью нагрузочного тестирования.

Настройка лимита сессий происходит в справочнике `environmentVariable` в переменной `sessionsLimit`.

## 6.4.2 Применение сессии.

Применение сессии — это процесс переноса настроек из изолированной сессии конфигурирования в основную функциональную сессию (тенант), делая изменения видимыми для всех пользователей системы.

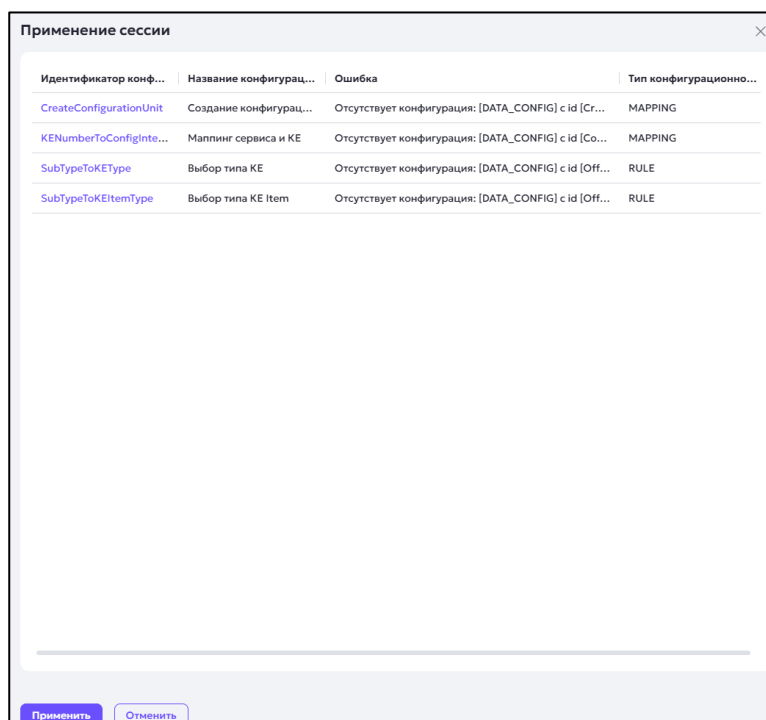
При применении конфигурационной сессии система выполняет следующую последовательность операций:

1. Объединение настроек: система объединяет настройки функциональной сессии с настройками конфигурационной сессии, разрешая конфликты в пользу конфигурационной сессии.
2. Создание нового слепка: создаётся новый слепок системы, объединяющий старые настройки с новыми. Этот снимок становится новой функциональной сессией.
3. Закрытие конфигурационной сессии: после успешного применения конфигурационная сессия закрывается и удаляется из списка доступных для конфигурирования сессий.
4. Регистрация изменений: Информация о применённой сессии сохраняется в истории изменений системы.

Необходимые шаги для применения сессии:

1. Зайти в подраздел «Управление конфигурациями».
2. Выполнить вход в одну из сессий. (Если нет сессий, то выполнить создание сессии)

3. Внести нужные изменения в конфигурацию системы внутри сессии.
4. Перейти к детальному представлению сессии.
5. Переход к детальному представлению сессии возможен:
  - Через дисклеймер кликом по кнопке «Перейти к детализации»
  - Через двойной клик по выбранной сессии в подразделе «Управление конфигурациями». В данном сценарии предварительно требуется войти в выбранную сессию.
6. Нажать кнопку «Применить».
7. Просмотреть отчёт об ошибках.
8. Наличие ошибок не блокирует применение сессии и носит информативный характер.



Идентификатор конф...	Название конфигурац...	Ошибка	Тип конфигурационно...
CreateConfigurationUnit	Создание конфигурац...	Отсутствует конфигурация: [DATA_CONFIG] с id [Cr...	MAPPING
KENumberToConfigInte...	Мэппинг сервиса и KE	Отсутствует конфигурация: [DATA_CONFIG] с id [Co...	MAPPING
SubTypeToKEType	Выбор типа KE	Отсутствует конфигурация: [DATA_CONFIG] с id [Off...	RULE
SubTypeToKEItemType	Выбор типа KE Item	Отсутствует конфигурация: [DATA_CONFIG] с id [Off...	RULE

Применить Отменить

**Рис. 6-3**– Отчёт об ошибках

9. Нажать кнопку «Применить» в шторке применения сессии. Дождаться завершения процесса применения.
10. Изменения из конфигурационной сессии были применены к тенанту, сессия исчезла из списка сессий, а система автоматически создала новый слепок конфигурации как точку восстановления и истории изменений.

