

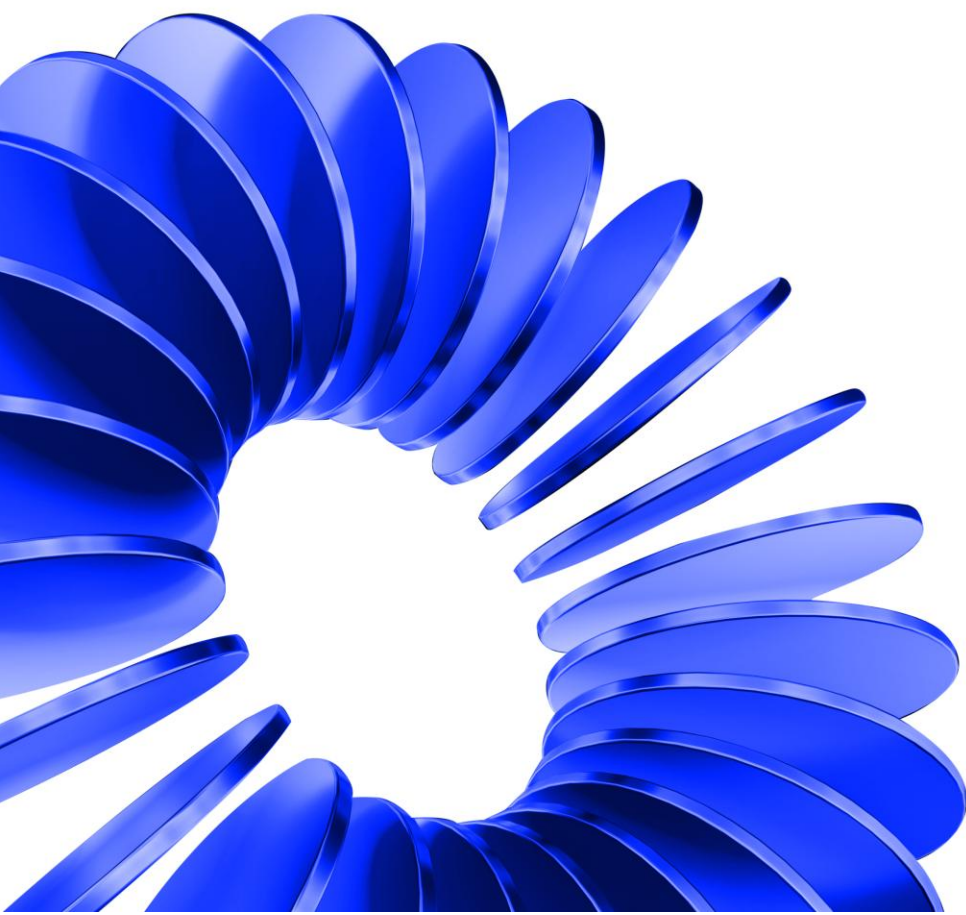
ОБЩЕСТВО С ОГРАНИЧЕННОЙ
ОТВЕТСТВЕННОСТЬЮ «РУСБИТЕХ-АСТРА»

TROK

СИСТЕМА ХРАНЕНИЯ ДАННЫХ TROK

РУКОВОДСТВО ПО НАСТРОЙКЕ

Москва, 2026г.



СОДЕРЖАНИЕ

1.	ОБЩАЯ ИНФОРМАЦИЯ	6
1.1.	Режим без локального диска	8
1.2.	Установка и настройка базовых компонентов	9
1.3.	Подготовка топологии сети и соблюдение аппаратных требований	12
1.4.	Сетевые интерфейсы	13
1.5.	Требования к сети.....	13
1.6.	Резервное копирование	14
2.	ИМЯ УЗЛА И ВОЗМОЖНЫЕ КОНФЛИКТЫ ПРИ СОЗДАНИИ.....	15
3.	ПОДГОТОВКА К РАЗМЕТКЕ ДИСКОВ.....	17
4.	РАЗМЕТКА ДИСКОВ LVM ДЛЯ ПУЛОВ ХРАНЕНИЯ	19
5.	РАЗМЕТКА ДИСКОВ LVM_THIN ДЛЯ ПУЛОВ ХРАНЕНИЯ	23
6.	ПРОВЕРКА И УСТАНОВКА ПАКЕТА TROK-WEBUI.....	26
7.	ПОДГОТОВКА SSL/TLS-СЕРТИФИКАТОВ	28
7.1.	Создание самоподписанного сертификата	28
7.2.	Установка сертификата внутреннего центра сертификации.....	30
7.3.	Установка готового fullchain-сертификата	31
7.4.	Проверка соответствия сертификата и закрытого ключа.....	32
7.5.	Установка корневого сертификата на клиентских устройствах	32
8.	ВКЛЮЧЕНИЕ HTTPS ДЛЯ TROK WEBUI	34
9.	ПРОВЕРКА И ПРИМЕНЕНИЕ КОНФИГУРАЦИИ	37
9.1.	Применение на сервере без HA.....	37
9.2.	Применение в HA-кластере	38
9.3.	Проверка HTTPS.....	38
10.	ОБЩИЕ СВЕДЕНИЯ О SDS UI (ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ДЛЯ УПРАВЛЕНИЯ СУЩНОСТЯМИ SDS).....	41
11.	ПРОЦЕДУРА АВТОРИЗАЦИИ ДЛЯ SDS UI.....	43
12.	ПАНЕЛЬ УПРАВЛЕНИЯ SDS UI (ДАШБОРД)	44
13.	УЗЛЫ КЛАСТЕРА.....	47
14.	ПУЛЫ ХРАНЕНИЯ.....	51
14.1.	Создание нового пула хранения	51
14.2.	Копирование пула (создание пула с теми же параметрами)	55
14.3.	Удаление пула хранения.....	56
14.4.	Теги и автоматическое размещение ресурсов	57

15. ГРУППЫ РЕСУРСОВ.....	59
15.1. Создание новой группы ресурсов	60
15.2. Удаление группы ресурсов.....	62
15.3. Приоритет размещения и тегирование	63
16. ШАБЛОНЫ РЕСУРСОВ	65
16.1. Создание шаблона ресурсов.....	66
16.2. Состояния шаблона ресурсов.....	67
16.3. Удаление шаблона ресурсов	69
17. ШАБЛОНЫ ТОМОВ.....	72
17.1. Замечания по операциям изменения размера томов (resize).....	74
18. СПИСОК ТОМОВ.....	78
18.1. Состояния тома (статусы).....	79
19. РЕСУРСЫ КЛАСТЕРА	81
20. СНИМКИ СОСТОЯНИЯ (СНАПШОТЫ).....	84
21. УЧЁТНЫЕ ДАННЫЕ ПРИЛОЖЕНИЯ (НАСТРОЙКИ)	86
21.1. Создание новой записи	86
21.2. Регенерация секрета.....	88
21.3. Удаление учётных данных	89
22. ВАЛИДАЦИЯ ПОЛЕЙ SDS UI.....	91
23. НАСТРОЙКА ДОСТУПА ПО ПРОТОКОЛАМ	92
23.1. Настройка отказоустойчивого хранилища NFS (на базе drbd-reactor).....	93
23.2. Настройка отказоустойчивого хранилища Samba (на базе drbd-reactor)	104
23.3. Настройка отказоустойчивого хранилища iSCSI (на базе drbd-reactor)	114
24. МОНИТОРИНГ PROMETHEUS – КОНТРОЛЛЕРОВ	122
24.1. Метрика trok_info	122
24.2. Метрика trok_node_state.....	122
24.3. Метрика trok_storage_pool_capacity_free_bytes	123
24.4. Метрика trok_storage_pool_capacity_total_bytes	124
24.5. Метрика trok_storage_pool_error_count	124
24.6. Метрика trok_resource_state.....	124
25. МОНИТОРИНГ PROMETHEUS – DRBD REACTOR	126
26. СТРУКТУРА И РАСПОЛОЖЕНИЕ ЛОГ-ФАЙЛОВ.....	127
27. НАСТРОЙКА ВЫСОКОДОСТУПНОГО КЛАСТЕРА.....	132
27.1. Перенос баз TROK на тома DRBD	134

27.2.	Создание конфигурации drbd-reactor (promoter) для ресурса ha1	137
27.3.	Проверка состояния drbd-reactor	140
27.4.	Настройка остальных узлов.....	141
28.	ДОСТУП ПО ПРОТОКОЛУ S3 К ХРАНИЛИЩУ SDS – ОБЩИЕ СВЕДЕНИЯ	145
28.1.	Архитектура интеграции S3 и SDS.....	145
29.	НАСТРОЙКА S3-ШЛЮЗА НА БАЗЕ DRBD И DRBD-REACTOR	147
29.1.	Подготовка к интеграции S3 и SDS.....	147
29.2.	Предварительная настройка операционной системы	148
29.3.	Подготовка файловой системы	150
29.4.	Настройка drbd-reactor для trok-s3-gw.....	151
29.5.	Проверка работоспособности.....	156
30.	ОБЩИЕ СВЕДЕНИЯ О S3 UI (ГРАФИЧЕСКИЙ ИНТЕРФЕЙС S3).....	158
31.	ПРОЦЕДУРА АВТОРИЗАЦИИ ДЛЯ S3 UI	160
32.	ПАНЕЛЬ УПРАВЛЕНИЯ S3 UI (ДАШБОРД)	162
33.	УПРАВЛЕНИЕ УЧЁТНЫМИ ЗАПИСЯМИ S3	164
33.1.	Создание нового пользователя.....	165
33.2.	Редактирование данных пользователя	167
33.3.	Удаление учётной записи	169
34.	БАКЕТЫ	171
34.1.	Создание бакета.....	172
34.2.	Изменение владельца бакета	174
35.	ПРОВОДНИК (ОБЗОР БАКЕТОВ)	176
35.1.	Добавление бакета в избранное	177
35.2.	Получение информации о бакете и добавление тегов.....	179
35.3.	Настройка политик бакета.....	181
35.4.	Управление версионированием бакета	188
35.5.	Включение и настройка блокировки объектов.....	190
35.6.	Просмотр и управление незавершёнными multipart-загрузками.....	192
35.7.	Удаление бакета	193
36.	ОПЕРАЦИИ С ОБЪЕКТАМИ / ФАЙЛАМИ	195
36.1.	Загрузка файлов.....	197
36.2.	Просмотр информации об объекте	199
36.3.	Скачивание объекта	201
36.4.	Генерация предварительно подписанной ссылки (Presigned URL).....	202

36.5. Удаление объекта	204
------------------------------	-----

1. ОБЩАЯ ИНФОРМАЦИЯ

Руководство описывает процесс создания кластерной системы хранения данных на примере конфигурации из трёх узлов. Кластер состоит из трёх физических серверов с установленной операционной системой Astra Linux.

Для обеспечения устойчивости к сбоям и достижения кворума при принятии решений рекомендуется использовать не менее трёх серверов. Такая конфигурация позволяет минимизировать риск разделения сети и повысить общую отказоустойчивость системы.

Все серверы должны находиться в одной локальной сети и быть объединены общей подсетью. Это необходимо для корректной работы кластера и обеспечения сетевого взаимодействия между узлами.

Для передачи данных между серверами рекомендуется использовать отдельную выделенную сеть для репликации. Это решение позволяет изолировать рабочий трафик приложений от сетевой нагрузки, создаваемой процессами кластера. В результате минимизируется взаимное влияние различных процессов и исключается снижение производительности из-за конкуренции за сетевые ресурсы.

Для повышения отказоустойчивости и устранения единой точки отказа данные реплицируются между всеми тремя узлами.

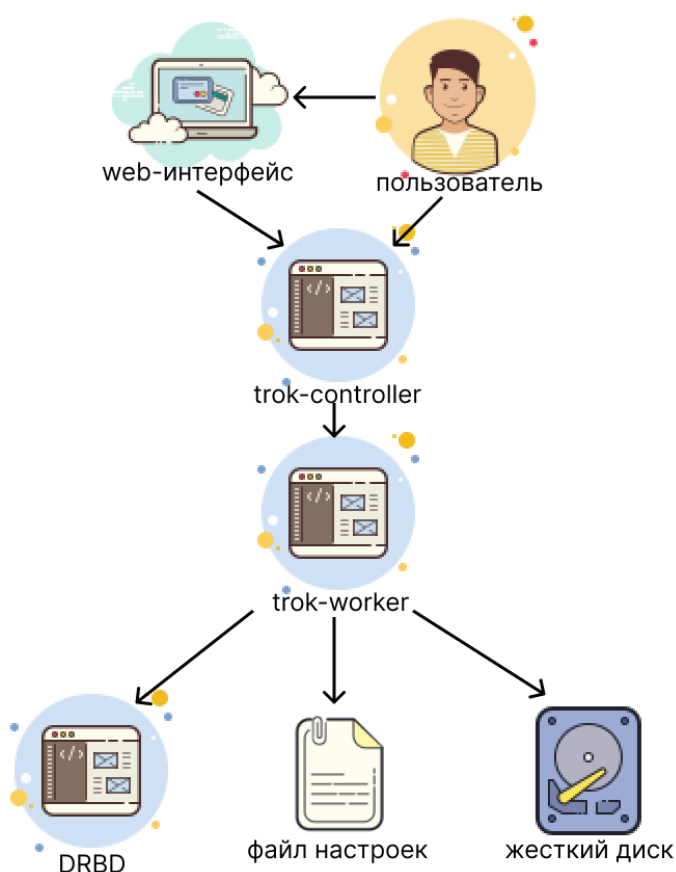


Рисунок 1 – Взаимодействие основных компонентов программного обеспечения

Техническая реализация программного обеспечения включает интеграцию четырех основных компонентов:

- trok-controller – центральный модуль, отвечающий за оркестрацию кластерных ресурсов и управление конфигурацией распределенного хранилища данных.

- trok-worker – служба, которая работает на каждом узле (сервере), где должны быть размещены реплицированные данные. Worker отвечает за управление локальными ресурсами хранения и взаимодействует с trok-controller.

- trok web GUI – веб-интерфейс, предназначенный для управления функциями системы хранения данных.

- trok-harbor – готовый программный модуль, который объединяет три ключевых элемента:

- drbd-reactor – событийно-ориентированная служба автоматизации для DRBD, которая в реальном времени отслеживает состояние узлов и ресурсов

DRBD и автоматически выполняет заданные действия (сценарии агентов ресурсов) при изменении этих состояний, например, при переключении ролей Primary/Secondary в кластере, обеспечивая автоматическое управление ресурсами.

- resource-agents — это набор готовых сценариев (агентов OCF), которые используются в TROK для управления высокодоступными хранилищами на DRBD. Агенты позволяют автоматизировать контроль состояния ресурсов и обеспечивают отказоустойчивость сервисов доступа.

- сервисы доступа – поддержка протоколов: iSCSI, NFS, SMB.

trok-harbor **поставляется в формате отдельного мета-пакета** и должен быть установлен на всех узлах типа Worker.

1.1. Режим без локального диска

Diskless (бездисковый) режим – это режим работы DRBD, когда узел не имеет локального дискового хранилища, но участвует в кластере как узел доступа к DRBD-ресурсу.

Если выразаться более простым языком, Diskful-сервер (с дисками) – выступает в качестве аналога склада. Он хранит данные. Diskless-сервер (без дисков) может использоваться как некий пункт выдачи. Он не хранит, но даёт доступ к тому, что есть на складе.

Бездисковый узел может выполнять функции арбитра (решающего голоса) для поддержания кворума в кластере, предотвращая ситуации с разделением ресурсов, когда узлы не могут договориться о том, какие данные являются авторитетными. Он также может выступать в роли клиента, считывающего и записывающего данные с других узлов.

Преимущества режима Diskless:

- не требуется локальный диск на Diskless нодах;
- позволяет размещать Diskless узлы на любой платформе (например, на гипервизорах в конвергентных системах);
- позволяет быстро добавлять и удалять узлы;

- обеспечивает больше возможных точек доступа к данным;
- быстрый failover;
- не теряет связь с ресурсом, если вышел из строя диск на Diskful-узле.

Бездисковый узел имеет следующие особенности:

- нет локального хранения данных – только метаданные в оперативной памяти;
- зависимость от сети – при потере связи данные недоступны;
- требуется минимум 32MB RAM на ресурс для метаданных (32MB на 1 TB DRBD);
- метаданные хранятся в памяти – исчезают после перезагрузки;
- автоматическое восстановление при восстановлении связи.

1.2. Установка и настройка базовых компонентов

Базовые компоненты программного обеспечения, необходимые, для работы системы указаны в таблице ниже:

Таблица 1 – Базовые компоненты системы

Компонент	Требования	Проверка
trok-controller	Версия, указанная в названии дистрибутива	systemctl status trok-controller
trok-worker	Версия, указанная в названии дистрибутива	systemctl status trok-worker
trok-cp-endpoint	Предназначен для подключения к API TROK. Версия, указанная в названии дистрибутива	systemctl status trok-cp-endpoint
Ядро ОС	Версия 6.1	sudo uname -r

DRBD	Версия, совместимая с версией ядра. ≥ 9.0 Несовпадение версии ядра и версии DRBD может привести к неработоспособности хранения.	<pre>sudo apt list --installed grep drbd</pre> Пример вывода (также см. примечание под таблицей): <pre>drbd-6.1.152-1-generic/stable,stable,now 9.2.12-4 amd64 [установлен] # Установлен связанный с ядром модуль DRBD версии 9.2.12.</pre> <pre>drbd-reactor/stable,now 1.4.1-1 amd64 [установлен] # Установлен компонент реактора для управления ресурсами.</pre> <pre>drbd-ueficert/now 9.2.11-1 all [установлен, локальный] # Установлен сертификат UEFI для DRBD.</pre> <pre>drbd-utils/stable,now 9.28.0-1 amd64 [установлен] # Установлены утилиты DRBD версии 9.28.0.</pre>
drbd-reactor	Версия > 1.4. Требует в качестве зависимости Python ≥ 3.6. Конфигурационные файлы в <code>/etc/drbd-reactor/</code>	Версия drbd-reactor будет указана в выводе предыдущей команды, но также существует отдельная команда для проверки: <pre>sudo drbd-reactor --version</pre>
Загруженные модули ядра DRBD	DRBD должен быть не просто установлен, а также загружен в ядро ОС как модуль и готов к работе. Если в выводе строки drbd нет: DRBD не загружен – либо из-за ошибки, либо	<pre>sudo lsmod grep -i drbd</pre> Пример вывода: <pre>drbd 819200 0 # Основной модуль DRBD загружен (размер 819200 байт, используется 0 процессами).</pre>

	несовместимости. Нужно выполнить установочную команду <code>trok-drbd-autoinstall</code> от имени суперпользователя или разбираться с ошибкой ядра.	<code>lru_cache 16384 1 drbd # Зависимый модуль LRU-кэша, используемый DRBD.</code> <code>libcrc32c 16384 6 nf_conntrack, nf_nat, dm_persistent_data, nf_tables, drbd, sctp # Модуль CRC32C, включая DRBD как одного из клиентов.</code>
LVM	LVM version 2.x.x	<code>sudo lvs --version</code>
WEB-UI	Веб-сервер (Apache) с HTTPS Порт 80 (или кастомный)	<code>sudo systemctl status apache2</code>
resource-agents	Набор OCF-скриптов для управления ресурсами кластера Поддержка агентов: DRBD, Filesystem, IPaddr2, portblock, nfsserver, exportfs, iSCSITarget, iSCSILogicalUnit	<code>sudo apt list --installed grep resource-agents</code> <code>ls /usr/lib/ocf/resource.d/heartbeat/</code> # Проверка наличия основных OCF-агентов для DRBD и файловых систем <code>ls /usr/lib/ocf/resource.d/heartbeat/ grep -E "(DRBD Filesystem IPaddr2 portblock)"</code> # Проверка наличия основных агентов для сервисов доступа <code>ls /usr/lib/ocf/resource.d/heartbeat/ grep -E "(nfsserver exportfs iSCSI smb-share)"</code>

Примечание. Вывод команды `sudo apt list --installed | grep drbd` зависит от того, на каком этапе установки был сделан запрос. Если вы установили `trok-worker` и `trok-controller` в результате выполнения команды вы получите только 2 строки:

```
drbd-6.1.152-1-generic/stable,stable,now 9.2.12-4 amd64 [установлен] # Установлен
связанный с ядром модуль DRBD версии 9.2.12.
```

```
drbd-utils/stable,now 9.28.0-1 amd64 [установлен] # Установлены утилиты DRBD
версии 9.28.0
```

Если помимо trok-worker и trok-controller вы установили trok-harbor в результате выполнения команды вы получите 3 строки:

```
drbd-6.1.152-1-generic/stable,stable,now 9.2.12-4 amd64 [установлен] # Установлен
связанный с ядром модуль DRBD версии 9.2.12.
```

```
drbd-reactor/stable,now 1.4.1-1 amd64 [установлен] # Установлен компонент
реактора для управления ресурсами.
```

```
drbd-utils/stable,now 9.28.0-1 amd64 [установлен] # Установлены утилиты DRBD
версии 9.28.0
```

Если вы выполнили все предыдущие условия и настроили сертификат, в результате выполнения команды вы получите вывод, который был указан в таблице.

1.3. Подготовка топологии сети и соблюдение аппаратных требований

Для комфортной работы желательно использовать минимум 3 серверные ноды, имеющие следующие характеристики:

- Процессор: 4+ ядер (x86_64), частота: от 2 ГГц;
- Размер оперативной памяти:
 - Минимальный объем: 8 ГБ;
 - Рекомендуемый объем: 64 ГБ и более;
- Тип оперативной памяти: DDR4, DDR5, ECC (защита от сбоев).
- Объем диска: 2 накопителя:
 - Дисковое пространство для операционной системы и программного обеспечения емкостью 128 ГБ и более (SAS/SATA HDD/SSD);
 - Дисковое пространство для хранения данных TROK (серверы хранения данных): 1 накопитель (и более) с емкостью не менее 1 ТБ (SAS/SATA HDD).;

– Для программного обеспечения TROK: 1 накопитель (и более) с емкостью не менее 128 ГБ (SAS/SATA HDD).

1.4. Сетевые интерфейсы

Рекомендации по скорости соединения:

- Минимум: 1 Гбит/с (1 GbE);
- Рекомендуется: 10 Гбит/с (10 GbE) и выше (особенно для репликации DRBD, чтобы избежать задержек при синхронизации данных между узлами)

Максимальный размер блока данных (в байтах), который может быть передан через сетевой интерфейс без фрагментации (размер MTU):

- Минимум: 1500 байт (стандартный Ethernet);
- Рекомендуется: Jumbo Frames (MTU = 9000) (критично для NVMe-oF, так как увеличивает пропускную способность и снижает накладные расходы при передаче данных).

Для обеспечения стабильной производительности и низкой задержки рекомендуется использовать отдельные сетевые пути для синхронизации данных DRBD и клиентского доступа. Это предотвращает конфликт трафика и повышает надежность системы.

1.5. Требования к сети

Все ноды должны находиться в одной подсети без NAT (например, 192.168.100.1/24).

Каждое устройство должно иметь статический IP-адрес или использовать резервирование DHCP.

Рекомендуется настроить параметр MTU (Maximum Transmission Unit) на сетевых интерфейсах для оптимизации передачи данных и предотвращения фрагментации пакетов. Для высокоскоростных сетей (10 Гбит/с и выше) рекомендуется использовать значение MTU 9000 (Jumbo Frames) для повышения пропускной способности и снижения нагрузки на процессор. Для корректной настройки параметров обратитесь к официальной документации вашего

дистрибутива, так как конфигурация может зависеть от конкретной реализации и версии ОС.

1.6. Резервное копирование

Перед выполнением любых операций по настройке или модификации системы хранения, если в системе хранения у вас уже размещены какие-то данные, настоятельно рекомендуется предварительно создать резервную копию всей информации, находящейся в системе. Это необходимо для предотвращения потери важных файлов и исключения риска их повреждения в случае ошибочных действий или сбоев в процессе изменения конфигурации. Такой подход позволяет обезопасить ваши ресурсы даже при возникновении непредвиденных ситуаций.

2. ИМЯ УЗЛА И ВОЗМОЖНЫЕ КОНФЛИКТЫ ПРИ СОЗДАНИИ

Перед тем как перейти к описанию правил именования, необходимо чётко разграничить используемые термины:

- имя хоста – сетевой идентификатор физического или виртуального сервера в операционной системе, обычно используемый для обращения к серверу в локальной сети или через DNS;

- имя узла типа воркер TROK – уникальный идентификатор, который присваивается каждому компоненту (узлу типа воркер) в программном обеспечении TROK. Данное имя используется для управления ресурсами кластера.

При выборе уникального имени узла хранения (узел типа воркер) TROK рекомендуется учитывать следующие правила:

- имя узла воркера в системе TROK и имя хоста должны совпадать;
- разрешенные базовые символы – латинские буквы: a-z, A-Z; цифры: 0-9; спецсимволы: - (дефис); . (точка);
- начало имени не может начинаться со спецсимвола (например, имя .server недопустимо);
- длина уникального имени должна составлять не более 48 символов;
- регистр букв не учитывается (например, Node1 и node1 считаются идентичными);
- запрещенные символы и форматы – пробелы, кириллица, символы: _, @, #, \$, %, ^, &, *, (), {}, [], /, \, :, ;, ", ', ~, !;
- запрещено использовать IP-адреса в качестве имён узла (например, 192.168.1.1 недопустимо).

Если имя узла воркера в системе TROK и имя хоста отличаются, можно воспользоваться следующими сценариями:

Приоритетный сценарий:

- Изменить имя хоста, используйте команду:

```
sudo hostnamectl set-hostname <имя_хоста>
```

Важно. Данная операция может повлечь за собой необходимость корректировки конфигурационных файлов иных программных компонентов, установленных в операционной системе. Поэтому перед выполнением изменения следует тщательно оценить потенциальное влияние на функционирование стороннего программного обеспечения и убедиться в безопасности данной процедуры для текущих процессов.

Сценарий, который используется в редких случаях и только в определенных условиях:

- Изменить имя узла воркера в системе TROK.

Важно. Если узел участвовал в репликации, необходимо выполнить её перенастройку. Рекомендуется применять данную процедуру только в ситуациях, когда узел более не должен входить в состав кластера, перед переустановкой системы или выводом сервера из эксплуатации, а также если узел был добавлен в самом начале настройки ошибочно. Перед выполнением этих операций рекомендуется провести резервное копирование конфигурации и убедиться, что ни один из ресурсов не останется в состоянии ошибки.

– Система TROK не имеет команды, изменяющей имя узла напрямую, поскольку имя узла тесно связано с её уникальным идентификатором в базе данных контроллера. Чтобы изменить имя узла, необходимо:

- Удаление старого узла;
- Добавление нового узла с правильным именем, используя тот же IP-адрес, что у старого узла.

Если поменять порядок шагов (например, сначала добавить новый узел с тем же IP-адресом), возникает ошибка вида: `rpc error: code = Unknown desc = address <IP-адрес> and port 50002 are already in use by net interface uuid.`

3. ПОДГОТОВКА К РАЗМЕТКЕ ДИСКОВ

Предварительная разметка дисков выполняется на узлах, которые используются для хранения данных (воркер-узлы). Для начала выведите детальный список устройств хранения используя команду:

```
sudo fdisk -l
```

```

success-express@astra-1:~$ sudo fdisk -l
[sudo] пароль для success-express:
Диск /dev/sda: 20 GiB, 21474836480 байт, 41943040 секторов
Модель диска: VBOX HARDDISK
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: dos
Идентификатор диска: 0x0c6a8fd7

Устр-во   Загрузочный  Начало    Конец     Секторы  Размер  Идентификатор  Тип
/dev/sda1 *          2048    39942143  39940096    19G      83 Linux
/dev/sda2          39944190  41940991  1996802    975M      5 Расширенный
/dev/sda5          39944192  41940991  1996800    975M      82 Linux подкачка / Solaris

Диск /dev/sdb: 7 GiB, 7516192768 байт, 14680064 секторов
Модель диска: VBOX HARDDISK
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
success-express@astra-1:~$

```

Рисунок 2 – Пример результата выполнения команды `sudo fdisk -l`

В текущем примере диск «sdb» свободен и готов к разметке. Именно он используется в дальнейшей работе.

Существуют альтернативные способы распределения дискового пространства. На изображении ниже мы видим **нежелательный вариант конфигурации системы**: один виртуальный диск на 20 ГБ (/dev/sda) и три его раздела:

- /dev/sda1 (19 ГБ, основной раздел, тип Linux) – системный раздел с ОС и всеми файлами;
- /dev/sda2 (975 МБ, расширенный);
- /dev/sda5 (975 МБ, подкачка – swap).

```

success-express@astra-1:~$ sudo fdisk -l
[sudo] пароль для success-express:
Диск /dev/sda: 20 GiB, 21474836480 байт, 41943040 секторов
Модель диска: VBOX HARDDISK
Единицы: секторов по 1 * 512 = 512 байт
Размер сектора (логический/физический): 512 байт / 512 байт
Размер I/O (минимальный/оптимальный): 512 байт / 512 байт
Тип метки диска: dos
Идентификатор диска: 0xc6a8fd7

Устр-во   Загрузочный  Начало     Конеч      Секторы  Размер  Идентификатор  Тип
/dev/sda1 *      2048 39942143 39940096    19G      83 Linux
/dev/sda2          39944190 41940991 1996802     975M      5 Расширенный
/dev/sda5          39944192 41940991 1996800     975M      82 Linux подкачка / Solaris
success-express@astra-1:~$

```

Рисунок 3 – Пример нежелательного набора устройств хранения

Единственный способ отделить необходимое количество дискового пространства в таком случае под TROK на этом диске – уменьшить текущий системный раздел `/dev/sda1`.

Это опасная процедура, требующая:

- Полного резервного копирования данных и ОС!
- Перезапуска системы через live-среду (например, GParted Live USB/CD), так как нельзя уменьшать смонтированный root-раздел (где работает ваша ОС).

Важно. Чтобы не допускать подобную ситуацию, при установке системы требуется обязательно производить предварительную разметку дисков.

4. РАЗМЕТКА ДИСКОВ LVM ДЛЯ ПУЛОВ ХРАНЕНИЯ

Разметка диска (fdisk или parted) требуется только в том случае, если под хранилище TROK выделяется не весь диск целиком, а его часть — отдельный раздел. Если для хранилища используется выделенный физический диск, на котором нет других данных и системных разделов, то можно указывать в операции pvcreate устройство /dev/sdX без предварительной разметки через fdisk.

LVM (Logical Volume Manager) – это система управления разделами диска в Linux, которая позволяет объединять несколько физических дисков или их разделов в одну большую группу хранения (volume group). Внутри этой группы можно создавать логические тома (logical volumes), которые, по сути, работают как обычные разделы, но могут быть динамически увеличены или уменьшены, перемещены между дисками, а также быстро сохранены как снимки (снапшоты). Использование LVM делает структуру хранения данных более гибкой, удобной для резервного копирования и масштабирования без необходимости переформатировать весь диск или останавливать систему.

С помощью классического LVM каждый логический том, выделяемый TROK для виртуальной машины или контейнера, полностью резервирует необходимое дисковое пространство заранее. Это гарантирует предсказуемое использование ресурсов и надёжность работы, так как том никогда не «закончится» неожиданно из-за нехватки места.

Сначала определите диски, с которыми планируется работать. Чтобы получить подробную информацию о дисках в системе, используйте команду:

```
lsblk
```

Процесс разметки дисков производится следующим образом:

Подготовьте место на диске для дальнейшей работы с помощью утилиты для работы с разделами fdisk:

```
sudo fdisk /dev/<имя_диска>
```

Важно. Все данные на диске будут удалены. Убедитесь, что выбрали правильный диск.

Внутри fdisk последовательно введите необходимые команды, подтверждая ввод нажатием клавиши ENTER:

n → создать новый раздел.

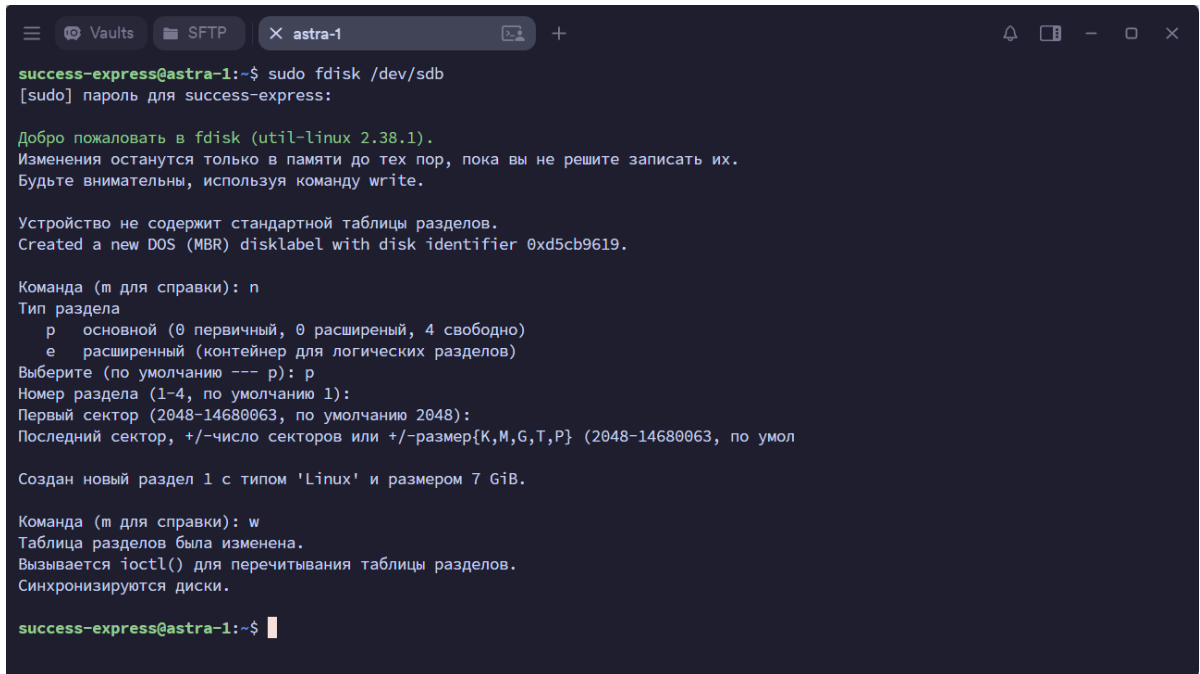
p → выбрать тип раздела: основной (primary).

Enter → принять номер раздела по умолчанию.

Enter → принять первый сектор по умолчанию (2048).

Enter → принять последний сектор по умолчанию (весь диск).

w → сохранить изменения и выйти.



```

success-express@astra-1:~$ sudo fdisk /dev/sdb
[sudo] пароль для success-express:

Добро пожаловать в fdisk (util-linux 2.38.1).
Изменения останутся только в памяти до тех пор, пока вы не решите записать их.
Будьте внимательны, используя команду write.

Устройство не содержит стандартной таблицы разделов.
Created a new DOS (MBR) disklabel with disk identifier 0xd5cb9619.

Команда (m для справки): n
Тип раздела
  p  основной (0 первичный, 0 расширенный, 4 свободно)
  e  расширенный (контейнер для логических разделов)
Выберите (по умолчанию --- p): p
Номер раздела (1-4, по умолчанию 1):
Первый сектор (2048-14689063, по умолчанию 2048):
Последний сектор, +/-число секторов или +/-размер{К,М,Г,Т,Р} (2048-14689063, по умол

Создан новый раздел 1 с типом 'Linux' и размером 7 GiB.

Команда (m для справки): w
Таблица разделов была изменена.
Вызывается ioctl() для перечитывания таблицы разделов.
Синхронизируются диски.

success-express@astra-1:~$

```

Рисунок 4 – Процесс подготовки места на диске для дальнейшей работы

Для инициализации раздела под LVM создайте физический том (PV) используя команду:

```
sudo pvcreate /dev/<имя_раздела>
```

pvcreate – команда для инициализации диска/раздела под LVM.

/dev/<имя_раздела> – созданный раздел (после fdisk).

```

success-express@astra-1:~$ sudo pvcreate /dev/sdb1
Physical volume "/dev/sdb1" successfully created.
success-express@astra-1:~$

```

Рисунок 5 – Пример вывода при успешном выполнении команды `sudo pvcreate /dev/sdb1`
 Для просмотра созданного физического тома используйте команду:

```
sudo pvdisplay
```

```

success-express@astra-1:~$ sudo pvdisplay
"/dev/sdb1" is a new physical volume of "<7,00 GiB"
--- NEW Physical volume ---
PV Name               /dev/sdb1
VG Name
PV Size                <7,00 GiB
Allocatable            NO
PE Size                0
Total PE               0
Free PE                0
Allocated PE           0
PV UUID                ngZI2f-tK4G-9ZF4-nviz-0b79-aaDX-tCXt3n

success-express@astra-1:~$

```

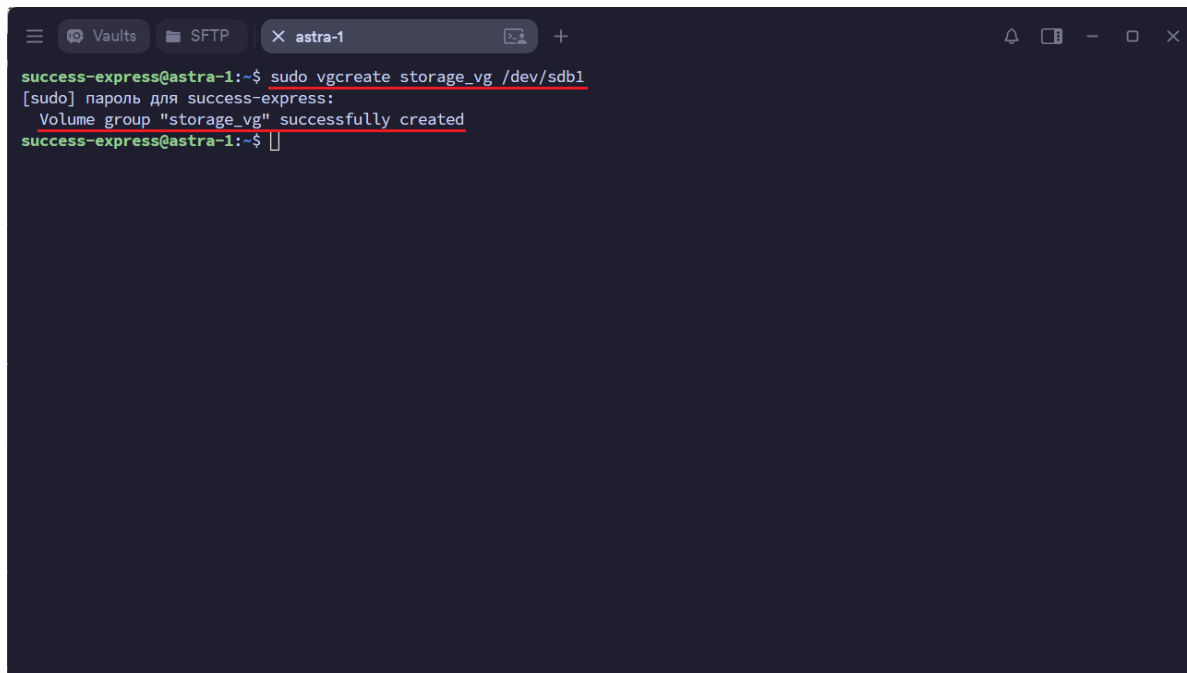
Рисунок 6 – Отображение созданного тома (PV)

Создайте группу томов (VG):

```
sudo vgcreate <имя_группы_ресурсов> /dev/<имя_раздела>
```

vgcreate – создает группу томов (логический контейнер для дисков).

Имя группы (должно быть уникальным в системе, можно выбрать любое, например, data_vg).

A screenshot of a terminal window with a dark background. The window title bar shows 'Vaults', 'SFTP', and 'astra-1'. The terminal text shows a user 'success-express' at host 'astra-1' running the command 'sudo vgcreate storage_vg /dev/sdb1'. The output shows a password prompt, followed by 'Volume group "storage_vg" successfully created', and then the prompt returns to the user. The command and the success message are underlined in red in the original image.

```
success-express@astra-1:~$ sudo vgcreate storage_vg /dev/sdb1
[sudo] пароль для success-express:
Volume group "storage_vg" successfully created
success-express@astra-1:~$
```

Рисунок 7 – Уведомление об успешном создании группы томов (VG)

Для проверки созданных объектов используйте команды:

```
sudo pvdisplay | grep -B 1 <имя_раздела> # Фильтрует вывод, показывая строки с
интересующим именем и дополнительную строку перед ними для контекстного понимания
информации.
```

```
sudo vgdisplay | grep -B 1 <имя_группы_ресурсов> # Аналогичный вывод с фильтрацией
по имени тома
```

5. РАЗМЕТКА ДИСКОВ LVM_THIN ДЛЯ ПУЛОВ ХРАНЕНИЯ

Разметка дисков производится на нодах, на которых размещена служба trok-worker.

Как и в предыдущем случае, операция необходима только в случае, если для хранения планируется использовать не весь диск, а его часть.

LVM_THIN – это усовершенствованная технология внутри LVM, реализующая так называемое «тонкое выделение» (thin provisioning). В обычных томах LVM всё выделенное под логический том пространство резервируется сразу и полностью, даже если данные ещё не записаны. В LVM_THIN пространство для томов выделяется динамически, по мере необходимости: создаётся большой «thin pool» – специальный пул хранения, из которого каждый логический том получает столько реального места, сколько в действительности занимает его содержимое в данный момент. Это позволяет более эффективно использовать дисковое пространство и хранить, например, множество виртуальных машин или контейнеров, не резервируя под каждую физически максимально возможный объём, а выделяя ресурсы по факту использования.

LVM_THIN применяется в TROK для более гибкого и эффективного использования диска в крупных кластерах. При тонком выделении (LVM_THIN) TROK может создать множество томов, которые будут занимать ровно столько пространства, сколько им нужно в данный момент времени. Это особенно полезно для сценариев с динамическими виртуальными машинами и контейнерами – пользователей можно не ограничивать по размеру тома, а общий пул хранит только фактические данные. Таким образом, LVM_THIN экономит место и обеспечивает лёгкое масштабирование, а автоматизация TROK позволяет отслеживать использование.

Определите диски, с которыми планируется работать. Чтобы получить подробную информацию о дисках в системе, используйте команду:

```
lsblk
```

Для начала создания LVM_THIN подготовьте место на диске для дальнейшей работы с помощью утилиты для работы с разделами fdisk:

```
sudo fdisk /dev/<имя_диска>
```

Важно. Все данные на диске будут удалены. Убедитесь, что выбрали правильный диск.

Внутри fdisk введите команды по порядку:

n → создать новый раздел.

p → выбрать тип раздела: основной (primary).

Enter → принять номер раздела по умолчанию.

Enter → принять первый сектор по умолчанию (2048).

Enter → принять последний сектор по умолчанию (весь диск).

w → сохранить изменения и выйти.

Для создания LVM Thin Pool необходимо выполнить следующую последовательность команд:

Создать физический том (PV):

```
sudo pvcreate /dev/<имя_раздела> # помечаем раздел как «физический том» для LVM
```

pvcreate – команда для инициализации диска/раздела под LVM.

/dev/<имя_раздела> – созданный раздел (после fdisk).

Создать группу томов (VG):

```
sudo vgcreate <имя_группы_ресурсов> /dev/<имя_раздела>
```

vgcreate – создает группу томов (логический контейнер для дисков).

Имя группы (должно быть уникальным в системе, можно выбрать любое, например, data_vg).

Внимание: для пулов хранения типа LVM_THIN в рамках одной группы томов (VG) допускается создание только одного тонкого пула (thin pool). Наличие двух и более thin pool внутри одной VG недопустимо. Если требуется несколько независимых thin pool, для каждого должна быть создана отдельная VG (группа томов). Перед добавлением нового пула

проверьте существующие thin pool в целевую VG и спланируйте топологию хранилища заранее.

Создать «тонкий пул» в группе (Thin Pool):

```
sudo lvcreate -l 100%FREE -T <имя_группы_томов>/<имя_тома_lvm_thin>
```

lvcreate – создает логический том.

-l 100%FREE – использовать всё свободное место в группе.

-T – указание создать Thin Pool (пул с динамическим выделением места).

<имя_группы_ресурсов>/<имя_тома_lvm_thin> – том располагается внутри группы.

Для проверки созданных сущностей воспользуйтесь командой:

```
sudo lvdisplay | grep -E "<имя_логического_тома_в_LVM>|<имя_группы_томов_в_LVM>"
```

В случае возникновения необходимости в удалении созданных ресурсов, можно воспользоваться следующими командами:

– удаление группы томов:

```
sudo vgremove <имя_группы_ресурсов>
```

– удаление физического тома:

```
sudo pvremove /dev/<имя_раздела>
```

6. ПРОВЕРКА И УСТАНОВКА ПАКЕТА TROK-WEBUI

Прежде чем приступать к настройке HTTPS, необходимо убедиться, что пакет `trok-webui` установлен на всех узлах, на которых должен работать веб-интерфейс TROK.

Для точной проверки состояния пакета выполните:

```
dpkg-query -W \
    -f='${binary:Package} ${db:Status-Abbrev} ${Version} ${Architecture}\n' \
    trok-webui
```

Если пакет установлен, команда вернёт строку следующего вида:

```
trok-webui ii 2.1~rc.4 all
```

Здесь:

- `trok-webui` — имя пакета;
- `ii` — пакет установлен и настроен;
- `2.1~rc.4` — установленная версия пакета, которая может отличаться в зависимости от используемого репозитория TROK;
- `all` — пакет не зависит от архитектуры процессора.

Если пакет отсутствует, команда завершится сообщением:

```
dpkg-query: no packages found matching trok-webui
```

На сервере без НА установите пакет из настроенного репозитория TROK:

```
sudo apt update
sudo apt install trok-webui
```

Пакет `trok-webui` устанавливает Apache HTTP Server как зависимость, размещает файлы веб-интерфейса в каталоге `/var/www/trok-webui`, включает необходимые модули Apache и активирует сайт `/etc/apache2/sites-available/trok-webui.conf`.

Пакет конфликтует с пакетами `nginx`. Если `nginx` установлен, перед подтверждением операции внимательно проверьте список пакетов, который предлагает удалить APT.

На HA-кластере не следует устанавливать `trok-webui` вручную отдельной командой `apt install`. Установка должна выполняться штатным сценарием развёртывания TROK или соответствующей ролью Ansible. В HA-кластере служба `apache2.service` управляется `drbd-reactor` и должна работать только на активном контроллере.

После установки проверьте конфигурацию Apache:

```
sudo apache2ctl configtest
```

Успешный результат:

```
Syntax OK
```

Проверьте установленный пакет и конфигурационный файл:

```
dpkg-query -W -f='${Status} ${Version}\n' trok-webui
ls -l /etc/apache2/sites-available/trok-webui.conf
ls -l /etc/apache2/sites-enabled/trok-webui.conf
ls -l /var/www/trok-webui/index.html
```

На сервере без HA служба Apache должна быть запущена:

```
systemctl is-active apache2
systemctl is-enabled apache2
```

Ожидаемый результат:

```
active
enabled
```

В HA-кластере Apache должен быть активен только на текущем активном контроллере. На пассивных контроллерах состояние `inactive` является ожидаемым. Не запускайте и не включайте Apache вручную на пассивных контроллерах.

7. ПОДГОТОВКА SSL/TLS-СЕРТИФИКАТОВ

Для защиты соединения между браузером и веб-интерфейсом TROK рекомендуется использовать HTTPS.

Сертификат должен содержать DNS-имя или IP-адрес, по которому пользователи открывают WebUI. Это значение должно присутствовать в расширении Subject Alternative Name, сокращённо SAN.

В этой инструкции используются следующие пути:

```
/etc/apache2/ssl/trok-webui.crt  
/etc/apache2/ssl/trok-webui.key
```

Файл trok-webui.crt должен содержать сертификат сервера и, при необходимости, промежуточные сертификаты. Файл trok-webui.key должен содержать соответствующий закрытый ключ.

Создайте отдельный каталог:

```
sudo install -d \  
-o root \  
-g root \  
-m 0700 \  
/etc/apache2/ssl
```

Закрытый ключ должен быть доступен только пользователю root. Добавлять пользователя www-data в группу для чтения ключа не требуется: Apache читает ключ при запуске до снижения привилегий рабочих процессов.

7.1. Создание самоподписанного сертификата

Самоподписанный сертификат подходит для тестовых стендов. Браузеры не будут доверять ему автоматически, пока сертификат или выпустивший его центр сертификации не добавлен в доверенное хранилище клиента.

Для сертификата на DNS-имя замените trok.example.com на фактическое имя WebUI и выполните:

```

sudo openssl req \
    -x509 \
    -nodes \
    -newkey rsa:3072 \
    -sha256 \
    -days 365 \
    -keyout /etc/apache2/ssl/trok-webui.key \
    -out /etc/apache2/ssl/trok-webui.crt \
    -subj "/C=RU/O=TROK/OU=WebUI/CN=trok.example.com" \
    -addext "subjectAltName=DNS:trok.example.com" \
    -addext "basicConstraints=critical,CA:FALSE" \
    -addext "keyUsage=critical,digitalSignature,keyEncipherment" \
    -addext "extendedKeyUsage=serverAuth"

```

Если пользователи обращаются к WebUI по IP-адресу, замените параметры CN и subjectAltName, например:

```

-subj "/C=RU/O=TROK/OU=WebUI/CN=192.168.1.10"
-addext "subjectAltName=IP:192.168.1.10"

```

После создания сертификата установите права:

```

sudo chown root:root \
    /etc/apache2/ssl/trok-webui.crt \
    /etc/apache2/ssl/trok-webui.key

sudo chmod 0644 /etc/apache2/ssl/trok-webui.crt

sudo chmod 0600 /etc/apache2/ssl/trok-webui.key

```

Проверьте срок действия и SAN:

```

openssl x509 \
    -in /etc/apache2/ssl/trok-webui.crt \
    -noout \
    -subject \
    -issuer \
    -dates \
    -ext subjectAltName

```

7.2. Установка сертификата внутреннего центра сертификации

Для установки сертификата внутреннего центра сертификации потребуются:

- server.crt — сертификат WebUI;
- server.key — закрытый ключ WebUI;
- intermediateCA.crt — промежуточный сертификат, если он используется;
- rootCA.crt — доверенный корневой сертификат.