### 6.4.3 Отчёт об ошибках.

При открытии отчёта через нажатие кнопки "Проверить" в детальном представлении сессии или при нажатии на кнопку "Применить" выполняются проверки на валидность сессии. В момент валидации объектов сессии проверяются преобразования данных, на вход которых были получены конфигурации с типами: INTEGRATION, MAPPING, PROCESS, RULE, а на выход выведен результат SnapshotValidationResult.

Запрос для получения списка объектов результата валидации сессии:

method: POST

request URL: `https://sfera.ppcps-stable.dev.sfera-t1.ru/app/catalog/api/api/v1/session/validate?sessionId=default_cab63395-58dd-4ea4-8270-b45f4bd84c6c`

query parameters: sessionId (Id сессии, для которой необходимо провести валидацию)

В результате валидации приходит набор предупреждений и ошибок. Если валидация прошла успешно, то вернётся пустой объект.

**Таблица 6-2** – Поля объекта *SnapshotValidationResult*

№	Идентификатор	Тип	Описание
1.	id	string	Уникальный идентификатор результата валидации.
2.	state	reference	Запись из справочника <code>snapshotValidationState</code>
3.	validatorName	string	Название проверки, сгенерировавшего результат.
4.	configurationId	string	Идентификатор проверенной конфигурации.
5.	configurationName	string	Имя проверенной конфигурации.
6.	message	string	Текст сообщения с описанием найденной проблемы или рекомендации
7.	configurationType	string	Тип проверенной конфигурации (INTEGRATION, MAPPING, PROCESS, RULE)

#### 6.4.4 Экспорт конфигурации.

Операция используется для сохранения конфигурации системы в виде файла. Файл может быть перенесён на другое окружение для использования при установке приложения.

##### 6.4.4.1 Экспорт конфигураций из сессии.

Необходимые шаги для экспорта сессии:

1. Зайти в подраздел «Управление конфигурациями».
2. Войти в детальное представление сессии. Для этого необходимо войти в сессию с помощью трёх точек в блоке «Сессия» и сделать двойной клик по строке с необходимой сессией.
3. В блоке «Экспорт конфигурации» нажать кнопку «Экспорт».
4. В открывшейся форме выбрать все типы объектов или конкретные объекты, которые требуется экспортировать (см. **Таблица 6-3**).
5. Список представлен в виде дерева. Для просмотра дочерних записей необходимо нажать на иконку в виде стрелки рядом с родительской записью.
6. Кнопка «Выбрать всё» позволяет выделить весь список. Кнопка «Очистить всё» – снять выделение со всего списка.

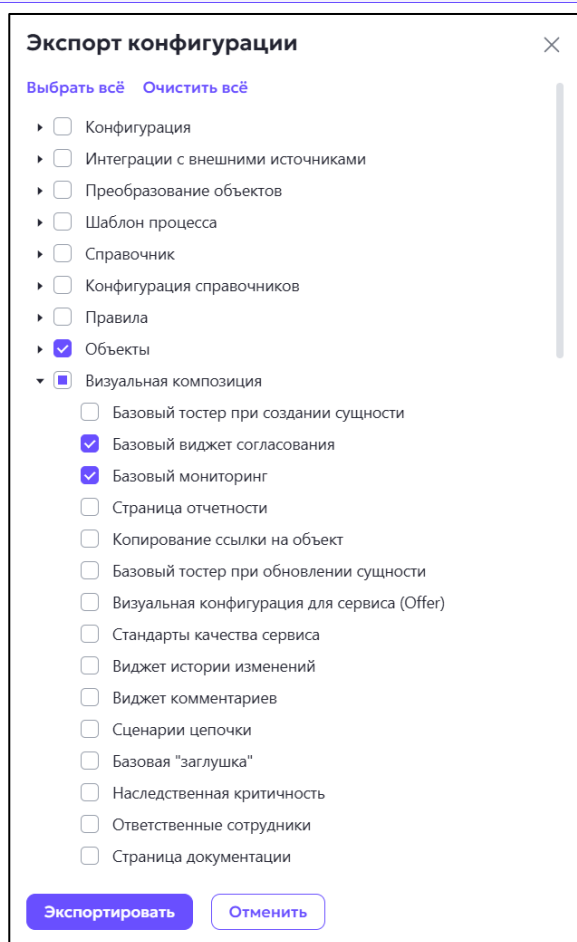
7. В списке приводятся только названия объектов – не коды или идентификаторы.

### **❗ Консистентность данных при экспорте и импорте конфигурации**

При выборе данных экспорта и импорта необходимо учитывать необходимость соблюдения консистентности.

Например, если выгружаемая интеграция будет содержать преобразование данных, которое не было включено в выгрузку, то интеграция не будет работать – так как начнёт ссылаться на отсутствующий в комплекте настроек объект.

Также необходимо проверять совпадение идентификаторов объектов. Если на окружении, где осуществляется импорт настроек, связанный объект имеет идентификатор, отличающийся от сохранённого в импортируемом объекте, потребуется перезаписать его с новым идентификатором связанного объекта.



**Рис. 6-4 – Форма «Экспорт конфигурации»**

8. Нажать «Экспортировать».
9. Запустится процесс подготовки и скачивания файла. Файл конфигурации будет упакован в архив с разрешением zip.
10. Маска названия файла – systemconfiguration\_<дата><время>.
11. Убедиться в корректности структуры выгруженного файла.
12. Папки соответствуют типам объектов или объектам. Отдельные записи сохранены в формате

json.

Размер	Сжат	Тип	Изменен	CRC32
5 697	2 721	Папка с файлами		
593	297	Папка с файлами		
21 590	6 389	Папка с файлами		
113 696	22 394	Папка с файлами		
5 679	2 807	Папка с файлами		
15 756	5 284	Папка с файлами		
3 402	2 047	Папка с файлами		
14 398	5 567	Папка с файлами		
21 295	6 448	Папка с файлами		
83 905	19 151	Папка с файлами		
670	329	Папка с файлами		
5 855	1 859	Папка с файлами		
19 138	6 223	Папка с файлами		
72 763	17 773	Папка с файлами		
10 756	3 433	Папка с файлами		
5 382	2 710	Папка с файлами		
5 589	2 765	Папка с файлами		
8 451	4 182	Папка с файлами		
1 861	1 160	Папка с файлами		
6 201	3 698	Папка с файлами		
280	210	Файл \".JSON\"	13.06.2025 12:57	00228FF3
245	168	Файл \".JSON\"	13.06.2025 12:57	BAD1220C
177	111	Файл \".JSON\"	13.06.2025 12:57	7F70D9C7
265	191	Файл \".JSON\"	13.06.2025 12:57	724F6A6E
531	296	Файл \".JSON\"	13.06.2025 12:57	96F7696A
401	227	Файл \".JSON\"	13.06.2025 12:57	8EFC4D2
500	262	Файл \".JSON\"	13.06.2025 12:57	7C428674
223	162	Файл \".JSON\"	13.06.2025 12:57	4E8E8542
277	201	Файл \".JSON\"	13.06.2025 12:57	EDA203C1
399	239	Файл \".JSON\"	13.06.2025 12:57	ED9EEA16
544	308	Файл \".JSON\"	13.06.2025 12:57	F85E063D
425	244	Файл \".JSON\"	13.06.2025 12:57	AC88961A
432	253	Файл \".JSON\"	13.06.2025 12:57	478211DC
271	189	Файл \".JSON\"	13.06.2025 12:57	060FC2F
256	188	Файл \".JSON\"	13.06.2025 12:57	87128E63
530	296	Файл \".JSON\"	13.06.2025 12:57	58D906DC
434	246	Файл \".JSON\"	13.06.2025 12:57	0C6DD826

Рис. 6-5 – Пример структуры файла конфигурации

#### 6.4.4.2 Экспорт конфигураций из тенанта.

**Примечание:** Пользователь не должен находиться в сессии для выгрузки конфигурации из тенанта.

1. Зайти в подраздел «Управление конфигурациями».
2. В блоке «Экспорт конфигурации» нажать кнопку «Экспорт». В данном кейсе будет выгружены настройки из основной рабочей системы (тенанта).
3. В открывшейся форме выбрать все типы объектов или конкретные объекты, которые требуется экспортировать (см. **Таблица 6-3**) и нажать кнопку «Экспортировать».

Таблица 6-3 – Типы объектов в структуре конфигурации

№	Раздел	Папка в выгрузке	Описание
1.	Конфигурация	DataConfig	Представление объектов. Все сущности, которые создаются в подразделе «Представление объектов» – включая настройки полей.
2.	Интеграции с внешними источниками	IntegrationConfiguration	Настройки интеграций. Все сущности, которые создаются в подразделе «Интеграции».
3.	Преобразование данных	MappingConfiguration	Преобразование данных. Все сущности, которые создаются в подразделе «Преобразование данных».
4.	Справочник	Reference	Записи справочников. Содержимое справочников, создаваемых в подразделе «Конфигурация справочников».