Сначала проверьте сертификат сервера.

Если сертификат подписан через промежуточный центр сертификации, выполните:

```
openssl verify \  
-CAfile rootCA.crt \  
-untrusted intermediateCA.crt \  
server.crt
```

Ожидаемый результат:

```
server.crt: OK
```

Если сертификат подписан непосредственно корневым центром сертификации, выполните:

```
openssl verify \  
-CAfile rootCA.crt \  
server.crt
```

Установите закрытый ключ:

```
sudo install \  
-o root \  
-g root \  
-m 0600 \  
server.key \  
/etc/apache2/ssl/trok-webui.key
```

Если используется промежуточный центр сертификации, сформируйте fullchain. Первым должен идти сертификат сервера, затем промежуточные сертификаты в порядке от ближайшего к серверу до вышестоящего:

```
cat server.crt intermediateCA.crt |
```

```
sudo tee /etc/apache2/ssl/trok-webui.crt >/dev/null
```

Корневой сертификат обычно не включается в передаваемый сервером fullchain.

Установите права на fullchain:

```
sudo chown root:root /etc/apache2/ssl/trok-webui.crt
sudo chmod 0644 /etc/apache2/ssl/trok-webui.crt
```

Если сертификат подписан непосредственно корневым центром сертификации и промежуточного сертификата нет, выполните:

```
sudo install \
-o root \
-g root \
-m 0644 \
server.crt \
/etc/apache2/ssl/trok-webui.crt
```

7.3. Установка готового fullchain-сертификата

Если центр сертификации выдал готовые файлы trok-webui.fullchain.crt и trok-webui.key, установите их следующими командами:

```
sudo install \
-o root \
-g root \
-m 0644 \
trok-webui.fullchain.crt \
/etc/apache2/ssl/trok-webui.crt

sudo install \
-o root \
-g root \
-m 0600 \
trok-webui.key \
/etc/apache2/ssl/trok-webui.key
```

Проверьте сертификат:

```
openssl x509 \
-in /etc/apache2/ssl/trok-webui.crt \
-noout \
-subject \
-issuer \
-dates \
```

```
-ext subjectAltName
```

7.4. Проверка соответствия сертификата и закрытого ключа

Получите контрольную сумму публичного ключа из сертификата:

```
openssl x509 \
-in /etc/apache2/ssl/trok-webui.crt \
-pubkey \
-noout |
openssl pkey -pubin -outform DER |
sha256sum
```

Получите контрольную сумму публичной части закрытого ключа:

```
sudo openssl pkey \
-in /etc/apache2/ssl/trok-webui.key \
-pubout \
-outform DER |
sha256sum
```

Контрольные суммы должны совпадать. Если они различаются, сертификат и закрытый ключ не образуют пару, и Apache не сможет использовать их вместе.

7.5. Установка корневого сертификата на клиентских устройствах

Корневой сертификат внутреннего центра сертификации необходимо устанавливать на клиентских устройствах, с которых пользователи открывают WebUI.

Установка корневого сертификата на сервер TROK не требуется только для того, чтобы Apache мог передавать сертификат WebUI. Сервер должен передавать сертификат WebUI и промежуточные сертификаты, а клиент должен доверять корневому сертификату.

Debian, Ubuntu и Astra Linux

Скопируйте только корневой сертификат:

```
sudo install \
-o root \
-g root \
-m 0644 \
rootCA.crt \
/usr/local/share/ca-certificates/trok-root-ca.crt
```

```
sudo update-ca-certificates
```

Промежуточный сертификат обычно не следует устанавливать как отдельный доверенный корень. Он должен входить в fullchain, который передаёт Apache.

Windows

Для установки сертификата для всех пользователей откройте оснастку сертификатов локального компьютера:

```
certlm.msc
```

Импортируйте rootCA.crt в хранилище:

Доверенные корневые центры сертификации

Для централизованного развёртывания в домене используйте групповые политики.

8. ВКЛЮЧЕНИЕ HTTPS ДЛЯ TROK WEBUI

Штатная конфигурация WebUI находится в файле:

```
/etc/apache2/sites-available/trok-webui.conf
```

Не создавайте отдельный минимальный HTTPS-vhost с DocumentRoot /var/www/html. HTTPS-конфигурация должна сохранять все штатные правила WebUI: обслуживание SPA, проксирование API, заголовки, тайм-ауты и сжатие.

Перед изменением создайте резервную копию:

```
sudo cp -a \
  /etc/apache2/sites-available/trok-webui.conf \
  /etc/apache2/sites-available/trok-webui.conf.before-tls
```

Откройте файл:

```
sudo nano /etc/apache2/sites-available/trok-webui.conf
```

Замените его содержимое следующей конфигурацией. Вместо trok.example.com укажите фактическое DNS-имя или IP-адрес, присутствующий в SAN сертификата:

```
# TROK выполняет аутентификацию через собственный control-plane API.
# На Astra Linux отсутствие этой директивы может включить обязательную
# аутентификацию Apache и привести к ошибке AM00001.
AstraMode off

<VirtualHost *:80>
    ServerName trok.example.com

    Redirect permanent / https://trok.example.com/
</VirtualHost>

<VirtualHost *:443>
    ServerName trok.example.com

    DocumentRoot /var/www/trok-webui
    DirectoryIndex index.html

    <Directory /var/www/trok-webui>
        Options -Indexes
        AllowOverride None
        Require all granted
        FallbackResource /index.html
    </Directory>
```

```

<Files "index.html">
    Header always set Cache-Control "no-cache, no-store, must-revalidate"
    Header always set Pragma "no-cache"
    Header always set Expires "0"
</Files>

ProxyRequests Off
ProxyPreserveHost On
ProxyAddHeaders On

SetEnvIf Remote_Addr "^(.+) $" trok_client_addr=$1
RequestHeader set X-Real-IP "%{trok_client_addr}e"
RequestHeader set X-Forwarded-Proto "https"

ProxyPass          /v1/          http://127.0.0.1:8080/v1/ connectiontimeout=60
timeout=60
ProxyPassReverse   /v1/          http://127.0.0.1:8080/v1/

ProxyPass          /v2/          http://127.0.0.1:8080/v2/ connectiontimeout=60
timeout=60
ProxyPassReverse   /v2/          http://127.0.0.1:8080/v2/

ProxyPass          /metrics http://127.0.0.1:8080/metrics connectiontimeout=60
timeout=60
ProxyPassReverse   /metrics http://127.0.0.1:8080/metrics

DeflateCompressionLevel 5
AddOutputFilterByType DEFLATE      text/plain      application/javascript
text/javascript text/xml text/css

SSLEngine on
SSLCertificateFile /etc/apache2/ssl/trok-webui.crt
SSLCertificateKeyFile /etc/apache2/ssl/trok-webui.key

SSLProtocol -all +TLSv1.2 +TLSv1.3
SSLCompression off
SSLSessionTickets off
</VirtualHost>

```

Если WebUI открывается по IP-адресу, замените, например:

```

ServerName 192.168.1.10

Redirect permanent / https://192.168.1.10/

```

В сертификате при этом должно присутствовать:

```
subjectAltName=IP:192.168.1.10
```

Не добавляйте SSLSessionCache внутрь <VirtualHost>. На Debian и Astra Linux общий кеш TLS-сессий настраивается модулем Apache в файле /etc/apache2/mods-available/ssl.conf.

Включите необходимые модули:

```
sudo a2enmod \  
  ssl \  
  proxy \  
  proxy_http \  
  headers \  
  deflate \  
  setenvif \  
  dir \  
  mime
```

Активируйте сайт TROK:

```
sudo a2ensite trok-webui
```

Проверьте список виртуальных хостов:

```
sudo apache2ctl -S
```

Если активирован стандартный сайт default-ssl, он может конфликтовать с TROK WebUI на порту 443. Если этот сайт не используется другими приложениями, отключите его:

```
sudo a2dissite default-ssl
```

Не отключайте сторонние виртуальные хосты без предварительной проверки их назначения.

9. ПРОВЕРКА И ПРИМЕНЕНИЕ КОНФИГУРАЦИИ

Перед применением конфигурации обязательно проверьте синтаксис:

```
sudo apache2ctl configtest
```

Успешный результат:

```
Syntax OK
```

Дополнительно проверьте список модулей и виртуальных хостов:

```
sudo apache2ctl -M | grep -E 'ssl|proxy|proxy_http|headers|deflate'

sudo apache2ctl -S
```

9.1. Применение на сервере без НА

На обычном сервере без НА примените конфигурацию без полной остановки Apache:

```
sudo systemctl reload apache2
```

Проверьте состояние службы:

```
systemctl is-active apache2

systemctl status apache2 --no-pager
```

Ожидаемое состояние:

```
active
```

Если reload завершился ошибкой, не выполняйте повторные перезапуски.

Проверьте журнал:

```
sudo journalctl \
  -u apache2 \
  --since "10 minutes ago" \
  --no-pager
```

9.2. Применение в HA-кластере

Сертификат, закрытый ключ и одинаковый файл `/etc/apache2/sites-available/trok-webui.conf` должны быть установлены на всех контроллерах, которые могут стать активными.

На каждом контроллере выполните:

```
sudo apache2ctl configtest
```

Успешный результат на каждом узле:

```
Syntax OK
```

Затем проверьте состояние Apache на каждом контроллере:

```
systemctl is-active apache2
```

Apache должен быть активен только на текущем активном контроллере. На пассивных контроллерах он должен оставаться остановленным.

На каждом контроллере можно безопасно выполнить следующую команду:

```
if systemctl is-active --quiet apache2; then
    sudo systemctl reload apache2
fi
```

Команда перезагрузит конфигурацию только там, где Apache уже активен, и не запустит службу на пассивном узле.

Не выполняйте на HA-контроллерах:

```
sudo systemctl enable apache2
sudo systemctl start apache2
sudo systemctl restart apache2
```

без отдельной процедуры управления HA. Жизненным циклом Apache в HA управляет drbd-reactor.

9.3. Проверка HTTPS

Проверьте перенаправление HTTP на HTTPS:

```
curl -I http://trok.example.com/
```

Ожидается ответ:

```
HTTP/1.1 301 Moved Permanently
Location: https://trok.example.com/
```

Проверьте главную страницу:

```
curl -fI https://trok.example.com/
```

Ожидается успешный HTTP-ответ, например:

```
HTTP/1.1 200 OK
```

Для самоподписанного сертификата можно явно указать доверенный сертификат:

```
curl \
  --cacert /etc/apache2/ssl/trok-webui.crt \
  -fI \
  https://trok.example.com/
```

Параметр `-k` допускается только для временной диагностики. Он отключает проверку сертификата и не должен использоваться как подтверждение корректной настройки TLS.

Проверьте сертификат и его соответствие имени сервера:

```
openssl s_client \
  -connect trok.example.com:443 \
  -servername trok.example.com \
  -verify_hostname trok.example.com \
  -verify_return_error </dev/null
```

Для сертификата внутреннего центра сертификации, который ещё не установлен в системное хранилище клиента, добавьте корневой сертификат:

```
openssl s_client \
  -connect trok.example.com:443 \
  -servername trok.example.com \
  -verify_hostname trok.example.com \
  -verify_return_error \
  -CAfile rootCA.crt </dev/null
```

Успешная проверка должна содержать:

```
Verify return code: 0 (ok)
```

Откройте WebUI в браузере:

```
https://trok.example.com/
```

Проверьте:

- загрузку главной страницы;
- загрузку JavaScript- и CSS-файлов;
- переход по внутренним страницам WebUI;
- выполнение входа в TROK;
- выполнение запросов /v1/ и /v2/;
- отсутствие ошибок сертификата;
- отсутствие HTTP 500.

После проверки просмотрите ошибки Apache:

```
sudo tail -n 100 /var/log/apache2/error.log
```

На Astra Linux отдельно убедитесь, что после применения конфигурации не появились новые сообщения:

```
AM00001: authentication not configured
```

Если такое сообщение присутствует, проверьте наличие директивы:

```
AstraMode off
```

Она должна находиться перед блоками <VirtualHost> в активном файле /etc/apache2/sites-available/trok-webui.conf.

10. ОБЩИЕ СВЕДЕНИЯ О SDS UI (ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ДЛЯ УПРАВЛЕНИЯ СУЩНОСТЯМИ SDS)

Графический интерфейс системы управления хранилищами предоставляет интуитивно понятный инструмент для администрирования распределенных ресурсов хранения данных. Его основа состоит из нескольких ключевых разделов, каждый из которых решает специфические задачи в рамках единой архитектуры.

Параметры для входа:

Имя пользователя: "admin"

Пароль: "admin"

Для описания структуры интерфейса будем отталкиваться от основных сущностей TROK:

Resource (ресурс) – представляет собой блок хранения данных (набор реплицируемых DRBD томов), который с помощью технологии DRBD копируется на несколько узлов одновременно. Каждый ресурс состоит из одного или нескольких томов. Управление данной сущностью осуществляется в модуле «Ресурсы».

Snapshot (снимок состояния) – это совокупность снимков томов (snapshot volume). Каждый снапшот включает один или несколько томов и содержит копии файлов и каталогов файловой системы на конкретный момент времени. Для создания новых снимков состояния используйте вкладку «Снапшоты».

Node (узел) – основной компонент архитектуры системы хранения, представляющий собой физический или виртуальный сервер, на котором выполняются различные службы. Добавлять узлы в систему TROK вы можете на странице «Узлы». Для добавленных узлов реализован мониторинг состояния в реальном времени, а также фильтрация для получения более актуальных данных.

Storage Pool (хранилище или пул хранения) – логическая группа, которая объединяет физические устройства хранения (например, группы томов LVM или пулы LVM_THIN) в логические пулы для динамического развертывания. Модуль «Пулы хранения» позволяет работать с пулами хранения: объединять физические

или виртуальные носители в одну логическую единицу с заданными характеристиками.

Resource Group (группа ресурсов) – это логическое объединение для управления общими параметрами ресурсов DRBD. Она позволяет централизованно задавать свойства для множества ресурсов, упрощая администрирование кластера. Для управления данной сущностью используется вкладка «Группы ресурсов».

Resource Definition (шаблон ресурса) – набор параметров, описывающих свойства ресурса хранения, которые могут быть использованы для создания нового экземпляра ресурса. Вкладка «Шаблоны ресурсов» предназначена для изменения созданных режимов репликации для ранее созданных групп ресурсов.

Volume Definition (шаблон тома) – набор параметров, определяющих характеристики тома, который используется для создания новых томов на основе готовых конфигураций. Применение шаблонов позволяет стандартизировать выделение дискового пространства и ускорить развертывание. Управление данной сущностью осуществляется на вкладке «Шаблоны томов».

Volume (том) – физическое блочное устройство (например, /dev/drbd1000), поддерживаемое хранилищами. Является минимальной единицей, доступной для использования и управления. На вкладке «Список томов» вы можете увидеть все тома, созданные в системе TROK.

11. ПРОЦЕДУРА АВТОРИЗАЦИИ ДЛЯ SDS UI

После ввода URL-адреса, предназначенного для доступа к интерфейсу в формате `http://hostname`, в командной строке браузера открывается стартовая страница. Для входа в панель администратора введите логин учетной записи в поле «Имя пользователя» и нажмите «Войти».

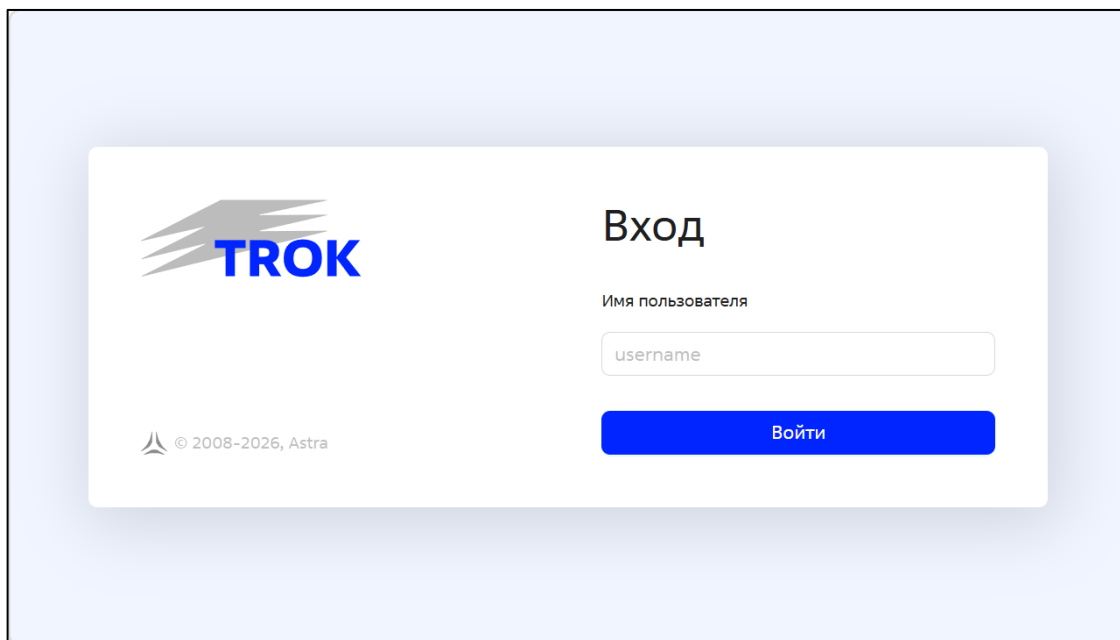


Рисунок 8 – Стартовая страница

В открывшемся окне введите пароль и нажмите «Войти» для завершения процесса авторизации.

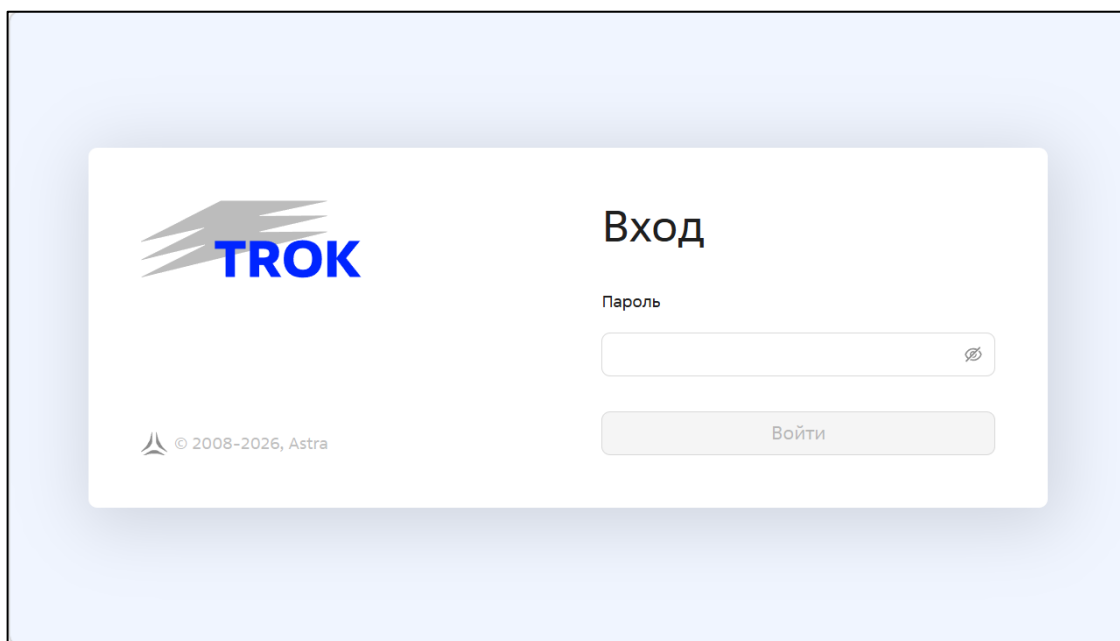


Рисунок 9 – Завершение процесса авторизации

12. ПАНЕЛЬ УПРАВЛЕНИЯ SDS UI (ДАШБОРД)

При первой авторизации по базовому URL, например, `http://<имя_хоста>/`, вы по умолчанию попадаете на главную панель управления (дашборд). В случае же, если переход осуществляется непосредственно по адресу конкретного раздела, например, `http://<имя_хоста>/storage-pools`, то после успешной авторизации вы будете перенаправлены именно в этот раздел – пулов хранения.

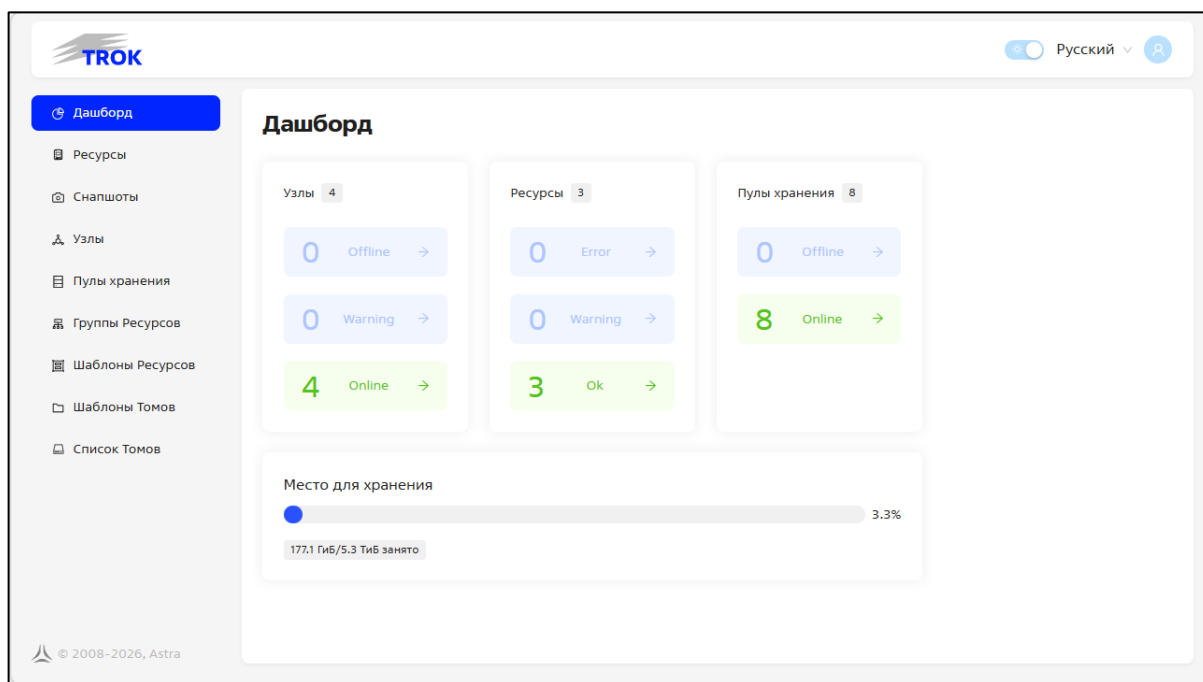


Рисунок 10 – Дашборд

Дашборд предоставляет пользователю набор статистической информации о состоянии всех компонентов системы, а также визуализирует объем доступного пространства для хранения данных.

В дополнение к этому, интерфейс включает визуальные индикаторы, которые обозначают количество узлов, ресурсов и пулов хранения, требующих внимания администратора.

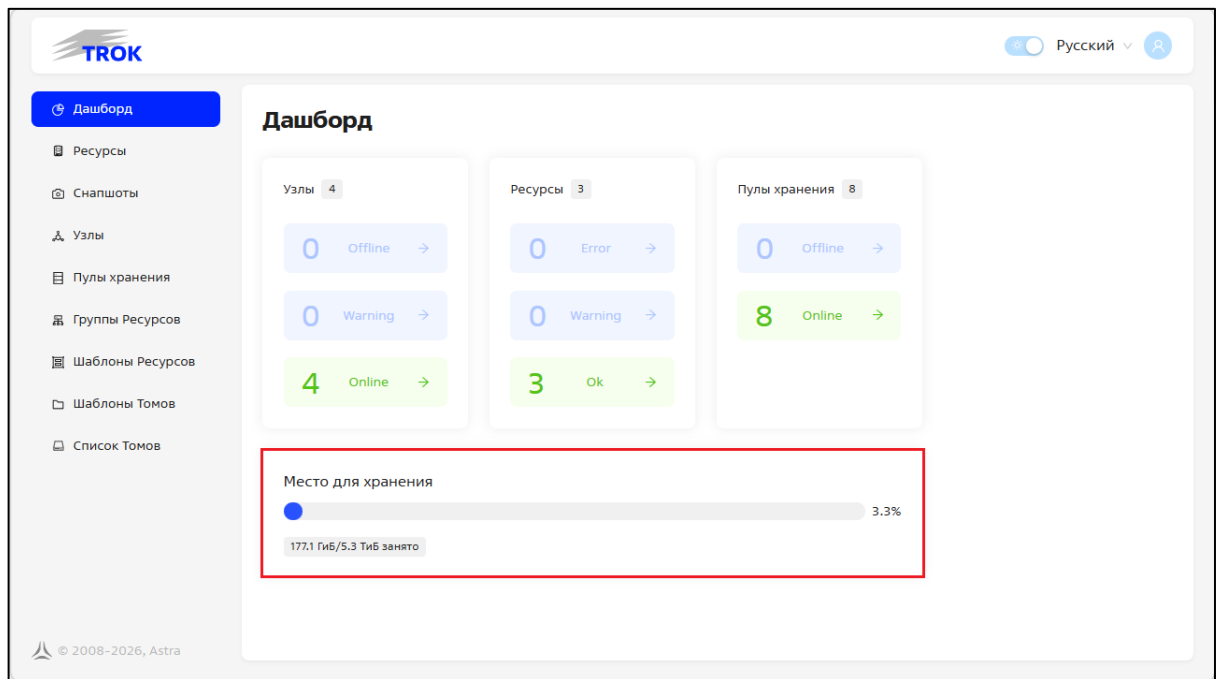


Рисунок 11 – Визуализация объема доступного пространства для хранения данных
Интерфейс программного обеспечения позволяет переключаться на английский язык в случае необходимости.

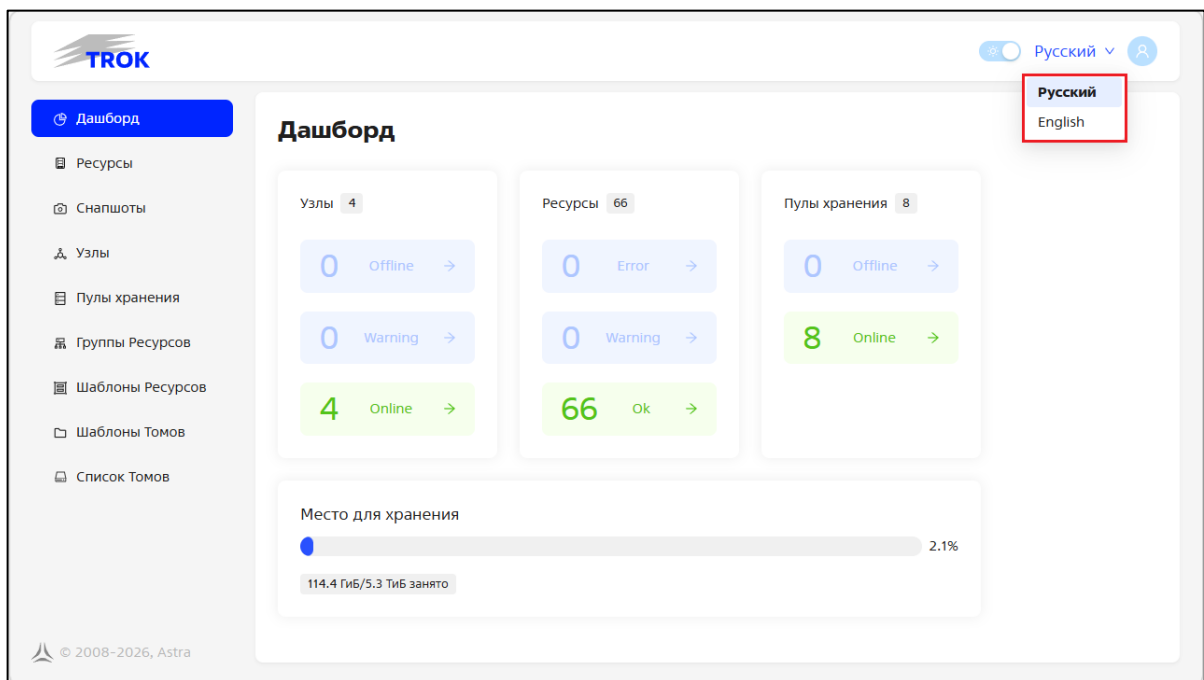


Рисунок 12 – Выпадающее меню для выбора языка интерфейса
Также предусмотрена возможность переключения на темную цветовую тему.

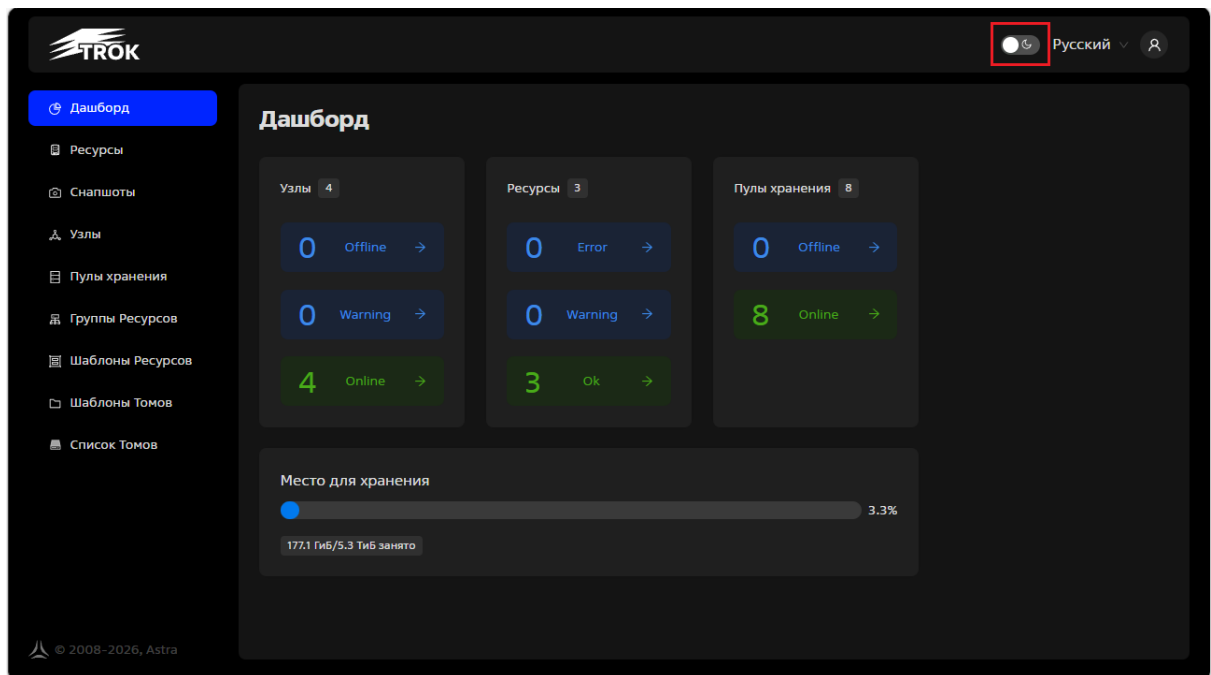


Рисунок 13 – Переключатель для цветовой темы

13. УЗЛЫ КЛАСТЕРА

Вкладка «Узлы» предназначена для управления и мониторинга вычислительных узлов в системе хранения данных.

При создании нового узла программное обеспечение требует ввода следующих данных:

Рисунок 14 – Добавление нового узла

– Выпадающий список «Тип узла» указывает на настройку узла типа worker – для хранения данных. Контроллер настраивается автоматически.

– Поле «Имя узла» предназначено для ввода уникального идентификатора (например, node-01, ctrl-main). **Важно!** Имя узла в системе TROK должно совпадать с именем хоста сервера. Совпадающие имя узла и имя хоста обеспечивают точное соответствие между управлением и фактическим узлом, стабильность операций и корректную работу репликаций и CSI-драйвера.

– Для того, чтобы узнать имя хоста, введите в командной строке операционной системы команду:

```
uname -n
```

или, для более развернутого вывода:

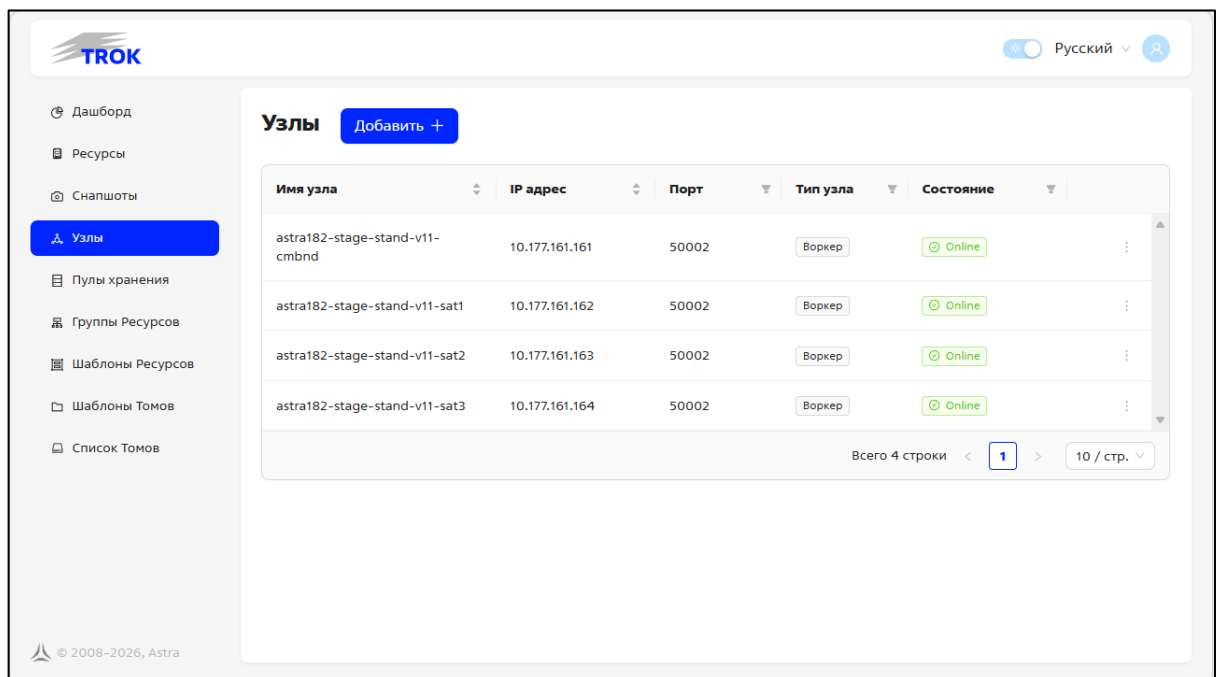
```
hostnamectl
```

Более подробная информация о валидации всех полей указана в пункте 22.

- **Имя узла:** Уникальное название узла (например, worker-01).
- **Поле IP-адрес** предназначено для ввода сетевого адреса узла для последующей связи его внутри кластера.
- **Поле «Порт»** для узла worker имеет значение по умолчанию: 50002.

Созданные узлы отображаются в таблице с заголовками:

- **Имя узла:** Такое же, как имя хоста.
- **IP адрес:** Адрес, по которому узел доступен в сети.
- **Порт:** Номер порта, используемого для связи с узлом.
- **Тип узла:** Классификация узла (Комбинированный, Воркер, Контроллер).
- **Состояние:** Информация о работоспособности узла и его текущем состоянии.



Имя узла	IP адрес	Порт	Тип узла	Состояние
astra182-stage-stand-v11-cmbnd	10.177.161.161	50002	Воркер	Online
astra182-stage-stand-v11-sat1	10.177.161.162	50002	Воркер	Online
astra182-stage-stand-v11-sat2	10.177.161.163	50002	Воркер	Online
astra182-stage-stand-v11-sat3	10.177.161.164	50002	Воркер	Online

Рисунок 15 – Список узлов

Для уже созданных узлов доступны три способа взаимодействия:

- удалить;
- пометить как потерянный;

– подключить повторно – данный способ взаимодействия появляется только в том случае, если узел перешел в состояние Evicted. (Evicted (Исключён) – состояние, при котором узел теряет связь с кластером или не проходит контрольные проверки состояния. В этом состоянии узел исключается из числа активных участников кластера и не выполняет задачи. Таймаут состояния составляет по умолчанию 1500 секунд.)

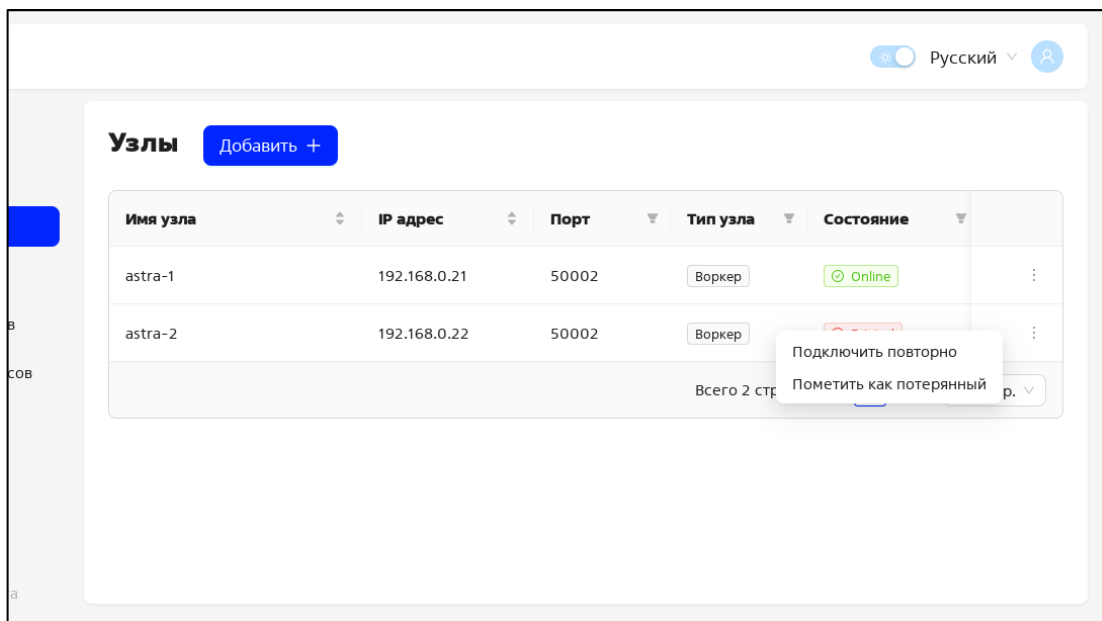


Рисунок 16 – Способы взаимодействия с узлом в состоянии Evicted

Для внесения изменений нажмите правой кнопкой мыши на строку с описанием узла и в выпадающем меню выберите нужное действие: «подключить повторно»

или «пометить как потерянный».

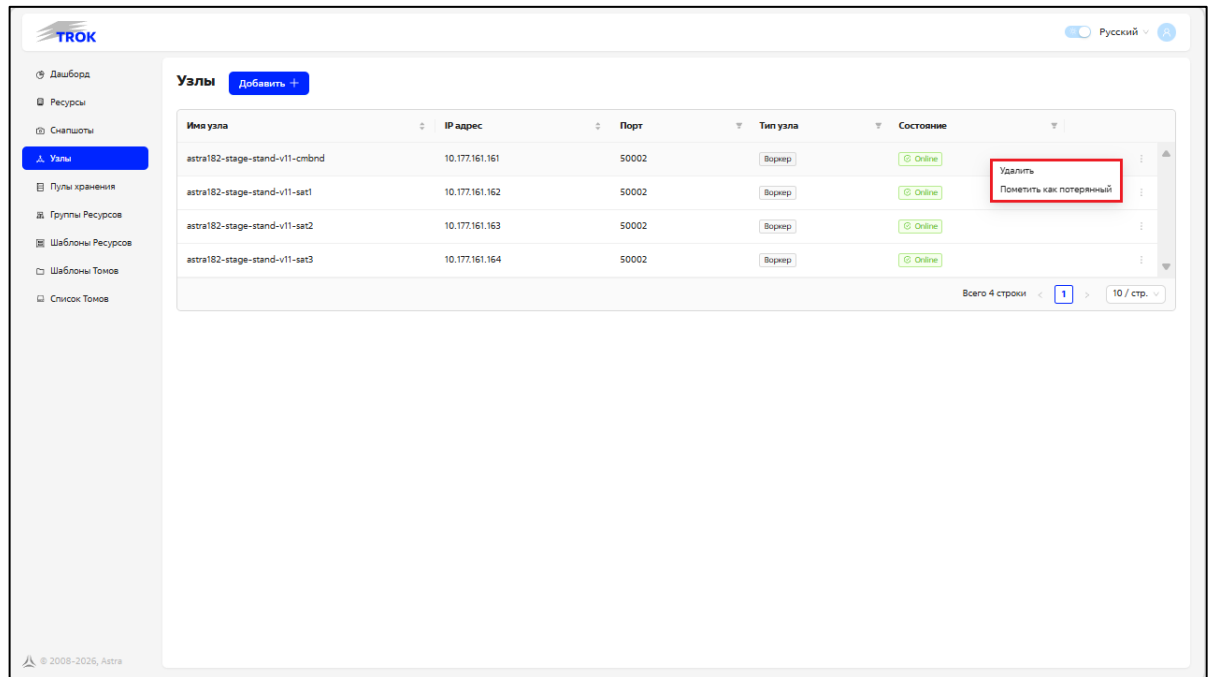


Рисунок 17 – Способы взаимодействия с созданным узлом

Также можно использовать иконки управления   или , которые появляются в правой части строки при наведении указателем мыши.

14. ПУЛЫ ХРАНЕНИЯ

Раздел «Пулы хранения» предназначен для управления и мониторинга логических групп дисков или томов — объединений физических и/или виртуальных носителей, на которых размещаются данные. На странице отображается общий список пулов, доступных узлов, их типы и текущие характеристики (ёмкость, свободное место, путь и метки).

Экземпляры одного и того же пула хранения могут быть размещены на нескольких узлах. Чтобы система корректно идентифицировала одинаковый пул на разных узлах, на каждом узле должен быть создан пул с одинаковым именем.

В основной части страницы приведена таблица со строками по каждому пулу. Колонки таблицы:

- **Имя пула** – уникальное текстовое имя пула;
- **Имя узла** – узел (node) где расположен пул;
- **Тип тома** – конфигурация тома (LVM, LVM_THIN и т. п.);
- **Путь** – Имя группы томов или группы томов/тонкого пула;
- **Теги** – метки, используемые для автоматического размещения ресурсов;
- **Ёмкость** – общая заявленная ёмкость пула;
- **Свободно** – объём доступного пространства;
- **Действия** – кнопки/меню для операций над пулом (копировать, удалить и т. д.).

Для быстрого доступа к операциям используйте иконки справа от строки или вызовите контекстное меню (кликните правой кнопкой по строке с именем пула хранения).

14.1. Создание нового пула хранения

Чтобы добавить пул, нажмите кнопку «Добавить» («Add»), расположенную в верхней части страницы.

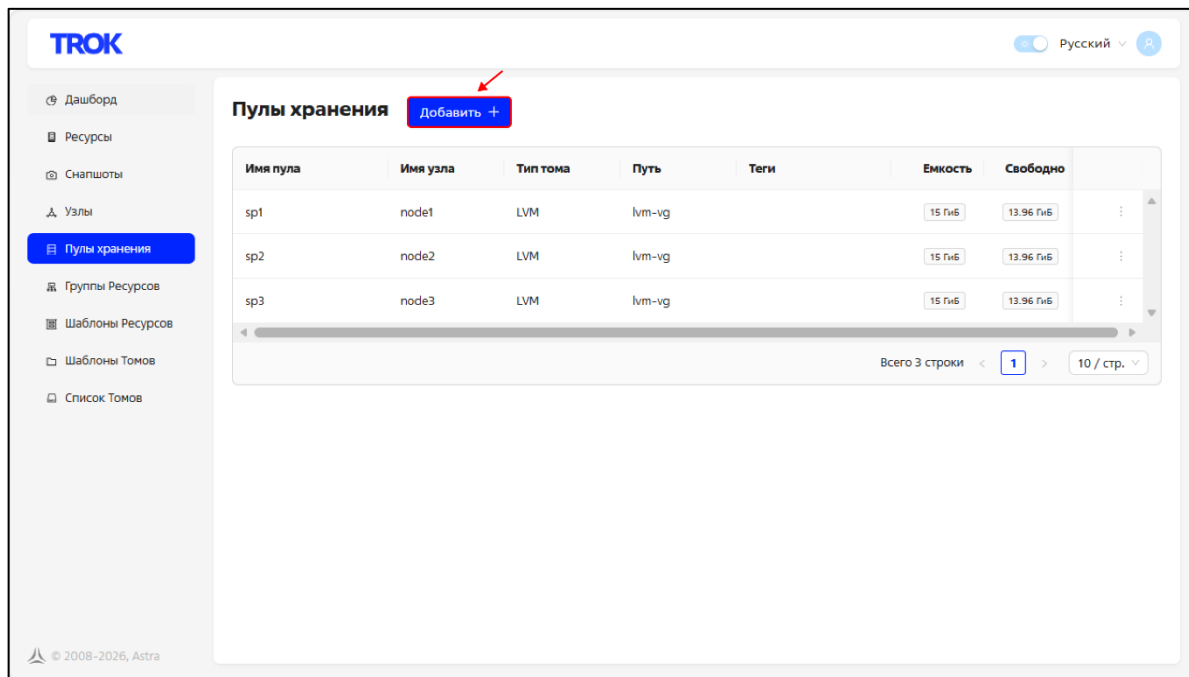


Рисунок 18 – Активация процедуры добавления нового пула

Откроется модальное окно с полями для ввода параметров нового пула.

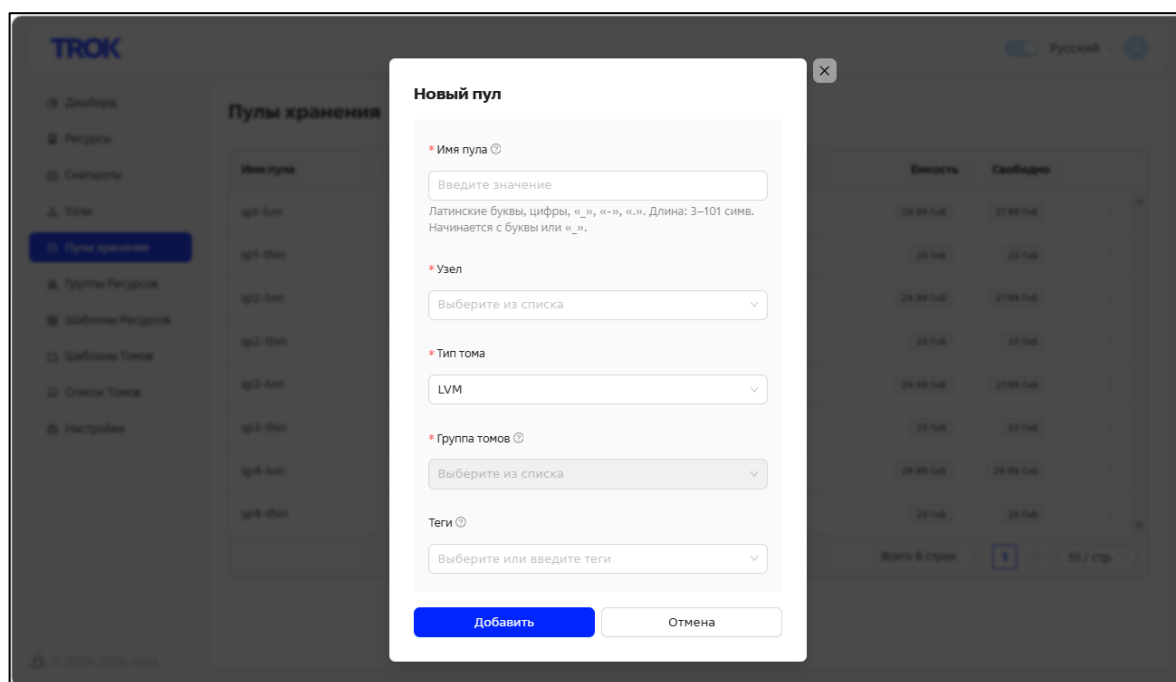


Рисунок 19 – Блок добавления нового пула

В форме создания предусмотрены следующие поля:

– Поле «Имя пула» («Pool Name») – уникальный идентификатор пула, обязателен для заполнения;

- **Выпадающий список «Узел» («Node»)** позволяет выбрать узел, на котором будет расположен пул. Это может быть физическая или виртуальная машина, предоставляющая ресурсы для хранения;
- **Выпадающий список «Группа томов» («Volume Group / VG»)** – адрес физического диска или логического тома, который будет включён в пул;
- **Выпадающий список «Тип тома»** – вариант организации тома.