№	Раздел	Папка в выгрузке	Описание
5.	Конфигурация справочников	ReferenceConfiguration	Справочники. Сами справочники и их настройки, которые создаются в подразделе «Конфигурация справочников».
6.	Правила	Rule	Правила. Все сущности, которые создаются в подразделе «Правила».
7.	Объекты	TargetObjectType	Жизненный цикл. Все сущности, которые создаются в подразделе «Жизненный цикл» – включая сам жизненный цикл, настройки статусов и переходов между статусами.
8.	Визуальная композиция	UIComposition	Визуальная композиция. Все сущности, которые создаются в подразделе «Визуальная композиция».

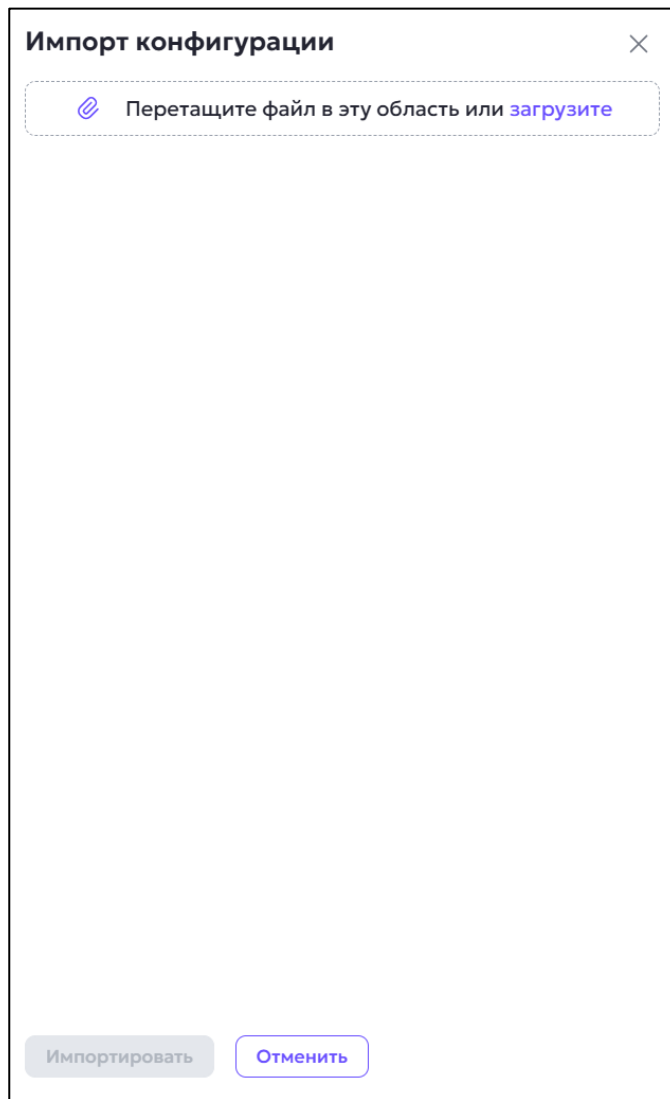
#### 6.4.5 Импорт конфигурации в сессию.

Импорт доступен только в рамках сессии.

Операция используется при установке предварительно сохранённой конфигурации системы. Если на окружении уже была применена какая-либо конфигурация системы, при импорте она будет заменена данными из импортируемого файла.

Необходимые шаги:

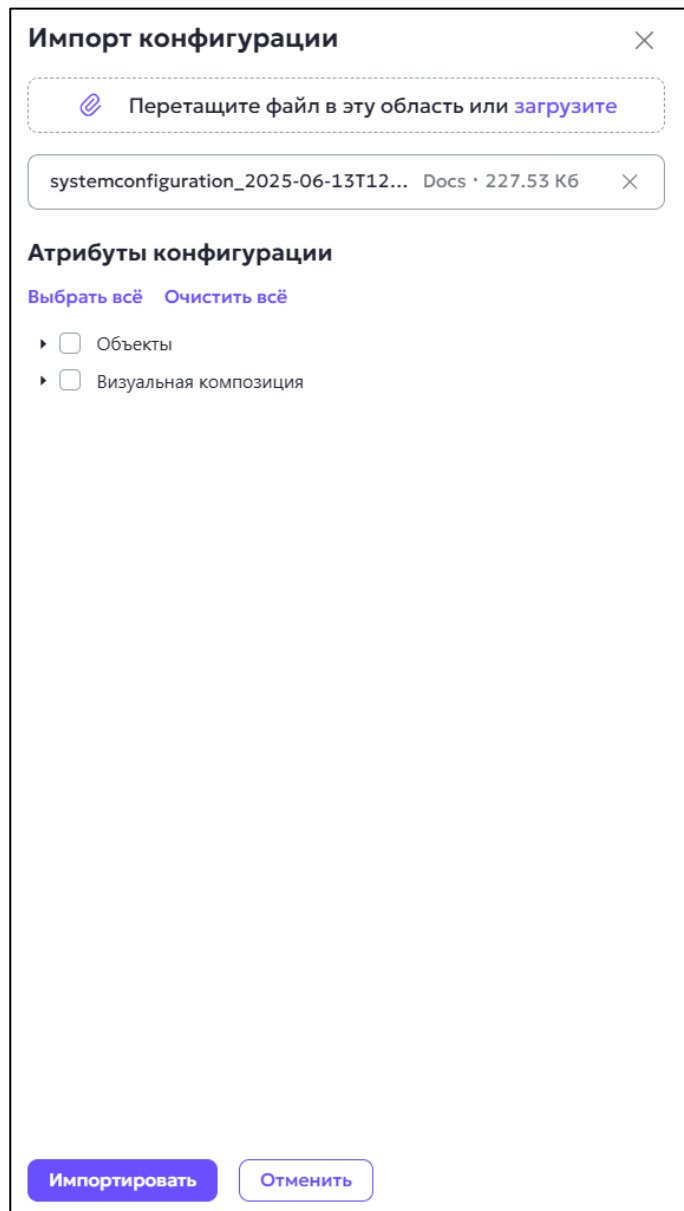
1. Зайти в подраздел «Управление конфигурациями».
2. Войти в детальное представление сессии. Для этого необходимо войти в сессию с помощью трёх точек в блоке «Сессия» и сделать двойной клик по строке с необходимой сессией.
3. В блоке «Импорт конфигурации» нажать кнопку «Импорт». Откроется форма «Импорт конфигурации». Файлы будут импортированы в сессию, в которой находится пользователь.



**Рис. 6-6** – Форма «Импорт конфигурации», файл не выбран

4. Перетащить подготовленный файл в область, обозначенную пунктиром, или нажать на кнопку «Загрузить» в этой области.
5. Во втором случае откроется системное окно операционной системы для выбора файла вложения. При выборе файла по умолчанию будет выставлено ограничение на его тип – архив. Выбрав файл, необходимо нажать кнопку «Открыть» в модальном окне.
6. Выбранный файл будет инициализирован системой. По окончании процесса в форме импорта отобразится структура файла, аналогичная дереву объектов при экспорте конфигурации.
7. Если был выбран файл, не относящийся к конфигурации системы, структура не отобразится, опция импорта не будет доступна.
8. В этом случае можно загрузить новый файл, повторив действия, описанные в п. 3. Предварительно можно очистить ранее добавленный файл нажатием на иконку в виде крестика в правой части строки файла.





**Импорт конфигурации**

Перетащите файл в эту область или [загрузите](#)

systemconfiguration\_2025-06-13T12... Docs · 227.53 K6

**Атрибуты конфигурации**

[Выбрать всё](#) [Очистить всё](#)

☐ Объекты

☐ Визуальная композиция

**Импортировать** **Отменить**

**Рис. 6-7** – Форма «Импорт конфигурации», файл выбран

9. Выбрать в дереве атрибутов конфигурации те типы объектов или конкретные объекты, которые требуется импортировать.
10. Нажать «Импортировать».
11. Если процесс обновления завершится успешно, выйдет системное уведомление об обновлении конфигурации.
12. Если ни один объект не был выбран для импорта, процесс обновления все равно запустится, однако в результате никаких изменений не произойдёт.
13. Изменения коснутся только объектов, которые находились в файле импорта. Сопоставление объектов при обновлении происходит по кодам и идентификаторам.
14. Убедиться в применении импортированных изменений.

#### **❶ Обновление пустым значением.**

При импорте конфигурации реализовано обновление пустым значением.

Например, если в импортируемом объекте поле не было заполнено, а на окружении, куда устанавливается настройка, это поле аналогичного объекта заполнено, то при импорте поле очистится (заполнится пустым значением).

#### **❷ Обратная совместимость файлов конфигураций.**

Из-за расширения модели объектов системы от версии к версии формат выгружаемой конфигурации может изменяться. Несмотря на заложенные в развитие системы мероприятия по миграции данных для обеспечения обратной совместимости, корректное применение старых конфигурационных данных на версии, превышающую версию создания конфигурации более чем на 3, не может быть гарантировано без последовательной установки промежуточных версий системы согласно инструкции.