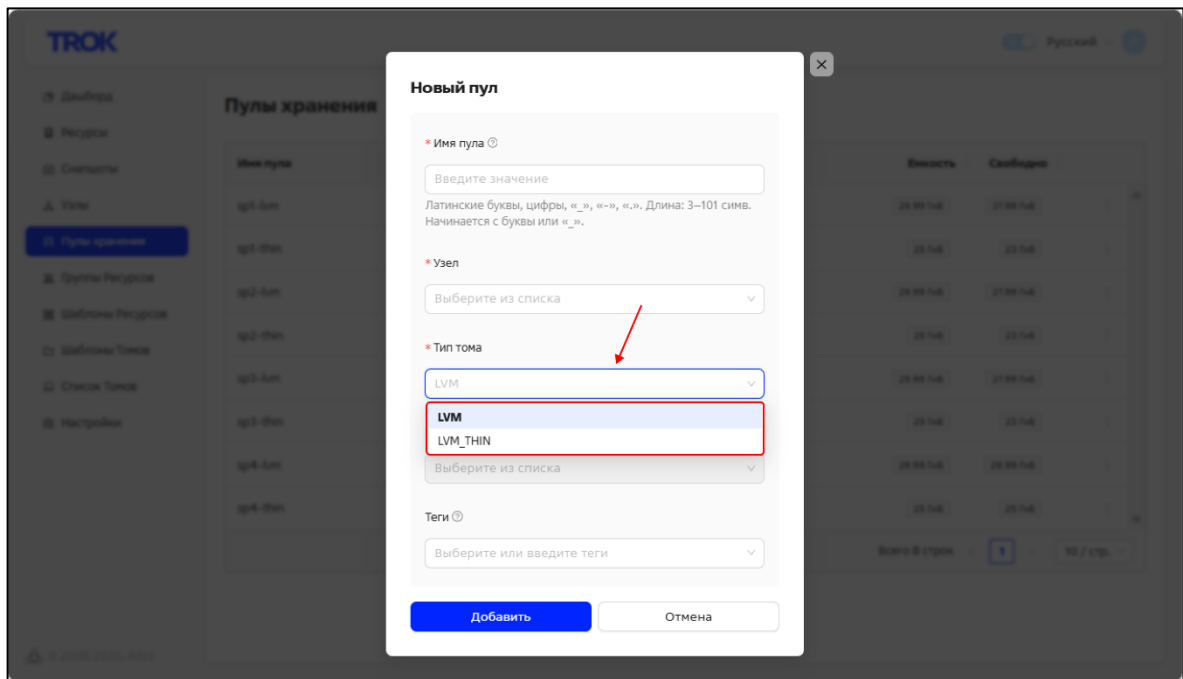


Рисунок 20 – Выпадающий список с выбором типа тома

Варианты:

- **«LVM» (Logical Volume Manager)**. Классические логические тома с фиксированным выделением пространства. Физические диски объединяются в группы томов (VG), из которых далее создаются логические тома (LV). Тома данного типа подходят для статичных нагрузок, где размер данных предсказуем (например, СУБД).
- **«LVM_THIN»**. Тонкие (thin) логические тома с динамическим выделением места. При таком способе организации дискового пространства память выделяется по мере записи данных. Поддерживает снапшоты и клонирование. Чаще всего используется для создания виртуальных машин, где размер диска может быть

заявлен, но реально используется меньше. При выборе «LVM_THIN» появится дополнительное выпадающее поле, в котором нужно указать группу томов (VG), в рамках которой будет создан thin-pool (см. главу 5).

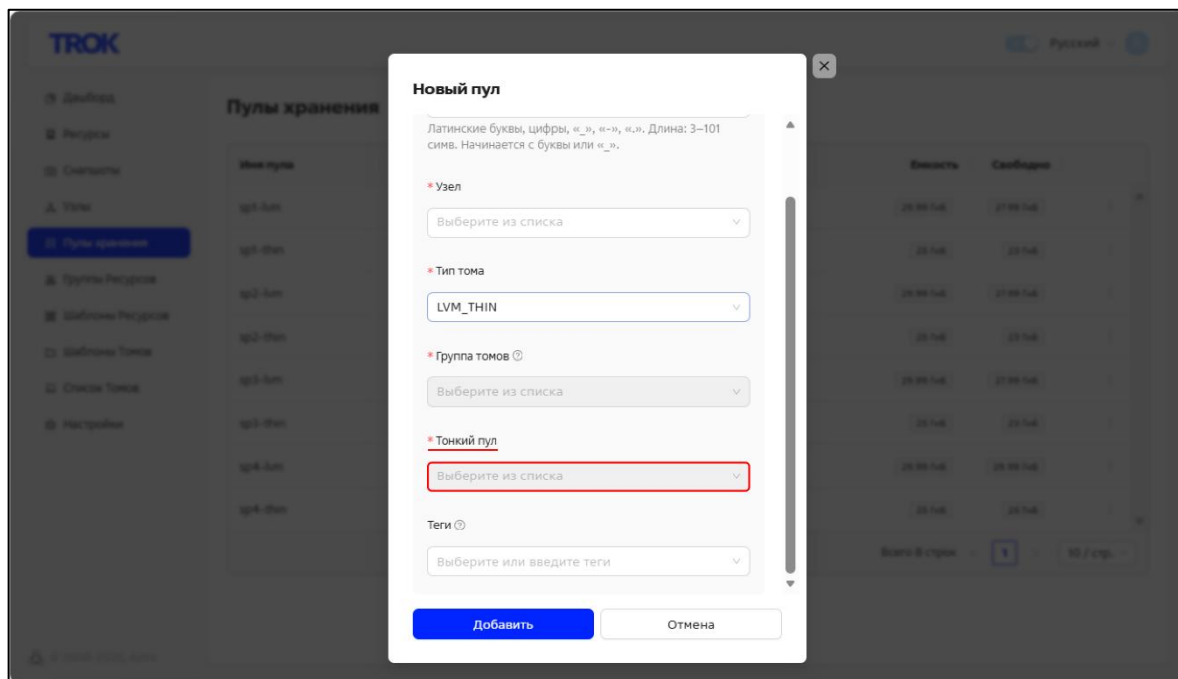


Рисунок 21 – Выпадающий список с выбором тонкого пула

– Поле «Теги» («Tags») — метки, которые позже используются системой для автоматического направления создаваемых ресурсов в нужные пулы/узлы.

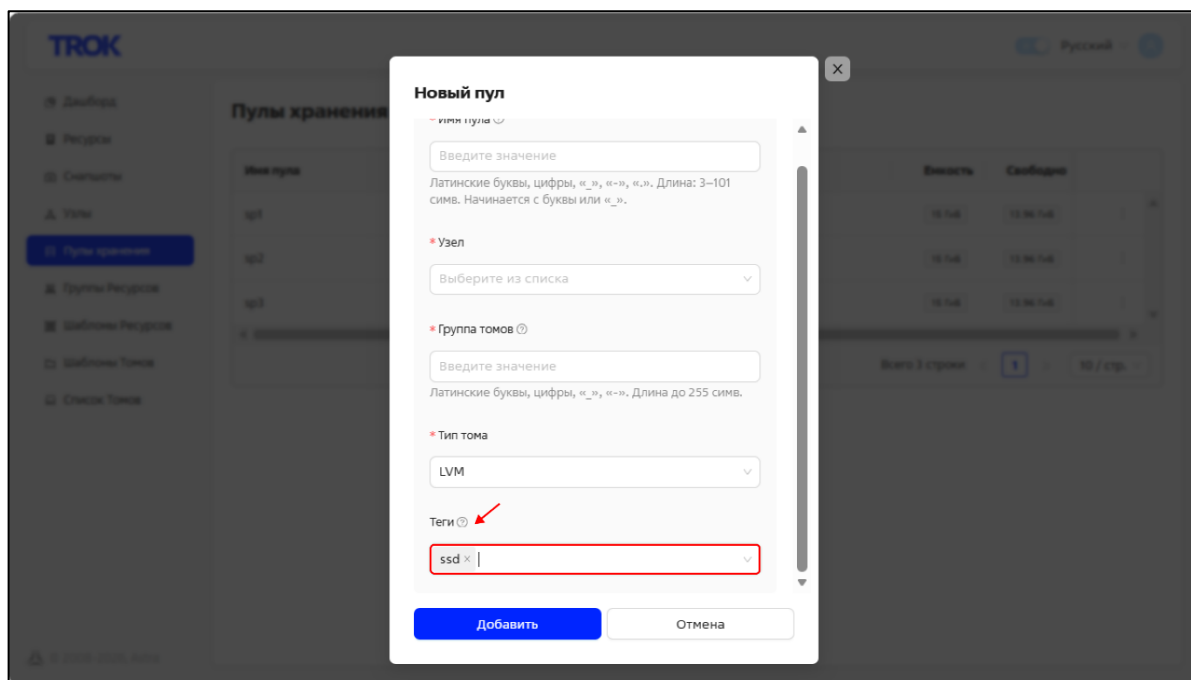


Рисунок 22 – Поле для ввода (или выбора ранее созданных) тегов

После заполнения нажмите «Добавить» («Add»). Для отмены используйте «Отмена» («Cancel»). Созданный пул появится в таблице и сразу станет доступен для использования.

14.2. Копирование пула (создание пула с теми же параметрами)

Если нужно создать пул с параметрами, идентичными существующему, используйте функцию «Копировать». Существует два способа вызова копирования:

Щёлкните правой кнопкой мыши по строке с нужным пулом и в появившемся контекстном меню выберите пункт «Копировать» («Сору»).

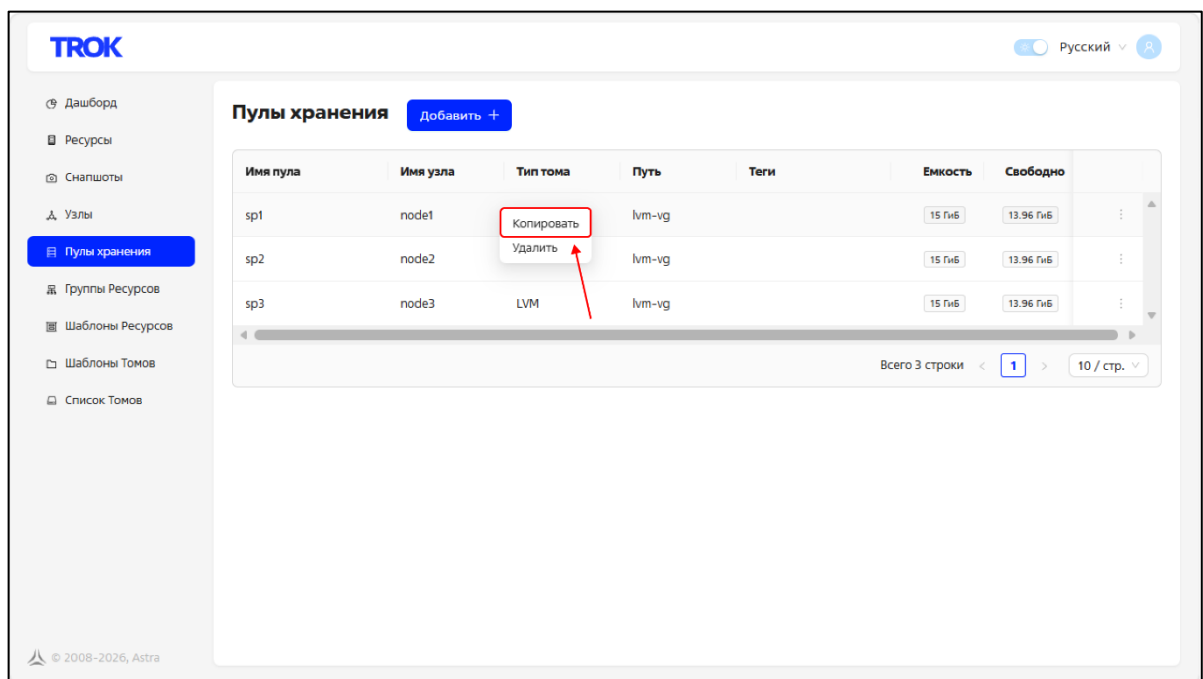


Рисунок 23 – Контекстное меню строки пула с пунктом «Копировать»

Либо наведите курсор на троеточие в правой части строки – появятся две иконки; нажмите иконку «Копировать» (в виде двух листов).

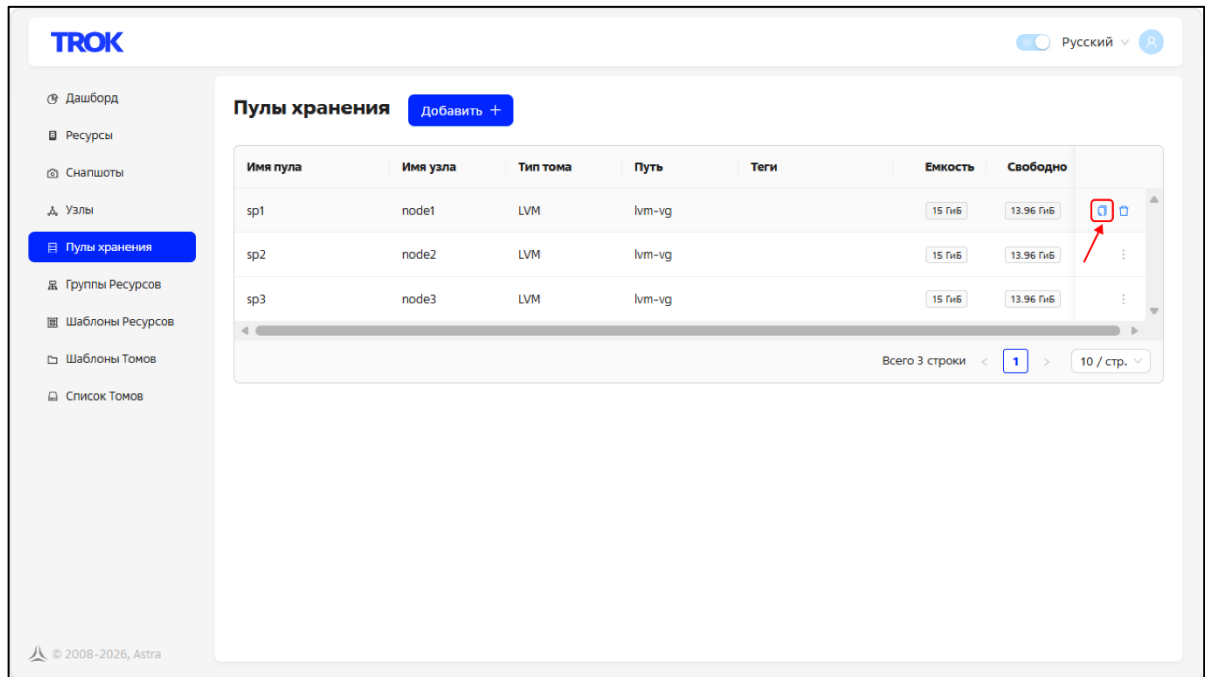


Рисунок 24 – Иконка для вызова функции копирования

В открывшейся форме будут предзаполнены параметры выбранного пула. Внесите необходимые изменения (например, имя или узел) и нажмите «Добавить» («Add»).

14.3. Удаление пула хранения

Для удаления пула в таблице выберите соответствующее действие (иконка корзины или пункт «Удалить» в меню).

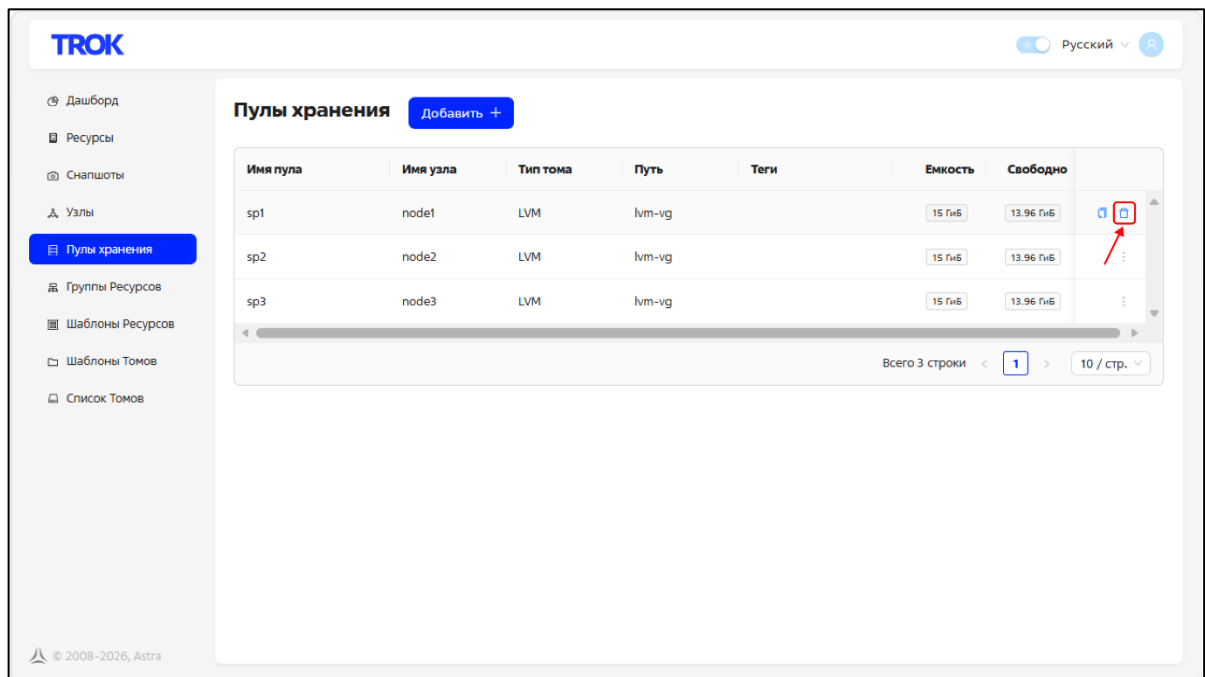


Рисунок 25 – Иконка для вызова функции удаления

После выбора откроется диалог подтверждения удаления с предупреждением о необратимости операции.

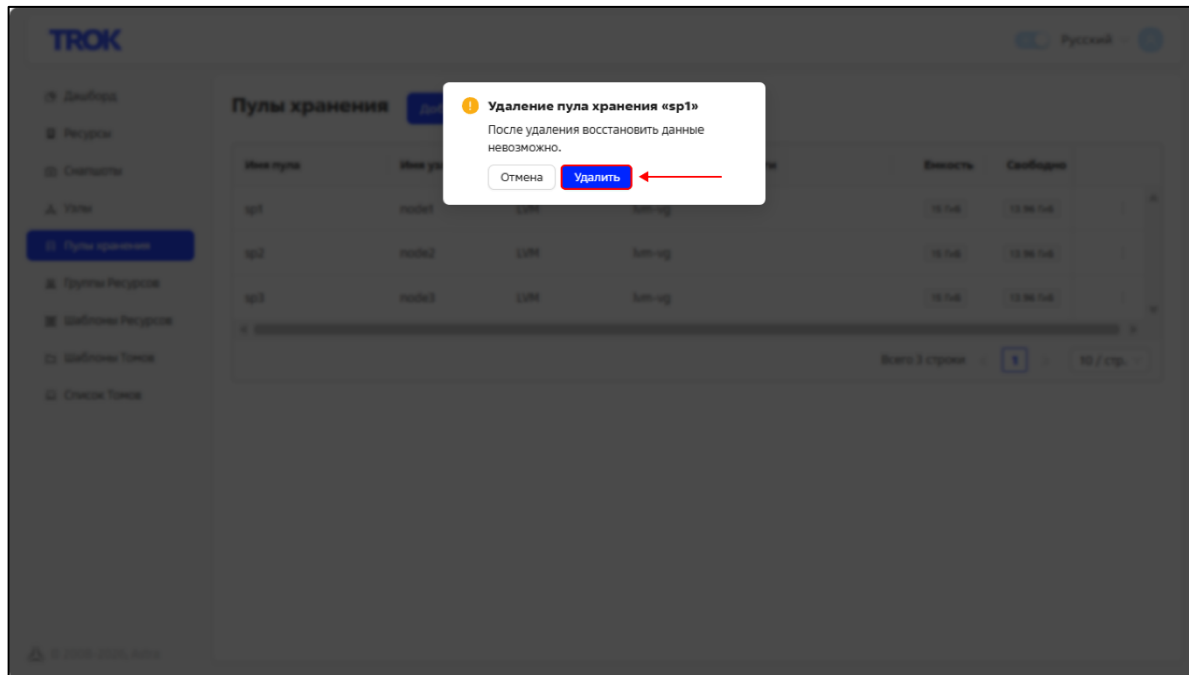


Рисунок 26 – Подтверждение удаления пула

В диалоге проверьте имя удаляемого пула и подтвердите действие кнопкой «Удалить» («Delete»). После подтверждения пул будет удалён, и доступ к данным, размещённым на нём, будет прекращён. Таблица обновится автоматически.

Примечания по удалению:

- Операция удаления необратима.
- Если на пуле присутствуют активные тома или ресурсы, система может запретить удаление – предварительно удалите или перенесите ресурсы.
- Перед удалением убедитесь, что права владения томами/ресурсами переданы или более не требуются.

14.4. Теги и автоматическое размещение ресурсов

Теги (метки) используются для автоматизации размещения ресурсов: при создании томов, шаблонов или других сущностей система может направлять их в определённые пулы/узлы на основе заданных меток. Это удобно для разделения

нагрузки, гарантирования расположения данных в нужной подсистеме хранения и для соблюдения политик размещения (например, «все бэкапы – в пулах с тегом backup»).

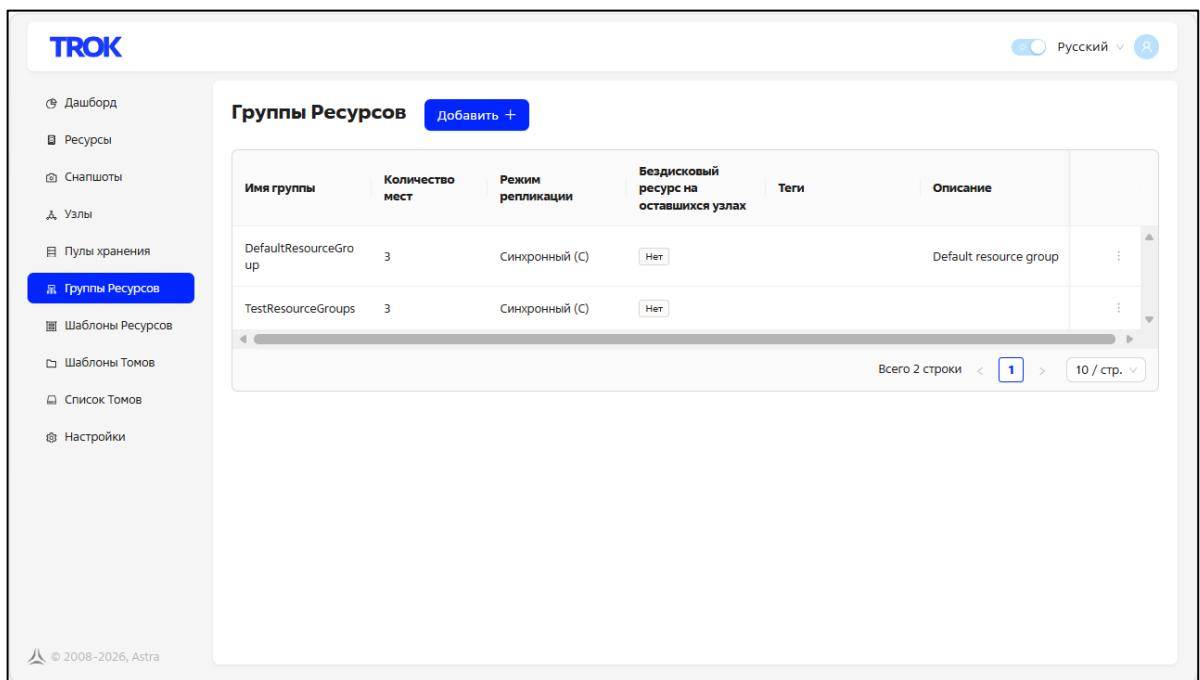
Рекомендации:

- Присваивайте осмысленные теги при создании пула (например, SSD, HDD, DATABASE, BACKUP).
- Документируйте используемые теги в вашей организации, чтобы избежать непреднамеренного размещения ресурсов.

15. ГРУППЫ РЕСУРСОВ

Раздел «Группы ресурсов» предназначен для создания, управления и мониторинга логических сущностей, объединяющих тома с одинаковыми параметрами репликации, размещения и доступа. На странице отображается таблица существующих групп, кнопка создания и быстрые операции над каждой строкой.

В основной части страницы приведена таблица групп ресурсов. Для каждой группы отображаются ключевые характеристики, что позволяет оперативно оценить конфигурацию и состояние.



Имя группы	Количество мест	Режим репликации	Бездисковый ресурс на оставшихся узлах	Теги	Описание
DefaultResourceGroup	3	Синхронный (C)	Нет		Default resource group
TestResourceGroups	3	Синхронный (C)	Нет		

Всего 2 строки < 1 > 10 / стр.

Рисунок 27 – Список групп ресурсов: общий вид таблицы

Колонки таблицы:

- **Имя группы** – уникальный идентификатор группы ресурсов;
- **Количество мест** – количество реплик (копий) для ресурсов из данной группы;
- **Режим репликации** – выбранный тип синхронизации между репликами;
- **Бездисковый ресурс на оставшихся узлах** – индикатор включённого режима (если применимо);
- **Теги** – метки, используемые для направленного размещения ресурсов;

- **Описание** – краткое пояснение назначения группы;
- **Действия** – иконки/меню для операции удалить.

15.1. Создание новой группы ресурсов

Чтобы добавить новую группу, нажмите кнопку «Добавить» («Add»), расположенную в верхней части страницы.

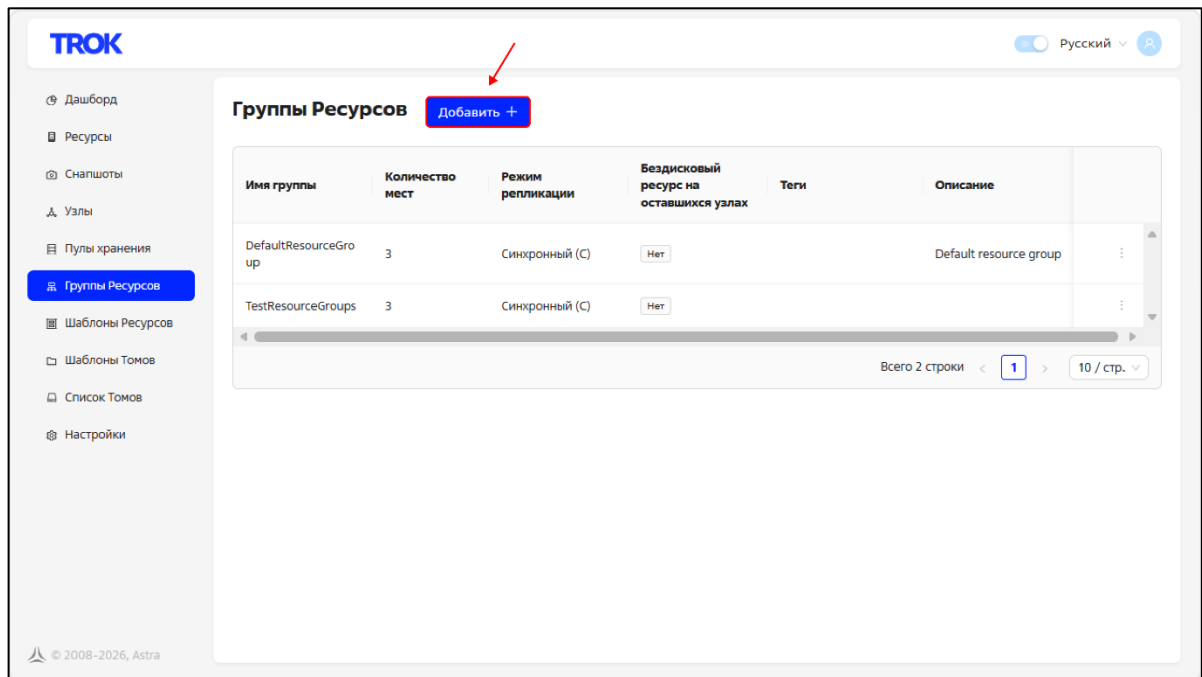


Рисунок 28 – Активация процедуры добавления новой группы ресурсов

Откроется модальное окно с полями для ввода параметров группы ресурсов.

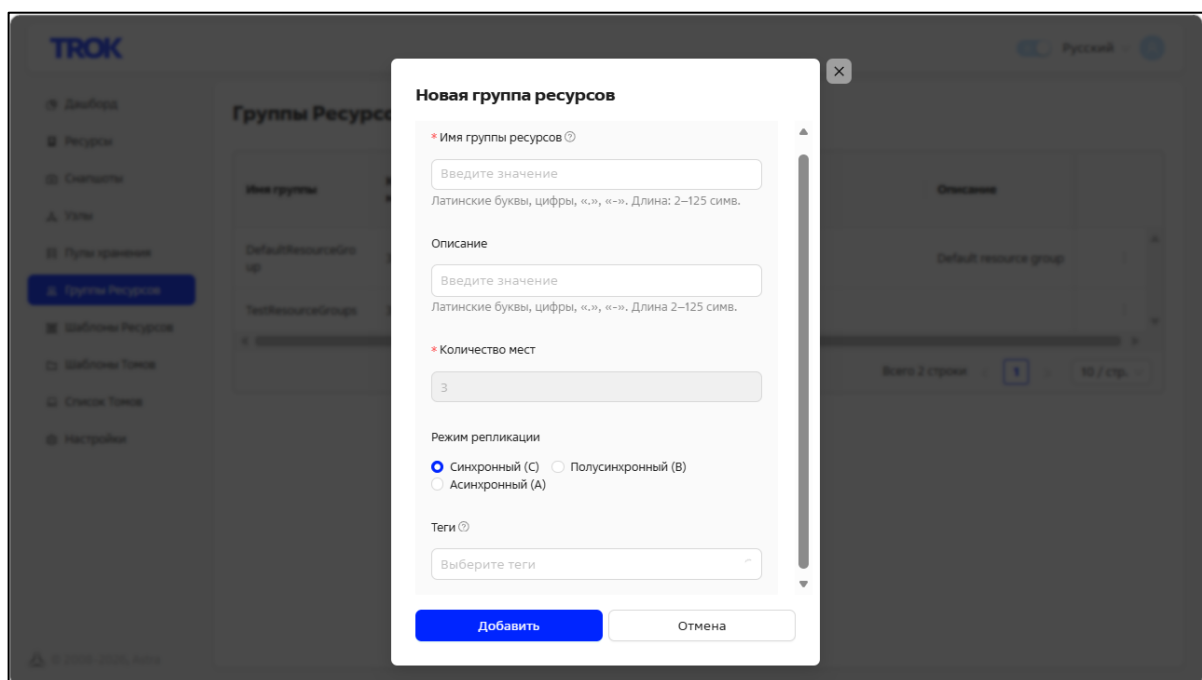


Рисунок 29 – Блок добавления новой группы ресурсов

В форме создания предусмотрены следующие поля:

- **Поле «Имя группы ресурсов»** предназначено для ввода уникального идентификатора, который будет использоваться для идентификации группы в системе.

- Краткое текстовое **поле «Описание»** предполагает ввод поясняющих данных для группы (например, «Высоконагруженные БД») и помогает идентифицировать назначение группы среди других.

- **Поле «Количество мест»** задается автоматически и означает количество копий данного ресурса.

- **Радиокнопка «Режим репликации»** определяет базовый алгоритм записи данных:

- **Синхронный (С)** – данные записываются на все реплики одновременно. Гарантирует абсолютную целостность данных.

- **Полусинхронный (В)** – запись подтверждается после фиксации хотя бы на одной реплике. Баланс между надёжностью и производительностью.

- **Асинхронный (А)** – запись подтверждается немедленно, репликация происходит в фоне. Обеспечивает высокую скорость, но допускает задержки актуализации данных.

- **Поле «Теги» («Tags»)** — метки, которые позже используются системой для автоматического направления создаваемых ресурсов в нужные пулы/узлы, а также для сопоставления группы и пулов хранения.

Теги группы будут применимы только при условии, что такие же теги уже существуют в пулах хранения. Иными словами, теги группы должны совпадать с тегами соответствующих пулов – только тогда автоматическое размещение сработает корректно. Если вы пытаетесь сохранить теги, которых нет в пулах хранения, система не примет изменения и вернёт ошибку.

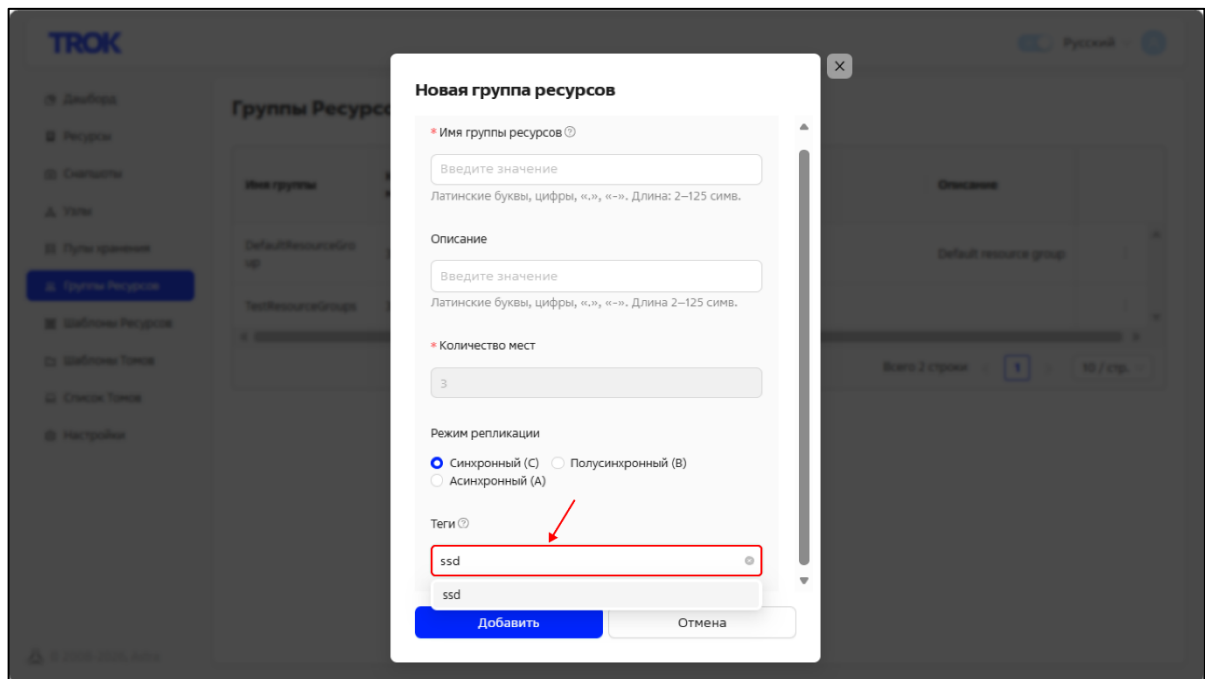


Рисунок 30 – Поле для ввода (или выбора ранее созданных) тегов

15.2. Удаление группы ресурсов

Удаление группы выполняется из списка групп. Доступные способы вызова операции удаления:

Щёлкните правой кнопкой мыши по строке с нужной группой и в появившемся контекстном меню выберите «Удалить».

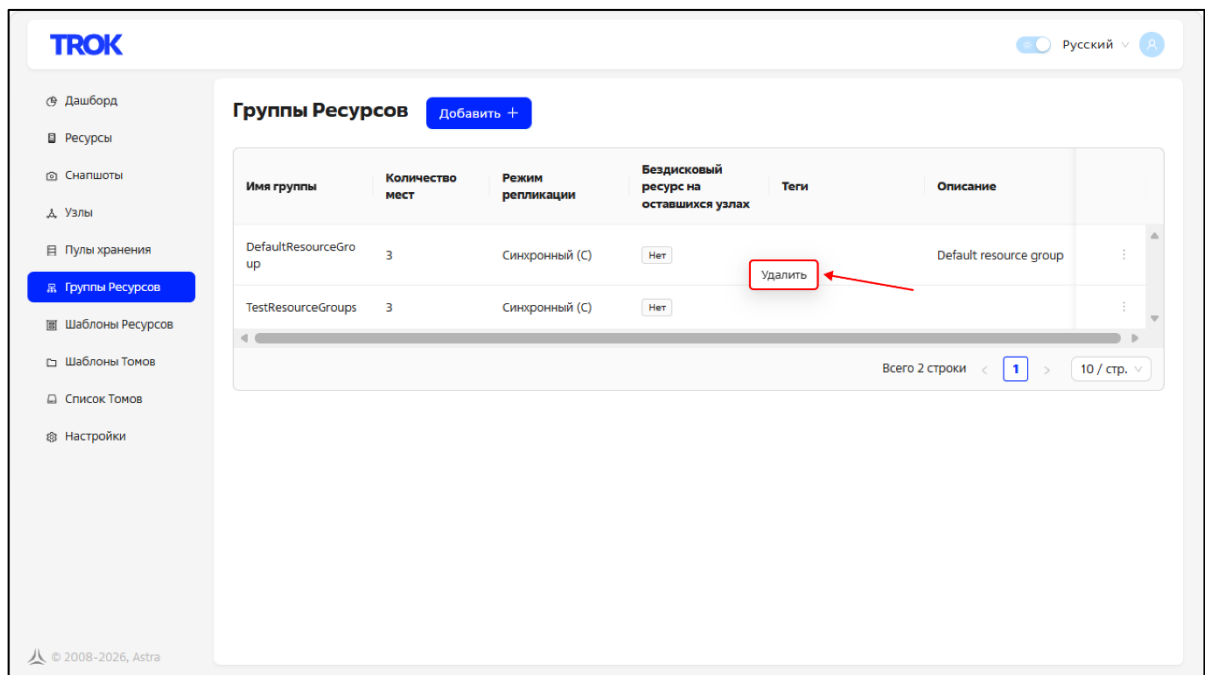


Рисунок 31 – Контекстное меню строки группы ресурсов с пунктом «Удалить»

Либо наведите курсор на трёхточечное меню в правой части строки: появится иконка быстрых действий – нажмите иконку «Удалить» (корзина).

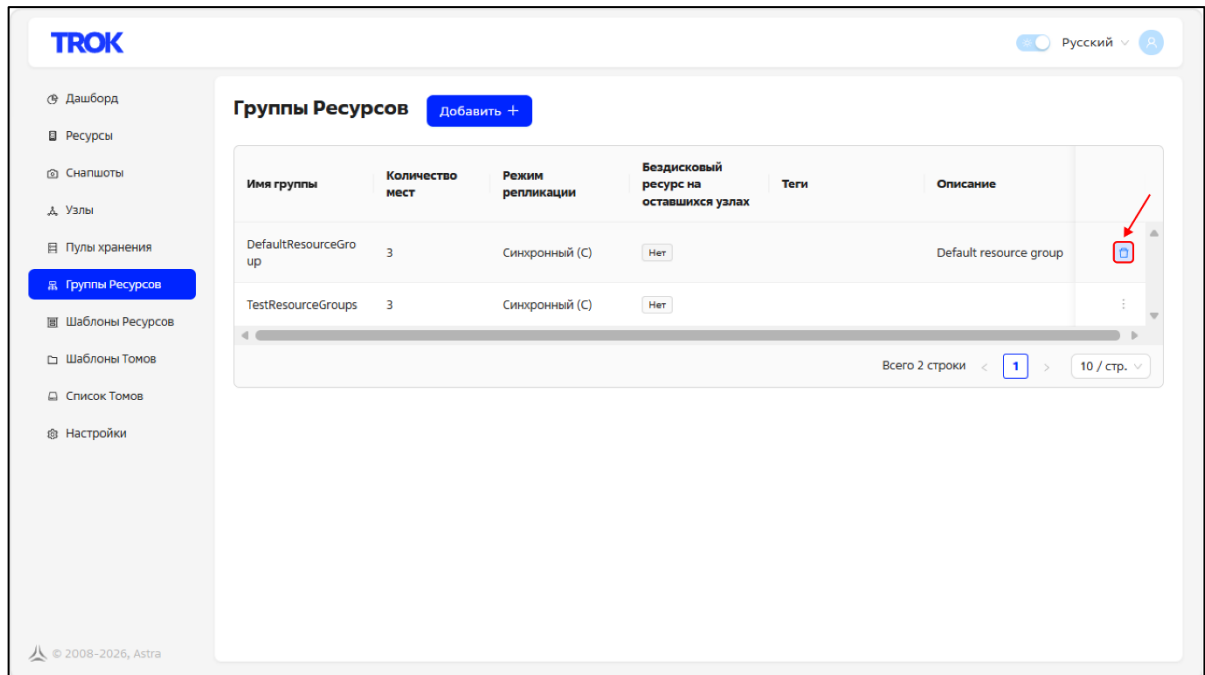


Рисунок 32 – Иконка для вызова функции удаления

После выбора команды откроется модальное окно подтверждения удаления с предупреждением об утрате данных и необратимости операции. Проверьте имя группы и подтвердите нажатием «Удалить» («Delete») или отмените действие.

Операция необратима – убедитесь, что все ресурсы/тома удалены или перенесены.

15.3. Приоритет размещения и тегирование

При автоматическом размещении ресурсов система в первую очередь выполняет фильтрацию доступных пулов по тегам, отбирая только те, чьи теги полностью соответствуют заданным требованиям ресурса, после чего из отобранных кандидатов исключаются пулы с недостаточным свободным пространством для размещения запрашиваемой ёмкости, и на финальном этапе из оставшихся пулов система выбирает тот, в котором осталось наименьшее количество свободного места, что позволяет максимально заполнить существующие пулы перед созданием новых и минимизировать фрагментацию.

Если найдены пул/узлы, удовлетворяющие по тегам и объёму, система размещает ресурс на одном из них.

При отсутствии пулов с совпадающими тегами система может выполнить размещение в любом пуле с достаточным местом. Для корректной работы автоматического размещения важно поддерживать актуальность тегов и отслеживать свободную ёмкость пулов – так обеспечивается предсказуемость и безопасность размещения данных.

16. ШАБЛОНЫ РЕСУРСОВ

Раздел «Шаблоны ресурсов» предназначен для управления шаблонами, на основе которых автоматически создаются экземпляры ресурсов с заранее заданными параметрами размещения, репликации и сетевой привязки. В верхней части страницы расположена кнопка «Добавить» («Add»), по нажатию которой открывается форма создания нового шаблона.

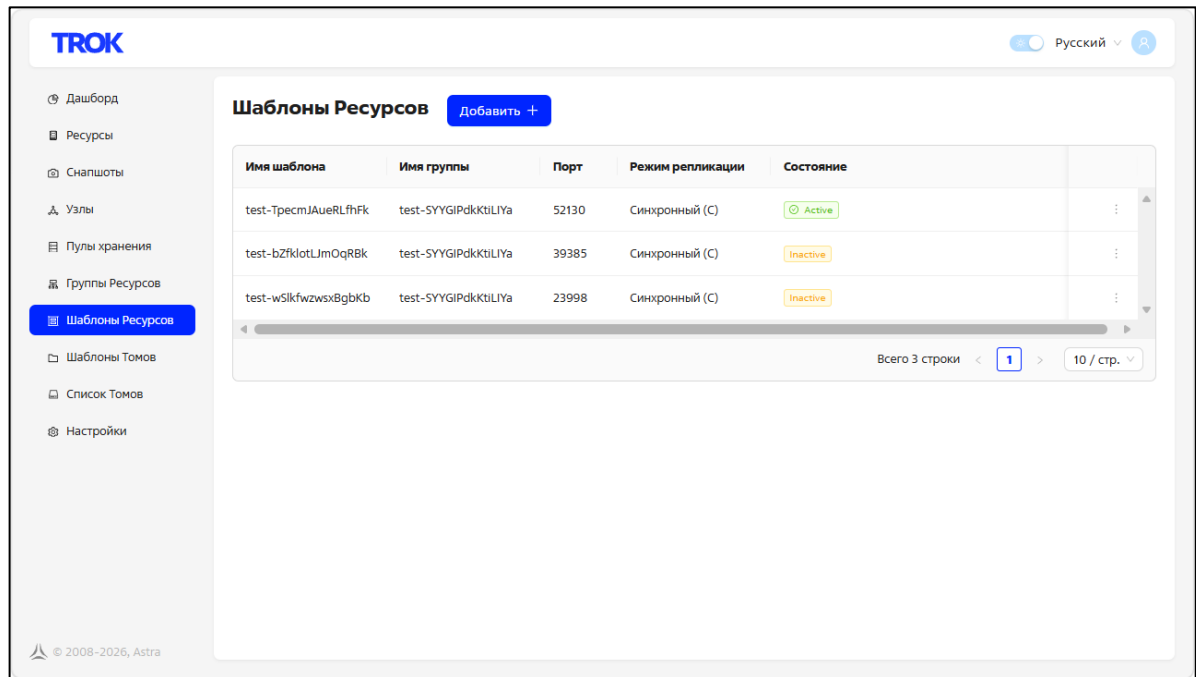


Рисунок 33 – Список шаблонов ресурсов: общий вид таблицы

Основная часть страницы содержит таблицу существующих шаблонов с ключевыми характеристиками и быстрыми операциями для каждой строки:

- **Имя шаблона** – уникальный идентификатор шаблона ресурса.
- **Имя группы** – уникальный идентификатор связанной группы ресурсов, для которой предназначен шаблон.
- **Порт** – идентификатор сетевого интерфейса узла, через который осуществляется размещение и доступ к ресурсам группы (используется при создании шаблона).
- **Текущий статус шаблона** (например, активен, неактивен, ошибка), отражающий его готовность к созданию экземпляров.

16.1. Создание шаблона ресурсов

Для создания шаблона нажмите «Добавить» («Add»).

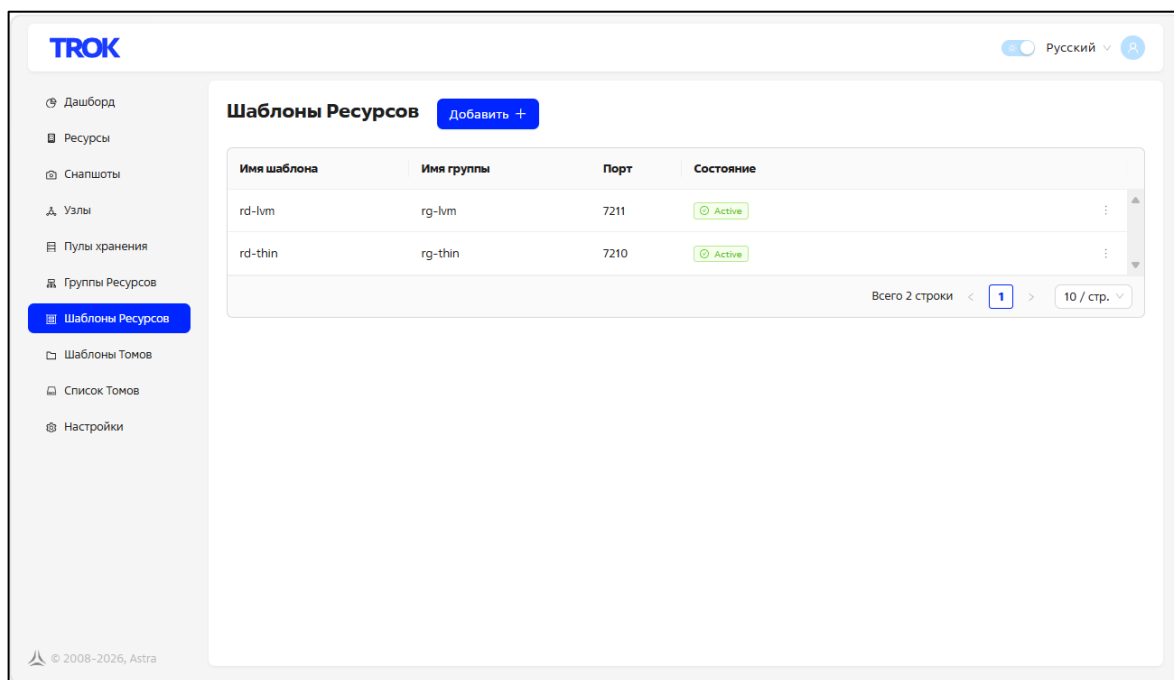


Рисунок 34 – Активация процедуры добавления нового шаблона ресурсов

Откроется модальное окно с полями для ввода параметров.

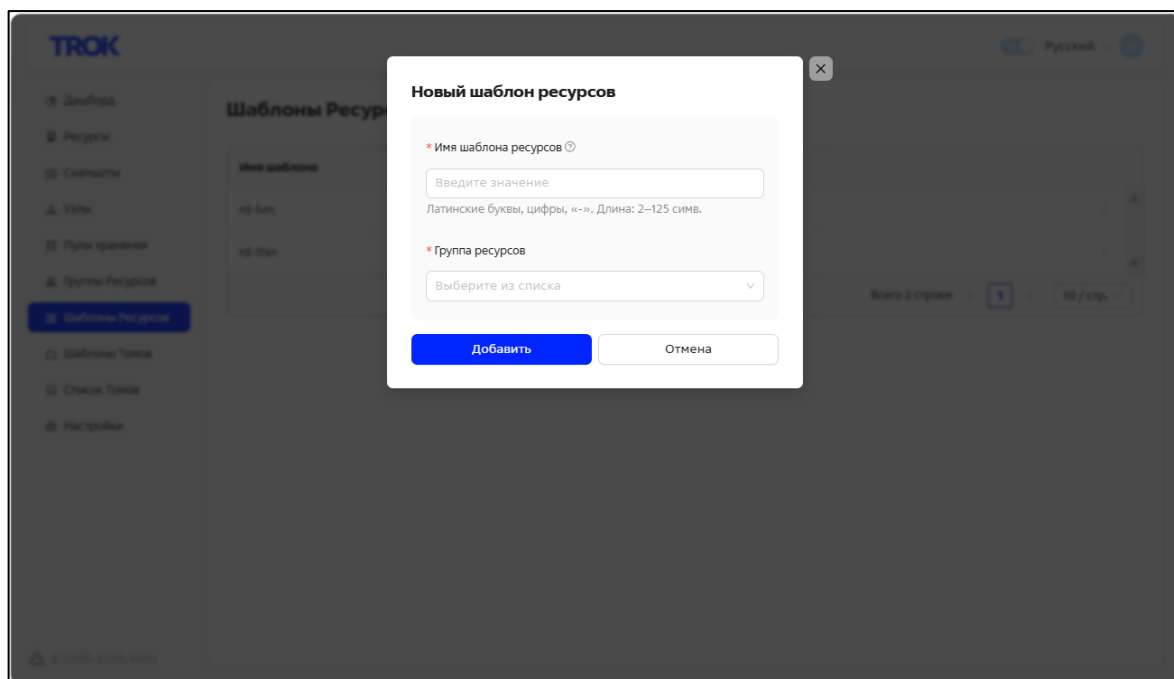


Рисунок 35 – Блок добавления нового шаблона ресурсов

Укажите корректные данные:

– **Поле «Имя шаблона ресурсов»** предназначено для ввода уникального идентификатора, шаблона.