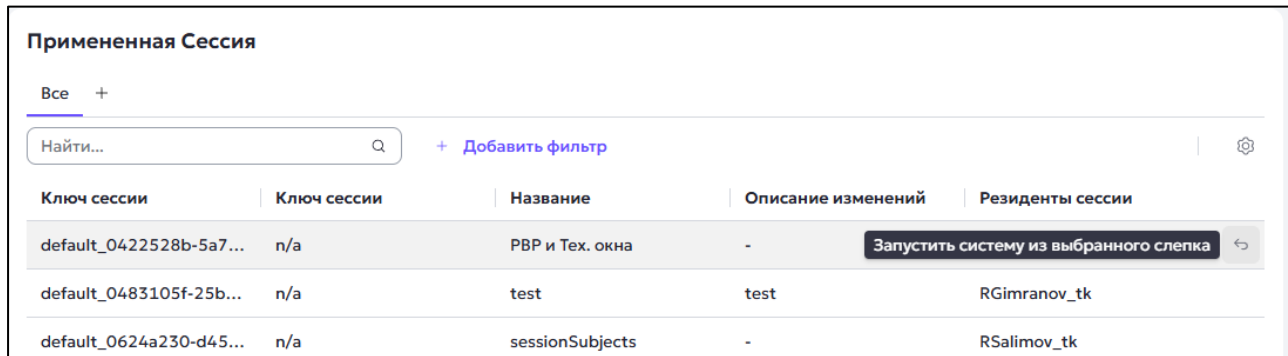
**Снэпшоты** (слепки) системы позволяют сохранять промежуточные настройки конфигурации и сравнивать их между собой. Слепки системы не могут быть перенесены напрямую на другое окружение, как файл конфигурации. Их назначение – управление конфигурацией для конкретного окружения. Снэпшоты конфигурации создаются автоматически при каждом успешном применении сессии к тенанту.

Если в конфигурации были сделаны изменения, то детализация снэпшота позволяет увидеть все изменённые объекты за заданный промежуток времени и откатить все или некоторые изменения до желаемого состояния, которое было зафиксировано в одном из слепков.

### **6.4.6 Полное восстановление конфигурации системы из снэпшота.**

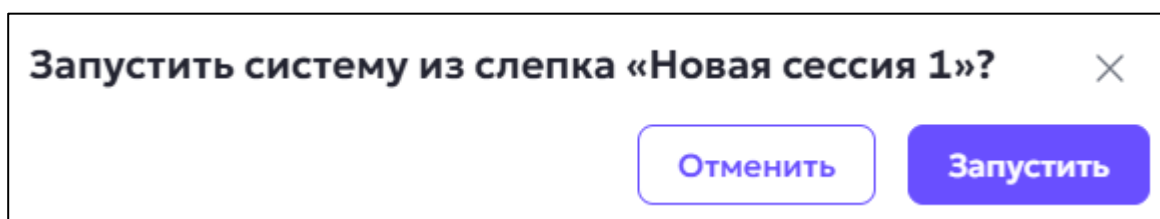
Необходимые шаги:

1. Зайти в подраздел «Управление конфигурациями».
2. Найти в блоке «Применённая сессия» нужную запись.
3. Навестись на строку и нажать «Запустить систему из выбранного слепка».



**Рис. 6-8** – Восстановление системы из применённой сессии (слепка)

- Подтвердить решение в вышедшем модальном окне. Появится уведомление о запуске процесса восстановления. Если процесс завершится успешно, об этом также выйдет подтверждающее уведомление.



**Рис. 6-9** – Модальное окно подтверждения восстановления данных из слепка конфигурации системы

- Убедиться в восстановлении настроек из снимка

**⚠ Восстановление конфигурации системы из снимка.**

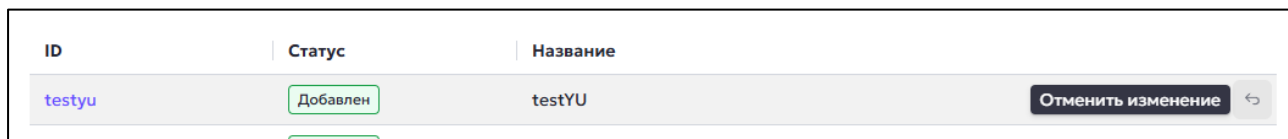
Данные из слепка конфигурации системы полностью заменяют текущую конфигурацию.

#### 6.4.7 Отмена изменений в рамках сессии.

Операция применяется, если необходимо восстановить настройки конкретного объекта в сессии. Отмена изменений действует только внутри конфигурационной сессии и не затрагивает уже применённые к тенанту изменения.

Необходимые шаги:

- Зайти в подраздел «Управление конфигурациями».
- Войти в сессию и перейти в детальное представление сессии.
- Найти объект, изменения которого необходимо отменить.
- При наведении на строку в правой части строки записи выбрать опцию «Отменить изменения».



**Рис. 6-10** – Кнопка отмены изменений

- Подтвердить восстановление объекта в модальном окне. Убедиться в отмене изменений.



## 7 Рекомендации по обеспечению надёжности и восстановлению работы продукта.

### 7.1 Диагностика продукта.

Рекомендуемые процедуры проверки работоспособности системы и локализации неработающих компонентов представлены в таблице ниже.

**Таблица 7-1** — Перечень планово-предупредительных мероприятий

№	Содержание мероприятия	Периодичность	Ответственный
1.	Мониторинг работоспособности различных компонентов системы	Определяется Администратором (рекомендованная частота: один раз в сутки, 5 дней в неделю)	Администратор
2.	Локализация неработающих компонентов системы	Определяется Администратором	Администратор
3.	Проверка наличия сертификатов	Перед запуском пилотной или целевой версии системы в промышленную среду	Администратор
4.	Проверка в консоли администратора Kubernetes (или аналогичной системы, используемой в данном решении) состояния подов (все поды запущены в требуемом количестве)	После запуска пилотной или целевой версии системы в промышленную среду	Администратор
5.	Проверка лог-файлов запуска компонентов продукта (микросервисов) на наличие сообщений с типом «ERROR»	После запуска пилотной или целевой версии системы в промышленную среду	Администратор
6.	Проверка работоспособности микросервисов на маршрутах (/healthcheck)	После запуска пилотной или целевой версии системы в промышленную среду	Администратор
7.	Проверка работоспособности микросервисов	Ежедневно	Администратор
8.	Проверка работоспособности компонента БД Grafana (или аналогичной системы, используемой в данном решении)	Ежедневно	Администратор
9.	Проверка статуса запуска серверов	Ежедневно	Администратор
10.	Проверка статуса мониторинга использования серверов	Раз в месяц	Администратор
11.	Проверка статуса мониторинга выполнения задач планировщика	Ежедневно	Администратор
12.	Мониторинг использования серверов: анализ статистики по лог-файлам	Раз в месяц	Администратор

## 7.2 Резервное копирование.

В продуктивной среде резервирование БД осуществляется стандартными средствами PostgreSQL (или аналогичной СУБД, используемой в решении).

Для обеспечения возможности аварийного восстановления данных системы регулярно производится резервное копирование БД:

- полное копирование – один раз в сутки;
- дифференциальное копирование – каждый час.

## 7.3 Восстановление из резервной копии

Инструментов восстановления БД из резервной копии в системе не предусмотрено - восстановление БД настраивается штатными средствами PostgreSQL (или аналогичной СУБД, используемой в решении).

Восстановление конфигурационных данных может выполняться следующими способами:

- через импорт SCM (System Configuration Management);
- через восстановление снимка системы.

Описание названных способов см. в 6.4. Управление конфигурациями.

## 7.4 Устранение проблем

Аварийно-восстановительные работы проводятся с целью поддержки бесперебойной работы системы и восстановления работоспособности системы в максимально сжатые сроки без потери данных.

### 7.4.1 Критерии аварийного события

1. Недоступность приложения «Сфера. Каталог сервисов» через интерфейс пользователя Платформы Сфера.
2. Обращение любого из пользователей системы в службу поддержки.
3. Выявление неисправности в результате планово-предупредительных работ со стороны службы поддержки.
4. Обращение в службу поддержки со стороны сотрудников мониторинга, администраторов СУБД и серверов приложений.
5. Не пройдены проверки работоспособности при вводе системы в эксплуатацию.

К потенциально аварийным событиям также относятся:

1. Недоступность серверов приложений.
2. Недоступность баз данных.
3. Некорректная работа системы.

При аварийном событии необходимо оценить состояние системы, данными для анализа проверки

состояния системы могут быть:

- данные по состоянию серверов из консолей администраторов,
- средства мониторинга,
- содержимое логов системы.

Эти данные анализируются и обобщаются. На основе анализа принимается решение по составу отказавших элементов системы и о мерах по их восстановлению.

Производится попытка запуска отказавших элементов в следующем порядке: сначала запускается сервер базы данных, затем все остальное окружение. После успешного запуска вновь проводится проверка состояния системы. По результатам проверки принимается решение о работоспособности системы или о переходе к её восстановлению.