– **Выпадающий список «Группа ресурсов»** предназначен для выбора ранее созданной группы ресурсов.

После заполнения нажмите «Добавить» («Add»). Для отмены используйте «Отмена» («Cancel»). Созданный шаблон ресурсов появится в таблице и сразу станет доступен для использования.

16.2. Состояния шаблона ресурсов

Шаблон может находиться в нескольких состояниях, отражающих его текущий жизненный цикл и операции управления. Некоторые статусы шаблонов ресурсов могут проходить слишком быстро для отображения в web-интерфейсе, поэтому их визуальное отслеживание невозможно. При необходимости мониторинга статуса используйте логи системы. Базовые состояния (статусы) шаблонов ресурсов:

Inactive (Неактивен) – состояние, в котором создается шаблон, и в которое шаблон переходит при удалении всех связанных с ним ресурсов. Также шаблон остается в этом состоянии до тех пор, пока количество ресурсов не достигнет заданного значения (требуемое количество экземпляров ресурсов).

Place count reached (Достигнуто необходимое количество ресурсов) – состояние, в которое переходит шаблон после создания количества ресурсов, равного требуемому значению.

Syncing metadata (Синхронизация метаданных) – состояние, при котором происходит обновление и согласование метаданных шаблона.

Active (Активен) – основное рабочее состояние шаблона. В данном состоянии метаданные синхронизированы, количество ресурсов соответствует требуемому значению, а все ресурсы находятся в состоянии UpToDate (актуальны).

Marked for autohealing (Помечен для автовосстановления) – состояние, в котором шаблон помечается системой мониторинга для выполнения процедуры автовосстановления. Шаблон находится в этом состоянии до истечения таймаута (заданного интервала ожидания) на автовосстановление.

Autohealing canceled (Автовосстановление отменено) — состояние, возникающее, если шаблон находился в статусе «Помечен для автовосстановления» (Marked for autohealing), и в этот момент администратор вручную удалил все ресурсы, относящиеся к шаблону.

После срабатывания таймаута на автовосстановление, вне зависимости от добавления новых ресурсов, шаблон автоматически перейдёт в состояние «Неактивен» (Inactive).

Autohealing (Автовосстановление) — состояние активного процесса автоматического восстановления шаблона. Переход в это состояние осуществляется из состояния Marked for autohealing (Помечен для автовосстановления) по завершении таймаута.

Snapshot rollback failed (Ошибка отката к снапшоту) — состояние, отражающее неудачный откат шаблона к выбранному снапшоту (моментальной копии). Является конечным состоянием, дальнейшие действия требуют дополнительного анализа.

Autoplacing (Автоплейсинг) — состояние, при котором происходит автоматическое размещение ресурсов: шаблон самостоятельно создаёт экземпляры ресурсов, пока их количество не сравняется с требуемым значением (place count), без ручного вмешательства.

Deleting (Удаление) — состояние, в котором выполняется удаление самого шаблона ресурсов. Операция может быть инициирована администратором вручную или запущена автоматически. В этом состоянии производится полная очистка шаблона и всех связанных с ним компонентов.

Дополнительные состояния:

Adding resource (Добавление ресурса) — состояние, при котором происходит добавление нового ресурса к шаблону (операция может быть выполнена вручную или инициирована автовосстановлением).

Deleting resource (Удаление ресурса) – состояние, связанное с удалением ресурса (операция может быть выполнена вручную или инициирована автовосстановлением).

Cloning (Клонирование) – процесс создания копии существующего ресурса.

Cloning from snapshot (Клонирование из снапшота) – процесс клонирования ресурса на основе существующего снапшота (моментальной копии).

Cloning volume (Клонирование тома) – процесс клонирования отдельного тома ресурса.

Creating snapshot (Создание снапшота) – состояние создания моментальной копии данных.

Deleting snapshot (Удаление снапшота) – состояние удаления ранее созданного снапшота (моментальной копии).

Rolling snapshot (Откат к снапшоту) – выполнение процедуры возврата шаблона к выбранному состоянию, зафиксированному снапшотом (моментальной копией).

Updating (Обновление) – состояние, в котором происходит изменение параметров или свойств шаблона.

16.3. Удаление шаблона ресурсов

Удаление шаблона ресурсов выполняется из списка шаблонов. Доступные способы вызова операции удаления:

Щёлкните правой кнопкой мыши по строке с нужным шаблоном и в появившемся контекстном меню выберите «Удалить».

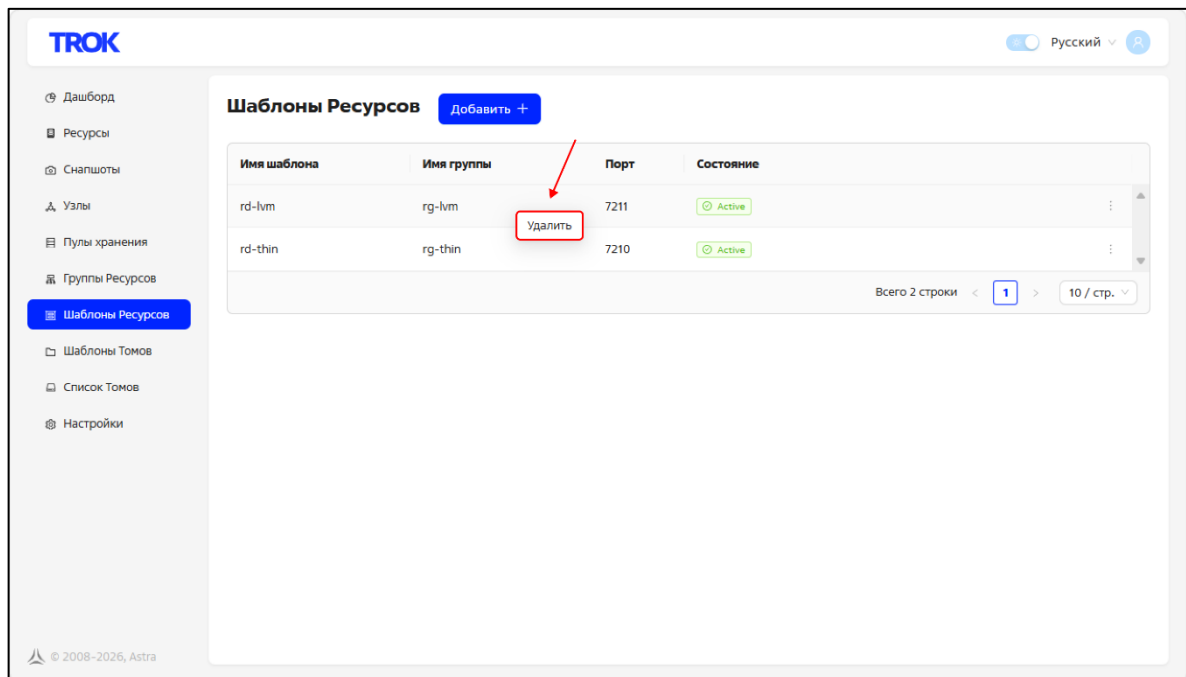


Рисунок 36 – Контекстное меню строки шаблона с пунктом «Удалить»

Либо наведите курсор на трёхточечное меню в правой части строки: появится меню быстрых действий – нажмите значок «Удалить» (корзина).

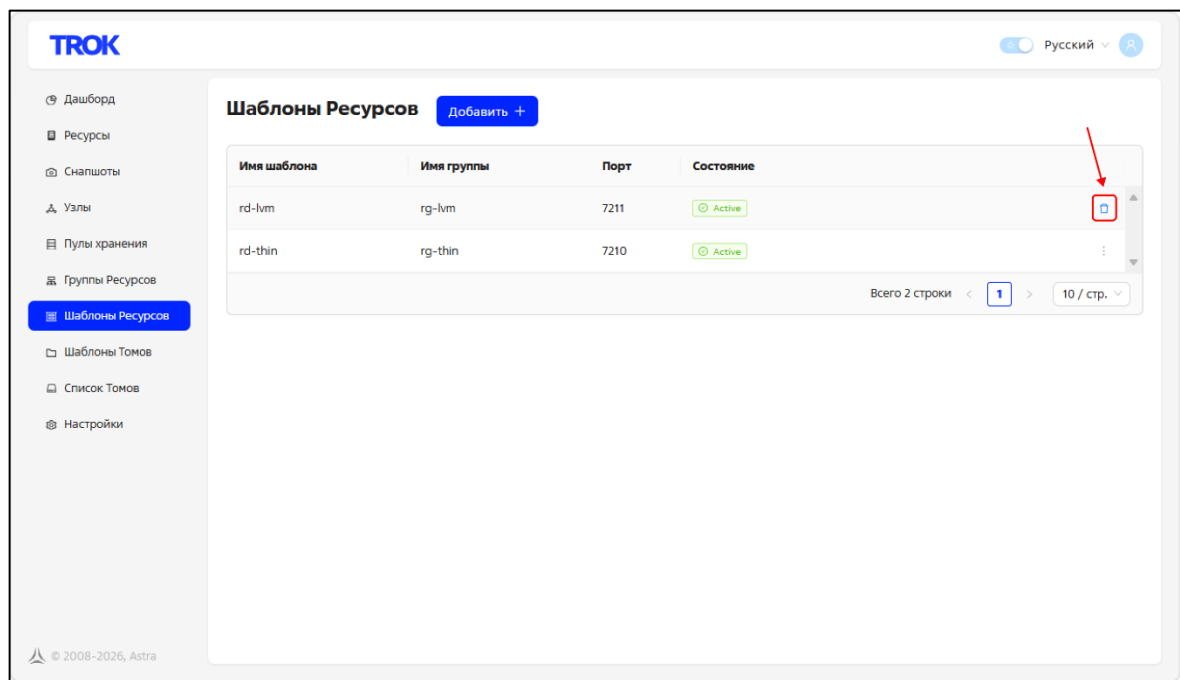


Рисунок 37 – Иконка для вызова функции удаления

После выбора команды откроется модальное окно подтверждения удаления с предупреждением о возможной потере данных и необратимости операции. В диалоге:

- убедитесь, что выбрано правильное имя шаблона;

- прочитайте предупреждение о последствиях удаления;
- подтвердите операцию нажатием «Удалить» («Delete») или отмените её.

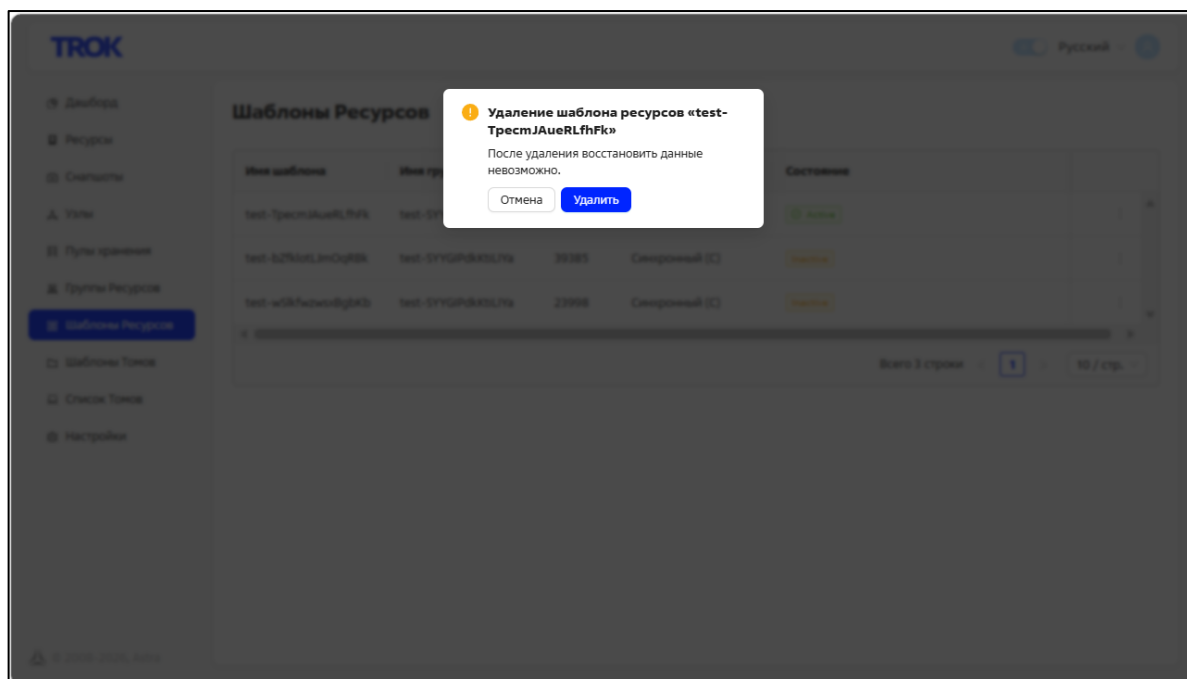


Рисунок 38 – Окно подтверждения для удаления шаблона ресурсов

17. ШАБЛОНЫ ТОМОВ

Страница «Шаблоны томов» предоставляет возможность задавать базовые шаблоны, которые впоследствии могут быть преобразованы в тома при создании ресурса. Вы можете создавать несколько томов для одного ресурса.

Для создания нового шаблона тома необходимо нажать кнопку «Добавить» и ввести требуемые данные в появившемся окне:

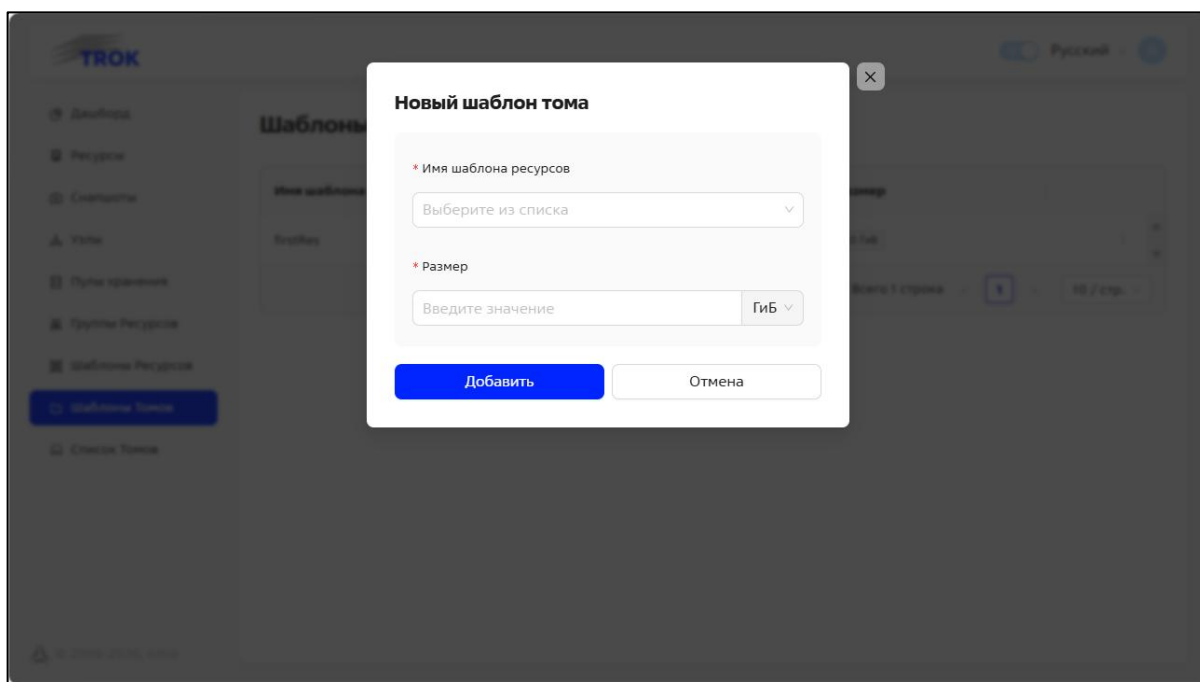


Рисунок 39 – Функция добавления шаблона тома

– **Выпадающий список «Имя шаблона ресурсов»** позволяет выбрать шаблон ресурсов, для которого вы планируете создать шаблон тома.

– **Поле «Размер»** используется для указания размера будущего тома. Также следует не забыть выбрать единицы измерения из предложенного выпадающего списка.

Созданные шаблоны томов отображаются в таблице с следующими заголовками:

- **Имя шаблона ресурсов;**
- **Номер тома;**
- **Размер.**

Для удаления шаблона томов выберите функцию «Удалить» в выпадающем меню (либо нажмите на иконку , которая появляется при наведении на строку с именем шаблона ресурсов).

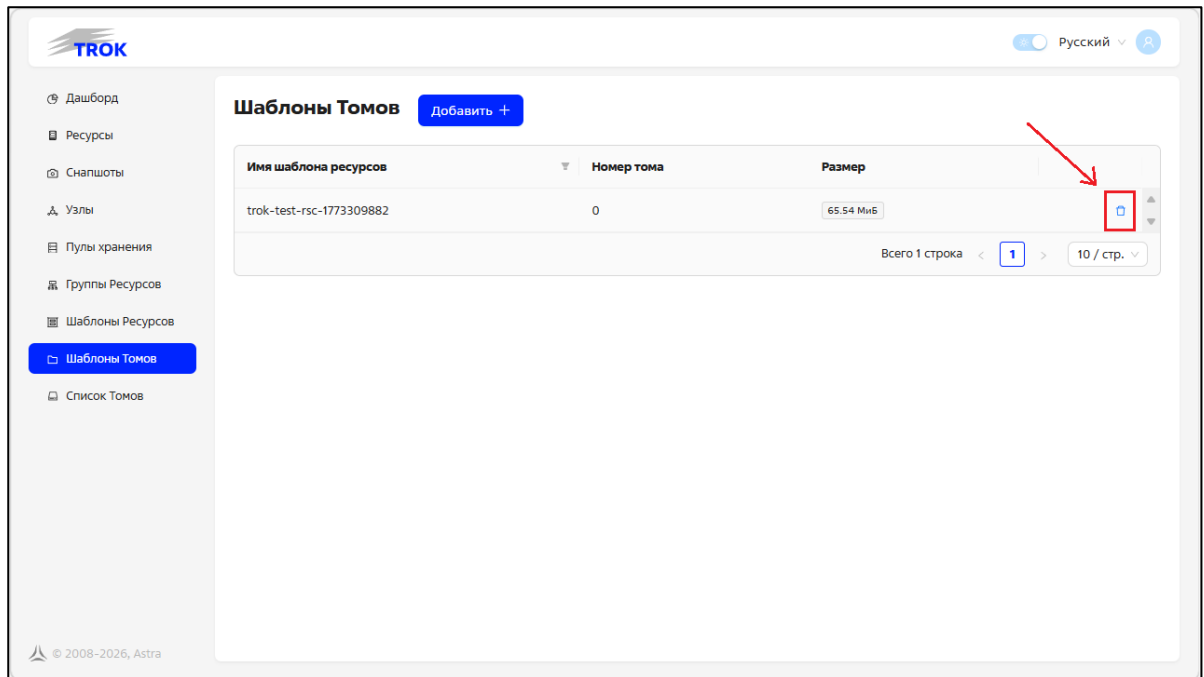


Рисунок 40 – Удаление шаблона ресурсов

А также подтвердите свое действие в появившемся окне подтверждения.

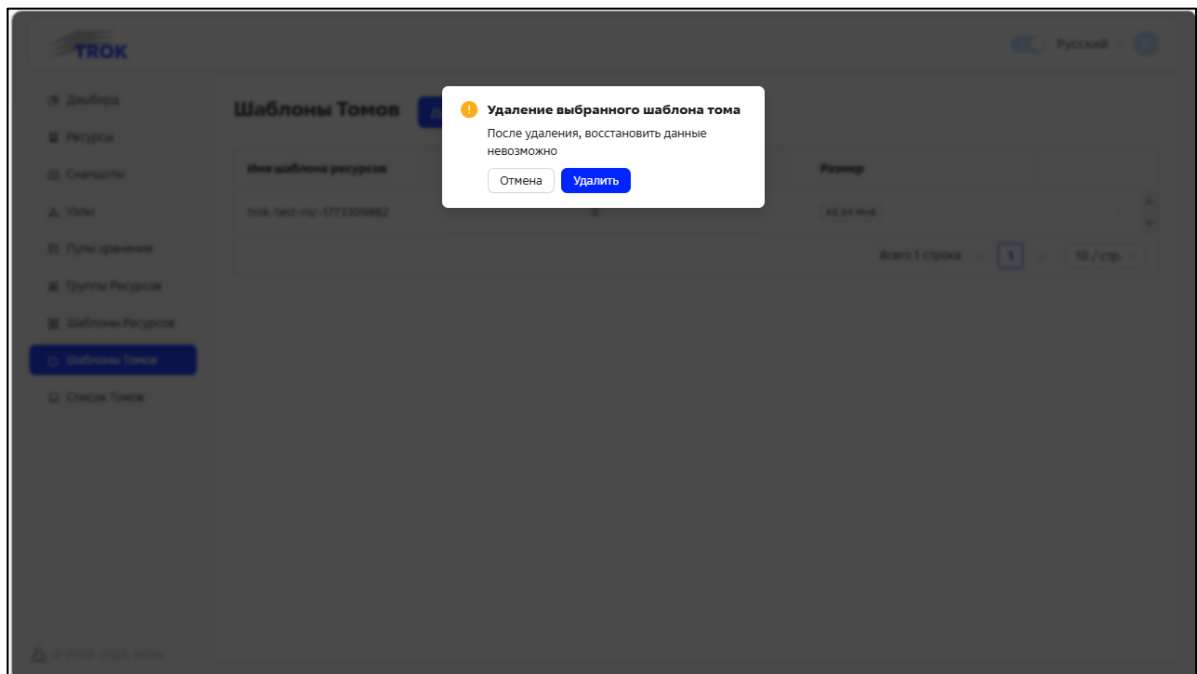


Рисунок 41 – Окно подтверждения для удаления шаблона ресурсов

Примечание. При организации доступа к хранилищу с использованием NFS необходимо создать один дополнительный том, предназначенный для хранения

метаданных NFS. Рекомендуемый размер тома метаданных должен составлять не менее 1% от общего объема системы хранения. Сохраняемые в томе метаданные обеспечивают корректное восстановление соединения с NFS-сервером при возникновении сбоев или переключении между основными узлами.

17.1. Замечания по операциям изменения размера томов (resize)

В шаблоне томов предусмотрена возможность изменения размера тома, однако **выполнять эту операцию можно только при соблюдении строгих условий и последовательности действий** во избежание потери данных (перед началом изменений прочтите раздел до конца).

Изменение размера разрешается только тогда, когда том находится в безопасном состоянии без активных фоновых операций — в частности, в статусах UpToDate или Inactive (подробнее о статусах см. раздел 18.1). В состояниях Attaching, Detaching, Negotiating, Inconsistent, Outdated, Failed, DUnknown и т.п. resize недоступен: попытки изменения размера будут блокироваться системой или завершаться ошибкой.

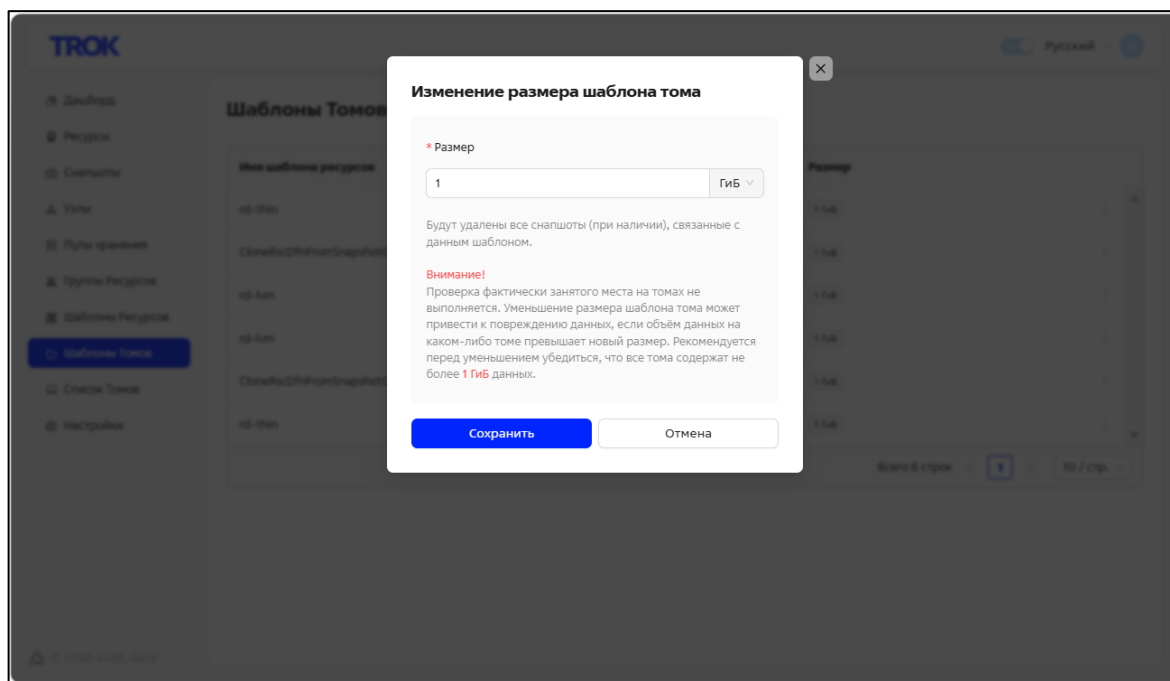


Рисунок 42 – Окно изменения размера тома для шаблона тома

Перед началом любых операций обязательно убедитесь, что все реплики находятся в состоянии UpToDate и нет выполняющихся синхронизаций,

клонирования, откатов или других фоновых задач. При увеличении размера тома проверьте наличие свободного места в целевых пулах/VG, а при уменьшении – убедитесь, что фактически используемый объём данных не превышает планируемый новый размер. По возможности выполняйте операции в окне минимальной нагрузки и временно останавливайте приложения, использующие том.

Обязательно соблюдать порядок операций при уменьшении размера. Сначала уменьшается файловая система внутри тома, только затем – сам блочный том (LV/раздел). **Категорическое ограничение:** XFS не поддерживает уменьшение размера; для XFS уменьшение невозможно и требует миграции данных на новый том или создания новой файловой системы.

Для ext2/3/4 при уменьшении размера последовательность действий на узле с Primary выглядит так:

Проверьте и исправьте ошибки файловой системы перед изменением размера

```
sudo e2fsck -f /dev/drbd/by-res/<res_def>/<том>
```

Уменьшите файловую систему до значения, не превышающего целевой размер тома (например, 900M). Файловая система должна быть уменьшена до размера $\leq$ нового размера LV.

```
sudo resize2fs /dev/drbd/by-res/<res_def>/<том> <новый_размер>
```

После успешного уменьшения файловой системы уменьшите сам логический том (пример для LVM):

```
sudo lvreduce -L <новый_размер> /dev/<vg>/<lv>
```

или

```
sudo lvresize -L <новый_размер> /dev/<vg>/<lv>
```

Уменьшение LV выполняйте только после уменьшения файловой системы (ФС); опция `--resizefs` может менять ФС автоматически, **использовать с пониманием последствий.**

Затем выполните повторную проверку ФС и примонтируйте её:

```
sudo e2fsck -f /dev/drbd/by-res/<res_def>/<том>
sudo mount /dev/drbd/by-res/<res_def>/<том> /mount/point
```

Порядок операций при увеличении размера (обратный). Сначала увеличивается блочный том, затем расширяется файловая система внутри него. Примерная последовательность:

```
sudo lvextend -L +<добавляемый_объём> /dev/<vg>/<lv>
```

или

```
sudo lvresize -L <новый_размер> /dev/<vg>/<lv>
```

Перед расширением ext-FS рекомендуется проверить ФС:

```
sudo e2fsck -f /dev/drbd/by-res/<res_def>/<том>
```

Для ext2/3/4 расширьте файловую систему:

```
sudo resize2fs /dev/drbd/by-res/<res_def>/<том>
```

без указания размера `resize2fs` увеличит ФС до максимально доступного размера тома.

Для XFS (ФС должна быть смонтирована) расширение выполняется по точке монтирования:

```
sudo xfs_growfs /mount/point
```

указывать нужно путь монтирования, а не блочное устройство (пример: `xfs_growfs /var/lib/trok-controller`).

При проверке результата убедитесь, что доступный объём соответствует ожидаемому; при необходимости выполните `e2fsck` для ext-ФС:

```
sudo df -h /mount/point
```

Ключевые предупреждения и рекомендации:

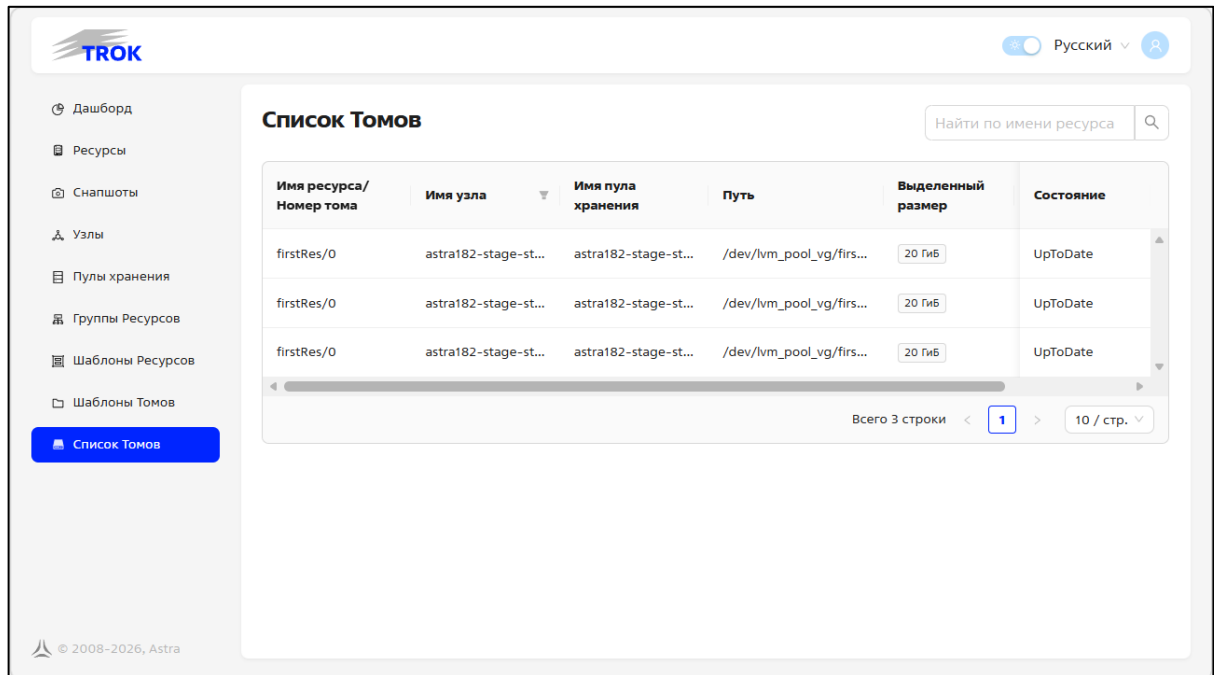
- Всегда делайте резервную копию или снапшот перед изменением размера.

- Никогда не уменьшайте LV без предварительного уменьшения файловой системы – это гарантированно приведёт к повреждению данных.
- XFS нельзя уменьшить – для уменьшения требуется миграция данных.
- Выполняйте операции только на Primary и только при отсутствии фоновых синхронизирующих операций.
- Учтите особенности thin-pool и LVM-метаданных; при использовании оркестрации (LINSTOR, LVM API) предпочтительно выполнять resize средствами оркестратора для согласованности на всех узлах.
- mkfs уничтожает данные – использовать только на пустых устройствах.

18. СПИСОК ТОМОВ

Страница с названием «Список томов» представляет собой интерфейс для созданных томов в системе хранения данных. Тома будут созданы и станут доступны после завершения процесса создания ресурса. Основным элементом страницы является таблица, в которой отображается подробная информация о каждом томе.

Таблица включает следующие столбцы:



Имя ресурса/ Номер тома	Имя узла	Имя пула хранения	Путь	Выделенный размер	Состояние
firstRes/0	astra182-stage-st...	astra182-stage-st...	/dev/lvm_pool_vg/firs...	20 ГиБ	UpToDate
firstRes/0	astra182-stage-st...	astra182-stage-st...	/dev/lvm_pool_vg/firs...	20 ГиБ	UpToDate
firstRes/0	astra182-stage-st...	astra182-stage-st...	/dev/lvm_pool_vg/firs...	20 ГиБ	UpToDate

Рисунок 43 – Список томов

– **Имя ресурса / Номер тома** – идентификатор тома, состоящий из названия шаблона ресурса и номера тома, что позволяет однозначно идентифицировать том в системе.

– **Имя узла** – название вычислительного узла, на котором размещён данный том, что помогает отслеживать распределение ресурсов по инфраструктуре.

– **Имя пула хранения** – наименование пула хранения, к которому относится том, что облегчает управление и мониторинг используемых хранилищ.

– **Путь** – файловый путь к тому на узле, указывающий на место расположение данных.

– **Выделенный размер** – объём дискового пространства, зарезервированного под данный том, выраженный в удобных единицах измерения (например, ГБ или ТБ).

– **Статус использования** – текущий режим эксплуатации тома, позволяющий быстро оценить доступность ресурса.

– **Состояние** – техническое состояние тома, что важно для своевременного выявления и устранения проблем.

18.1. Состояния тома (статусы)

Каждый статус отражает текущее состояние тома и указывает, какие операции в этот момент допустимы или требуют вмешательства.

Базовые состояния (статусы) томов:

Diskless (Без диска) – состояние тома, при котором на данном узле отсутствует локальное бэкенд-устройство: DRBD-ресурс существует, но данных на этом узле нет, и все I/O перенаправляются по сети к узлам с физическими репликами.

Attaching (Подключение) – выполняется процесс подключения диска к ресурсу. В это время происходит монтирование/инициализация устройства; до завершения операции ресурс частично недоступен для рабочих нагрузок.

Detaching (Отключение) – выполняется процесс безопасного отключения диска от ресурса. Операция гарантирует завершение текущих I/O и консистентное отсоединение; до её завершения не рекомендуется выполнять операции, изменяющие данные.

Failed (Ошибка) – диск или связанный с ним ресурс находятся в состоянии ошибки и не пригодны для работы. Требуется диагностика причин отказа (повреждение устройства, сетевая ошибка, проблемы репликации) и восстановление или замена тома.

Negotiating (Согласование) – происходит обмен и согласование состояния между узлами кластера (обычно при инициализации, переподключении реплик

или восстановлении связи). В этот период может выполняться проверка метаданных и конфигурации реплики.

Inconsistent (Несогласовано) – содержимое диска не согласуется с другими репликами (есть расхождения в данных). Требуется синхронизация и/или восстановление целостности данных перед переводом в согласованное состояние.

Outdated (Устарело) – данные на данном диске отстают от актуальной версии; реплика требует синхронизации, чтобы привести данные в соответствие с мастер-копией или достичь состояния кворума.

DUnknown (Неизвестное состояние) – временное состояние, указывающее на отсутствие достоверной информации о состоянии диска (например, во время опроса или при ошибке коммуникации). Требуется повторная проверка статуса.

Consistent (Согласовано) – данные на диске согласованы с другими репликами, но, возможно, не являются полностью актуальными (есть риск отставания). Обычно это промежуточное безопасное состояние до завершения синхронизации.

UpToDate (Актуально) – диск полностью синхронизирован и содержит самую последнюю версию данных. Это рабочее состояние, при котором том готов к обслуживанию запросов и может считаться «активным» с точки зрения согласованности данных.

19. РЕСУРСЫ КЛАСТЕРА

Перед созданием нового ресурса необходимо выполнить следующую последовательность шагов:

- 1 шаг – создать узлы;
- 2 шаг – создать пулы хранения;
- 3 шаг – создать группы ресурсов;
- 4 шаг – создать шаблоны ресурсов;
- 5 шаг – создать шаблоны томов.

При создании нового ресурса программное обеспечение требует ввода следующих данных:

– **Выпадающий список «Имя шаблона ресурсов»** позволяет выбрать ранее созданный шаблон, который определяет характеристики нового ресурса. Шаблоны ресурсов создаются во вкладке «Шаблоны ресурсов» и служат для стандартизации конфигураций, упрощая процесс развертывания.

– **Переключатель «Автоматический выбор узла хранения»** при создании ресурса делегирует системе выбор наиболее подходящих узлов для размещения данных. В приоритете оказываются узлы с наибольшим объёмом свободного пространства в пулах хранения. При отключении переключателя узлы можно назначить вручную.

– **Выпадающий список «Узел»** позволяет выбрать конкретный узел, на котором будет размещен ресурс. Узлы представляют собой физические или виртуальные машины, ранее созданные во вкладке «Узлы», и обеспечивают вычислительные мощности для работы с ресурсами.

– **Выпадающее меню «Пул хранения»** позволяет выбрать группу хранения, которая была создана на вкладке «Пулы хранения». Пулы хранения представляют собой группы физических устройств, объединенных для оптимизации использования пространства и повышения производительности.

– **Переключатель «Добавить бездисковую копию»** позволяет создать реплику данных для уже существующего ресурса, которая не использует

локальное дисковое пространство узла, а подключается к диску другого узла по сети для повышения отказоустойчивости.

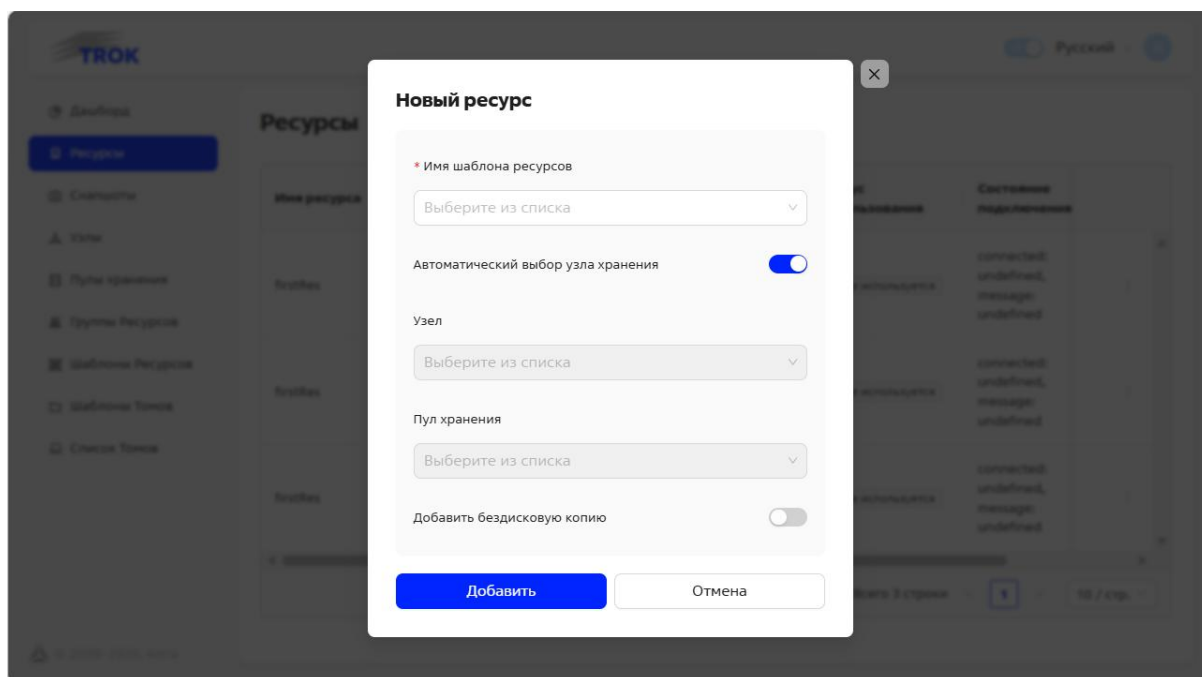


Рисунок 44 – Добавление ресурса

Созданные ресурсы отображаются в таблице с заголовками:

- **Имя ресурса:** Имя шаблона ресурса, для которого создан ресурс.
- **Имя узла:** Указывает узел, на котором размещен ресурс.
- **Время создания:** Дата и время, когда ресурс был создан.
- **Порт:** Номер порта, используемого для подключения к ресурсу.
- **Статус использования:** Информация о текущем состоянии использования ресурса.
- **Состояние подключения:** Указывает статус подключения ресурса к системе.
- **Состояние:** Общая информация о работоспособности и состоянии ресурса.

Для удаления ресурса наведите курсор мыши на строку с его названием и кликните правой кнопкой мыши для появления выпадающего меню с функцией удаления. Также при наведении указателя на мыши на строку с описанием ресурса в правой части строки появляется иконка удаления. Вы можете использовать его и в появившемся окне подтверждения подтвердить удаление.

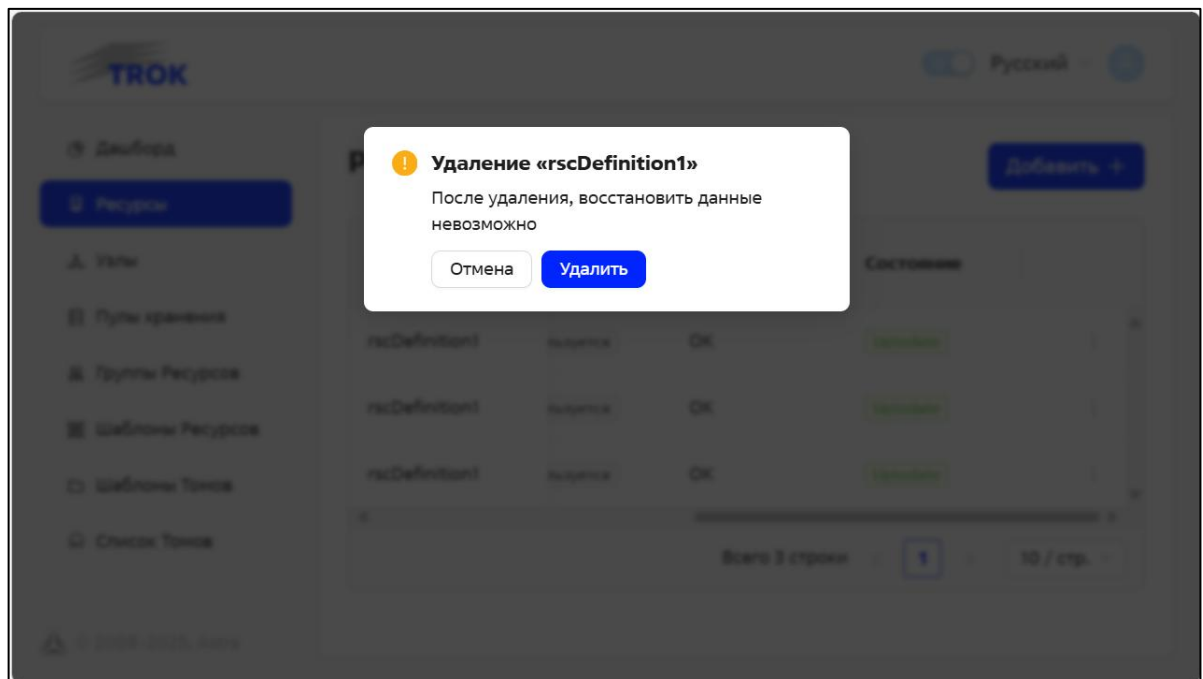


Рисунок 45 – Удаление ресурса

20. СНИМКИ СОСТОЯНИЯ (СНАПШОТЫ)

Вкладка «Снапшоты» предназначена для создания снимков текущего состояния ресурсов.

При создании нового снапшота требуется ввести следующие данные:

Рисунок 46 – Создание снапшота

- **Выпадающее меню «Определение ресурса»** позволяет выбрать из списка имя шаблона ресурсов, по которому вы хотите сделать снимок состояния.
- **Поле «Имя снапшота»** предназначено для ввода уникального идентификатора для снимка состояния.

Созданные снапшоты отображаются в таблице с заголовками:

- **Имя снапшота:** Произвольное имя снапшота (например, snap-01).
- **Определение ресурса:** Шаблон ресурсов, по которому созданы ресурсы.
- **Узлы:** Узлы кластера, на которых доступен снапшот (например, worker-01, worker-02).
- **Время создания:** Дата и время создания снапшота (например, 2026-03-05T12:34:56Z).
- **Состояние:** Информация о статусе снапшота:

- **Pending (Ожидание):** Процесс создания идёт, данные копируются/захватываются, но снимок пока недоступен для использования.
- **OK (Готов):** Снимок полностью создан и готов к использованию. Можно использовать его для восстановления данных.
- **Rollback (Откат):** Выполняется процедура возврата целевого ресурса к состоянию, зафиксированному выбранным снимком.
- **Invalid (Не удалось):** Создание снимка завершилось с ошибкой. Нельзя использовать данные снимка до устранения проблемы.

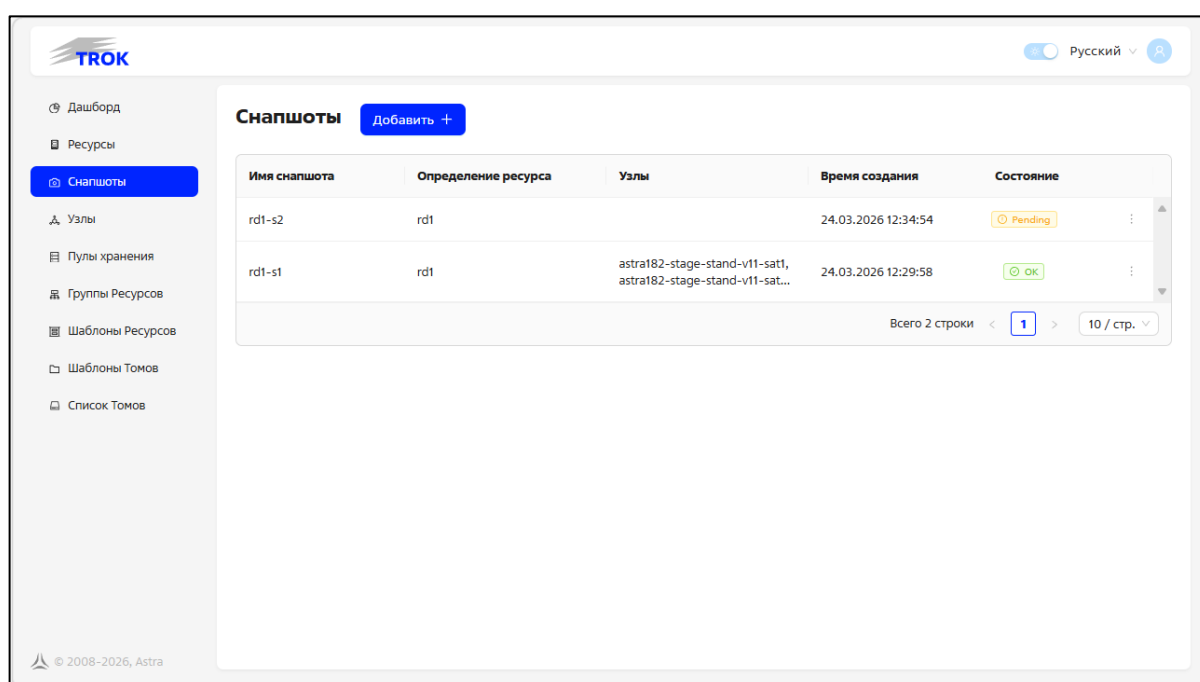


Рисунок 47 – Список снимотов

Чтобы восстановить ресурс до сохранённой в выбранном снимоте конфигурации текущего состояния, наведите курсор на строку нужного снимота и в появившемся контекстном меню нажмите на иконку отката (↶).

Для удаления снимота используйте иконку 🗑️.

21. УЧЁТНЫЕ ДАННЫЕ ПРИЛОЖЕНИЯ (НАСТРОЙКИ)

Раздел «Настройки» предназначен для управления учётными записями сервисов и их секретными ключами. В верхней части страницы располагается кнопка «Добавить» («Add»), по нажатию которой открывается форма для создания новой записи. В основной части приведена таблица зарегистрированных учётных данных. Для каждой записи отображаются основные параметры и доступны быстрые операции – регенерация секрета и удаление.

В таблице для каждой записи отображаются следующие параметры:

- **Имя (Name)** – уникальный идентификатор записи учётных данных;
- **Описание (Description)** – краткая служебная заметка, заполняется по желанию;
- **Создал (Created By)** – пользователь, создавший запись;
- **Выдан (Issued)** – дата и время создания/выдачи;
- **ID клиента (Client ID)** – идентификатор клиента, выдаваемый системой;
- **Истекает (Expires)** – дата и время окончания срока действия секрета;
- **Действия (Actions)** – в правой части строки при наведении появляются иконки для регенерации секрета (значок ключа/обновления) и удаления (значок корзины).

21.1. Создание новой записи

Для создания новой учётной записи нажмите кнопку «Добавить» («Add») в верхней части экрана.

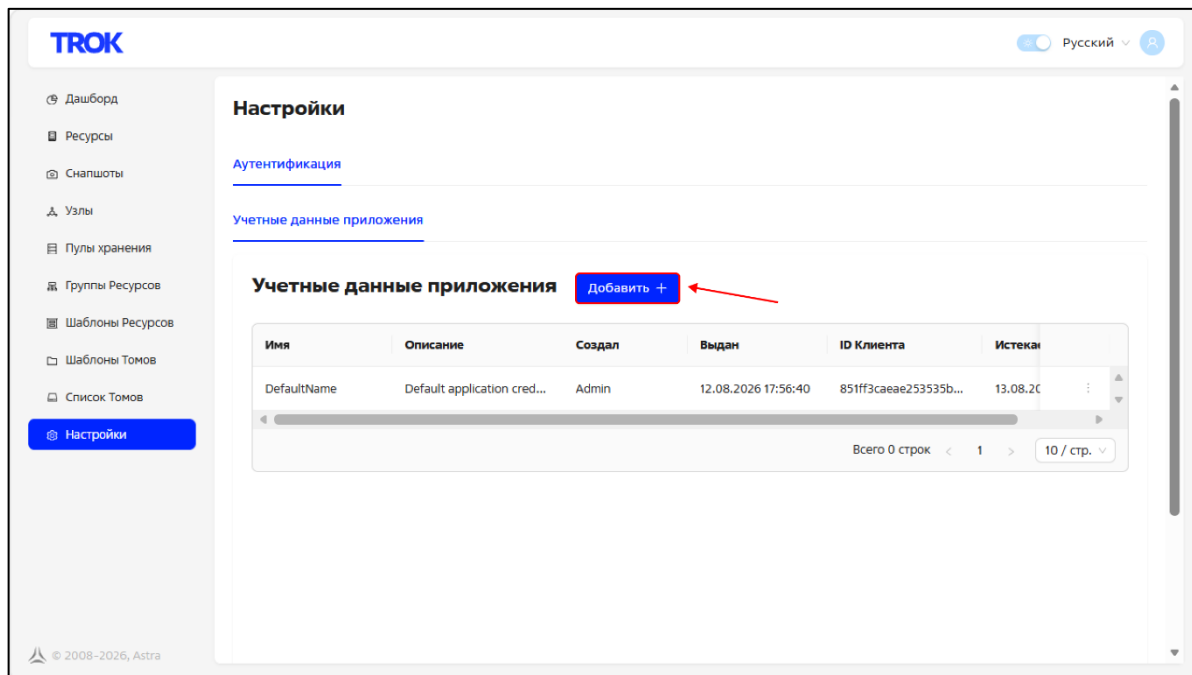


Рисунок 48 – Активация процедуры добавления новой учетной записи

Откроется модальное окно для ввода параметров новой записи. В поле «Имя» укажите уникальное имя – оно будет использоваться при идентификации учётных данных. При необходимости заполните поле «Описание», чтобы пояснить назначение записи. Выберите «Время жизни» учетной записи из предложенных вариантов в выпадающем списке.

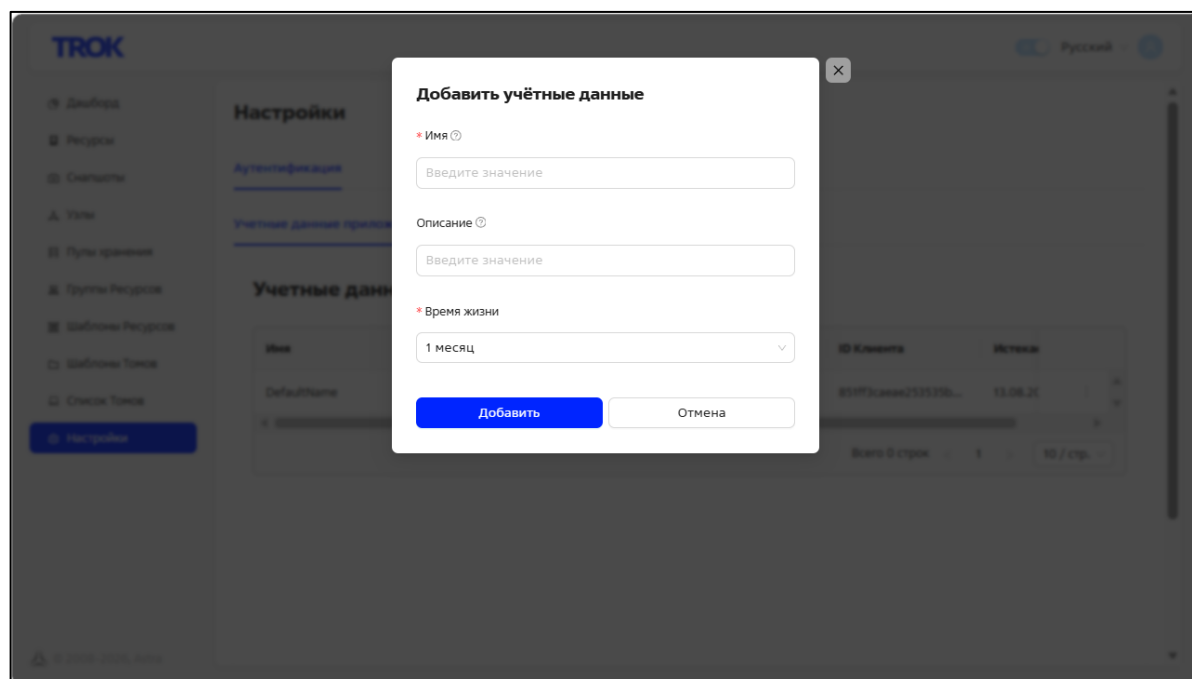


Рисунок 49 – Модальное окно для добавления новой учетной записи

После подтверждения система сгенерирует пару значений – Client ID и Secret; секрет отображается только один раз, поэтому его нужно сохранить немедленно.

21.2. Регенерация секрета

Регенерация секрета выполняется из строки таблицы. Наведите курсор на трёхточечное меню в правой части строки – появятся иконки быстрых действий, нажмите иконку регенерации (ключ/стрелка).

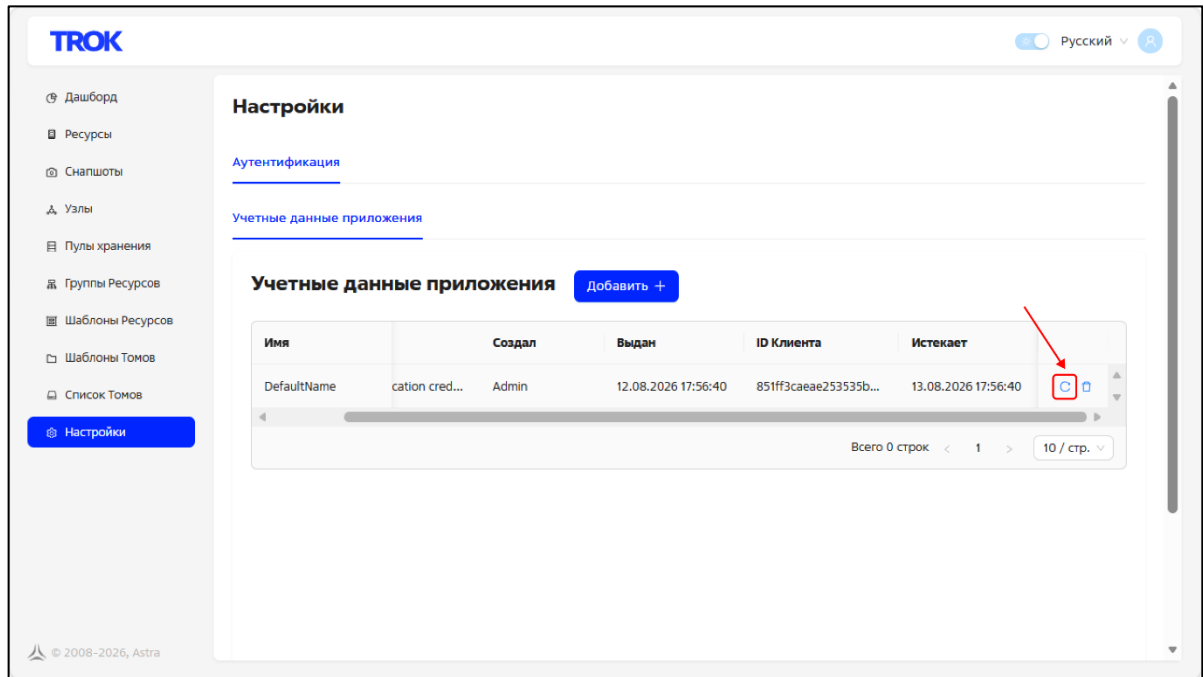


Рисунок 50 – Иконка для вызова функции регенерации секрета

При выборе команды откроется модальное окно регенерации с предупреждением о немедленной инвалидизации текущего секрета и с возможностью выбрать срок действия нового. После подтверждения текущий секрет перестает работать, а новый отображается единожды – сохраните его.

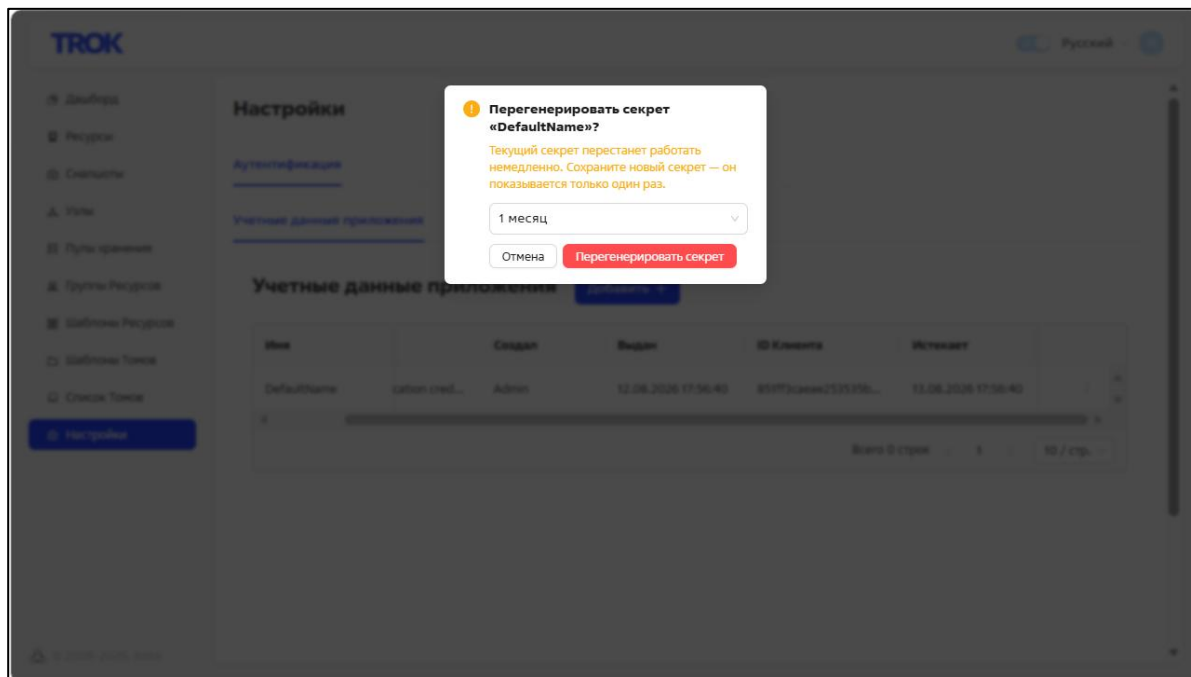


Рисунок 51 – Модальное окно регенерации секрета

21.3. Удаление учётных данных

Удаление учётных данных также выполняется из таблицы через иконку удаления в правой части строки.

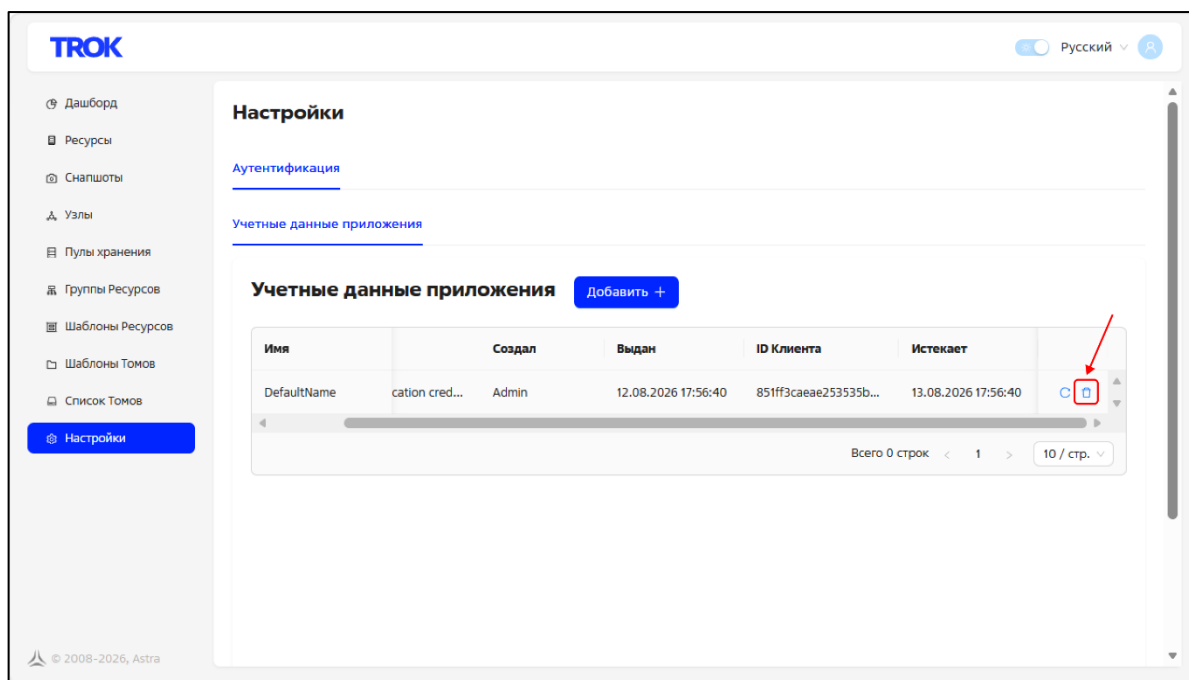


Рисунок 52 – Иконка для вызова функции удаления учетной записи

После выбора команды откроется модальное окно подтверждения удаления с предупреждением о необратимости операции.

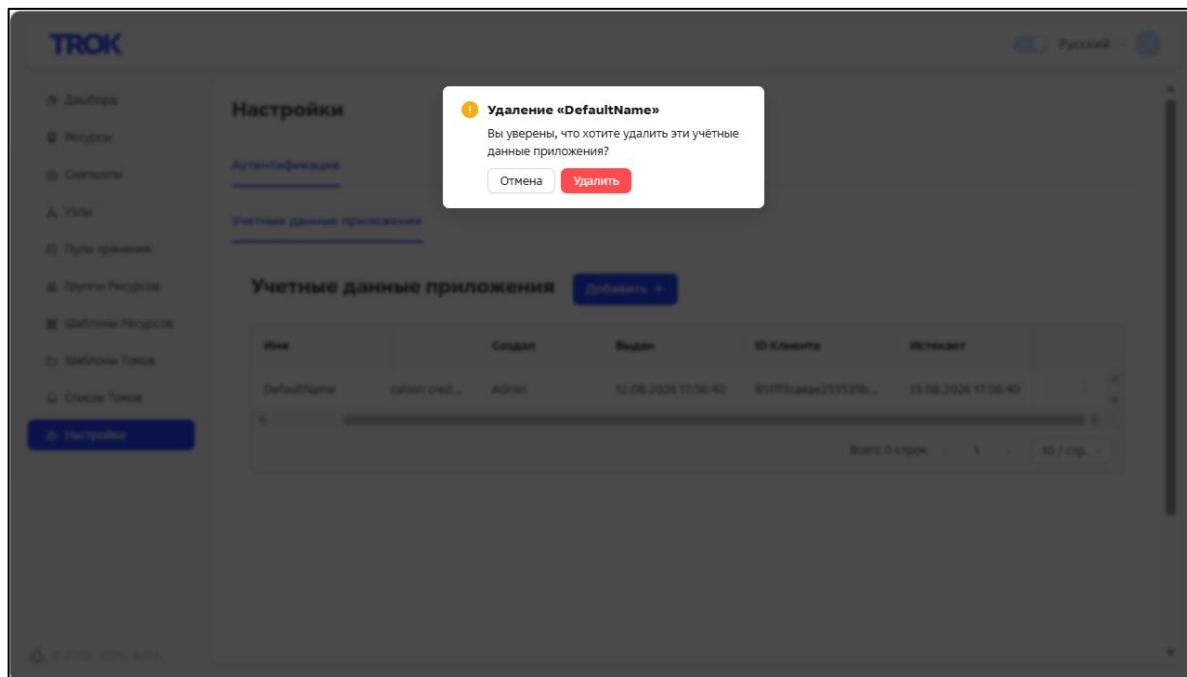


Рисунок 53 – Окно подтверждения для удаления учетной записи

Проверьте корректность выбранной записи и подтвердите удаление кнопкой «Удалить» или отмените операцию. После подтверждения доступ по соответствующему Client ID/Secret прекращается немедленно, запись исчезает из таблицы.

Примечания: операция удаления необратима. Если учётные данные используются в каких-то автоматизированных задачах или интеграциях, предварительно обновите конфигурации, чтобы избежать простоя. Имя учётных данных должно быть уникальным в рамках системы.

22. ВАЛИДАЦИЯ ПОЛЕЙ SDS UI

При создании новых сущностей обязательным для заполнения является каждый параметр, отмеченный специальным символом (звёздочкой). Поля, предназначенные для указания имён сущностей (таких как узлы, тома и прочие), должны содержать только символы латинского алфавита. Максимальная длина имени ограничена 255 символами. Использование пробелов и следующих специальных символов запрещено: !#\$%^&*()=+{}[].

Поле для ввода IP-адреса принимает числовые значения, соответствующие стандартам IPv4 или IPv6. Валидация IPv4-адреса включает проверку структуры на наличие четырёх октетов, разделённых точками, при этом каждый октет должен находиться в диапазоне от 0 до 255. Для IPv6-адреса проверяется корректность шестнадцатеричного представления с учётом допустимых сокращений и разделителей (двоеточий).

Значение в поле «Порт» должно быть целым числом и находиться в диапазоне от 1 до 65535, что соответствует допустимым значениям портов протоколов TCP и UDP.

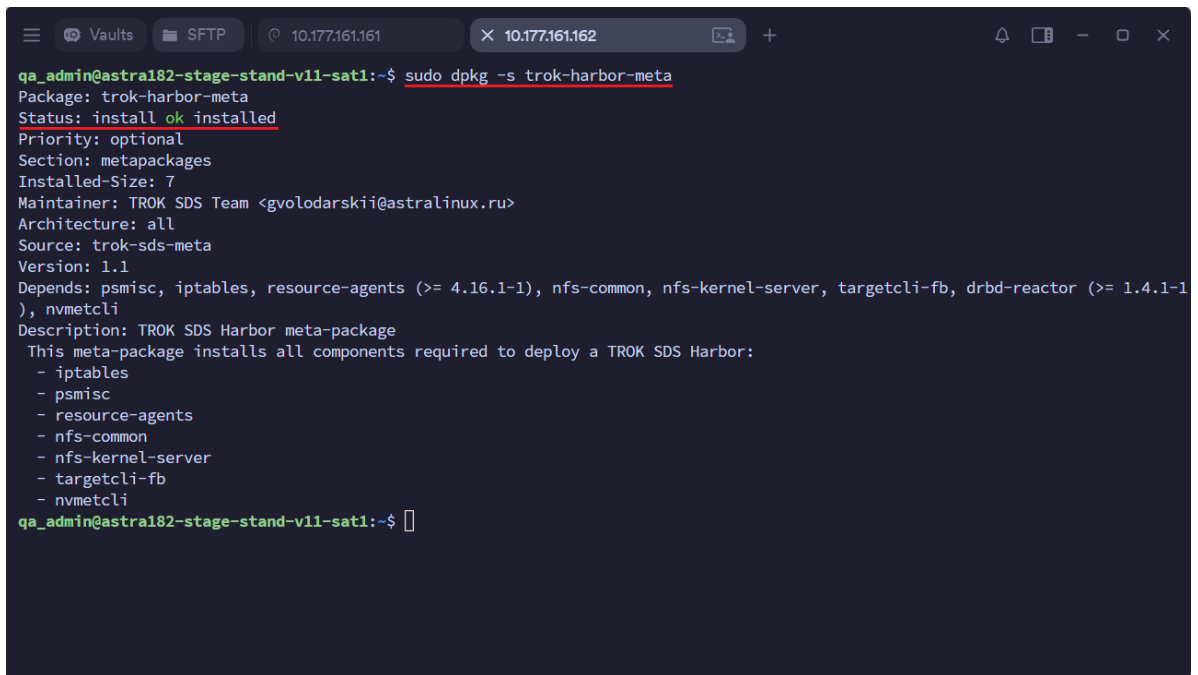
Поле «Физический диск или том» предполагает ввод абсолютного пути к блочному устройству, например, /dev/sdb, /dev/nvme0n1 или к разделу, например, /dev/sdb1. Введённый путь должен соответствовать стандартному синтаксису файловых путей в Unix-подобных операционных системах.

23. НАСТРОЙКА ДОСТУПА ПО ПРОТОКОЛАМ

Для работы с протоколами доступа на узлы-воркеры должен быть установлен компонент trok-harbor.

Проверить его наличие можно с помощью команды:

```
sudo dpkg -s trok-harbor-meta
```



```
qa_admin@astra182-stage-stand-v11-sat1:~$ sudo dpkg -s trok-harbor-meta
Package: trok-harbor-meta
Status: install ok installed
Priority: optional
Section: metapackages
Installed-Size: 7
Maintainer: TROK SDS Team <gvolodarskii@astralinux.ru>
Architecture: all
Source: trok-sds-meta
Version: 1.1
Depends: psmisc, iptables, resource-agents (>= 4.16.1-1), nfs-common, nfs-kernel-server, targetcli-fb, drbd-reactor (>= 1.4.1-1), nvmetcli
Description: TROK SDS Harbor meta-package
 This meta-package installs all components required to deploy a TROK SDS Harbor:
  - iptables
  - psmisc
  - resource-agents
  - nfs-common
  - nfs-kernel-server
  - targetcli-fb
  - nvmetcli
qa_admin@astra182-stage-stand-v11-sat1:~$
```

Рисунок 54 – Результат выполнения команды `sudo dpkg -s trok-harbor-meta`

Если пакет не установлен, используйте команду (подробная информация об установке пакетов указана в документе «TROK Инструкция по установке» в 8 пункте):

```
sudo apt install -y trok-harbor-meta
```

Настройка доступа к ресурсам осуществляется только после создания самих ресурсов.

Важно. В рамках одной системы хранения один ресурс может быть назначен только одному сервису предоставления доступа к данным (например, Samba, iSCSI или NFS); попытка одновременного назначения нескольких сервисов к одному ресурсу приведёт к конфликту, и функционирование второго сервиса будет невозможно.

23.1. Настройка отказоустойчивого хранилища NFS (на базе drbd-reactor)

Служба NFS (Network File System) развёртывается в виде сервиса на каждом узле хранения, на котором расположена реплика ресурса, для обеспечения доступа к данным.

Для корректной работы NFS-кластера рекомендуется заранее создать отдельные DRBD-тома:

- На каждом узле хранения должен быть минимум один ресурс с томом для метаданных NFS, который предназначен для хранения служебной информации файловой системы. Обычно данный том имеет относительно небольшой объём (не менее 1% от общего объема системы хранения), так как метаданные занимают значительно меньше места по сравнению с пользовательскими данными.

- Ресурс (или ресурсы) с томами для данных NFS – используется для размещения непосредственно пользовательских данных, доступных через файловые шары (экспортируемые каталоги).

Все перечисленные DRBD-ресурсы должны быть предварительно созданы и полностью подготовлены к эксплуатации на всех узлах хранения, входящих в кластер.

Первый этап настройки включает в себя подготовку операционной системы.

Выполните команды для отключения и остановки стандартного NFS-сервиса на всех узлах:

```
sudo systemctl disable nfs-server  
sudo systemctl stop nfs-server
```

Активируйте службу drbd-reactor:

```
sudo systemctl enable drbd-reactor
```

Вторым этапом следует подготовка файловой системы.

Переведите DRBD-ресурс в первичное (Primary) состояние. Если предполагается организация доступа по NFS к нескольким DRBD-ресурсам, необходимо перевести в состояние Primary все соответствующие ресурсы.

Для просмотра списка доступных DRBD-ресурсов и их текущего состояния используйте команду:

```
sudo drbdadm status
```

Результат выполнения данной команды содержит информацию о каждом ресурсе, его роли и состоянии дисков на всех узлах. Например:

```
rd1 role:Secondary
  disk:UpToDate
  astra182-stage-stand-v11-sat2 role:Secondary
    peer-disk:UpToDate
  astra182-stage-stand-v11-sat3 role:Secondary
    peer-disk:UpToDate

rd2 role:Secondary
  disk:UpToDate
  astra182-stage-stand-v11-sat2 role:Secondary
    peer-disk:UpToDate
  astra182-stage-stand-v11-sat3 role:Secondary
    peer-disk:UpToDate
```

Для перевода выбранного DRBD-ресурса в состояние Primary используйте команду:

```
sudo drbdadm primary <имя_ресурса>
```

Примечание. Ресурс имеет то же имя, что и шаблон ресурса.

После изменения режима тома в web-интерфейсе изменяет свой статус на «Используется».

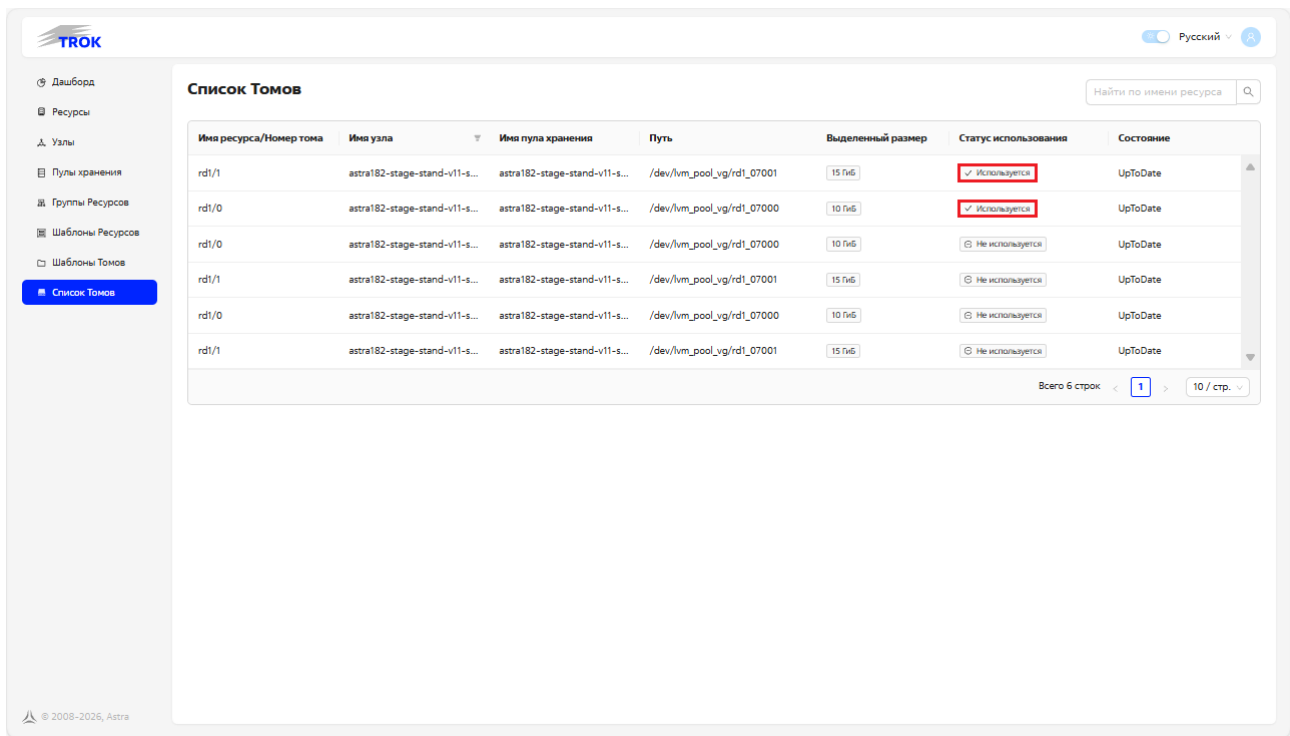


Рисунок 55 – Пример отображения статуса томов в web-интерфейсе

Создайте файловую систему на томах DRBD-ресурсов, к которым требуется доступ по NFS. Форматирование выполняется однократно на узле, где ресурс в состоянии Primary:

```
sudo mkfs -t ext4 /dev/drbd/by-res/<имя_ресурса>/<порядковый_номер_тома_ресурса>
```

```

qa_admin@astral82-stage-stand-v11-sat1:~$ sudo mkfs -t ext4 /dev/drbd/by-res/rd1/0
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 2499835 4k blocks and 625856 inodes
Filesystem UUID: 8917b01b-d43b-4fd3-86e2-f1e2f1238109
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

qa_admin@astral82-stage-stand-v11-sat1:~$

```

Рисунок 56 – Результат выполнения команды монтирования тома DRBD в файловую систему ext4

Создайте каталог для хранения метаданных NFS. Такой каталог должен быть создан на каждом узле хранения (узел-воркер). Хранящиеся в данном каталоге метаданные используются для восстановления соединения с NFS-сервером в случае переключения роли primary-узла.

```
sudo mkdir -p /srv/ha/internal/nfs
```

Создайте структуру каталогов, которая будет использоваться для экспорта и служебных целей. Подготовьте каталоги для монтирования и служебные директории на всех узлах, где находятся drbd-ресурсы с томами для раздачи доступа по NFS. Сначала смонтируйте том для хранения метаданных NFS на первичном узле:

```
sudo mount /dev/drbd/by-res/<имя_ресурса_для_метаданных>/<номер_тома_для_метаданных> /srv/ha/internal/nfs
```

Создайте необходимые подкаталоги и установите права доступа на томе drbd-ресурса для хранения метаданных NFS используя команды:

```
sudo mkdir /srv/ha/internal/nfs/nfsinfo
sudo chmod -R 0755 /srv/ha/internal/nfs
```

Создайте каталог на всех узлах:

```
sudo mkdir -p /srv/nfs-share/<имя_ресурса>
```

Смонтируйте тома для хранения пользовательских данных NFS на Primary узле:

```
sudo mount /dev/drbd/by-res/<имя_ресурса>/<номер_тома_для_данных> /srv/nfs-share/<имя_ресурса>
sudo chmod 0755 /srv/nfs-share/<имя_ресурса>
```

Размонтируйте тома ресурсов для данных NFS:

```
sudo umount /srv/nfs-share/<имя_ресурса>
```

Размонтируйте тома и переведите ресурсы в состояние Secondary:

```
sudo umount /srv/ha/internal/nfs

sudo drbdadm secondary <имя_ресурса_1>

sudo drbdadm secondary <имя_ресурса_2> # Если у вас на узле больше 1 ресурса на
котором нужно настроить NFS)
```

Далее следуют шаги для конфигурации службы drbd-reactor.

Создайте конфигурационный файл для promoter'a на всех узлах, где находятся drbd-ресурсы:

```
sudo touch /etc/drbd-reactor.d/nfs.toml
```

Откройте файл /etc/drbd-reactor.d/nfs.toml и добавьте следующую конфигурацию (универсальную для NFS-таргетов):

```
[[promoter]]

[promoter.resources]

[promoter.resources.<имя_ресурса_для_метаданных>]
  on-drbd-demote-failure = "reboot-immediate"
  runner = "systemd"
  preferred-nodes = [<"<имя_приоритетного_узла>">]

  start = [
    ""ocf:heartbeat:Filesystem fs_cluster_private device=/dev/drbd/by-
res/<имя_ресурса_для_метаданных>/<номер_тома_для_метаданных>
directory=/srv/ha/internal/nfs fstype=ext4 run_fsck=no""",
    ""ocf:heartbeat:Filesystem fs_1 device=/dev/drbd/by-
res/<имя_ресурса>/<номер_тома_для_данных> directory=/srv/nfs-
share/<имя_ресурса_для_метаданных> fstype=ext4 run_fsck=no""",
    ""ocf:heartbeat:nfsserver nfsserver nfs_ip=<плавающий_ip>
nfs_server_scope=<плавающий_ip>
nfs_shared_infodir=/srv/ha/internal/nfs/nfsinfo""",
    ""ocf:heartbeat:exportfs exp_nfs_a directory=/srv/nfs-
share/<имя_ресурса_для_метаданных> fsid=101
clientspec=<подсеть_для_подключения_клиентов>
options='rw,no_subtree_check,no_root_squash'""",
    ""ocf:heartbeat:IPaddr2 virtual_ip cidr_netmask=<маска_подсети>
ip=<плавающий_ip> """,
  ]
```

Пример конфигурации:

```
[[promoter]]
```

```
[promoter.resources]

[promoter.resources.nfs1]
    on-drbd-demote-failure = "reboot-immediate"
    runner = "systemd"
    preferred-nodes = ["astra182-stage-stand-v11-cmbnd","astra182-stage-stand-
v11-sat1"
    ]

    start = [
        "ocf:heartbeat:Filesystem fs_cluster_private device=/dev/drbd/by-
res/nfs1/0 directory=/srv/ha/internal/nfs fstype=ext4 run_fsck=no",
        "ocf:heartbeat:Filesystem fs_1 device=/dev/drbd/by-res/nfs-data1/0
directory=/srv/nfs-share/nfs-data1 fstype=ext4 run_fsck=no",
        "ocf:heartbeat:nfsserver      nfsserver      nfs_ip=10.177.161.170
nfs_server_scope=10.177.161.170 nfs_shared_infodir=/srv/ha/internal/nfs/nfsinfo",
        "ocf:heartbeat:exportfs      exp_nfs_data_1  directory=/srv/nfs-
share/nfs-data1      fsid=101      clientspec=10.177.161.0/24
options='rw,no_subtree_check,no_root_squash'",
        "ocf:heartbeat:IPaddr2      virtual_ip      cidr_netmask=24
ip=10.177.161.170",
    ]
```

Разбор конфигурации:

– **[promoter]**

Открывает новый блок описания промоутера. Позволяет вам определить отдельный набор ресурсов/действий для работы с кластером.

– **[promoter.resources]**

Блок описания ресурсов, которыми управляет этот promoter.

– **[promoter.resources.rd1]**

Блок описания ресурсов с именем ресурса.

– **on-drbd-demote-failure = "reboot-immediate"**

Если при переключении ресурса DRBD с режима Primary в режим Secondary (demote) происходит ошибка, этот узел будет немедленно перезагружён.

– **runner = "systemd"**

Параметр задаёт способ управления всеми вспомогательными сервисами и ресурсами для DRBD-promoter. В данном случае использование systemd позволяет запускать, останавливать, отслеживать статус нужных служб через системный сервис-менеджер (systemd)

– **preferred-nodes = ["<имя_узла_1>", "<имя_узла_2>"]**

Определяет список узлов (серверов) – кандидатов для промодутирования ресурса в primary (активная роль). Кластер учитывает порядок и предпочтения, где в первую очередь держать ресурс в активном состоянии. Если что-то случилось с первым узлом в списке – промодут происходит на втором, и так далее. Например, если preferred-nodes = ["node1", "node2"] – DRBD-ресурс будет стараться быть primary на node1, и если недоступен – только тогда на node2.

```
– "ocf:heartbeat:Filesystem fs_metadata device=/dev/drbd/by-res/<имя_ресурса>/1 directory=/srv/ha/internal/nfs fstype=ext4 run_fsck=no",
```

Монтирование приватной служебной директории, содержащей метаданные NFS, осуществляется с использованием выделенного тома DRBD-ресурса. При создании ресурса вы создаете в нем помимо основных томов, дополнительный том специально для NFS. Он служит исключительно для хранения служебной информации и не используется для размещения пользовательских данных, представляя собой метатом для функциональности NFS-кластера. В предварительно созданную директорию /srv/ha/internal/nfs монтируется том, например, <имя_ресурса>/1 (номер тома не имеет значения, вы выделяете его исходя из собственной архитектуры), в котором размещаются кластерные служебные данные, включая каталог nfs_shared_infodir.

```
– "ocf:heartbeat:Filesystem fs_1 device=/dev/drbd/by-res/<имя_ресурса>/<номер_тома> directory=/srv/nfs-share/<имя_ресурса> fstype=ext4 run_fsck=no",
```

Правило подключает том с пользовательскими данными.

device – устройство DRBD-ресурса <имя_ресурса>/<номер_тома> реальные данные для первой шары. Его можно подключить, используя 2 вида ссылок: первая: /dev/drbd/by-res/<имя_ресурса>/<номер_тома> либо вторая: /dev/drbd<minor-номер_устройства>. Minor-номер устройства – уникальный идентификатор, используется DRBD для адресации своих блочных устройств. Например, drbd7000 – это уникальный номер 700 для защиты от конфликтов и

том №0 (rd1/0), а drbd7001 – это аналогично уникальный номер 700 и том №1 (rd1/1). Посмотреть весь список блочных устройств можно используя команду

```
ls /dev/ | grep drbd
```

Но в случае, если на сервере более одного ресурса, тома DRBD создаются со сквозным нумерованием. То есть, если у вас помимо ресурса rd1 также есть ресурс rd2 и тома в нем rd2/0 и rd2/1, то minor-номер устройств для них будет drbd7002 и drbd7003 соответственно. Чтобы не путаться в наименованиях, рекомендуется использовать более простой вариант ссылки – /dev/drbd/by-res/rd1/0.

Важно. В веб-интерфейсе отображаются тома LVM (путь тома LVM), однако их адреса не следует использовать при настройке правил. Для корректной работы необходимо указывать адреса томов DRBD, которые описаны выше.

Директория /srv/nfs-share/<имя_ресурса> – точка монтирования, где будут лежать файлы, которые NFS будет раздавать.

Если вы монтируете второй ресурс, то для него дополнительно нужно будет создать заранее каталог для монтирования и служебные директории. Пример каталога `sudo mkdir -p /srv/nfs-share/<имя_второго_ресурса>`, а также выдать соответствующие права, как описано выше.

Правило для него будет выглядеть аналогично:

```
""ocf:heartbeat:Filesystem          fs_2          device=/dev/drbd/by-
res/<имя_второго_ресурса>/0    directory=/srv/nfs-share/<имя_второго_ресурса>
fstype=ext4 run_fsck=no""
```

```
– ""ocf:heartbeat:nfsserver        nfsserver        nfs_ip=<плавающий_ip>
nfs_server_scope=<плавающий_ip>
nfs_shared_infodir=/srv/ha/internal/nfs/nfsinfo""
```

Запускает фоновые процессы (демоны) NFS (nfsd, rpc.mountd, rpc.statd и так далее).

nfs_shared_infodir=/srv/ha/internal/nfs/nfsinfo – критически важная директория, которая должна находиться на общем хранилище (на выделенном томе DRBD).

В ней NFS-сервер хранит служебную информацию, уникальные идентификаторы (например, для statd) и другие файлы состояния. Она необходима для корректного переключения сервиса между узлами, чтобы новый активный узел мог восстановить состояние NFS (например, уведомить клиентов о переезде сервиса).

NFS-сервер запущен, но порт 2049 всё ещё заблокирован (см. правило 1), поэтому клиенты пока не могут подключиться.

`nfs_ip=<плавающий_ip>` (ip можно выбрать произвольный, но он обязательно должен отличаться от ip адресов, использованных в других внутренних сетях организации для избежания конфликтов, например 192.168.10.199)

IP-адрес, на котором NFS-сервер будет принимать соединения. Это `плавающий_virtual_ip`, который также управляется кластером (отдельным ресурсом `IPaddr2`). NFS-сервер привязывается к этому адресу.

`nfs_server_scope=<плавающий_ip>`

Определяет, на каком интерфейсе или адресе NFS-сервер будет публиковать свои сервисы (через `rpc.nfsd`). Обычно совпадает с `nfs_ip`.

```
– ""ocf:heartbeat:exportfs          exp_nfs_1          directory=/srv/nfs-
share/<имя_ресурса>                  fsid=1d0b09e1-1f61-5ae4-98c1-1661f84a5321
clientspec=<подсеть_клиентов>/24
options='rw,sync,no_subtree_check,no_root_squash'""",
```

Правило выполняет команду:

```
exportfs          -o          rw,sync,no_subtree_check,no_root_squash
<общая_подсеть_клиентов>/24:/srv/nfs-share/<имя_ресурса>
```

fsid: Уникальный идентификатор экспортируемой файловой системы. Критически важно, чтобы он совпадал на обоих узлах кластера и оставался уникальным в пределах одного NFS-сервера. Это обеспечивает корректное функционирование NFS-клиентов и предотвращает потерю доступа к экспортируемой файловой системе при переключении между узлами.

clientspec: Подсеть, из которой разрешен доступ клиентов. Например, 192.168.10.0/24

options: Специфический набор опции NFS.

Для второго ресурса правило будет выглядеть аналогично:

```
""""ocf:heartbeat:exportfs                exp_nfs_2                directory=/srv/nfs-
share/<имя_второго_ресурса>                fsid=2d0b09e2-1f62-5ae4-98c2-2661f84a5322
clientspec=<подсеть_клиентов>/24
options='rw,sync,no_subtree_check,no_root_squash""""',
    - """"ocf:heartbeat:IPaddr2            virtual_ip                cidr_netmask=24
ip=<плавающий_ip> """"',
```

Плавающий IP-адрес активируется только после завершения всех подготовительных этапов, за исключением процедуры разблокировки сетевого порта.

Перезапустите службу drbd-reactor для применения конфигурации:

```
sudo systemctl restart drbd-reactor
```

Проверьте статус drbd-reactor с помощью команды:

```
sudo drbd-reactorctl status
```

Убедитесь, что promoter активен и все сервисы запущены.

Для проверки NFS экспорта выполните команду:

```
sudo exportfs
```

Убедитесь, что экспорт настроен для вашей сети.

Пример вывода команды `sudo drbd-reactorctl status` для primary узла:

```
etc/drbd-reactor.d/nfs.toml:
Promoter: Currently active on this node
● drbd-services@nfs1.target
● └─ drbd-promote@nfs1.service
● └─ ocf.rs@fs_cluster_private_nfs1.service
● └─ ocf.rs@fs_1_nfs1.service
● └─ ocf.rs@nfsserver_nfs1.service
● └─ ocf.rs@exp_nfs_data_1_nfs1.service
● └─ ocf.rs@virtual_ip_nfs1.service
```

Пример вывода команды `sudo drbd-reactorctl status` для secondary узла:

```
/etc/drbd-reactor.d/nfs.toml:
Promoter: Currently active on node 'astra182-stage-stand-v11-cmbnd'
○ drbd-services@nfs1.target
○ └─ drbd-promote@nfs1.service
○ └─ ocf.rs@fs_cluster_private_nfs1.service
○ └─ ocf.rs@fs_1_nfs1.service
○ └─ ocf.rs@nfsserver_nfs1.service
○ └─ ocf.rs@exp_nfs_data_1_nfs1.service
○ └─ ocf.rs@virtual_ip_nfs1.service
```

Протестируйте подключение смонтировав NFS-шару для проверки.

Создайте временный каталог для монтирования экспортируемого ресурса:

```
sudo mkdir -p /mnt/test-nfs
```

Выполните команду для монтирования NFS-шары:

```
sudo mount -t nfs <плавающий_ip>:/srv/nfs-share/<имя_ресурса> /mnt/test-nfs
```

Проверьте содержимое раздела:

```
ls /mnt/test-nfs
```

После проверки можно размонтировать шару:

```
sudo umount /mnt/test-nfs
```

Для отключения экспорта NFS удалите конфигурационный файл.

Перейдите в каталог с конфигурациями drbd-reactor:

```
cd /etc/drbd-reactor.d/
```

Просмотрите список файлов, чтобы найти конфиг, связанный с нужным ресурсом:

```
ls -l
```

Обычно файл имеет имя, связанное с названием ресурса, например: `nfs.toml`, `iscsi_rd2.toml` и т.д.

Удалите соответствующий конфигурационный файл:

```
sudo rm <имя_конфигурации>.toml
```

Перезапустите службу drbd-reactor, чтобы изменения вступили в силу:

```
sudo systemctl restart drbd-reactor
```

23.2. Настройка отказоустойчивого хранилища Samba (на базе drbd-reactor)

Первый этап настройки включает в себя установку необходимых пакетов. Действия необходимо производить на узлах типа воркер.

Для обновления кэша пакетов выполните команду на всех узлах кластера:

```
sudo apt update
```

Установите необходимые программы и компоненты: Samba, DRBD Reactor и OCF-агенты на всех узлах:

```
sudo apt install samba samba-client drbd-reactor resource-agents
```

Поскольку службой Samba будет управлять drbd-reactor, отключите запуск сервиса Samba при включении узла:

```
sudo systemctl disable smbd
```

Остановите сервис Samba:

```
sudo systemctl stop smbd
```

Следующим шагом выполняется настройка поддержки современных версий SMB.

Отредактируйте файл конфигурации Samba на всех узлах:

```
sudo vim /etc/samba/smb.conf
```

Вставьте следующие параметры в секцию [global] (добавьте секцию, если она отсутствует):

```
[global]
  client min protocol = SMB2_02
  server min protocol = SMB2_02
  client max protocol = SMB3_11
  server max protocol = SMB3_11
```

Разбор конфигурации:

– **[global]**

Начало глобального раздела в файле настроек. Всё, что указано ниже, касается всех пользователей и всех ресурсов сервера Samba.

– **client min protocol = SMB2_02**

client min protocol: определение минимального протокола (самого старого), который Samba будет использовать при подключении к другим серверам как «клиент».

SMB2_02: это первая версия современного безопасного протокола SMB2. С данного узла нельзя подключаться (к другому узлу) по старой, небезопасной версии протокола SMB1.

– **server min protocol = SMB2_02**

server min protocol: определение минимального протокола, который Samba принимает от клиентов (других узлов).

SMB2_02: также разрешено подключение с использованием протокола не ниже SMB2.

– **client max protocol = SMB3_11**

client max protocol: максимально новая версия протокола, которую Samba использует для подключения (к другим узлам) как «клиент».

SMB3_11 – одна из последних (и самых защищённых) версий протокола SMB3.

– **server max protocol = SMB3_11**

server max protocol – максимально допустимая версия протокола, если к этому узлу подключаются другие узлы.

SMB3_11 – тоже самая новая поддерживаемая версия.

Добавьте тестовый ресурс общего доступа Samba в конец файла /etc/samba/smb.conf на сервере:

```
[public]
comment = Public Share
path = /var/lib/samba/share
browseable = yes
read only = no
```

```

guest ok = yes
create mask = 0755
directory mask = 0755
force user = nobody
force group = nogroup

```

Разбор конфигурации:

– **[public]**

Блок конфигурации общей (публичной) директории общего доступа.

– **comment = Public Share**

Это описание, которое может отображаться в интерфейсах управления или сетевых списках.

– **path = /var/lib/samba/share**

Путь на диске сервера к каталогу, который будет доступен через этот сетевой общий ресурс. Физическая папка на сервере.

– **browseable = yes**

Параметр, определяющий видимость сетевого ресурса в списке доступных через сеть ресурсов.

– **read only = no**

Запрет на запись отсутствует – можно читать и записывать файлы в этот общий сетевой ресурс.

– **guest ok = yes**

Гостевой доступ разрешён – можно подключаться без входа в систему. При необходимости можно отключить гостевой доступ и ограничить вход только для авторизованных пользователей.

– **create mask = 0755**

Права по умолчанию для создаваемых файлов внутри общего сетевого ресурса: владелец может читать, писать и выполнять, остальные – читать и выполнять.

– **directory mask = 0755**

Права по умолчанию для создаваемых каталогов внутри общего сетевого ресурса: аналогично, владелец имеет полный доступ, остальные – чтение и выполнение.

– **force user = nobody**

Все новые файлы будут принадлежать пользователю nobody. Этот параметр влияет на доступ к файлам, и кто может их редактировать.

– **force group = nogroup**

Все новые файлы будут принадлежать группе nogroup. Этот параметр влияет на права группы и доступ к файлам.

Создайте каталог для общего доступа, где будут храниться файлы, доступные с помощью протокола Samba.

```
sudo mkdir -p /var/lib/samba/share
```

Установите владельца и группу каталога. Для общего доступа чаще используют «nobody» как анонимного пользователя и «nogroup» как группы:

```
sudo chown nobody:nogroup /var/lib/samba/share
```

Если вы хотите, чтобы внутри каталога и все файлы внутри унаследовали этот же владельца/группу, можно применить рекурсивно:

```
sudo chown -R nobody:nogroup /var/lib/samba/share
```

Установите права доступа на каталог:

```
sudo chmod 0777 /var/lib/samba/share
```

0777 – это восьмеричное (octal) представление прав доступа к файлу или каталогу в UNIX-подобных системах. Данная запись читается поразрядно, слева направо:

- Первый разряд (слева) – специальные биты. В случае "0" специальные биты не установлены. Если вместо него будет, например, 1, 2 или 4, то это будет означать установку соответствующих специальных флагов ("4": setuid, "2": setgid, "1": sticky bit).
- Второй разряд – права владельца (user).
- Третий разряд – права группы (group).
- Четвёртый разряд – права остальных пользователей (others).

Каждый из правовых разрядов (кроме первого) формируется как сумма трёх базовых разрешений:

- Чтение (read) – 4
- Запись (write) – 2
- Выполнение (execute) – 1

Получается:

$$7 = 4 \text{ (чтение)} + 2 \text{ (запись)} + 1 \text{ (выполнение)}$$

То есть для владельца, группы и остальных задан полный набор прав, поэтому три последних цифры – 7 7 7.

Полная запись 0777 значит:

"0" – не установлены специальные биты,

"7" – владелец может читать, записывать и запускать (или заходить в каталог),

"7" – группа также может читать, записывать и запускать (или заходить в каталог),

"7" – остальные пользователи имеют те же права.

Полные права (0777) предоставляют максимальный доступ для всех пользователей системы. Для повышения безопасности используют более строгие варианты, например, 0775 (только владелец и группа имеют полный доступ, остальные — только чтение и выполнение) или 0770 (только владелец и группа, остальные — без доступа). Дополнительные ограничения доступа можно настроить в файле конфигурации `smb.conf` для служб Samba (например, параметры `guest ok`, `writable` и др.).