#### 7.4.2 Порядок действий при выявлении аварийного события

Общий порядок действий при возникновении аварийного события:

1. Диагностика возникшей ситуации, первичный сбор информации.
2. Оповещение координатора, ответственного за выполнение аварийного восстановления, если самостоятельное устранение проблемы невозможно.
3. Определение специфики проблемы, привлечение специалистов.
4. Выполнение действий по восстановлению согласно выбранному сценарию.
5. Рассылка оповещений об аварийной ситуации:
  - всем заинтересованным лицам (например, Бизнес-заказчик, Технолог, Аналитик ИС);
  - каналы оповещения (например, электронная почта, SMS-сообщение, звонок) выбираются на усмотрение службы поддержки в зависимости от критичности ситуации;
  - в оповещении указываются данные о зарегистрированном инциденте: описание аварийного события, периметр события, возможные причины возникновения, плановые сроки устранения, влияние на смежные ИС и возможные последствия.

К процессам поддержки и восстановления работоспособности системы должны быть привлечены следующие сотрудники:

- Системные администраторы;
- Администраторы серверов приложений (в части сопровождения и эксплуатации систем поддержки бизнеса).

Работоспособность системы определяется способностью выполнения системой предписанных ей функций текущим набором средств при отсутствии критических ошибок в логах. В зависимости от ситуации, характеризующей состояние системы после сбоя, процесс восстановления может состоять из нескольких этапов.

Краткий перечень возможных действий при сбое в системе:

- Проверить статус работоспособности приложения, при необходимости перезапустить.
- Закончилось место на диске - необходимо освободить место при помощи архивирования и удаления данных с дисков.

- Истёк срок действия пароля для ТУЗ - необходимо актуализировать пароль для ТУЗ.
- Сломался сервер с виртуализацией - необходимо проанализировать ситуацию и устранить проблему.

### 7.4.3 Типовые аварийные ситуации и пути их решения

#### 7.4.3.1 Аварийные ситуации системы оркестровки контейнеризированных приложений.

1. Ошибка пуллинга образа из менеджера репозитория в систему оркестровки контейнеризированных приложений – неправильный путь к образу.

Субъект: система оркестровки контейнеризированных приложений.

Проблема: при деплое пода не происходит пуллинг образа в систему оркестровки контейнеризированных приложений, по причине ошибки ссылки на образ.

Решение: скорректировать путь к образу, свериться с репозиторием в менеджере репозитория.

2. Ошибка пуллинга образа из менеджера репозитория в систему оркестровки контейнеризированных приложений – нет прав на пуллинг.

Субъект: система оркестровки контейнеризированных приложений.

Проблема: при деплое пода не происходит пуллинг образа в систему оркестровки контейнеризированных приложений.

Решение: сделать новый пулл-секрет для репозитория, из которого не проходит пуллинг. Выполнить биндинг пулл-секрета и к текущей учётной записи для деплоя.

В случае проблем зайти под ТУЗ Teamcity в менеджере репозитория, проверить наличие прав на репозиторий докер.

3. Ошибка деплоя HelmChart в систему оркестровки контейнеризированных приложений – UPGRADEFAILED.

Субъект: система оркестровки контейнеризированных приложений.

Проблема: при деплое HelmChart деплой всегда падает, сколько бы раз ни перезапускать пайплайн. Новый деплой даже не запускается, никакие изменения в системе оркестровки контейнеризированных приложений не применяются.

Решение: найти и удалить секрет. После перезапустить деплой.

#### 7.4.3.2 Аварийные ситуации СУБД

Ошибка подключения к PostgreSQL.

Субъект: Cluster PostgreSQL.

Проблема: ошибки подключения к кластеру PostgreSQL. Пример: (лог пода cjava)

org.postgresql.util.PSQLException: Connection to {имя сервера}:5432 refused. Check that the hostname and port are correct and that the postmaster is accepting TCP/IP connections.

Решение: проверка доступности сервера PostgreSQL, проверить фаерволл на сервере PostgreSQL, сетевую доступность по заданному имени и порту.



### 7.4.3.3 Аварийные ситуации сервера непрерывной интеграции

Ошибка подключения версионных настроек.

Субъект: сервер непрерывной интеграции.

Проблема: сервер непрерывной интеграции не загружает пайплайн из VCSroot, ссылаясь на то, что нет первой версии пайплайна.

Решение: выключить синхронизацию версионных настроек в следующем порядке: Synchronization enabled → Synchronization disabled → Use settings from a parent project.

Включить синхронизацию.