Перезапустите службы Samba, чтобы применить внесённые изменения в конфигурацию:

```
sudo systemctl restart smbd
```

Примечание. На некоторых системах также может потребоваться перезапуск `nmbd` (сетевой сервис NetBIOS):

```
sudo systemctl restart nmbd
```

Проверьте настройки вашего файла конфигурации smb.conf с помощью команды:

```
sudo testparm
```

Если в smb.conf есть ошибки, testparm вернёт сообщение об ошибке, например:

```
ERROR: invalid parameter 'guest ok' in section [share]
```

Подготовьте DRBD ресурсы и миграцию данных Samba.

На одном из узлов переведите DRBD ресурс в первичный (active) режим:

```
sudo drbdadm primary <имя_ресурса>
```

Отформатируйте устройство DRBD в файловую систему XFS (создайте файловую систему).

Введите команду для форматирования:

```
sudo mkfs.xfs /dev/drbd/by-res/<имя_ресурса>/<номер_тома>
```

Внимание. После выполнения этой команды все старые данные на этом устройстве сотрутся.

Подготовьте директорию Samba. Создайте резервную копию данных и новую директорию. Сначала переместите старую папку с данными в «резервную»:

```
sudo mv /var/lib/samba /var/lib/samba.bak
```

Затем создайте новую пустую директорию:

```
sudo mkdir /var/lib/samba
```

Смонтируйте DRBD устройство и восстановите данные из резервной копии:

```
sudo mount /dev/drbd/by-res/<имя_ресурса>/<номер_тома> /var/lib/samba

sudo cp -av /var/lib/samba.bak/* /var/lib/samba/ # Команда копирует в том drbd-ресурса (вместе со скрытыми файлами и правами доступа).
```

Настройте drbd-reactor, выполнив следующую последовательность действий.

На всех узлах создайте файл конфигурации:

```
sudo vim /etc/drbd-reactor.d/<имя_ресурса>.toml
```

Отформатируйте файл, вставив следующий конфигурационный блок:

```
[[promoter]]
id = "<имя_ресурса>"
[promoter.resources.<имя_ресурса>]
on-drbd-demote-failure = "reboot-immediate"
runner = "systemd"
preferred-nodes = []
start = [
    "ocf:heartbeat:Filesystem fs_1 device=/dev/drbd/by-res/<имя_ресурса>/0
directory=/var/lib/samba fstype=xfs run_fsck=no",
    "ocf:heartbeat:IPaddr2 service_ip cidr_netmask=24 ip=<плавающий_ip-адрес>",
    "smbd.service",
]
stop-services-on-exit = true
target-as = "BindsTo"
```

Разбор конфигурации:

– **[promoter]**

Открывает новый блок описания промоутера. Позволяет определить отдельный набор правил для управления ресурсами в кластере. Каждый такой блок организует автоматизацию включения/выключения сервисов и управления ресурсом DRBD на выбранном сервере (узле).

– **id = "<имя_ресурса>"**

Указывает уникальное имя данного блока промоутера, обычно совпадает с именем DRBD-ресурса. Используется для идентификации и связи между конфигурационными секциями и сервисами.

– **[promoter.resources.<имя_ресурса>]**

Этот раздел описывает параметры для управления конкретным ресурсом с заданным именем. Все команды и действия ниже внутри этого блока будут относиться именно к названному ресурсу.

– **on-drbd-demote-failure = "reboot-immediate"**

Если возникла ошибка при попытке «понизить» (перевести в неактивное состояние) DRBD-ресурс, компьютер немедленно перезагрузится. Это помогает избежать проблем, когда одновременно два сервера считают себя «главными».

- **runner** = "systemd"

systemd – программа, которая управляет другими программами/службами в Linux. Запуск и остановка сервисов (Samba, ФС и т.д.) будет через systemd.

- preferred-nodes = []

Список «предпочтительных» серверов, на которых лучше всего держать этот ресурс активным. Пусто – значит, может быть активен на любом сервере.

```
– ""ocf:heartbeat:Filesystem fs_1 device=/dev/drbd/by-
res/<имя_ресурса>/<номер_тома> directory=/srv/lib/samba fstype=xfs
run fsck=no",,
```

Монтирование основной директории данных Samba происходит с использованием DRBD-устройства, которое представляет собой защищённый и реплицируемый диск между всеми узлами кластера.

Параметр `device=/dev/drbd/by-res/<имя_ресурса>/<номер_тома>` определяет путь к DRBD-устройству, связанному с данным ресурсом.

Монтирование выполняется в папку `/var/lib/samba` – стандартное расположение для сервисных и пользовательских данных Samba.

Указывается тип файловой системы `fstype=xfs`, что оптимально для больших файловых хранилищ с поддержкой современных возможностей.

Флаг `run_fsck=no` означает, что проверки целостности файловой системы при монтировании проводиться не будут, что ускоряет процедуру старта в целом.

```
– "ocf:heartbeat:IPaddr2 service_ip cidr_netmask=24 ip=<плавающий_ip-адрес>"
```

Плавающий (виртуальный) IP-адрес активируется на данном узле только после завершения всех обязательных подготовительных процедур: успешного монтирования DRBD-устройства и создания нужных сервисных папок.

Плавающий IP (ip=10.177.161.250) позволяет пользователям и клиентским компьютерам всегда подключаться к одному и тому же адресному объекту, даже если работающий в данный момент сервер меняется.

Параметр `cidr_netmask=24` соответствует стандартной маске для подсети 255.255.255.0.

Этот адрес автоматически переносится между серверами кластера в зависимости от того, какой из них в текущий момент активен («primary»), обеспечивая бесперебойный доступ к файловому ресурсу Samba.

– "smbd.service"

Запуск сервиса Samba (smbd), который делает возможным обмен файлами по сети между компьютерами и этим сервером.

Сервис иницируется только после успешного завершения всех предыдущих этапов: монтирования хранилища и назначения плавающего IP-адреса, что гарантирует правильную доступность всех данных для пользователей.

Служба smbd обеспечивает работу протокола SMB/CIFS и взаимодействие с сетевыми папками.

Последним шагом будет запуск проверка конфигурации и тестирование подключения.

Перезапустите службу drbd-reactor. Выполните команду на всех узлах:

```
sudo drbd-reactorctl restart
```

Проверьте состояние конфигурации и запущенных сервисов (также выполняется на каждом узле):

```
sudo drbd-reactorctl status
```

Ожидаемый вывод:

```
/etc/drbd-reactor.d/<имя_ресурса>.toml:
Promoter: Currently active on this node
● drbd-services@<имя_ресурса>.target
● └─ drbd-promote@<имя_ресурса>.service
● └─ ocf.rs@fs_1_<имя_ресурса>.service
● └─ ocf.rs@service_ip_<имя_ресурса>.service
● └─ smbd.service
```

Проверьте доступность Samba-ресурса, выполнив команду с любого клиента:

```
smbclient -L //<плавающий_ip-адрес> -m SMB3_11 -U%
```

Ожидаемый результат:

Sharename	Type	Comment
print\$	Disk	Printer Drivers
public	Disk	Public Share
IPC\$	IPC	IPC Service (Samba 4.21.4-Debian-4.21.4+dfsg-1ubuntu3.5astra1)
nobody	Disk	Home Directories
SMB1 disabled -- no workgroup available		
Sharename	Type	Comment
print\$	Disk	Printer Drivers
IPC\$	IPC	IPC Service (Samba 4.21.4-Debian-4.21.4+dfsg-1as tra4+b2)
nobody	Disk	Home Directories
SMB1 disabled -- no workgroup available		

Проверка возможности подключения к samba-ресурсу на удаленном хосте:

Установите пакет утилит cifs используя команду:

```
sudo apt install cifs-utils
```

Создайте точку монтирования:

```
sudo mkdir -p /mnt/samba_share
```

Выполните монтирование и проверьте результат:

```
sudo mount -t cifs //<плавающий_ip-адрес>/nobody /mnt/samba_share -o vers=3.11,guest,uid=$(id -u),gid=$(id -g),noperm
```

Набор параметров монтирования:

– vers=3.11 – версия протокола SMB, которая используется при монтировании.

– guest – монтирование выполняется как гость без ввода учётных данных. Если сервер позволяет гостевой доступ (guest ok = yes на сервере), не нужно вводить имя пользователя и пароль.

– uid=\$(id -u) – (подстановка оболочки) устанавливает идентификатор владельца файлов в точке монтирования равным текущему пользователю Linux.

– Этот параметр необходим, чтобы файлы и каталоги, созданные в шаре, отображались как принадлежащие вам, а не «nobody».

– `gid=$(id -g)` – устанавливает идентификатор группы для файлов в точке монтирования равным основной группе текущего пользователя, чтобы групповые права соответствовали вашему пользователю и было проще работать совместно с другими пользователями.

– `noperm` – отключает проверку прав доступа на стороне клиента.

Для проверки монтирования используйте команду:

```
findmnt /mnt/samba_share
```

Она выведет информацию о точке монтирования, если такая существует. Если ничего не найдено, значит монтирование не получилось.

Проверка состояния DRBD ресурсов осуществляется с помощью команды для мониторинга репликации и показывает состояние синхронизации и целостности хранилища:

```
sudo drbdadm status
```

Состояние сервисов под управлением `drbd-reactor` можно проверить, выполнив команду:

```
sudo drbd-reactorctl status
```

23.3. Настройка отказоустойчивого хранилища iSCSI (на базе `drbd-reactor`)

Первый этап настройки включает в себя установку необходимых пакетов.

Для тестирования подключения установите iSCSI инициатор на узлах-воркерах:

```
sudo apt install open-iscsi -y
```

Вторым этапом настройте конфигурации DRBD Reactor для iSCSI, а также конфигурацию iSCSI Initiator (клиентской части).

Создайте или отредактируйте файл конфигурации DRBD Reactor:

```
sudo vim /etc/drbd-reactor.d/iscsi-<имя_ресурса>.toml
```

Вставьте следующий конфигурационный блок iSCSI в файл:

```

[[promoter]]

[promoter.resources]

[promoter.resources.<имя_ресурса>]
on-drbd-demote-failure = "reboot-immediate"
runner = "systemd"
start = [
    "ocf:heartbeat:IPaddr2 service_ip0 cidr_netmask=24 ip=<плавающий_ip-адрес>",
    "ocf:heartbeat:iSCSITarget target allowed_initiators=''"
incoming_password=<пароль_iSCSI> incoming_username=<логин_iSCSI> iqn=iqn.2025-
12.ru.astra:<имя_ресурса> portals=<плавающий_ip-адрес>:3260",
    "ocf:heartbeat:iSCSILogicalUnit lu1 lun=1 path=/dev/drbd/by-
res/<имя_ресурса>/<номер_тома> product_id='TROK iSCSI' scsi_sn=7509fab1
target_iqn=iqn.2025-12.ru.astra:<имя_ресурса>",
]
stop-services-on-exit = true
target-as = "Requires"

```

Разбор конфигурации:

– **[promoter]**

Открывает новый блок описания промоутера. Позволяет определить отдельный набор правил для управления ресурсами в кластере. Каждый такой блок организует автоматизацию включения/выключения сервисов и управления ресурсом DRBD на выбранном сервере (узле).

– **[promoter.resources]**

Секция-«контейнер» для всех ресурсов промоутера. Здесь перечисляются конкретные блоки ресурсов. В данном примере этот раздел пуст, но он предусматривает существование секции [promoter.resources.<имя_ресурса>].

– **[promoter.resources.<имя_ресурса>]**

Описание параметров для управления конкретным ресурсом с заданным именем. Все команды и действия внутри этого блока относятся именно к названному ресурсу. Обратите внимание, что в этом примере явного поля id нет; имя ресурса задаётся в имени секции вместо обозначения <имя_ресурса>.

– **on-drbd-demote-failure = “reboot-immediate”**

Если возникает ошибка при попытке понизить DRBD-ресурс (перевести его в неактивное состояние), система немедленно перезагружается. Это предотвращает ситуацию, когда два узла считают себя «primary» одновременно.

– **runner = “systemd”**

Указывает систему-инициатор управления сервисами. systemd будет запускать/останавливать сервисы и блоки, указанные далее (например, IP-адрес, iSCSI-ресурсы и т.д.) через systemd Unit-файлы и цели.

– **stop-services-on-exit = true**

После завершения работы промоутера (или при выходе из блока) следует остановить запущенные сервисы. Это помогает корректно освободить ресурсы и не держать сервисы в запущенном состоянии при уходе узла из активной роли.

– **start = []**

Перечисление команд в стартовом блоке означает следующие шаги:

– **ocf:heartbeat:IPaddr2 service_ip0 cidr_netmask=24 ip=<плавающий_ip-адрес>**

Плавающий (виртуальный) IP-адрес, который будет активироваться на текущем активном узле. service_ip0 — имя или метка IP-адреса, который будет управляться навыком Failover-кластера. cidr_netmask=24 задаёт сетевую маску 255.255.255.0. Такой IP создаёт единый доступ к ресурсу независимо от того, на каком узле он запущен.

– **ocf:heartbeat:iSCSITarget target allowed_initiators=“incoming_password=12345678 incoming_username=astral iqn=iqn.2025-12.ru.astra:ssd1 portals=<плавающий_ip-адрес>:3260**

iSCSI Target – компонент, который делает доступным iSCSI-ресурс для инициаторов.

target – имя цели (target) внутри iSCSI Target-объекта.

allowed_initiators=“ – список разрешённых инициаторов; пустое значение означает любые инициаторы разрешены, или настройка по умолчанию.

incoming_username/incoming_password – учётные данные для аутентификации.

iqn=iqn.2025-12.ru.astra:<имя_ресурса> – iSCSI Qualified Name цели.

portals=<плавающий_ip-адрес>:3260 – список порталов (IP-адрес/порт) для подключения инициаторов к цели.

– **ocf:heartbeat:iSCSILogicalUnit** **lu1** **lun=1** **path=/dev/drbd/by-res/<имя_ресурса>/<номер_тома>** **product_id='TROK iSCSI'**
scsi_sn=7509fab1 target_iqn=iqn.2025-12.ru.astra:<имя_ресурса>

iSCSI Logical Unit (LU) – логический том, который будет использоваться как SCSI-устройство для инициатора.

lu1 – имя LU внутри целевой конфигурации.

lun=1 – номер LUN внутри этой LU.

path=/dev/drbd/by-res/<имя_ресурса>/<номер_тома> – путь к DRBD-устройству, которое реплицируется между узлами и будет экспортировано как LU.

product_id='TROK iSCSI' – идентификатор продукта для LU (обычно используется для описания на стороне инициатора).

scsi_sn=7509fab1 – SCSI-serial number LU.

target_iqn=iqn.2025-12.ru.astra:<имя_ресурса> – целевой IQN, к которому привязан LU.

– **stop-services-on-exit = true**

Завершение работы промоутера сопровождается остановкой запущенных связанных сервисов и ресурсов, чтобы корректно освободить узел и снизить риск конфликтов или «зацикливания» ресурсов.

– **target-as = “Requires”**

Указывает тип зависимости цели (target) от других сервисов. Значение “Requires” означает, что целевой блок будет активен только если зависимости-сервисы запущены; иначе запуск происходить не будет. Это формирует корректную последовательность запуска: сначала необходимые сервисы, затем целевые ресурсы.

Пример использования конфигурационного блока:

```
[[promoter]]
[promoter.resources]

[promoter.resources.disk1]
on-drbd-demote-failure = "reboot-immediate"
```

```
runner = "systemd"
start = [
    "ocf:heartbeat:IPaddr2 service_ip0 cidr_netmask=24 ip=10.177.161.168",
    "ocf:heartbeat:iscsiTarget target allowed_initiators=''"
incoming_password=12345678 incoming_username=astralqn=iqn.2026-02.ru.astra:disk1
portals=10.177.161.168:3260",
    "ocf:heartbeat:iscsiLogicalUnit lu1 lun=1 path=/dev/drbd/by-res/disk1/0
product_id='TROK iSCSI' scsi_sn=7509fab1 target_iqn=iqn.2026-02.ru.astra:disk1",
]
stop-services-on-exit = true
target-as = "Requires"
```

После внесения изменений перезапустите drbd-reactor командой:

```
sudo systemctl restart drbd-reactor
```

Выполните проверку состояния ресурса:

```
drbd-reactorctl status
```

Пример вывода:

```
/etc/drbd-reactor.d/iscsi-disk1.toml:
Promoter: Currently active on this node
● drbd-services@disk1.target
● └─ drbd-promote@disk1.service
● └─ ocf.rs@service_ip0_disk1.service
● └─ ocf.rs@target_disk1.service
● └─ ocf.rs@lu1_disk1.service
```

Разбор вывода:

– Promoter: Currently active on this node

Узел в настоящий момент выступает промотором (то есть активным узлом, который несёт роль ведущего для данного DRBD-ресурса).

Далее перечислены связанные элементы:

drbd-services@disk1.target – системная единица типа target, предназначенная для управления DRBD-сервисами ресурса disk1.

drbd-promote@disk1.service – сервис, который выполняет продвижение ресурса disk1 на текущем узле (приведение к состоянию primary).

ocf.rs@service_ip0_disk1.service – агент ресурса OCF (Open Cluster Framework) для IP-адреса, связанного с ресурсом disk1.

ocf.rs@target_disk1.service – управления ресурсом DRBD-таргета disk1.

ocf.rs@lu1_disk1.service – агент ресурса для связанного логического устройства (LU) disk1.

Далее следует блок настроек iSCSI на клиентских узлах.

Задайте уникальное имя для идентификации iSCSI инициатора:

```
echo "InitiatorName=iqn.2026-02.ru.astra:<имя_ресурса>" >
/etc/iscsi/initiatorname.iscsi
```

Отредактируйте файл конфигурации iSCSI и укажите учетные данные для настройки аутентификации CHAP:

```
sudo vim /etc/iscsi/iscsid.conf
```

Добавьте или измените следующие строки:

```
node.session.auth.username = <логин_iSCSI>
node.session.auth.password = <пароль_iSCSI>
```

Кандидаты логин/пароль должны соответствовать настройкам iSCSI Target, которые указаны в вашей конфигурации start блока файла iscsi-<имя_ресурса>.toml.

Выполните команду для обнаружения iSCSI Target:

```
sudo iscsiadm -m discovery -t sendtargets -p <плавающий_ip-адрес>
```

Пример успешного вывода:

```
10.177.161.168:3260,1 iqn.2026-02.ru.astra:disk1
```

Выполните команду для подключения к iSCSI Target:

```
sudo iscsiadm -m node --login
```

Пример успешного вывода:

```
Logging in to [iface: default, target: iqn.2026-02.ru.astra:disk1, portal:
10.177.161.168,3260]
```

```
Login to [iface: default, target: iqn.2026-02.ru.astra:disk1, portal:
10.177.161.168,3260] successful.
```

Определите метку диска TROK iSCSI. Введите команду:

```
sudo fdisk -l
```

Пример вывода:

```
Disk /dev/sda: 979,66 MiB, 1027252224 bytes, 2006352 sectors
Disk model: TROK iSCSI
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 1048576 bytes
```

Выполните команду для получения уникального SCSI ID:

```
sudo /usr/lib/udev/scsi_id -g -u /dev/sda # Для примера имя диска из вывода выше
```

Запишите полученный идентификатор (например: 360014057509fab100000000000000000).

Создайте файл правила для автоматического создания символической ссылки:

```
sudo cat > /etc/udev/rules.d/30-change-iscsi-name.rules << EOF
KERNEL=="sd[a-z]", SUBSYSTEM=="block", PROGRAM="/usr/lib/udev/scsi_id -g -u
/dev/\$name", RESULT=="360014057509fab10000000000000000", SYMLINK+="iscsi/disk_1"
EOF
```

Перезагрузите правила udev для немедленного применения:

```
sudo udevadm control --reload-rules
```

Отключитесь от iSCSI Target (при необходимости). Для удаления подключения выполните следующие команды:

```
iscsiadm -m node -p <плавающий_ip-адрес>:3260,0 -T iqn.2025-12.ru.astra:<имя_ресурса> --logout

iscsiadm -m node -p <плавающий_ip-адрес>:3260,0 -T iqn.2025-12.ru.astra:<имя_ресурса> -o delete
```

Пример команды с указанными данными:

```
sudo iscsiadm -m node -p 10.177.161.168:3260,0 -T iqn.2026-02.ru.astra:disk1 --logout

sudo iscsiadm -m node -p 10.177.161.168:3260,0 -T iqn.2026-02.ru.astra:disk1 -o delete
```

Пример успешного вывода команд:

```
Logging out of session [sid: 1, target: iqn.2026-02.ru.astra:disk1, portal:
10.177.161.168,3260]
```

```
Logout of [sid: 1, target: iqn.2026-02.ru.astra:disk1, portal:
10.177.161.168,3260] successful.
```

На финальном этапе следует настроить проверку и мониторинг.

Проверьте статус drbd-reactor. Убедитесь, что конфигурация загружена и ресурс активен:

```
sudo drbd-reactorctl status
```

Пример успешного вывода:

```
/etc/drbd-reactor.d/iscsi-disk1.toml:
Promoter: Currently active on this node
● drbd-services@disk1.target
● └─ drbd-promote@disk1.service
● └─ ocf.rs@service_ip0_disk1.service
● └─ ocf.rs@target_disk1.service
● └─ ocf.rs@lu1_disk1.service
```

Восстановите подключение к таргету:

```
sudo iscsiadm -m discovery -t sendtargets -p <плавающий_ip-адрес>

sudo iscsiadm -m node --login
```

Проверьте доступность iSCSI Target. С клиентского узла убедитесь, что диск доступен по символической ссылке:

```
sudo ls -la /dev/iscsi/<имя_ресурса>
```

Пример вывода команды:

```
lrwxrwxrwx 1 root root 6 map 2 11:40 /dev/iscsi/disk_1 -> ../sda
```

24. МОНИТОРИНГ PROMETHEUS – КОНТРОЛЛЕРОВ

Программное обеспечение предоставляет возможности для интеграции контроллеров с системой мониторинга Prometheus. В случае, если контроллер запущен, он автоматически экспортирует метрики в формате Prometheus, и данные могут быть извлечены посредством запроса к конечной точке `http://controller's-ip/metrics`.

24.1. Метрика `trok_info`

Метрика `trok_info` предоставляет базовую информацию о версии и сборке контроллера и содержит метки:

- `getid` – уникальный идентификатор сборки (хеш коммита Git);
- `buildtime` – время сборки программного обеспечения;
- `version` – версия программного обеспечения (семантическое версионирование).

Пример значения:

```
trok_info{getid="372c916b7d97fa10e8ea480b66ea3da665ab5849",buildtime="2025-10-13T14:22:02+03:00",version="1.29.2"} 1
```

Интерпретация:

Значение всегда равно 1 – подтверждает доступность метрик. Используется для проверки работоспособности экспортера и позволяет отслеживать версии в гетерогенных средах.

24.2. Метрика `trok_node_state`

Метрика `trok_node_state` отслеживает состояние узлов в кластере и имеет следующие коды состояний:

- 0="OFFLINE" – узел недоступен;
- 1="ONLINE" – узел работает нормально;
- 2="CONNECTING" – выполняется подключение к узлу;

- 3="HANDSHAKING" – выполняется обмен служебной информацией между узлами;
- 4="UNSUPPORTED_DRBD_VERSION" – обнаружена неподдерживаемая версия DRBD;
- 5="DRBD_TOOLS_ERROR" – произошла ошибка в работе инструментов DRBD;
- 6="EVICTED" – узел исключен из кластера.

Метки:

- node – имя узла в кластере;
- nodetype – тип узла (satellite или controller);
- address – IP-адрес узла;
- port – порт для подключения;
- encryption – тип шифрования (PLAIN = без шифрования).

Пример значения:

```
trok_node_state{node="nodeA",nodetype="SATELLITE",address="10.177.161.43",port="50002",encryption="PLAIN"} 1
```

24.3. Метрика trok_storage_pool_capacity_free_bytes

Метрика trok_storage_pool_capacity_free_bytes отображает свободное место в пулах хранения и содержат следующие метки:

- storage_pool – имя пула хранения;
- node – UUID узла, на котором расположен пул;
- driver – тип драйвера хранилища:
 - LVM – классический LVM;
 - LVM_THIN – тонкий LVM;
 - DISKLESS – бездисковый пул.

Пример значения:

```
trok_storage_pool_capacity_free_bytes{storage_pool="PoolA",node="0acfd509-93a6-4504-9df9-5f078095f42b",driver="LVM"} 1234000
```

```
trok_storage_pool_capacity_free_bytes{storage_pool="PoolB",node="0acfd509-93a6-4504-9df9-5f078095f42b",driver="LVM_THIN"} 1234000
```

24.4. Метрика trok_storage_pool_capacity_total_bytes

Метрика trok_storage_pool_capacity_total_bytes отображает общую емкость пулов хранения и использует набор меток аналогичный пункту 1.3.

Пример значения:

```
trok_storage_pool_capacity_total_bytes{storage_pool="PoolA",node="0acfd509-93a6-4504-9df9-5f078095f42b",driver="LVM"} 1236000
```

```
trok_storage_pool_capacity_total_bytes{storage_pool="PoolB",node="0acfd509-93a6-4504-9df9-5f078095f42b",driver="LVM_THIN"} 1236000
```

24.5. Метрика trok_storage_pool_error_count

Метрика trok_storage_pool_error_count является счетчиком ошибок в пулах хранения и использует метки аналогичные предыдущим метрикам пулов.

Пример значения:

```
trok_storage_pool_error_count{storage_pool="PoolA",node="0acfd509-93a6-4504-9df9-5f078095f42b",driver="LVM"} 0
```

```
trok_storage_pool_error_count{storage_pool="PoolB",node="0acfd509-93a6-4504-9df9-5f078095f42b",driver="LVM_THIN"} 0
```

Интерпретация:

- 0 – ошибок нет;
- >0 – имеются проблемы с пулом.

Используется для уведомлений о проблемах хранилища.

24.6. Метрика trok_resource_state

Метрика trok_resource_state отображает состояние ресурса на каждом узле кластера.

Тип метрики:

```
# TYPE trok_resource_state gauge
```

Пример значения:

```
trok_resource_state{name="e4953a7e-bdcd-4258-b87b-439064621737",node="28945d49-0188-42d8-b12c-463dec711272"} 0
```

```
trok_resource_state{name="d560722f-692b-4c01-91aa-151066c5c0e2",node="335bb340-fac0-4f6a-b774-20bf940b4379"} 0
```

```
trok_resource_state{name="2658b15a-31c5-407f-be2c-824bd17c3c9d",node="c41c403a-ced5-4faa-8407-3be6aa1e9ee4"} 0
```

Значения метрики:

- 0 – ресурс в нормальном состоянии (Ok);
- 1 – предупреждение (WARNING);
- 2 – ошибка (ERROR).

25. МОНИТОРИНГ PROMETHEUS – DRBD REACTOR

Настройка DRBD Reactor для экспорта метрик Prometheus так же проста. Следует разместить следующий файл конфигурации в директории конфигураций DRBD Reactor и перезапустить DRBD Reactor:

```
# cat << EOF > /etc/drbd-reactor.d/prometheus.toml
[[prometheus]]
enums = true
address = "0.0.0.0:9942"
EOF

# systemctl restart drbd-reactor.service
```

После перезапуска вы сможете отправить curl-запрос на конечную точку метрик и просмотреть метрики DRBD, предоставляемые DRBD Reactor.

```
# curl 127.0.0.1:9942
# TYPE drbd_device_alwrites_total counter
# HELP Number of updates of the activity log area of the meta data
drbd_device_alwrites_total{name="trok_db",volume="0",minor="1000"} 1
# TYPE drbd_resource_resources gauge
# HELP Number of resources
drbd_resource_resources 1
...snip...
```

26. СТРУКТУРА И РАСПОЛОЖЕНИЕ ЛОГ-ФАЙЛОВ

Настройка уровня логирования производится в конфигурационном файле соответствующего сервиса.

Важно. Для применения внесённых изменений требуется выполнить перезапуск данного сервиса.

Конфигурационные файлы располагаются в каталоге `/etc/<имя-сервиса>/`. Например, для открытия конфигурационного файла `trok-controller` можно воспользоваться следующей командой:

```
sudo vim /etc/trok-controller/config.yaml
```

В программном обеспечении используется несколько уровней подробности записываемых сообщений:

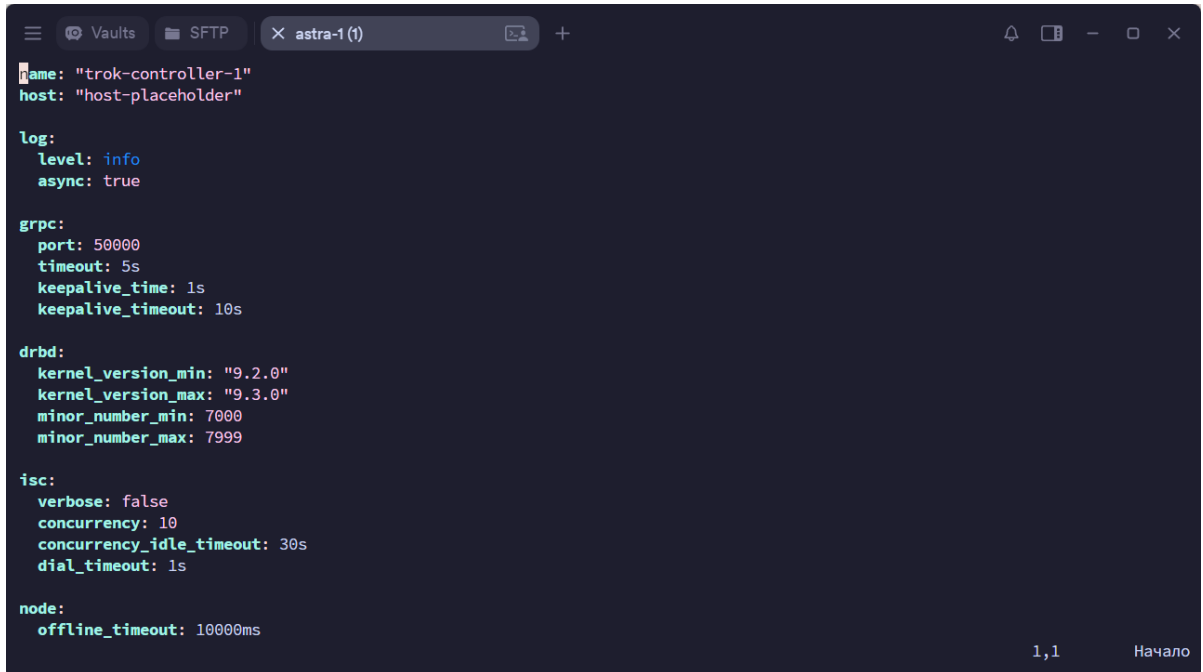
- Первый уровень – **ERROR** – фиксируются ошибки критического характера, которые приводят к нарушению работы системы или отдельных её компонентов. Сообщения этого уровня указывают на события, требующие немедленного вмешательства для восстановления штатной работы.

- Второй уровень – **WARN** – регистрируются предупреждения о потенциальных проблемах или аномалиях, которые пока не мешают функционированию системы, но могут впоследствии привести к ошибкам. Уровень позволяет своевременно выявлять и устранять возможные риски.

- Третий уровень – **INFO** – отображаются информационные сообщения о штатных событиях и процессах в системе. Этот уровень используется для отслеживания нормальной работы сервисов, фиксации ключевых этапов выполнения операций и общего контроля.

- Четвертый уровень – **DEBUG** – предоставляется наиболее подробная диагностическая информация, предназначенная главным образом для разработчиков и администраторов при отладке и глубоком анализе работы приложения. Сообщения этого уровня обычно включают подробные данные о выполнении программного кода.

Каждый последующий уровень логирования включает в себя сообщения всех предыдущих, менее подробных уровней. По умолчанию уровень логирования, заданный в конфигурационных файлах, устанавливается в значение INFO.



```

name: "trok-controller-1"
host: "host-placeholder"

log:
  level: info
  async: true

grpc:
  port: 50000
  timeout: 5s
  keepalive_time: 1s
  keepalive_timeout: 10s

drbd:
  kernel_version_min: "9.2.0"
  kernel_version_max: "9.3.0"
  minor_number_min: 7000
  minor_number_max: 7999

isc:
  verbose: false
  concurrency: 10
  concurrency_idle_timeout: 30s
  dial_timeout: 1s

node:
  offline_timeout: 10000ms
  
```

Рисунок 57 – Настройка уровня логирования в конфигурационном файле

Логи системы разделены по функциональным компонентам и хранятся в директории /var/log/trok/. Представленная структура облегчает поиск, анализ и ротацию записей, относящихся к конкретному модулю.

Структура каталогов по умолчанию:

```

/var/log/trok/
├── trok-auth/                                # Логи модуля аутентификации и авторизации
│   ├── trok-auth.log                        # Текущий активный лог-файл
│   └── trok-auth.log-2025-12-24            # Архивный лог-файл за указанную дату
├── trok-controller/                          # Логи основного управляющего контроллера
│   ├── trok-controller.log
│   └── trok-controller.log-2025-12-24
├── trok-cp-endpoint/                        # Логи Control Plane API
│   ├── trok-cp-endpoint.log
│   └── trok-cp-endpoint.log-2025-12-24
├── trok-installer/                          # Логи установки и обновления системы
│   └── install_20251124_131523.log          # Уникальный файл на каждую сессию установки
├── trok-worker/                             # Логи службы worker
│   ├── trok-worker.log
│   ├── trok-worker.log-2025-12-24.gz      # Сжатый архивный лог
│   └── trok-worker.log-2025-12-25
  
```

К числу основных событий, фиксируемых в журналах (логах), относятся следующие:

- Уведомление о запуске сервиса контроллера с идентификацией версии ПО:

```
INFO Starting trok-controller service {"version": "20260217.1403", "hash": "7b127e8"}
```

Это описание обозначает запуск службы контроллера, указывая её версию и хэш-коммит исходного кода.

При обращении в службу поддержки укажите полученные в данные. Это позволит специалистам оперативно идентифицировать используемую версию системы и ускорит процесс диагностики и устранения проблемы.

- Уведомление о подключении к основной базе данных сервиса:

```
opening storage... {"storage path": "/var/lib/trok-controller/data.db"}
```

В случае утраты базы данных вместо данного сообщения будет отображаться уведомление об ошибке подключения.

- Уведомление о запуске менеджера узлов, отвечающего за управление кластерными нодами:

```
starting nodes manager... {"nodes manager": "nodes manager"}
```

Начинается обработка состояния и взаимодействия между узлами.

- Уведомление об успешной инициализации:

```
initialization finished, loaded 5 nodes id 958 us {"nodes manager": "nodes manager"}
```

Сообщение на примере говорит об успешном завершении инициализации, в ходе которой было загружено 5 узлов за 958 микросекунд. Кластерные узлы успешно обнаружены и готовы к использованию.

- Уведомление об ошибке инициализации менеджера узлов:

```
error while initializing node manager
```

Система не смогла корректно загрузить информацию об узлах, необходима диагностика.

- Уведомление о запуске gRPC сервера:

```
Starting grpc server...
```

gRPC сервер отвечает за обработку удалённых вызовов и взаимодействие с клиентами/узлами.

- Уведомление о готовности к работе gRPC сервера:

```
grpc server started
```

– Уведомление об успешном корректном добавлении узла и регистрации его в кластере:

```
handshake response for hostname: astra17-dev1-sat2, drbd kernel version: 9.2.10
{"uuid": "36d0b11b-cd9b-4cc0-81df-5c45596799e1", "name": "node2"}
```

На примере кластерный контроллер подтверждает успешное установление соединения с узлом astra17-dev1-sat2, сообщая его DRBD-версию и уникальный идентификатор.

– Уведомление об установке канала связи для передачи событий между сервером и клиентом:

```
opening event stream for client dns:///192.168.161.45:50002 {"Events": ""}
```

Открытие потоковой передачи событий (event stream) для подключённого клиента по указанному адресу.

- Событие о состоянии ресурса:

```
got ResourceStateEvent, resource uuid: 2a116e74-a097-49af-be7c-bc251aecf45a,
role: Secondary {"Events": ""}
```

Указывается уникальный идентификатор ресурса и его роль (здесь – Secondary, то есть пассивная копия).

- Событие изменения состояния тома:

```
got DiskStateEvent, resource uuid: 2a116e74-a097-49af-be7c-bc251aecf45a, node
uuid: 8e14bed2-5ce0-4386-965f-3cf0e383ddf8, volume: 0, state: UpToDate {"Events":
""}
```

В примере указывается, что на определённом узле для конкретного ресурса и соответствующего тома изменено состояние на «UpToDate» (реплика данных синхронизирована).

– Событие изменения состояния соединения между двумя узлами кластера для конкретного ресурса:

```
got ConnectionStateEvent, resource uuid: 2a116e74-a097-49af-be7c-bc251aecf45a,
node from uuid: 8e14bed2-5ce0-4386-965f-3cf0e383ddf8, node to uuid: astra17-dev1-
sat2, state: Connected {"Events": ""}
```

В примере сообщение об установке соединения между двумя узлами (указаны уникальные идентификаторы и статус соединения).

– Уведомление о получении и регистрации вызова API-метода ListNode (список узлов) с заданными параметрами:

```
grpc method ListNode called {"reqID": "82377301-8553-41b5-8e34-cde6c2db6565",
"offset": 0, "limit": -1}
```

Внешний клиент или процесс запрашивает информацию о списке узлов.

– Уведомление о регистрации вызова API-метода ViewRscs, предназначенного для просмотра ресурсов кластера с определёнными фильтрами:

```
grpc method ViewRscs called {"reqID": "8f62a44a-9ac8-4e0e-b36d-dd0f79744911",
"filter by nodes": [], "filter by resources": [], "filter by storage pools": [],
"filter by props": [], "offset": 0, "limit": 10}
```

27. НАСТРОЙКА ВЫСОКОДОСТУПНОГО КЛАСТЕРА

Перед созданием шаблона ресурса убедитесь, что диски подготовлены и размечены в соответствии с инструкциями, описанными в главах 3–5 настоящего документа, а также все узлы кластера развернуты в комбинированной конфигурации, совмещающей роли контроллера и рабочего узла (controller + worker). Для этого необходимо выполнить первоначальную настройку физических томов, создать группы томов и логические тома, а также настроить DRBD-ресурсы. Только после завершения этих подготовительных этапов переходите к созданию шаблона ресурса в веб-интерфейсе.

Перейдите в веб-интерфейс и создайте новый шаблон ресурса, например, с именем ha1. В открывшейся форме укажите имя шаблона (ha1) и сохраните изменения.

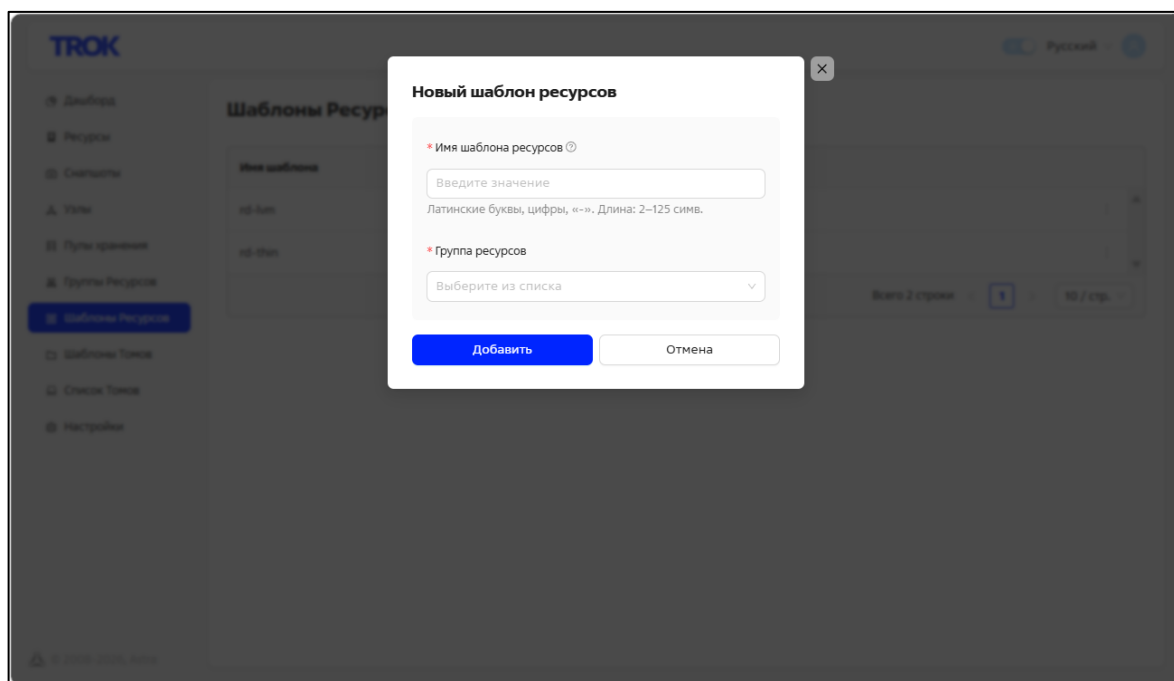


Рисунок 58 – Пример создания шаблона ресурса

Затем необходимо создать два шаблона тома для созданного ресурса ha1. В разделе «Шаблоны томов» нажмите кнопку «Добавить» и создайте сначала первый шаблон тома, привязав его к шаблону ресурса ha1.

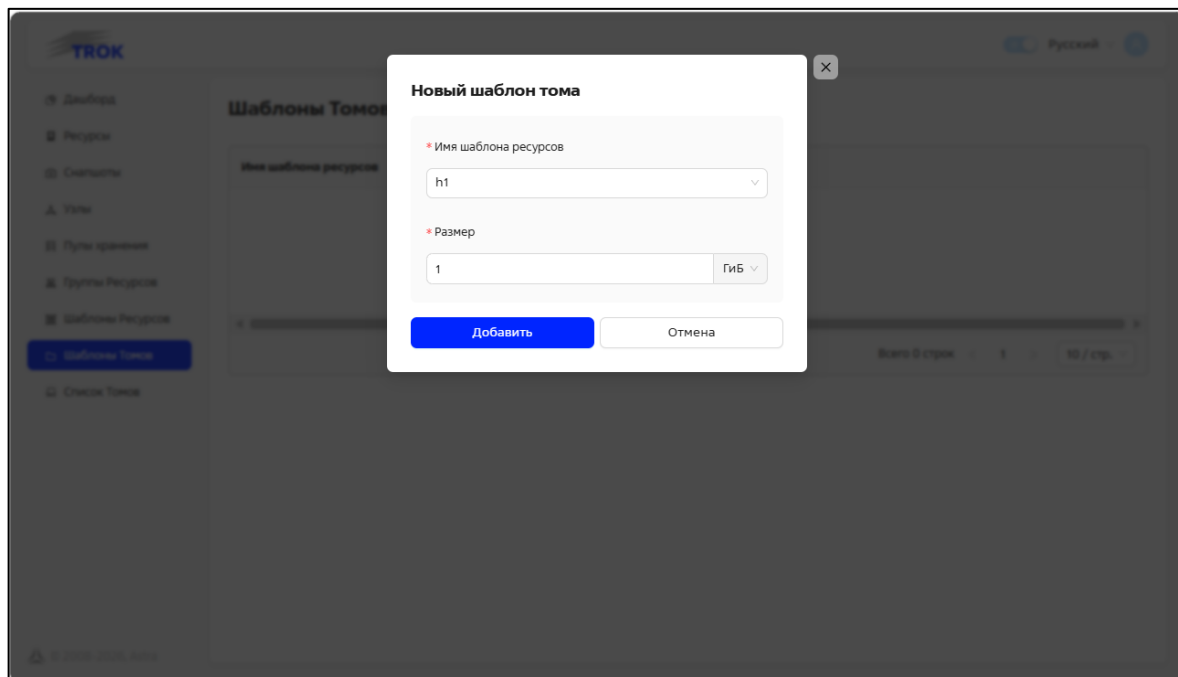


Рисунок 59 – Пример создания шаблона тома

Укажите размер тома и подтвердите создание. Повторите аналогичные действия для второго тома с номером 1.

В результате на вкладке «Шаблоны томов» должны отобразиться две записи, соответствующие созданному шаблону ресурса ha1 с номерами томов 0 и 1. Наличие двух томов подтверждается соответствующей информацией в интерфейсе.

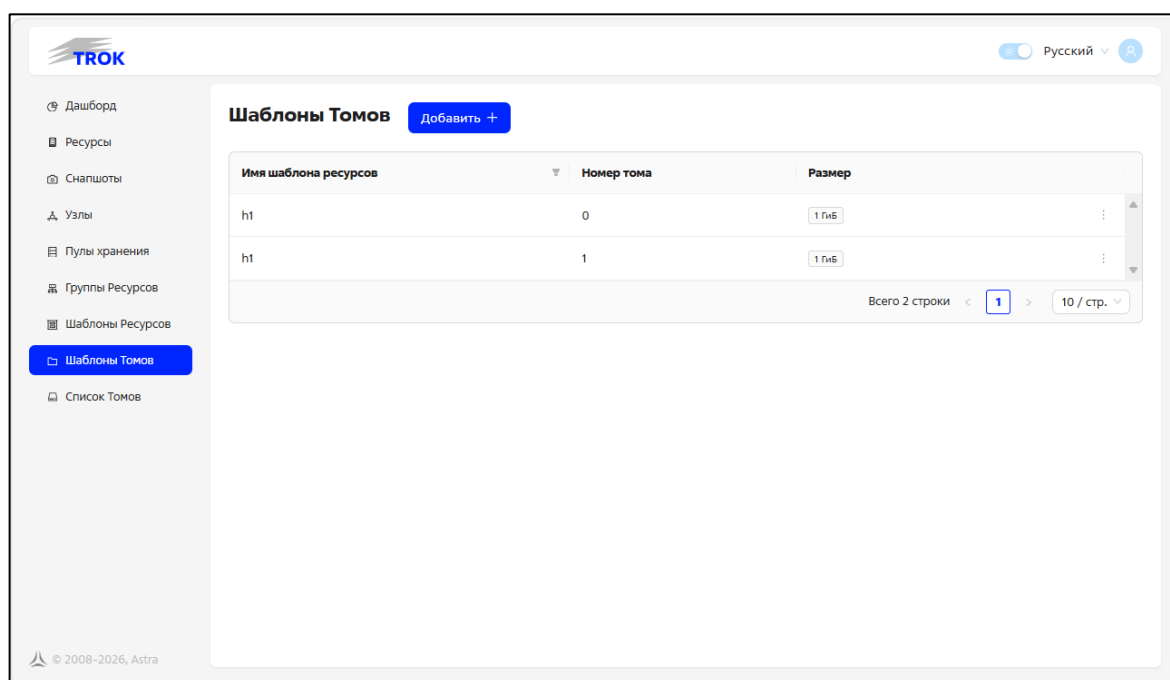


Рисунок 60 – Два созданных шаблона тома (номер тома присваивается автоматически)

На основании созданного шаблона ресурса ha1 создайте сами ресурсы, выбрав подготовленный шаблона ресурса. При создании вы можете оставить включенным параметр «Автоматический выбор узла хранения» либо вручную указать узлы или пулы хранения в соответствии с политикой вашей инфраструктуры. После заполнения полей и подтверждения ресурсы отобразятся в интерфейсе не мгновенно – сначала система покажет сообщение о том, что инициализация создания ресурса запущена, и только по окончании процесса ресурсы станут доступны для просмотра и управления. Дождитесь перехода ресурсов в состояние UpToDate, что свидетельствует о готовности к использованию.

27.1. Перенос баз TROK на тома DRBD

Для переноса файлов баз данных на тома созданного DRBD-ресурса ha1 выполните следующие шаги.

На первом узле из списка кластера переведите DRBD-ресурс ha1 в активное состояние (primary), чтобы сделать данные доступными для записи:

```
sudo drbdadm primary ha1
```

Определите имена томов, входящих в состав ресурса ha1, с помощью команды:

```
sudo lsblk
```

Утилита lsblk отображает древовидную структуру блочных устройств, показывая имена, размеры и точки монтирования.

Пример вывода:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPPOINTS
...						
sdc	8:32	0	1G	0	disk	
├─test_vg-ha1_07000	253:0	0	1G	0	lvm	
│ └─drbd7000	147:7000	0	1G	0	disk	
├─test_vg-ha1_07001	253:4	0	1G	0	lvm	
│ └─drbd7001	147:7001	0	1G	0	disk	

В выводе найдите устройства, соответствующие ресурсу ha1, например, drbd7000 и drbd7001.

Создайте файловую систему ext4 на каждом из найденных томов. Для первого тома:

```
sudo mkfs.ext4 /dev/drbd7000
```

И для второго:

```
sudo mkfs.ext4 /dev/drbd7001
```

Команда `mkfs.ext4` форматирует указанное устройство в файловую систему ext4, подготавливая его к хранению файлов.

Остановите сервисы `trok-controller`:

```
sudo systemctl stop trok-controller
```

и `trok-auth`:

```
sudo systemctl stop trok-auth
```

а также отключите их автоматический запуск при загрузке системы:

```
sudo systemctl disable trok-controller
```

```
sudo systemctl disable trok-auth
```

Команда `systemctl stop` завершает работу указанной службы, а `systemctl disable` удаляет её из автозагрузки, чтобы предотвратить конфликты во время переноса данных.

Примонтируйте тома ресурса `ha1` в соответствующие каталоги, предварительно переместив существующие базы данных во временное хранилище. Сначала переместите базу данных контроллера из рабочей папки во временное хранилище `/opt`, чтобы освободить каталог для монтирования.

```
sudo mv /var/lib/trok-controller/data.db /opt
```

Затем примонтируйте первый том DRBD-ресурса (устройство `/dev/drbd7000`) в каталог контроллера, подключая внешнее хранилище в нужную точку.

```
sudo mount /dev/drbd7000 /var/lib/trok-controller
```

Верните базу данных из временного хранилища обратно в каталог контроллера (но теперь она физически сохраняется на примонтированном DRBD-томе).

```
sudo mv /opt/data.db /var/lib/trok-controller
```

Переместите базу данных сервиса аутентификации из рабочей папки во временное хранилище /opt, освобождая каталог для монтирования.

```
sudo mv /var/lib/trok-auth/data.db /opt
```

Смонтируйте второй том DRBD-ресурса (устройство /dev/drbd7001) в каталог сервиса аутентификации, подключая внешнее хранилище.

```
sudo mount /dev/drbd7001 /var/lib/trok-auth
```

Верните базу данных аутентификации из временного хранилища обратно в каталог сервиса, теперь она сохраняется на втором DRBD-томе.

```
sudo mv /opt/data.db /var/lib/trok-auth
```

Установите верные права доступа и владельца для каталогов с базами данных:

Команда:

```
sudo chmod -R 755 /var/lib/trok-controller /var/lib/trok-auth
```

устанавливает права доступа 755 (владелец может читать, писать и выполнять; группа и остальные – только читать и выполнять) рекурсивно для всех файлов и подкаталогов в папках контроллера и аутентификации.

В качестве владельца и группы используйте пользователя trok рекурсивно для всех файлов и подкаталогов в указанных папках, чтобы сервисы TROK имели доступ к своим данным.

```
sudo chown trok:trok -R /var/lib/trok-controller /var/lib/trok-auth
```

Проверьте, что тома успешно примонтированы, выполнив команду:

```
sudo mount
```

Утилита `mount` без аргументов отображает список всех смонтированных файловых систем. В выводе должны присутствовать строки, указывающие на примонтированные устройства `/dev/drbd7000` и `/dev/drbd7001` с соответствующими точками монтирования.

Отмонтируйте тома DRBD-ресурса `ha1`, чтобы подготовить их к переключению на другой узел:

```
sudo umount /var/lib/trok-controller
sudo umount /var/lib/trok-auth
```

После выполнения команд устройство освобождено для дальнейших операций.

Переведите DRBD-ресурс `ha1` в пассивное состояние (`secondary`):

```
sudo drbdadm secondary ha1
```

Теперь ресурс доступен только для чтения и подготовлен к переключению на другой узел кластера.

27.2. Создание конфигурации `drbd-reactor (promoter)` для ресурса `ha1`

Для настройки автоматического управления ресурсами создайте конфигурационный файл `drbd-reactor` для ресурса `ha1`. В этом файле описывается, какие действия и сервисы должны быть запущены, когда на данном узле ресурс переходит в активное состояние. Выполните команду:

```
sudo cat << EOF > /etc/drbd-reactor.d/trok-controller.toml
[[promoter]]
id = "ha1"
[promoter.resources.ha1]
start = [
  "ocf:heartbeat:Filesystem ha_1 device=/dev/drbd/by-res/ha1/0
directory=/var/lib/trok-controller fstype=ext4 run_fsck=no",
  "ocf:heartbeat:Filesystem ha_2 device=/dev/drbd/by-res/ha1/1
directory=/var/lib/trok-auth fstype=ext4 run_fsck=no",
  "ocf:heartbeat:IPaddr2 virtual_ip cidr_netmask=24 ip=192.168.0.125",
  "trok-auth.service",
  "trok-controller.service",
]
```

EOF

Рассмотрим подробнее, что содержит эта конфигурация.

- **[[promoter]]**

Секция, которая определяет агент-промоутер, отвечающий за управление ресурсами.

- **id = "ha1"**

Уникальный идентификатор промоутера, соответствующий имени DRBD-ресурса.

- **[promoter.resources.ha1]**

Секция, описывающая ресурсы, которые должны быть запущены на узле, когда ресурс ha1 становится активным (primary).

- **start = [**

Список команд и сервисов, которые будут выполнены последовательно при активации ресурса. Рассмотрим каждый элемент списка:

- **"ocf:heartbeat:Filesystem ha_1 device=/dev/drbd/by-res/ha1/0
directory=/var/lib/trok-controller fstype=ext4 run_fsck=no"**

Это OCF-ресурс (Open Cluster Framework), который монтирует файловую систему. Он использует скрипт Filesystem из набора heartbeat. Аргументы:

ha_1 – уникальное имя ресурса.

device=/dev/drbd/by-res/ha1/0 – путь к блочному устройству первого тома DRBD-ресурса ha1.

directory=/var/lib/trok-controller – точка монтирования, куда будет примонтировано устройство.

fstype=ext4 – тип файловой системы.

run_fsck=no – отключает проверку файловой системы при монтировании, что ускоряет запуск.

- **"ocf:heartbeat:Filesystem ha_2 device=/dev/drbd/by-res/ha1/1
directory=/var/lib/trok-auth fstype=ext4 run_fsck=no"**

Аналогичный ресурс для второго тома DRBD-ресурса ha1. Отличие:

ha_2 – имя ресурса.

device=/dev/drbd/by-res/ha1/1 – путь ко второму тому.

directory=/var/lib/trok-auth – точка монтирования для сервиса аутентификации.

– **"ocf:heartbeat:IPaddr2 virtual_ip cidr_netmask=24 ip=192.168.0.125"**

OCF-ресурс для управления виртуальным IP-адресом. Аргументы:

virtual_ip – имя ресурса.

cidr_netmask=24 – маска подсети (в данном случае /24, что соответствует 255.255.255.0).

ip=192.168.0.125 – сам виртуальный IP-адрес, который будет привязан к активному узлу. Это позволяет клиентам обращаться к кластеру по одному фиксированному адресу независимо от того, на каком физическом узле работают сервисы.

– **"trok-auth.service"**

Запуск системной службы аутентификации TROK.

– **"trok-controller.service"**

Запуск системной службы контроллера TROK.

После сохранения конфигурационного файла перезагрузите процесс drbd-reactor, чтобы он применил новые настройки:

```
sudo systemctl restart drbd-reactor
```

Команда перезапускает указанную службу, в результате чего она перечитывает все конфигурационные файлы из каталога /etc/drbd-reactor.d/ и начинает управлять ресурсами в соответствии с новыми правилами.

Если после перезапуска drbd-reactor изменения не вступили в силу, выполните полную перезагрузку узла:

```
sudo reboot
```

Перезагрузка системы обеспечит повторную инициализацию всех системных демонов с загрузкой актуальных конфигураций. Используйте её только при необходимости и с учётом временной недоступности служб.

27.3. Проверка состояния drbd-reactor

Для проверки состояния drbd-reactor выполните команду, которая отображает текущий статус управления ресурсами и список активных служб:

```
sudo drbd-reactorctl status
```

Подкоманда status показывает информацию о загруженных конфигурациях и состоянии ресурсов, которыми управляет drbd-reactor. В выводе вы увидите следующую информацию:

```
/etc/drbd-reactor.d/trok-controller.toml:
Promoter: Currently active on this node
○ drbd-services@ha1.target
● └─ drbd-promote@ha1.service
● └─ ocf.rs@ha_1_ha1.service
● └─ ocf.rs@ha_2_ha1.service
● └─ ocf.rs@virtual_ip_ha1.service
● └─ trok-auth.service
● └─ trok-controller.service
```

– **/etc/drbd-reactor.d/trok-controller.toml:**

Указывает, какой конфигурационный файл загружен и используется для управления ресурсами. В данном случае это созданный ранее файл для ресурса ha1.

– **Promoter: Currently active on this node**

Означает, что данный узел является активным промоутером, то есть именно на нём выполняются все действия по запуску и управлению ресурсами, описанными в конфигурации.

Далее приводится список служб и ресурсов, управляемых drbd-reactor:

– ○ **drbd-services@ha1.target**

Целевая группа (target), объединяющая все ресурсы, связанные с DRBD-ресурсом ha1. Символ ○ означает, что это группа, а не отдельная служба.

– ● └─ **drbd-promote@ha1.service**

Служба, которая переводит DRBD-ресурс ha1 в активное состояние (primary). Символ ● перед службой означает, что она успешно запущена и активна.

– ● └─ **ocf.rs@ha_1_ha1.service**

Служба, управляющая первым томом DRBD-ресурса (монтирование файловой системы в /var/lib/trok-controller). ocf.rs указывает на то, что это OCF-ресурс. Служба активна и работает.

– ● |— **ocf.rs@ha_2_ha1.service**

Служба, управляющая вторым томом DRBD-ресурса (монтирование файловой системы в /var/lib/trok-auth). Также активна.

– ● |— **ocf.rs@virtual_ip_ha1.service**

Служба, отвечающая за настройку виртуального IP-адреса 192.168.0.125, чтобы обеспечить доступ к кластеру по постоянному адресу. Активна.

– ● |— **trok-auth.service**

Служба аутентификации TROK, которая была запущена в соответствии с конфигурацией. Символ ● подтверждает, что она активна и работает корректно.

– ● |— **trok-controller.service**

Служба контроллера TROK, также активна и запущена на данном узле.

Внимательно проверьте вывод команды: все службы, перечисленные в выводе, должны иметь символ ● (или зелёный цвет), что означает их активное состояние. Если какая-либо служба не запущена, это указывает на проблему, которую необходимо устранить для корректной работы кластера.

27.4. Настройка остальных узлов

Для настройки второго и третьего узлов кластера выполните аналогичные действия.

На каждом из оставшихся узлов остановите службы контроллера и аутентификации, чтобы они не мешали переносу данных:

```
sudo systemctl stop trok-controller
sudo systemctl stop trok-auth
```

Отключите автоматический запуск этих служб (команда `systemctl disable` удаляет службы из автозагрузки) при загрузке системы, чтобы они не стартовали до тех пор, пока не будут полностью настроены:

```
sudo systemctl disable trok-controller
sudo systemctl disable trok-auth
```

Удалите старые файлы баз данных контроллера и сервиса аутентификации, которые остались от предыдущих установок:

```
sudo rm -f /var/lib/trok-controller/data.db
sudo rm -f /var/lib/trok-auth/data.db
```

Команда `rm -f` принудительно удаляет указанные файлы без запроса подтверждения. Флаг `-f` (force) подавляет сообщения об ошибках, если файлы не существуют. Это необходимо, чтобы на узлах не осталось локальных копий баз данных, которые могут конфликтовать с данными на общем DRBD-хранилище.

Создайте конфигурационный файл `drbd-reactor` для ресурса `ha1` на текущем узле. Эта конфигурация описывает, какие ресурсы и службы должны быть запущены, когда узел становится активным:

```
sudo cat << EOF > /etc/drbd-reactor.d/trok-controller.toml
[[promoter]]
id = "ha1"
[promoter.resources.ha1]
start = [
  "ocf:heartbeat:Filesystem          ha_1          device=/dev/drbd/by-res/ha1/0
directory=/var/lib/trok-controller fstype=ext4 run_fsck=no",
  "ocf:heartbeat:Filesystem          ha_2          device=/dev/drbd/by-res/ha1/1
directory=/var/lib/trok-auth fstype=ext4 run_fsck=no",
  "ocf:heartbeat:IPaddr2 virtual_ip cidr_netmask=24 ip=192.168.0.125",
  "trok-auth.service",
  "trok-controller.service",
]
EOF
```

Подробное описание каждого параметра приведено в предыдущем разделе при создании конфигурации на первом узле.

Перезапустите процесс `drbd-reactor`, чтобы он применил новую конфигурацию:

```
sudo systemctl restart drbd-reactor
```

Команда `systemctl restart` перезапускает службу `drbd-reactor`, заставляя её перечитать все конфигурационные файлы из каталога `/etc/drbd-reactor.d/` и начать управлять ресурсами согласно новым настройкам.

Проверьте статус `drbd-reactor` на настроенном узле, чтобы убедиться, что конфигурация загружена корректно и все службы готовы к запуску:

```
sudo drbd-reactorctl status
```

В выводе должна отображаться загруженная конфигурация и список служб, которые будут активированы при переходе узла в состояние `primary`.

После настройки всех узлов перезагрузите первую ноду, чтобы инициировать переключение активного состояния на другой узел:

```
sudo reboot
```

Команда выполняет перезагрузку системы, что приводит к передаче роли `primary` другому узлу кластера в соответствии с настройками DRBD.

После перезагрузки проверьте состояние DRBD-ресурса `ha1` на любом из оставшихся узлов, чтобы убедиться, что роль `primary` успешно перешла на другой сервер:

```
sudo drbdadm status
```

В выводе вы должны увидеть, что на одном из узлов (например, `astra-1` или `astra-3`) ресурс имеет роль `Primary`, а его диски находятся в состоянии `UpToDate`, что означает полную синхронизацию данных. Пример вывода:

```
ha1 role:Primary
  volume:0 disk:UpToDate
  volume:1 disk:UpToDate
astra-1 role:Secondary
  volume:0 peer-disk:UpToDate
  volume:1 peer-disk:UpToDate
astra-3 role:Secondary
  volume:0 peer-disk:UpToDate
  volume:1 peer-disk:UpToDate
```

В этом примере текущий узел является primary, а два других узла – secondary, с полностью синхронизированными данными.

Наконец, проверьте доступность веб-интерфейса TROK по плавающему IP-адресу, который был настроен в конфигурации drbd-reactor. Откройте адрес, который вы ранее назначили в конфигурации, в браузере. Если все настройки выполнены корректно, вы должны увидеть страницу входа в WEBUI портал TROK, что подтверждает успешную работу кластера и доступность сервисов. Пример ввода:

```
http://192.168.0.125
```

28. ДОСТУП ПО ПРОТОКОЛУ S3 К ХРАНИЛИЩУ SDS – ОБЩИЕ СВЕДЕНИЯ

Использование стандартизированного REST-API обеспечивает интеграцию решений для резервного копирования, аналитики и мультимедиа с единым объектным хранилищем без установки дополнительных драйверов и без сложных процедур настройки.

SDS — наиболее удобная основа для S3, поскольку программно отделяет функции хранения от аппаратуры, обеспечивая независимость от поставщика, горизонтальное масштабирование и богатый набор сервисов данных.

Служба trok-s3-gw (S3-совместимый шлюз) разворачивается как системный сервис на каждом узле хранения, имеющем локальную реплику соответствующего DRBD-ресурса.

Для корректного функционирования trok-s3-gw в составе кластера рекомендуется предварительно создать один или несколько DRBD-томов. Том данных – содержит пользовательские объекты, метаданные IAM (учётные записи, ключи доступа) и каталог для версионирования. Допускается использование одного тома с логическим разделением по каталогам либо выделение отдельных томов для каждой категории; в типовой конфигурации предполагается единый общий том.

Минимально рекомендуемый объём тома определяется планируемым объёмом пользовательских данных; метаданные IAM занимают незначительный объём.

Все DRBD-ресурсы должны быть заранее созданы и полностью синхронизированы между всеми узлами кластера.

28.1. Архитектура интеграции S3 и SDS

Все три узла кластера являются равноправными шлюзовыми (gateway) узлами и в равной степени могут стать владельцами ресурса. На одном из них, а именно на узле <узел_шлюза_1>, размещён контроллер TROK (оркестратор DRBD). Однако это размещение не наделяет данный узел статусом выделенного

владельца S3-сервиса. Активный владелец (active owner) определяется автоматически на основе алгоритмов выбора и может оказаться на любом из трёх узлов. Резервный (spare) узел в данной конфигурации не предусмотрен: DRBD-ресурс развёрнут на всех трёх узлах одновременно (параметр place-count = 3), что обеспечивает тройную избыточность без выделения отдельного узла для горячего резерва.



Рисунок 61 – Модель взаимодействия подсистем S3 и SDS

29. НАСТРОЙКА S3-ШЛЮЗА НА БАЗЕ DRBD И DRBD-REACTOR

29.1. Подготовка к интеграции S3 и SDS

Базовые условия:

- Ресурсы предварительно созданы с использованием TROK, выступающего в роли DRBD-оркестратора.

- На каждом узле развернуты и функционируют службы Satellite.

- На одном или нескольких узлах развернута и активна служба Controller.

- Доступны и запущены утилиты уровня узла (инструменты контроля состояния и диагностики).

- drbdadm – управление и статус DRBD-ресурсов.

- drbd-reactorctl – управление наблюдателями и политиками drbd-reactor.

- lvm – управление физическими, групповыми и логическими томами.

- Обеспечены доступы и сетевая связность (адресация, маршрутизация, открытые порты); подготовлена требуемая дисковая разметка/конфигурация.

Перед началом работ обратите внимание: следующие системные имена и пути являются фиксированными и не подлежат изменению:

- trok-s3-gw@.service – шаблон системного юнита шлюза; экземпляры именуются по имени ресурса.

- /etc/trok-s3-gw.d/ – каталог конфигурации шлюза.

- /etc/drbd-reactor.d/ – каталог конфигурации drbd-reactor.

- ocf:heartbeat:Filesystem – агент OCF для монтирования файловой системы.

- ocf:heartbeat:IPaddr2 – агент OCF для управления виртуальным IP-адресом.

Убедитесь, что у вас есть SSH-доступ с правами администратора (root/sudo) ко всем трём узлам. На каждом узле откройте и проверьте доступность следующих TCP-портов:

Таблица 2 – Порты, используемые для настройки

Порт	Описание
------	----------

22/tcp	SSH
7000–7999/tcp	DRBD-репликация (в процессе настройки выделяется по одному порту на каждый DRBD-ресурс)
7070/tcp	S3 API
7080/tcp	Admin API
7071/tcp	WebUI S3
50000/tcp	сервис TROK Controller
50002/tcp	сервис TROK Worker
80/tcp	Веб-сервер (Apache) с HTTP
443/tcp/	HTTPS (веб-сервер Apache) с SSL/TLS
2049/tcp	NFS (при использовании NFS-транспорта; при необходимости могут потребоваться дополнительные порты, например rpcbind/mountd)
3260/tcp	iSCSI (при использовании iSCSI-транспорта)
445/tcp 139/tcp	SMB/CIFS (Samba; при использовании устаревшего NetBIOS могут дополнительно потребоваться 137/udp, 138/udp, 139/tcp)
4420/tcp	NVMe-oF (NVMe/TCP)

29.2. Предварительная настройка операционной системы

В первую очередь проверьте синхронизацию системного времени. Даже небольшое рассогласование часов вызывает ошибки подписи S3 (clock skew) и нарушает согласованность кластера. Настройте NTP на всех узлах с помощью chrony и проверьте состояние синхронизации.

Откройте терминал с правами администратора. Обновите список доступных пакетов и установите chrony – службу синхронизации системного времени (если она ранее не была установлена):

```
sudo apt-get update && sudo apt-get install -y chrony
```

Запустите службу и включите ее автозапуск:

```
sudo systemctl enable --now chrony
```

Проверьте сводку состояния синхронизации:

```
chronyc tracking
```

Просмотрите список источников времени и их статус:

```
chronyc sources
```

Синхронизация в Chrony происходит полностью автоматически в фоновом режиме. Для принудительной настройки можете использовать команду:

```
sudo chronyc -a makestep
```

Важно. В результате выполнения команды время может «прыгнуть» назад/вперёд – это влияет на таймеры, логи и некоторые сервисы. Лучше выполнять в окно обслуживания.

Проверьте, запущена ли служба drbd. Если служба не активна, выполните её запуск:

```
sudo systemctl is-active --quiet drbd || sudo systemctl start drbd
```

При необходимости просмотрите подробный статус службы для подтверждения корректного запуска:

```
sudo systemctl status drbd --no-pager -l
```

Проверьте, включена ли служба в автозагрузку. Если автозапуск не активирован, включите его:

```
sudo systemctl is-enabled --quiet drbd || sudo systemctl enable drbd
```

Подтвердите, что служба активна и включена в автозагрузку:

```
sudo systemctl is-active drbd
sudo systemctl is-enabled drbd
```

29.3. Подготовка файловой системы

Ниже представлена последовательность шагов по переводу DRBD-ресурса, содержащего данные для trok-s3-gw, в состояние Primary, созданию файловой системы и подготовке структуры каталогов. При использовании нескольких ресурсов повторите операции для каждого.

Откройте терминал с правами администратора. Проверьте список DRBD-ресурсов и их текущее состояние:

```
sudo drbdadm status
```

Пример вывода:

```
<имя_ресурса> role:Secondary
disk:UpToDate
<имя_узла_2> role:Secondary
peer-disk:UpToDate
<имя_узла_3> role:Secondary
peer-disk:UpToDate
```

Переведите ресурс в состояние Primary на узле, где будет выполняться форматирование:

```
sudo drbdadm primary <имя_ресурса>
```

Убедитесь, что роль ресурса изменилась на Primary:

```
sudo drbdadm status
```

Создайте файловую систему на томе DRBD-ресурса (операция выполняется один раз на узле с ролью Primary):

```
sudo mkfs -t ext4 /dev/drbd/by-res/<имя_ресурса>/<номер_тома>
```

Здесь <имя_ресурса> – имя DRBD-ресурса.

Важно: создание файловой системы удалит все данные на выбранном томе.

На каждом узле кластера создайте каталог для монтирования DRBD-тома:

```
sudo mkdir -p /mnt/<имя_ресурса>/<номер_тома>
```

Смонтируйте том на активном узле (где ресурс в состоянии Primary):

```
sudo mount /dev/drbd/by-res/<имя_ресурса>/<номер_тома> /mnt/<имя_ресурса>/<номер_тома>
```

Создайте внутри тома структуру каталогов, соответствующую конфигурации: data, versioning и iam. Если <директория_монтирования> – это /mnt/<имя_ресурса>/<номер_тома>, выполните команду:

```
sudo mkdir -p <директория_монтирования>/data  
<директория_монтирования>/versioning <директория_монтирования>/iam
```

Установите права доступа на созданные каталоги:

```
sudo chmod 0777 <директория_монтирования>/data  
<директория_монтирования>/versioning <директория_монтирования>/iam
```

После создания структуры каталогов размонтируйте том:

```
sudo umount /mnt/<имя_ресурса>/<номер_тома>
```

Переведите ресурс обратно в состояние Secondary (если не планируете сразу настраивать drbd-reactor):

```
sudo drbdadm secondary <имя_ресурса>
```

29.4. Настройка drbd-reactor для trok-s3-gw

Посмотреть содержимое шаблона unit-файла сервиса trok-s3-gw:

```
systemctl cat trok-s3-gw@.service
```

Шаблон юнита systemd (unit template) устанавливается вместе с пакетом; ручное создание unit-файла не требуется. Вручную создаётся только конфигурационный файл экземпляра (instance).

Заполните файл `/etc/trok-s3-gw.d/<имя_экземпляра>.conf`, который должен присутствовать на всех gateway-узлах, актуальной информацией. Используйте команду:

```
sudo tee /etc/trok-s3-gw.d/<имя_экземпляра>.conf >/dev/null <<EOF
VGW_BACKEND=posix
VGW_BACKEND_ARG=<директория_монтирования>/data
VGW_VERSIONING_DIR=<директория_монтирования>/versioning
VGW_PORT=<плавающий_виртуальный_IP_кластера>:7070
VGW_ADMIN_PORT=<плавающий_виртуальный_IP_кластера>:7080
VGW_WEBUI_PORT=<плавающий_виртуальный_IP_кластера>:7071
VGW_WEBUI_NO_TLS=true
VGW_HEALTH=/health
VGW_REGION=<регион_S3>
VGW_CORS_ALLOW_ORIGIN=http://<плавающий_виртуальный_IP_кластера>:7071
VGW_IAM_DIR=<STATE_ROOT>/iam
ROOT_ACCESS_KEY_ID=<имя_пользователя>
ROOT_SECRET_ACCESS_KEY=<секрет>
EOF
```

Выберите подходящее значение переменной `VGW_REGION`; в качестве примера можно использовать `ru-central-1`.

`VGW_WEBUI_NO_TLS=true` отключает TLS для WebUI. Используйте это только в изолированном окружении.

Для эксплуатации выберите один из вариантов:

Первый вариант – терминировать TLS на самом шлюзе (и для S3 на `<плавающий_виртуальный_IP_кластера>:7070`, и для WebUI):

Укажите пути к сертификату и ключам:

- `VGW_CERT=/etc/ssl/certs/your-cert.pem`;
- `VGW_KEY=/etc/ssl/private/your-key.pem`;
- `VGW_ADMIN_CERT=/etc/ssl/certs/your-cert.pem`;
- `VGW_ADMIN_CERT_KEY=/etc/ssl/private/your-key.pem`.

Уберите из конфига строку `VGW_WEBUI_NO_TLS=true`.

Задайте HTTPS в `VGW_CORS_ALLOW_ORIGIN` (например, `https://IP:7071`).

Пример с локально выпущенным самоподписанным сертификатом:

```
sudo mkdir -p /etc/ssl/certs /etc/ssl/private
sudo openssl req -x509 -newkey rsa:2048 -nodes \
-keyout /etc/ssl/private/your-key.pem \
```

```
-out /etc/ssl/certs/your-cert.pem \
-days 365 -subj "/CN=RU"
```

Пример конфигурации:

```
sudo tee /etc/trok-s3-gw.d/default.conf >/dev/null <<EOF
VGW_BACKEND=posix
VGW_BACKEND_ARG=/mnt/res1/0/data
VGW_VERSIONING_DIR=/mnt/res1/0/versioning
VGW_PORT=192.168.0.167:7070
VGW_ADMIN_PORT=192.168.0.167:7080
VGW_WEBUI_PORT=192.168.0.167:7071
VGW_HEALTH=/health
VGW_REGION=ru-central-1
VGW_CORS_ALLOW_ORIGIN=https://192.168.0.167:7071
VGW_IAM_DIR=/mnt/res1/0/iam
ROOT_ACCESS_KEY_ID=test
ROOT_SECRET_ACCESS_KEY=test
VGW_CERT=/etc/ssl/certs/your-cert.pem
VGW_KEY=/etc/ssl/private/your-key.pem
VGW_ADMIN_CERT=/etc/ssl/certs/your-cert.pem
VGW_ADMIN_CERT_KEY=/etc/ssl/private/your-key.pem
EOF
```

Второй вариант – завершить TLS на reverse-прокси перед <плавающий_виртуальный_IP_кластера>:

Настройте прокси на HTTPS и замените http:// на https:// в VGW_CORS_ALLOW_ORIGIN и во всех клиентских проверках/URL.

Смените владельца файла на root и группу на root:

```
sudo chown root:root /etc/trok-s3-gw.d/<имя_экземпляра>.conf
```

Установите режим доступа 0640 (осьмерично: gw-r-----): владельцу – чтение и запись; группе – только чтение; прочим пользователям – никаких прав.

```
sudo chmod 0640 /etc/trok-s3-gw.d/<имя_экземпляра>.conf
```

Один и тот же файл /etc/trok-s3-gw.d/<имя_экземпляра>.conf должен присутствовать на всех gateway-узлах.

После создания перезагрузите конфигурацию systemd:

```
sudo systemctl daemon-reload
```

Проверьте ручной запуск сервиса с указанием имени экземпляра (в примере default) на активном узле после монтирования тома:

```

sudo mount /dev/drbd/by-res/<имя_ресурса>/<номер_тома>
/mnt/<имя_ресурса>/<номер_тома>

sudo systemctl start trok-s3-gw@default.service

sudo systemctl status trok-s3-gw@default.service

```

Остановите сервис и размонтируйте том перед настройкой drbd-reactor:

```

sudo systemctl stop trok-s3-gw@<имя_конфигурации>

sudo umount /mnt/drbd

```

Создайте конфигурационный файл промоутера на каждом узле:

```

sudo touch /etc/drbd-reactor.d/trok-s3-gw.toml

```

Откройте файл и добавьте следующую конфигурацию (подставьте свои имена ресурсов, номера томов, плавающий IP и т.д.):

```

[[promoter]]
id = "trok-s3-gw"

[promoter.resources]

[promoter.resources.<имя_ресурса>]
  on-drbd-demote-failure = "reboot-immediate"
  runner = "systemd"
  stop-services-on-exit = true

  start = [
    "ocf:heartbeat:Filesystem          fs_state          device=/dev/drbd/by-
res/<имя_ресурса>/<номер_тома>          directory=/mnt/<имя_ресурса>/<номер_тома>
fstype=ext4",
    "ocf:heartbeat:IPaddr2      vip      ip=<плавающий_виртуальный_IP_кластера>
cidr_netmask=24",
    "trok-s3-gw@<имя_экземпляра>.service"
  ]

  on-drbd-demote-failure = "reboot"

```

Разбор конфигурации:

- **[[promoter]] id = "trok-s3-gw"**

Открывает новый блок описания промоутера с идентификатором trok-s3-gw (используется в логах/метриках и позволяет иметь несколько независимых промоутеров). Позволяет определить набор правил и действий, которые drbd-reactor будет выполнять при изменении роли ресурса (promote/demote).

– **[promoter.resources]**

Блок описания DRBD-ресурсов, которыми управляет данный промоутер.

– **[promoter.resources.<имя_ресурса>]**

Блок настроек конкретного DRBD-ресурса. Вместо <имя_ресурса> укажите реальное имя ресурса, для которого настраиваются действия при переходах ролей.

– **on-drbd-demote-failure = "reboot-immediate"**

Если при переключении ресурса DRBD из Primary в Secondary (demote) произойдёт ошибка, узел будет немедленно перезагружён для предотвращения split-brain и восстановления кластерного состояния.

– **runner = "systemd"**

Определяет механизм управления прикладными службами и OCF-ресурсами. Значение systemd означает, что запуск/остановка и мониторинг сервисов выполняются стандартным менеджером systemd.

– **stop-services-on-exit = true**

При остановке/выходе промоутера (или при демоуте ресурса) он останавливает все ранее запущенные сервисы и ресурсы, обеспечивая корректную уборку.

– **start = [...]**

Массив действий, выполняемых автоматически при промоуте ресурса в Primary. Порядок критичен: сначала готовится файловая система и точки монтирования, затем запускается прикладной сервис, после чего (опционально) поднимается плавающий IP.

– **""ocf:heartbeat:Filesystem <имя_fs_ресурса> device=/dev/drbd/by-res/<имя_ресурса>/<номер_тома> directory=/mnt/<имя_ресурса>/<номер_тома> fstype=ext4 run_fsck=no""**,

Монтирует том DRBD по стабильному пути /dev/drbd/by-res/<имя_ресурса>/<номер_тома> в /mnt/drbd с типом файловой системы ext4.

– ""

ocf:heartbeat:IPaddr2**vip****ip=<плавающий_виртуальный_IP_кластера> cidr_netmask=24""**,

Поднимает плавающий IP-адрес для клиентского доступа к сервису. Этот адрес управляется кластером и всегда активен только на текущем узле Primary. Убедитесь, что указаны корректные сеть/маска, адрес свободен и маршрутизация настроена на всех узлах одинаково.

– "" **trok-s3-gw@<имя_экземпляра>.service**""

Запускает экземпляр сервиса S3-шлюза через systemd.

– **on-drbd-demote-failure = "reboot"**

При ошибке демоута DRBD (перехода из Primary в Secondary) узел выполняет штатную перезагрузку (reboot) для предотвращения split-brain и восстановления кластера.

Запустите сервис drbd-reactor и включите автозапуск при старте ОС:

```
systemctl enable --now drbd-reactor
```

Проверьте текущее состояние сервиса (без перехода в постраничный просмотр):

```
systemctl status drbd-reactor --no-pager
```

Проверьте статус инстанса промотора с id “trok-s3-gw”; если инстанс ещё не создан/не запущен и команда вернёт ошибку, оператор || true подавит её (скрипт продолжит выполнение):

```
sudo drbd-reactorctl status trok-s3-gw || true
```

29.5. Проверка работоспособности

После первоначальной настройки убедитесь в корректной работе DRBD-ресурса и S3-шлюза. Выполните последовательность действий, описанную ниже.

Для подтверждения статуса Primary и наличия смонтированного тома выполните команду:

```
mount | grep /mnt/drbd
```

После выполнения команда должна вернуть строку с точкой монтирования /mnt/drbd.

Для проверки работы сервиса trok-s3-gw и прослушивания портов 7070 (S3) и 7071 (WebUI) выполните команду:

```
ss -tulpn | grep -E '7070|7071'
```

После выполнения в выводе должны отображаться процессы, слушающие указанные порты.

Для проверки доступности S3-API через шлюз выполните команду:

```
aws s3 --endpoint-url http://192.168.10.199:7070 ls
```

После выполнения в выводе должен появиться список бакетов или пустой результат без ошибки.

Откройте в браузере адрес <http://192.168.10.199:7071> (при наличии). Убедитесь, что страница WebUI загружается без ошибок.

Для проверки отказоустойчивости смоделируйте отказ активного узла. Используйте один из вариантов:

- Выключите активный узел.
- Переведите ресурс в состояние Secondary вручную.

После выполнения убедитесь в корректном переключении:

- Второй узел автоматически становится Primary.
- Том монтируется на втором узле.
- Сервис trok-s3-gw запускается и слушает порты 7070 и 7071.
- Плавающий IP 192.168.10.199 переходит на второй узел.

30. ОБЩИЕ СВЕДЕНИЯ О S3 UI (ГРАФИЧЕСКИЙ ИНТЕРФЕЙС S3)

S3-хранилища представляют собой современное решение для хранения, управления и обмена файлами в облачной инфраструктуре. В основе хранения лежит организация данных в виде контейнеров, называемых «бакетами» (buckets), каждый из которых предназначен для упорядоченного и безопасного размещения файлов – объектов.

Назначение S3-хранилищ

Основная задача S3-хранилища – централизованное хранение файлов и данных пользователя в защищённой среде с возможностью быстрого доступа из любой точки сети. S3 применяется для размещения резервных копий, хранения пользовательских данных, мультимедийных файлов, архивной информации и статического контента сайтов, а также в качестве среды для интеграции с различными сервисами и приложениями. Масштабируемость и надёжность S3 позволяют легко адаптировать хранилище под нужды как небольшого бизнеса, так и крупных компаний.

Способы применения

S3-хранилища широко используются для разнообразных задач:

- Бэкап (резервное копирование) и архивирование корпоративных данных.
- Организация доступных библиотек фото, видео или других медиафайлов.
- Хранение документов для совместной работы и обмена между сотрудниками.
- Размещение статических данных сайтов и мобильных приложений.
- Интеграция с аналитическими и другими внешними сервисами для автоматической загрузки и получения информации.

Версионирование в бакетах

Одной из ключевых функций S3-хранилищ является поддержка версионирования файлов (versioning). При включении данной опции, любые изменения или удаление файлов не приводят к их безвозвратной потере: система хранит все версии объекта, что позволяет в любой момент восстановить

предыдущее состояние файла или случайно удалённый документ. Версионирование играет важную роль для:

- Защиты данных от случайного или несанкционированного удаления/изменения.
- Контроля истории изменений и возможности возврата к ранее сохранённым версиям.
- Организации бизнес-процессов, предусматривающих обработку документов с отслеживанием их редактирования.

Основные элементы пользовательского интерфейса системы управления S3-хранилищем включают:

Панель навигации

Располагается с левой стороны экрана и содержит ссылки на ключевые разделы:

- Dashboard (Главная панель управления) – служит для обзора состояния системы, предоставляет информацию о пользователях и бакетах, содержит индикатор статуса подключения.
- Users (Пользователи) – предназначен для управления учетными записями пользователей.
- Buckets (Бакеты) – управление контейнерами для хранения данных.
- Explorer (Проводник) – инструмент для навигации по содержимому бакетов, просмотра и работы с файлами.

Главная рабочая область

Центральная часть окна динамически отображает содержимое выбранного раздела.

Панель действий и фильтров

В верхней части основной рабочей области размещены элементы для создания новых файлов и папок, включения версионирования, а также кнопки загрузки данных. Фильтрация и поиск обеспечивают быстрый доступ к нужным объектам по имени или параметрам.

31. ПРОЦЕДУРА АВТОРИЗАЦИИ ДЛЯ S3 UI

После ввода URL-адреса, предназначенного для доступа к интерфейсу (в формате `http://<имя_хоста>:7071`), в адресной строке браузера открывается стартовая страница входа в систему.

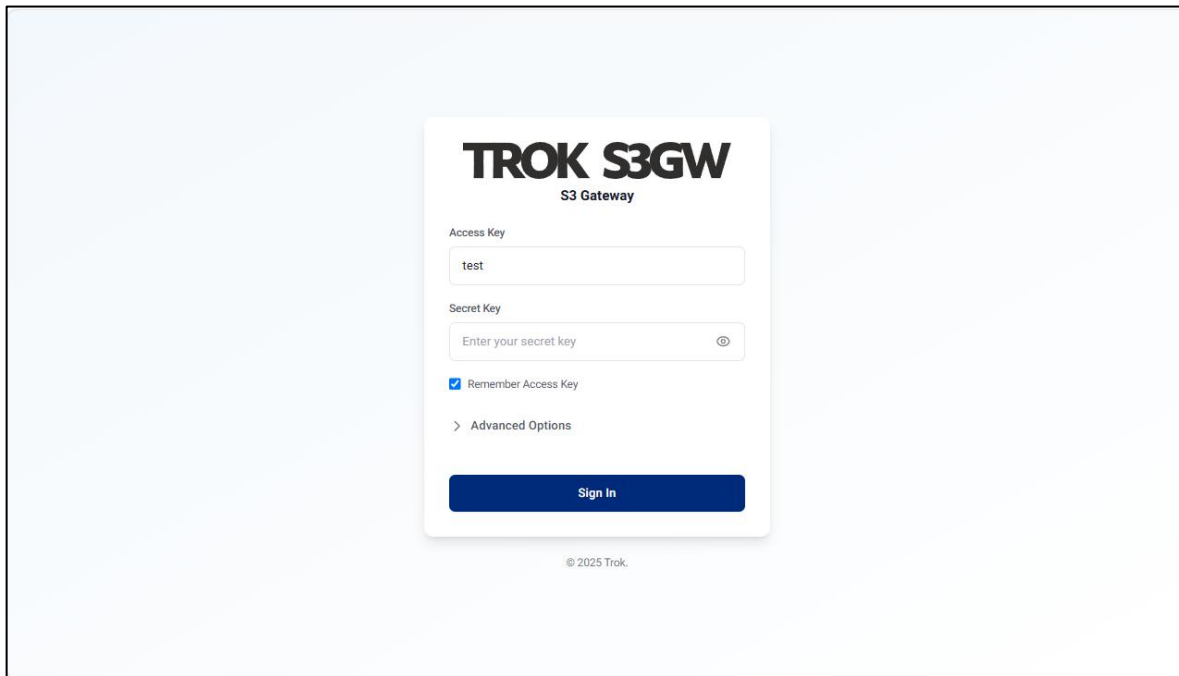


Рисунок 62 – Страница авторизации

Для авторизации на панели управления TROK S3 введите в поле «Ключ доступа» (Access Key) значение, заданное администратором или указанное при запуске сервиса, а в поле «Секретный ключ» (Secret Key) – соответствующий секретный пароль. Дополнительно можно отметить чекбокс «Запомнить ключ доступа» для сохранения используемых данных на этом устройстве. Для продолжения нажмите кнопку «Войти» (Sign In).

После успешного ввода ключа доступа и секретного ключа автоматически открывается главная панель управления системой.

При необходимости в раскрывающемся разделе «Расширенные настройки» (Advanced Options) можно указать дополнительные параметры подключения (например, адрес точки доступа S3 API, регион или вариант адресации бакетов).

The screenshot displays the 'S3 Gateway' configuration window. It features several input fields and a 'Sign In' button. The 'Access Key' field contains the text 'test'. The 'Secret Key' field has a placeholder 'Enter your secret key' and a toggle icon. Below it, the 'Remember Access Key' checkbox is checked. The 'Advanced Options' section is expanded, showing 'S3 API Endpoint' set to 'http://10.177.161.64:7070', 'Admin API Endpoint' set to 'http://10.177.161.64:7080', and 'Region' set to 'ru-central-1'. At the bottom, the 'Bucket Addressing Style' is set to 'Path Style' (indicated by a toggle switch). A 'Sign In' button is located at the bottom of the form. The footer of the window shows '© 2025 Trok'.

Рисунок 63 – Раздел «Расширенные настройки»

Примечание

Для первичной авторизации в системе ключи доступа и секретные ключи для администратора должны быть заранее заданы через переменные окружения при запуске сервиса администратором (например, `ROOT_ACCESS_KEY`, `ROOT_SECRET_KEY` для POSIX-бэкенда или `VGW_S3_ACCESS_KEY`, `VGW_S3_SECRET_KEY` для S3-бэкенда).

32. ПАНЕЛЬ УПРАВЛЕНИЯ S3 UI (ДАШБОРД)

После успешной авторизации открывается главная панель управления (Dashboard). Она предназначена для быстрого обзора состояния системы и перехода к основным операциям управления пользователями и бакетами.

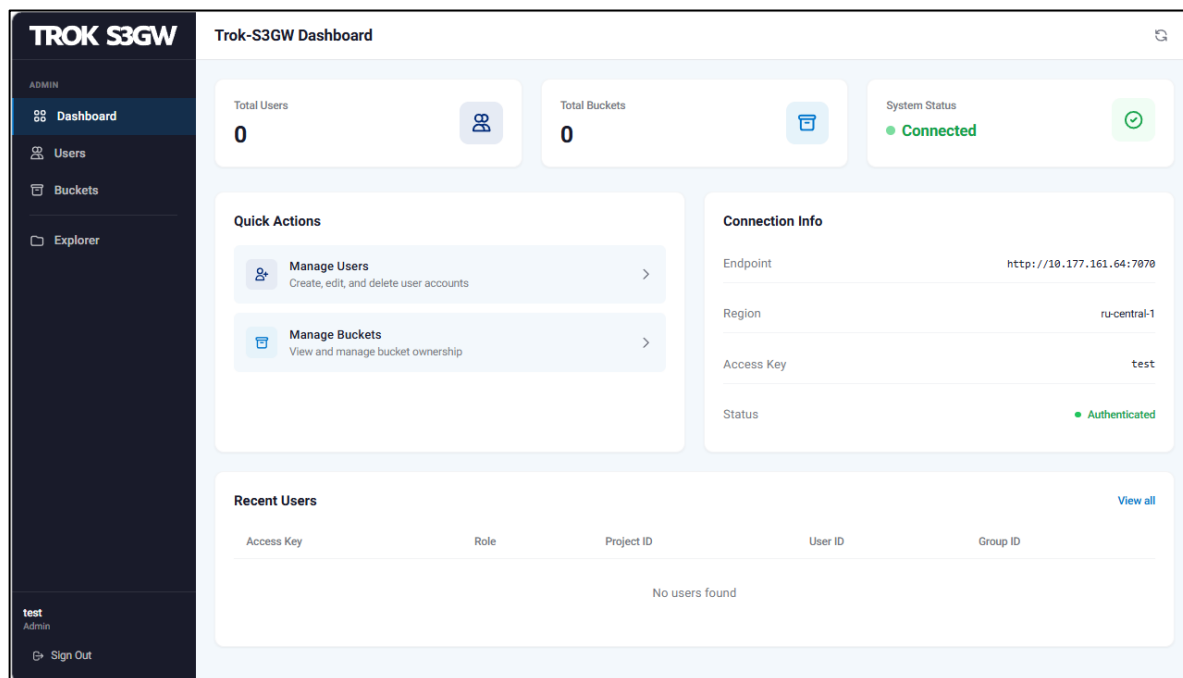


Рисунок 64 – Главная панель управления

В верхней части панели отображаются виджеты с краткой статистикой:

- Всего пользователей (Total Users) – отображает количество зарегистрированных пользователей.
- Всего бакетов (Total Buckets) – показывает текущее число созданных бакетов.
- Статус системы (System Status) – индикация текущего состояния подключения; зелёная отметка и надпись «Подключено» (Connected) свидетельствуют о корректной работе сервера.

В центральной части находится раздел «Быстрые действия» (Quick Actions), включающий два элемента:

- Управление пользователями (Manage Users) – переход к разделу управления учетными записями (создание, редактирование, удаление

пользователей). Кнопка кликабельна и ведёт туда же, куда и пункт «Users» на левой боковой панели.

– Управление бакетами (Manage Buckets) – переход к разделу управления бакетами (создание новых контейнеров, изменение прав доступа, удаление бакетов). Кнопка также кликабельна и аналогична по функционалу пункту «Buckets» в боковой панели.

Правее расположен виджет «Информация о подключении» (Connection Info), отображающий:

– Точка подключения (Endpoint) – адрес, используемый системой для обращения к API.

– Регион (Region) – наименование места размещения вашего хранилища (например, us-east-1).

– Ключ доступа (Access Key) – имя текущего ключа, использованного при авторизации.

– Статус (Status) – подтверждение того, что пользователь успешно аутентифицирован (Authenticated).

Ниже выведен дополнительный раздел «Последние пользователи» (Recent Users), где при необходимости можно просмотреть активность последних учётных записей.

33. УПРАВЛЕНИЕ УЧЁТНЫМИ ЗАПИСЯМИ S3

Раздел «Пользователи» («Users») предназначен для управления учетными записями и ключами доступа к системе. В верхней части расположена строка поиска по ключу доступа («Search by access key...») и выпадающий список фильтрации по ролям («All Roles»). Последний позволяет быстро выбрать отображение только тех пользователей, которые относятся к определённой роли — например, администраторам («Admin») или обычным пользователям («User», «User+»).

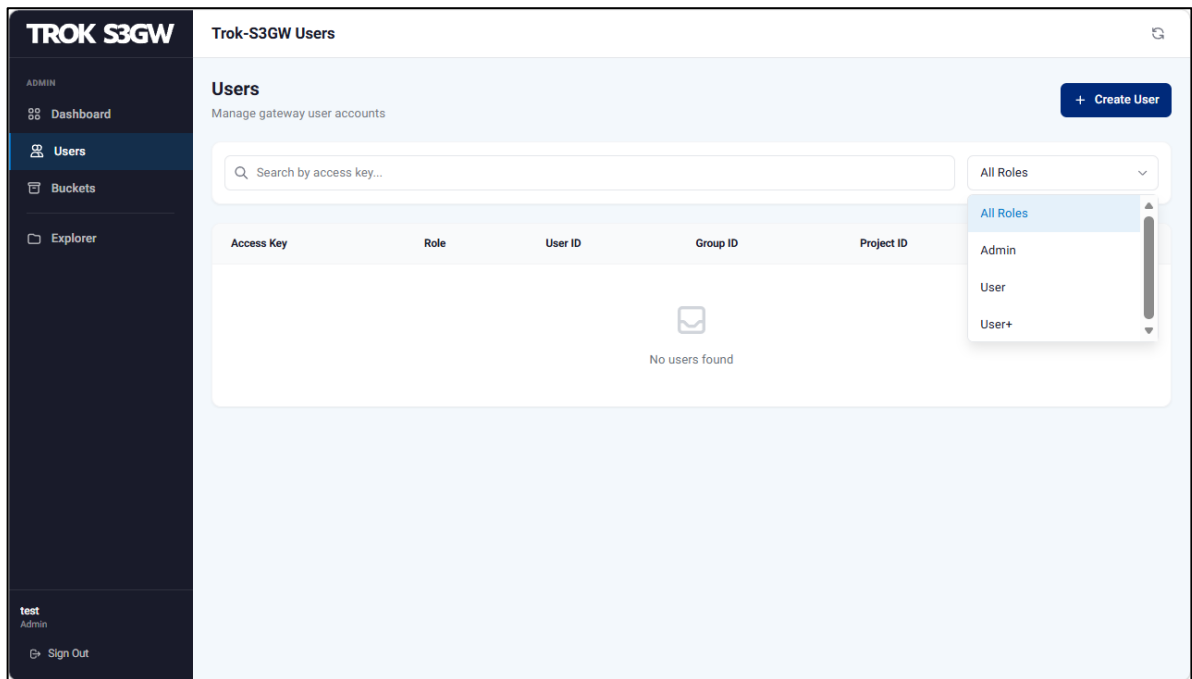


Рисунок 65 – Список выбора роли при фильтрации пользователей

Справа находится кнопка «Создать пользователя» («+ Create User»), по нажатию которой открывается форма для добавления новой учетной записи.

В основной части страницы представлен список зарегистрированных пользователей в табличном виде. Для каждого пользователя отображаются следующие параметры:

- Ключ доступа (Access Key) – уникальный идентификатор пользователя в системе;
- Роль (Role) – определяет уровень прав пользователя (например, Администратор, Пользователь и др.);

- Идентификатор пользователя (User ID) – уникальный числовой идентификатор;
- Идентификатор группы (Group ID) – если используется групповая политика, отражает принадлежность к группе;
- Идентификатор проекта (Project ID) – используется для разграничения пользователей по проектам;
- Действия (Actions) – справа для каждого пользователя доступны кнопки редактирования (иконка «карандаш») и удаления учетной записи (иконка «корзина»).

Для поиска нужного пользователя необходимо ввести часть ключа в строке поиска либо отфильтровать список по роли. Таблица автоматически обновляется в соответствии с установленными параметрами фильтрации.

33.1. Создание нового пользователя

Для выполнения операции создания пользователя нажмите кнопку «+ Create User» («Создать пользователя»), расположенную в правой части экрана.

После выбора команды «Создать пользователя» появляется модальное окно для заполнения параметров новой учетной записи.

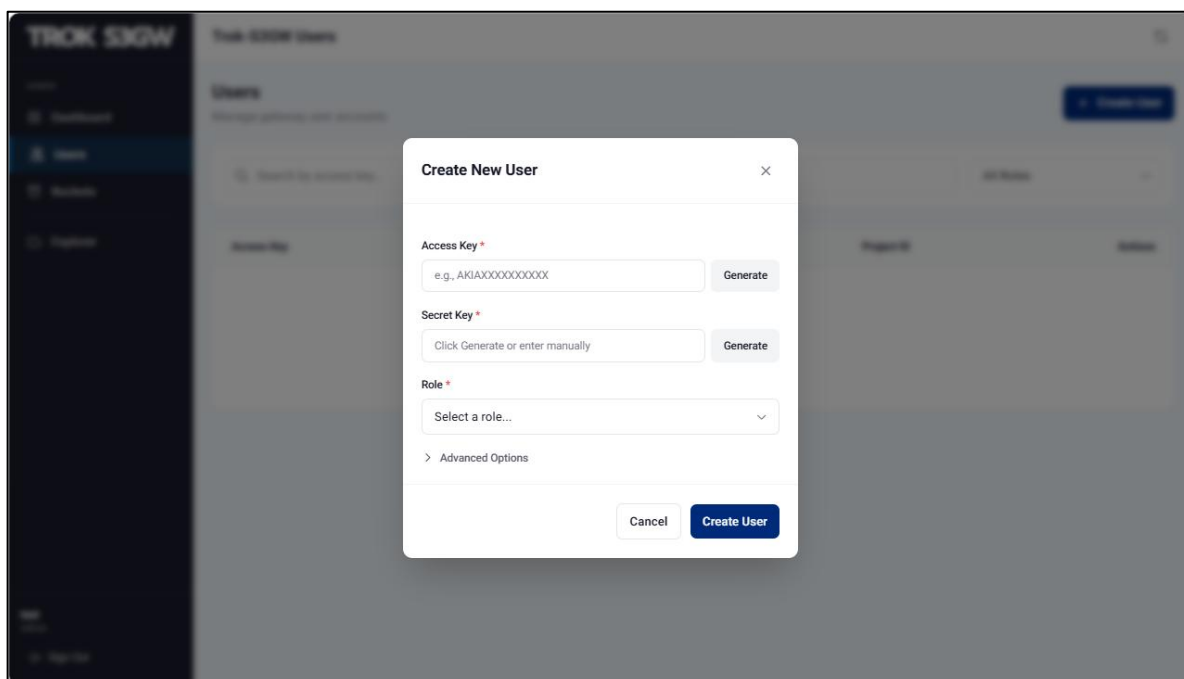


Рисунок 66 – Открытие формы создания пользователя

В появившейся форме предусмотрены поля для ввода ключа доступа (Access Key) и секретного ключа (Secret Key). Значения этих полей могут быть сгенерированы автоматически с помощью функции «Generate» («Сгенерировать»), либо введены вручную пользователем. Оба ключа являются обязательными для заполнения.

Следующий обязательный шаг – выбор роли пользователя. В поле «Role» («Роль») выберите подходящее значение из раскрывающегося списка. Доступны стандартные роли: «Admin» (полный административный доступ), «User» (стандартные пользовательские права), «User+» (расширенные пользовательские права).

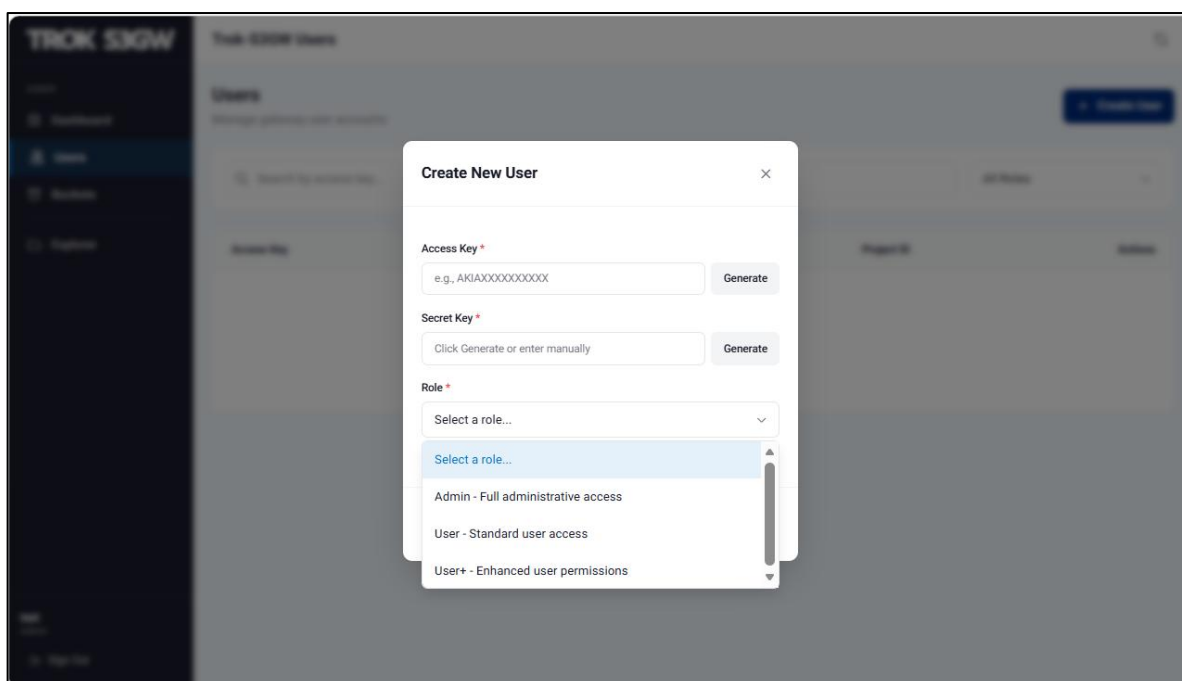


Рисунок 67 – Выбор роли для нового пользователя

При необходимости можно задать дополнительные параметры. Для этого разверните раздел «Advanced Options» («Дополнительные параметры»). В открывшихся полях доступны настройки идентификаторов: User ID (идентификатор пользователя), Group ID (идентификатор группы), Project ID (идентификатор проекта).

Значения этих параметров по умолчанию установлены в «0», однако при необходимости их можно изменить.

Рисунок 68 – Ввод дополнительных параметров нового пользователя

Для завершения процесса создания пользователя нажмите кнопку «Create User» («Создать пользователя»). Для отмены процедуры используйте кнопку «Cancel» («Отмена»).

После завершения операции новый пользователь появится в общем списке учетных записей и будет доступен для управления через соответствующий раздел.

33.2. Редактирование данных пользователя

Редактирование данных пользователя выполняется из раздела «Пользователи» («Users»).

Для начала операции найдите строку с нужным пользователем в общем списке и нажмите на значок редактирования в столбце «Действия» («Actions»). Значок выполнен в виде карандаша и расположен справа от параметров пользователя.

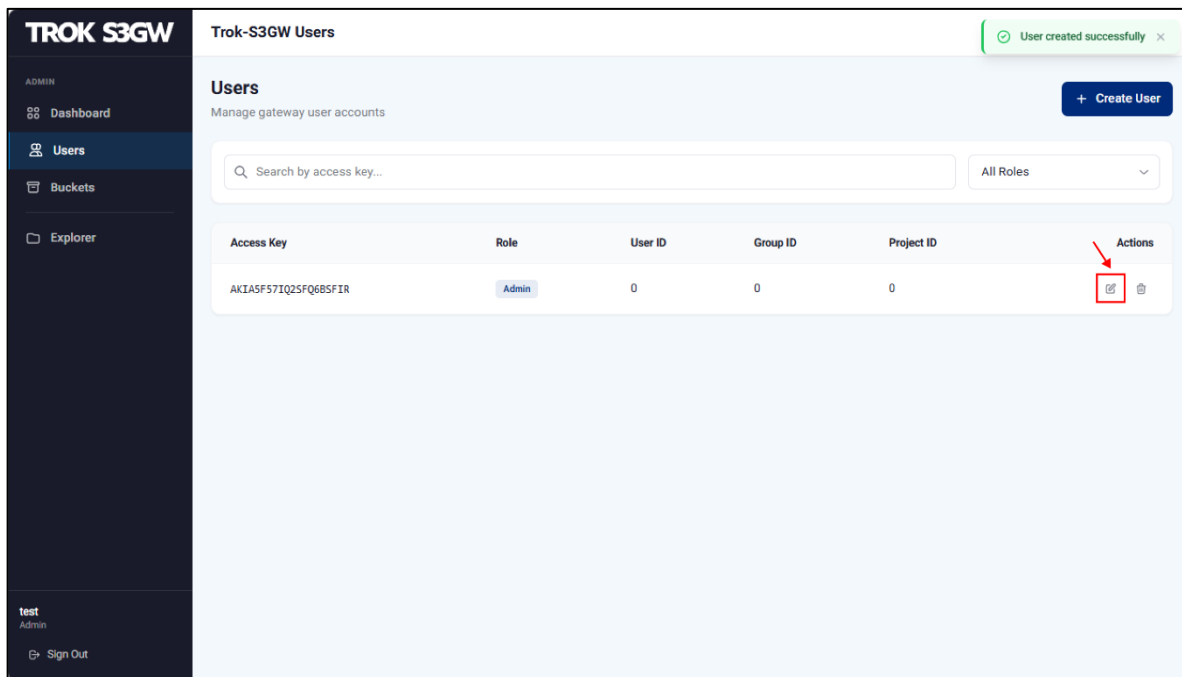


Рисунок 69 – Вызов формы редактирования учетной записи пользователя

Нажатие на значок редактирования открывает модальное окно с заполненной формой пользователя. В этой форме отображаются текущие значения ключа доступа, роли и дополнительных параметров.

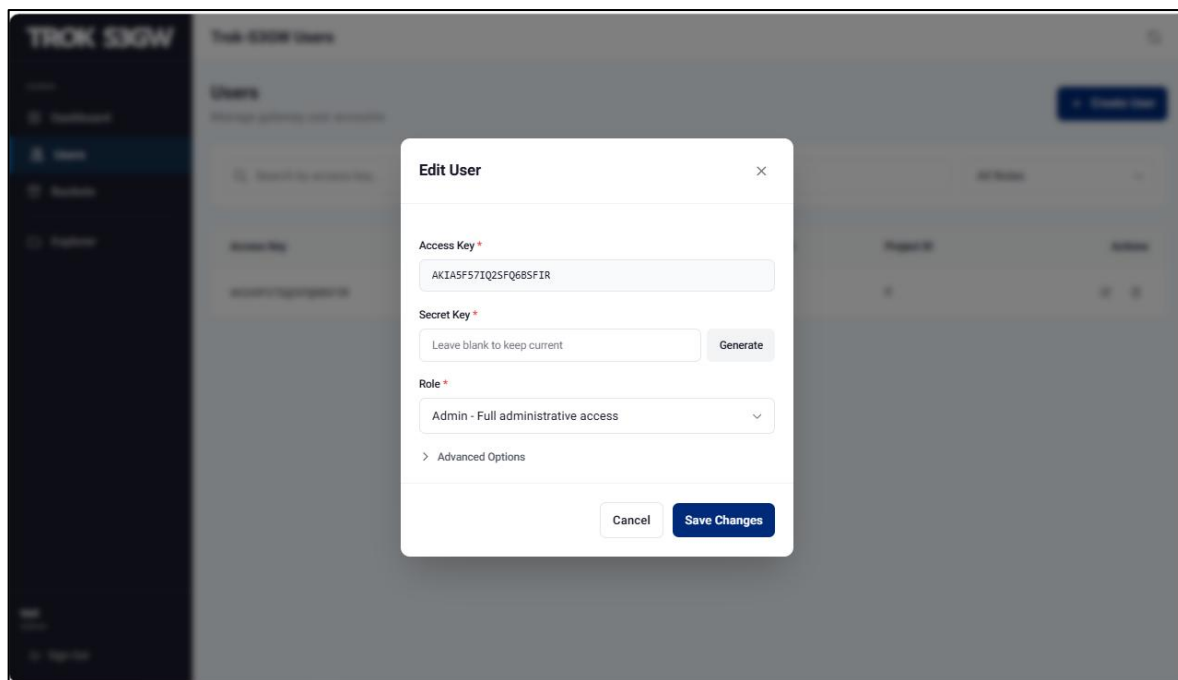


Рисунок 70 – Модальное окно для редактирования учетной записи пользователя

После внесения изменений завершите процедуру нажатием кнопки «Save Changes» («Сохранить изменения»). Для отмены редактирования используйте

кнопку «Cancel» («Отмена»). Изменения сразу применяются и становятся видны в общем списке пользователей.

33.3. Удаление учётной записи

Для выполнения операции удаления учетной записи откройте раздел «Пользователи» («Users»). В таблице найдите строку с нужным «Access Key» и в столбце «Действия» («Actions») нажмите иконку корзины («Delete»), чтобы инициировать удаление учетной записи.

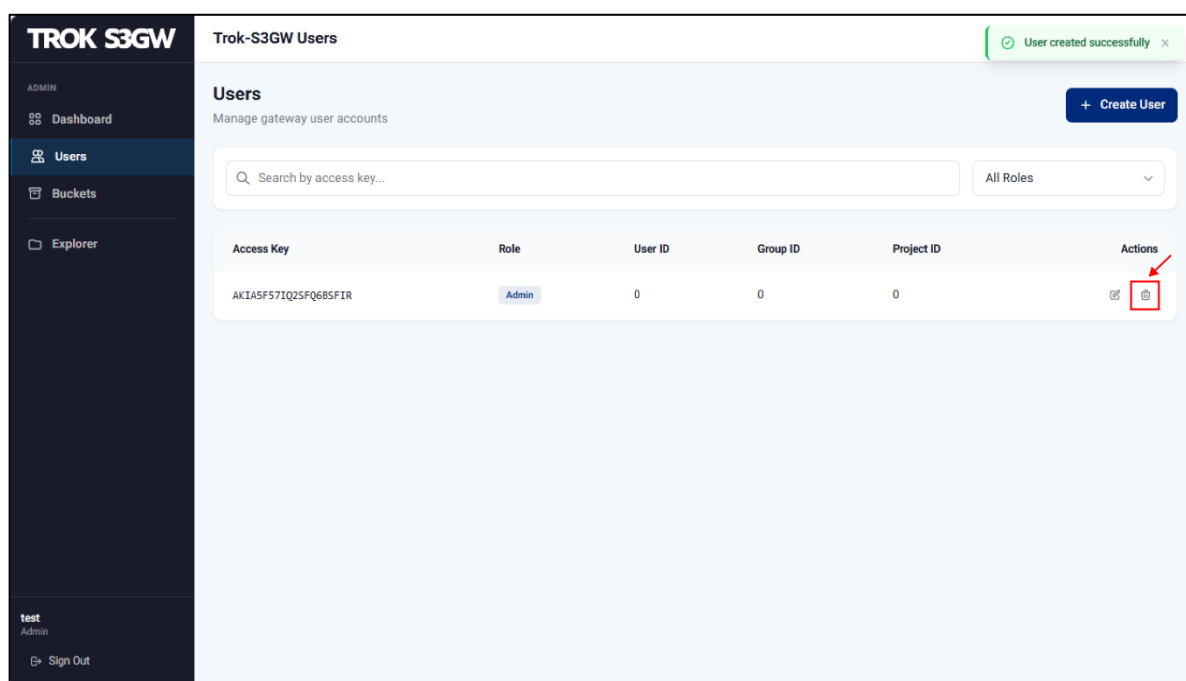


Рисунок 71 – Вызов диалога удаления учетной записи пользователя из списка «Users» («Пользователи»)

После выбора команды удаления откроется модальное окно «Delete User» («Удалить пользователя») с указанием удаляемого «Access Key» и предупреждением о необратимости действия. Проверьте корректность данных и нажмите кнопку «Delete User» («Удалить пользователя») для подтверждения операции. Для отмены действия используйте кнопку «Cancel» («Отмена»).

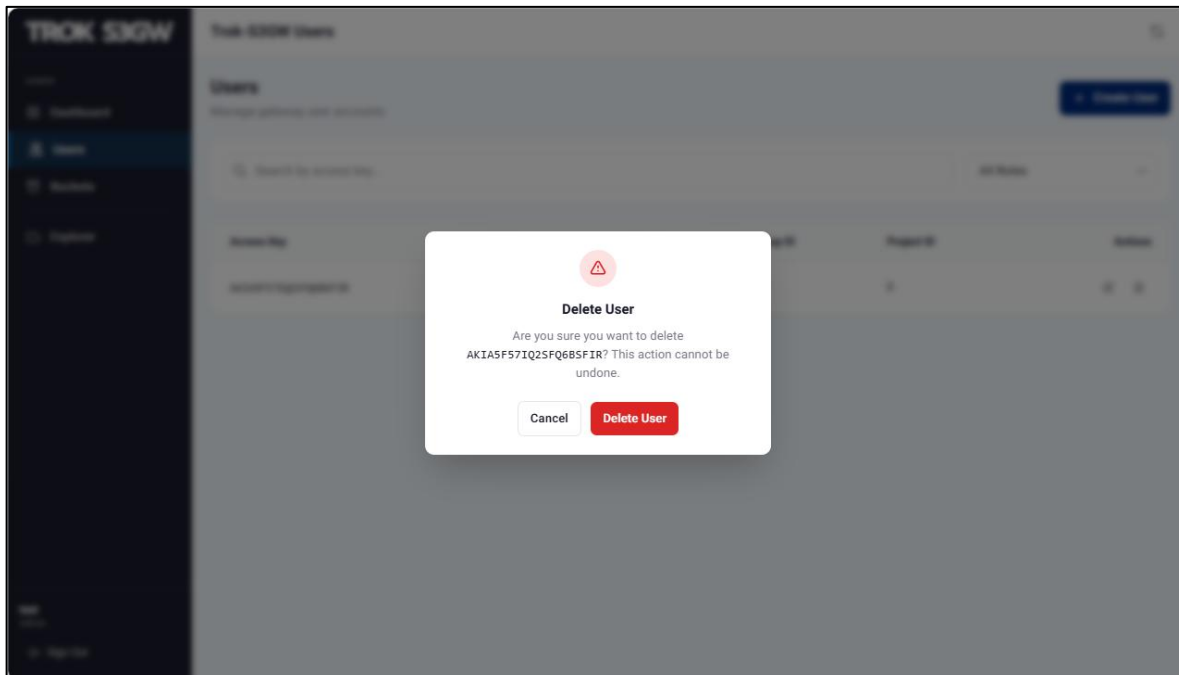


Рисунок 72 – Подтверждение удаления учетной записи

После подтверждения пользователь будет удалён из общего списка, а доступ по его «Access Key» и «Secret Key» будет немедленно прекращён. Таблица обновится автоматически.

Примечания:

- Операция удаления учетной записи необратима.
- Если пользователь является владельцем одного или нескольких бакетов, предварительно выполните передачу прав владения на эти бакеты (см. раздел «Изменение владельца бакета»). В противном случае удаление может быть отклонено политиками безопасности системы.

34. БАКЕТЫ

Раздел «Бакеты» («Buckets») предназначен для управления контейнерами данных в системе. В верхней части экрана расположен информационный блок с подсказкой о функциях: создание новых бакетов, просмотр существующих и передача прав владения между пользователями (Create buckets, view existing buckets, and transfer ownership between users.).

Под блоком подсказки находится строка поиска по имени бакета («Search buckets...») и выпадающий список фильтрации по владельцам (по умолчанию: «All Owners»).

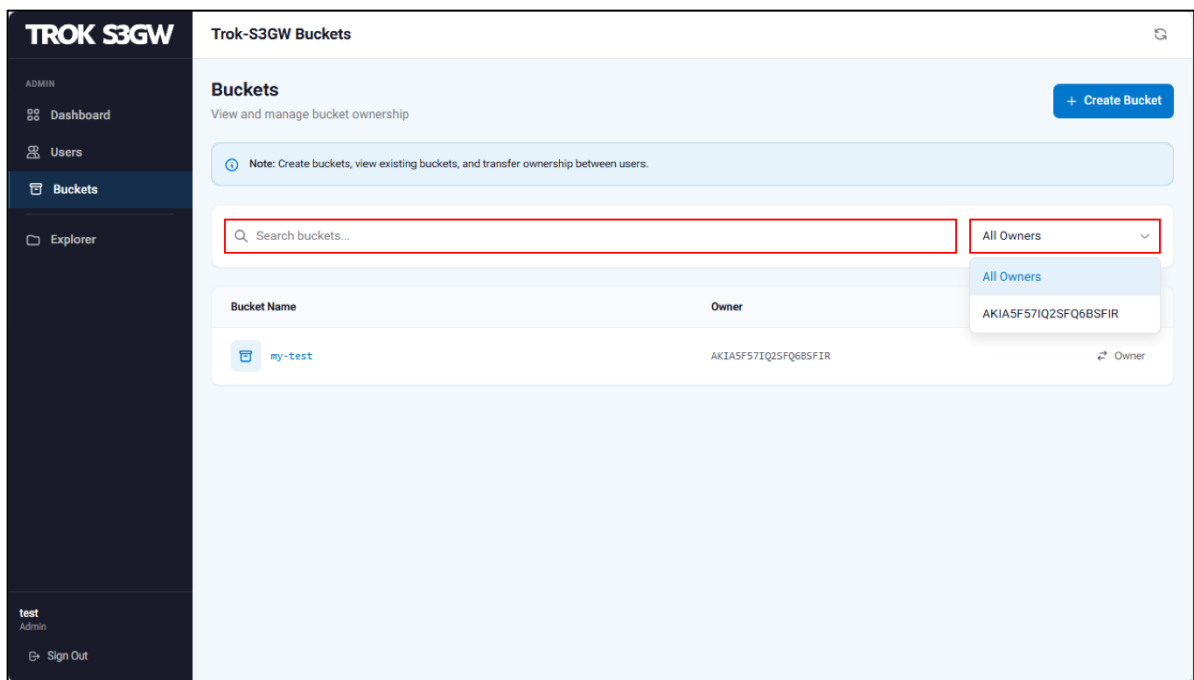


Рисунок 73 – Строка поиска и выпадающий список фильтрации бакетов

Справа расположена кнопка «+ Создать бакет» («+ Create Bucket»), по нажатию которой открывается форма для добавления нового контейнера данных.

В основной части страницы представлен список существующих бакетов в табличном виде. Для каждого бакета отображаются следующие параметры:

- Имя бакета (Bucket Name) – уникальное имя контейнера данных;
- Владелец (Owner) – ключ доступа пользователя, которому принадлежит бакет;

– Действия (Actions) – справа для каждого бакета доступны команды по изменению владельца (значок с круговой стрелкой) или быстрой идентификации текущего владельца (пометка «Owner»).

Для поиска нужного бакета введите фрагмент имени в строке поиска либо выберите владельца из выпадающего меню фильтрации. Список автоматически обновится согласно выбранным условиям.

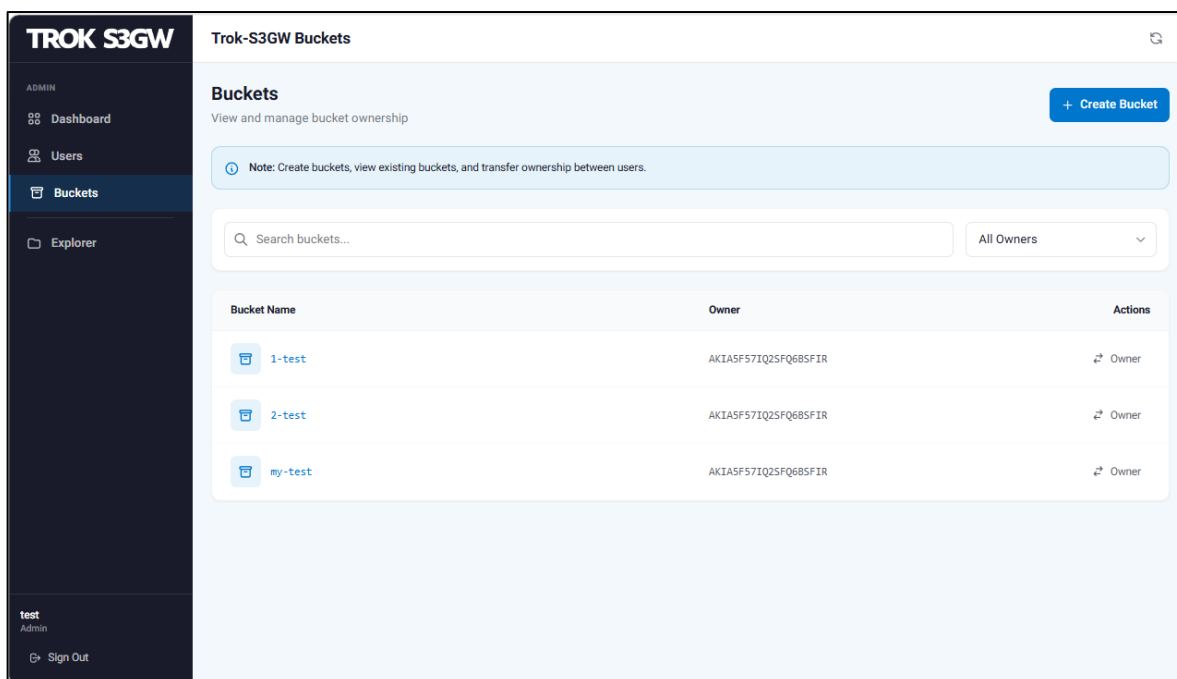


Рисунок 74 – Список бакетов с параметрами владельца и действиями

34.1. Создание бакета

Создание нового бакета осуществляется с помощью специальной формы, которая открывается по кнопке «+ Create Bucket» («+ Создать бакет») в разделе «Бакеты» («Buckets»). Последовательность действий пользователя включает ввод имени, выбор владельца и настройку дополнительных параметров бакета.

В открывшемся диалоговом окне заполните следующие поля:

– Имя бакета (Bucket Name) – имя, определяющее уникальный идентификатор создаваемого бакета. Имя должно содержать от 3 до 63 символов, быть в нижнем регистре, и может включать буквы, цифры и дефисы.

– Владелец (Owner) – указание пользователя-владельца бакета. Доступен выпадающий список с перечнем пользователей системы.

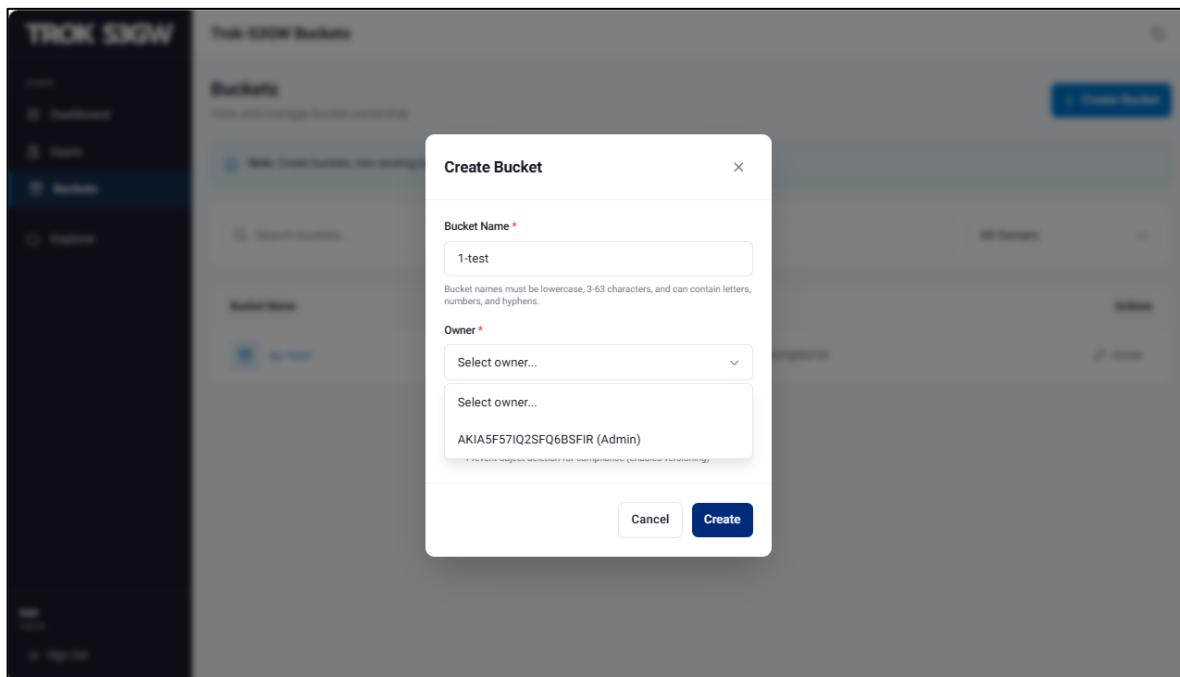


Рисунок 75 – Выбор владельца создаваемого бакета из выпадающего списка

Дополнительно в нижней части формы можно активировать опции:

- Enable Versioning – включение поддержки версионности объектов в бакете;
- Enable Object Lock – активация блокировки объектов для соответствия требованиям хранения и безопасности.

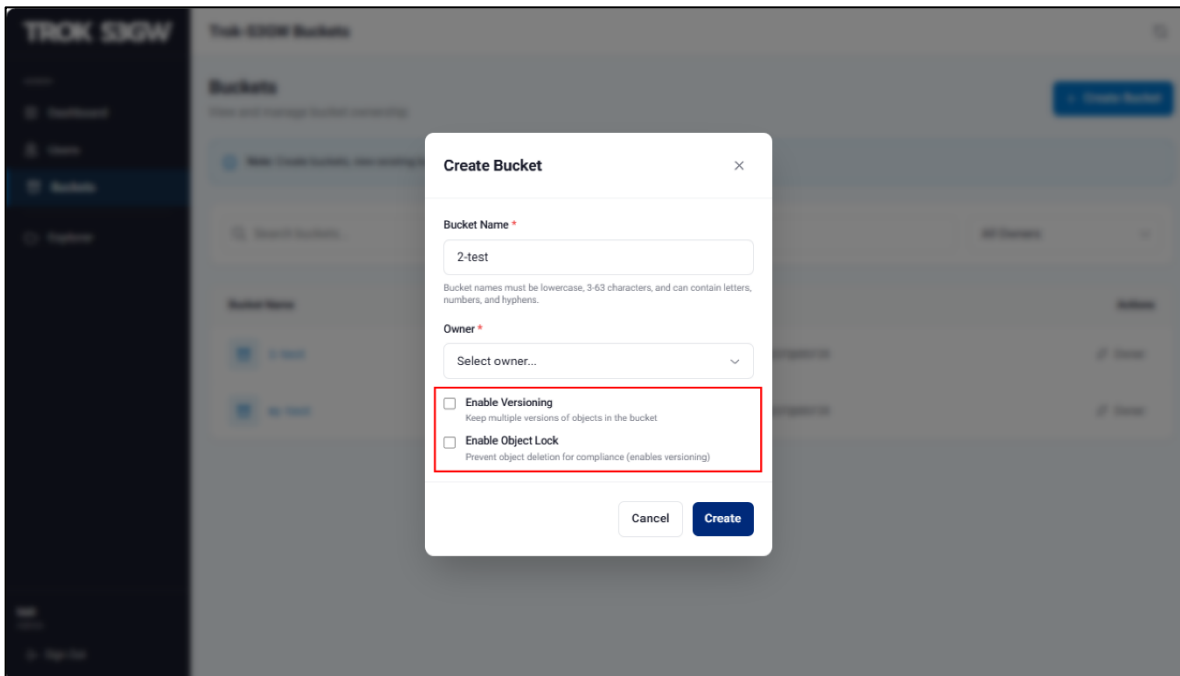


Рисунок 76 – Настройка дополнительных параметров при создании нового бакета

Для завершения процедуры создания бакета необходимо нажать кнопку «Create» («Создать») в правом нижнем углу окна. Если требуется отменить операцию, используется кнопка «Cancel» («Отмена»).

34.2. Изменение владельца бакета

Чтобы изменить владельца бакета, найдите нужный бакет в списке и перейдите к столбцу «Actions» («Действия»). Нажмите на кнопку «Owner» («Владелец») с иконкой смены владельца, расположенную справа от названия бакета.

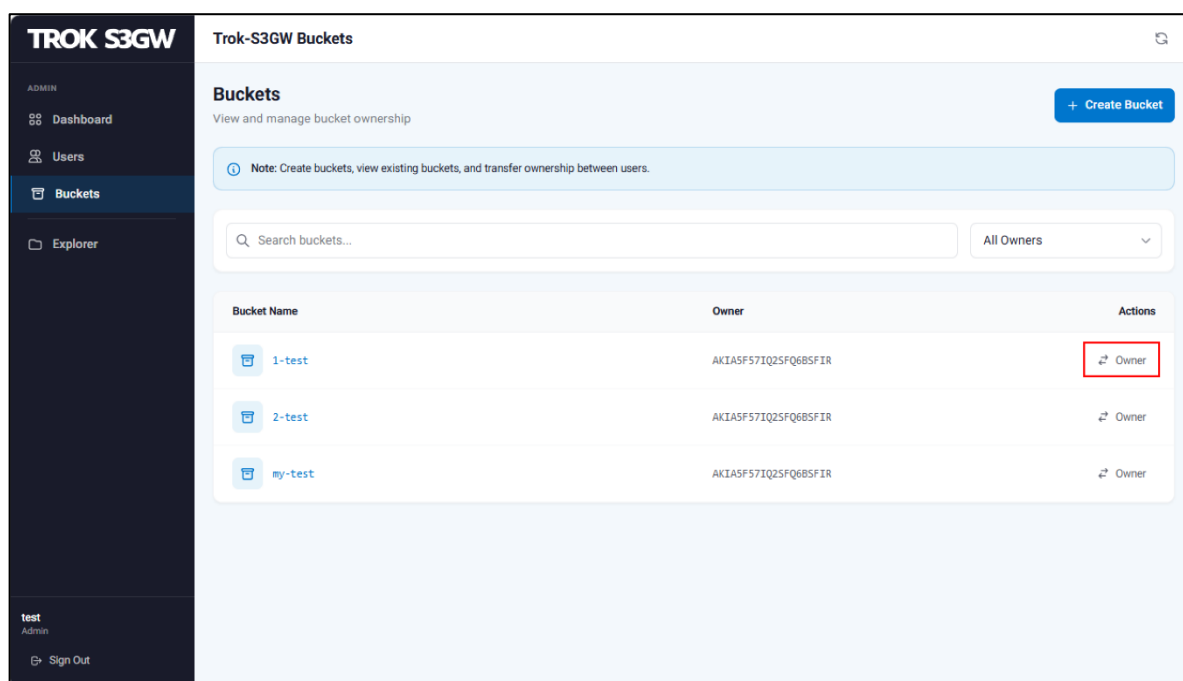


Рисунок 77 – Вызов функции изменения владельца бакета

В открывшемся диалоговом окне «Change Bucket Owner» («Изменение владельца бакета») ознакомьтесь с параметрами выбранного бакета и текущим владельцем. В поле «New Owner» («Новый владелец») выберите нового пользователя из выпадающего списка.

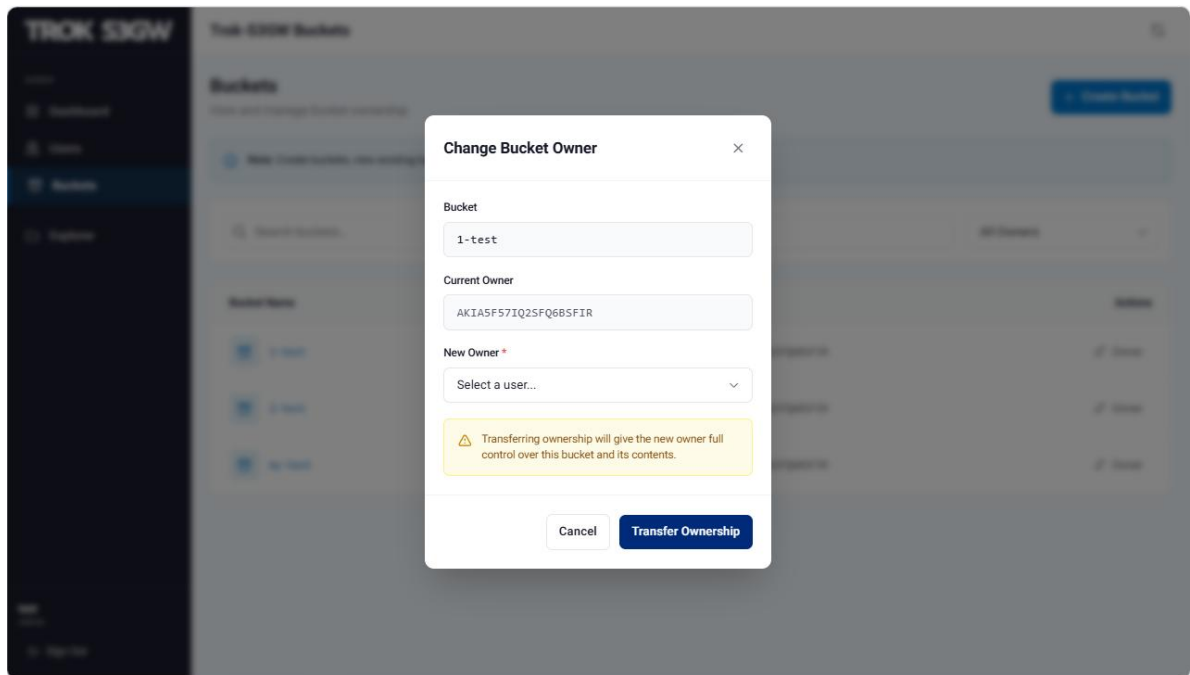


Рисунок 78 – Диалоговое окно выбора нового владельца и подтверждения передачи прав на бакет

Для завершения процедуры нажмите кнопку «Transfer Ownership» («Передать права»). Система выдаст предупреждение о том, что новый владелец получит полный контроль над бакетом и его содержимым. Если необходимо прекратить выполнение операции, воспользуйтесь кнопкой «Cancel» («Отмена»).

35. ПРОВОДНИК (ОБЗОР БАКЕТОВ)

Раздел «Проводник» («Explorer») предназначен для просмотра и управления бакетами в системе хранения данных.

В верхней части страницы размещена кнопка «+ Create Bucket» («+ Создать бакет»), которая позволяет добавить новый контейнер в систему. По нажатию открывается форма для ввода необходимых параметров нового бакета.

В основной области страницы представлен табличный список бакетов, доступных для просмотра и управления. Для каждого бакета отображаются следующие параметры:

- Имя (Name) – уникальное имя бакета, позволяющее идентифицировать контейнер в системе;

- Версионирование (Versioning) – статус включения поддержки версий объектов («Enabled» – Включено, «Disabled» – Отключено). Данная опция позволяет хранить несколько версий файлов в бакете;

- Блокировка объектов (Object Lock) – статус защиты объектов от удаления («Enabled» – Включено, «Disabled» – Отключено). Функция обеспечивает соответствие корпоративным требованиям безопасности;

- Действия (Actions) – справа для каждого бакета предусмотрен набор иконок, обеспечивающих доступ к дополнительным функциям управления контейнером, включая просмотр свойств, изменение прав, блокировку, а также удаление бакета (Remove).

Взаимодействие с конкретным бакетом осуществляется путём выбора соответствующей команды в столбце «Actions» (Действия).

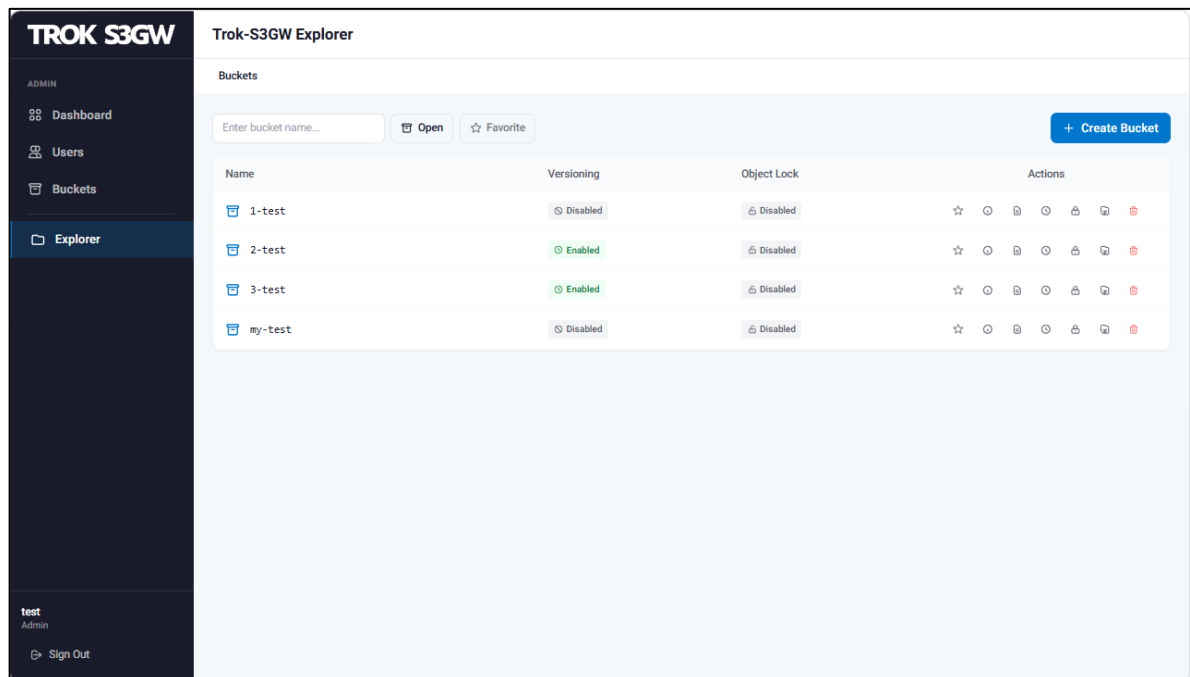


Рисунок 79 – Таблица бакетов

35.1. Добавление бакета в избранное

Для закрепления часто используемых бакетов в панели «Favorites» выполните следующие действия в разделе «Проводник» («Explorer»).

При необходимости воспользуйтесь строкой «Enter bucket name...» («Введите имя бакета...») для поиска нужного бакета в таблице. В таблице бакетов найдите нужный контейнер. В столбце «Actions» («Действия») нажмите иконку пустой звезды напротив выбранного бакета. Иконка изменит состояние на заполненную, указывая на добавление в избранное.

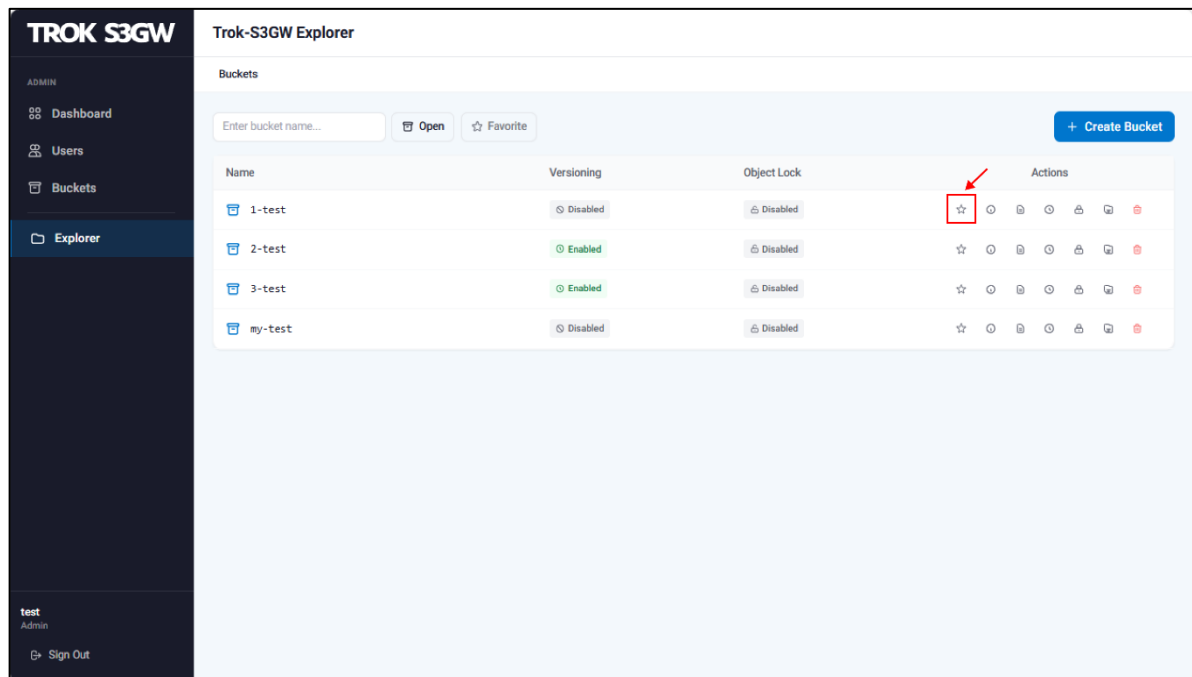


Рисунок 80 – Выбор иконки «звезда» в столбце «Actions» («Действия»)

В правом верхнем углу появится уведомление об успешном добавлении (например, «Added "1-test" to favorites»).

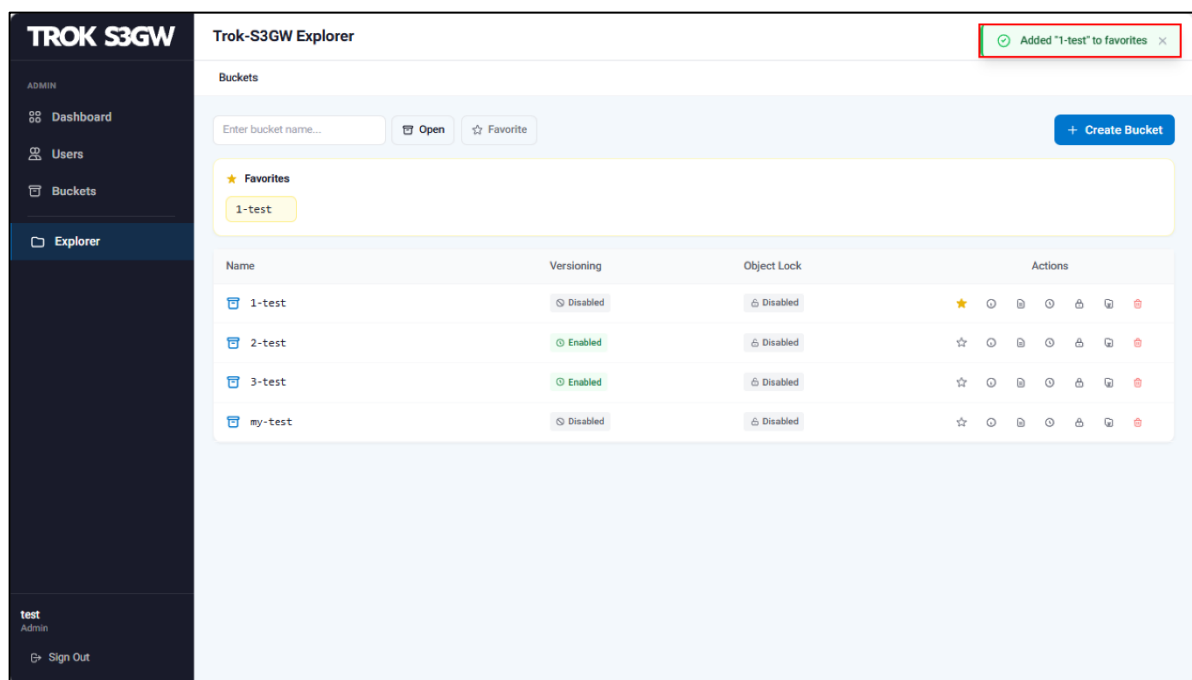


Рисунок 81 – Результат добавления бакета в список избранных

В верхней части страницы отобразится панель «Favorites» («Избранное») с бейджем добавленного бакета.

В блоке «Favorites» («Избранное») отображаются все закреплённые бакеты. При необходимости повторите шаги для добавления других контейнеров.

Для удаления бакета из избранного нажмите на заполненную звезду в его строке ещё раз.

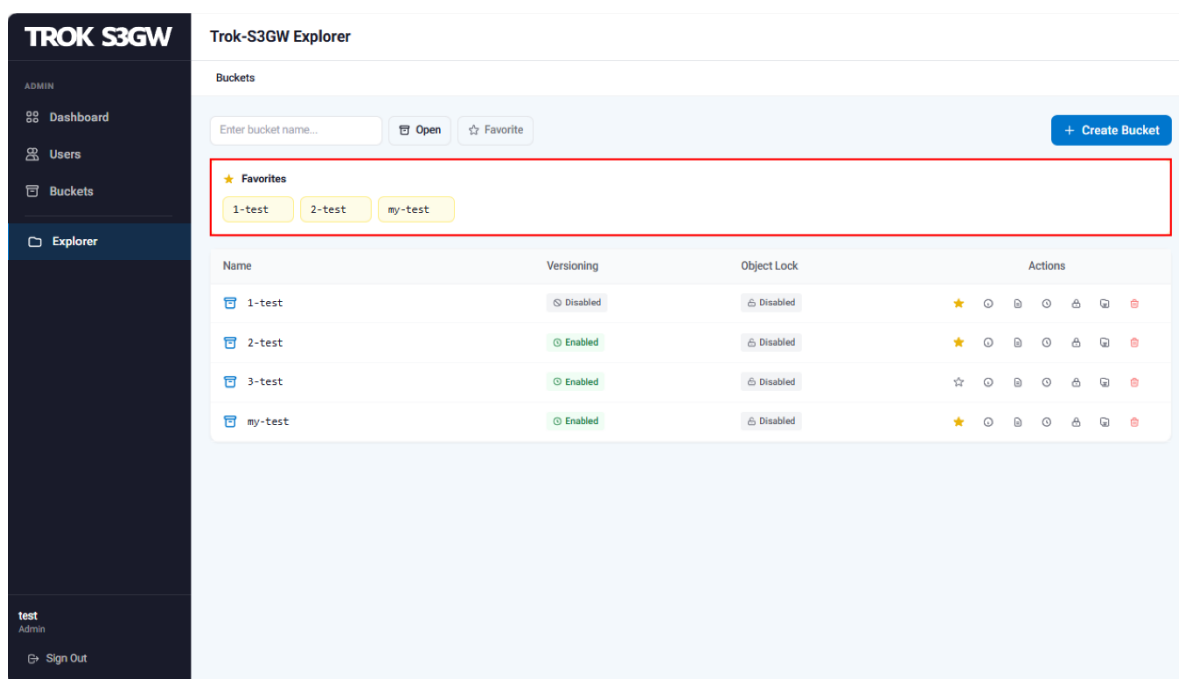


Рисунок 82 – Панель «Favorites» («Избранное») с несколькими закреплёнными бакетами

35.2. Получение информации о бакете и добавление тегов

Для получения подробной информации о бакете выполните следующие действия в разделе «Проводник» («Explorer»). Найдите нужный бакет в списке и перейдите к столбцу «Действия» («Actions»). Нажмите на иконку с символом информации («i»), расположенную справа от названия бакета в соответствующей строке.

В открывшемся окне «Bucket Information» («Информация о бакете») ознакомьтесь с основными параметрами:

- Ниже, в разделе «Tags» («Теги»), отображается список тегов, связанных с бакетом. Для добавления новых тегов используйте ссылку «Add Tag» («Добавить тег»). После внесения изменений сохраните их с помощью кнопки «Save Tags» («Сохранить теги»). Для закрытия окна используйте кнопку «Close» («Закрыть»).

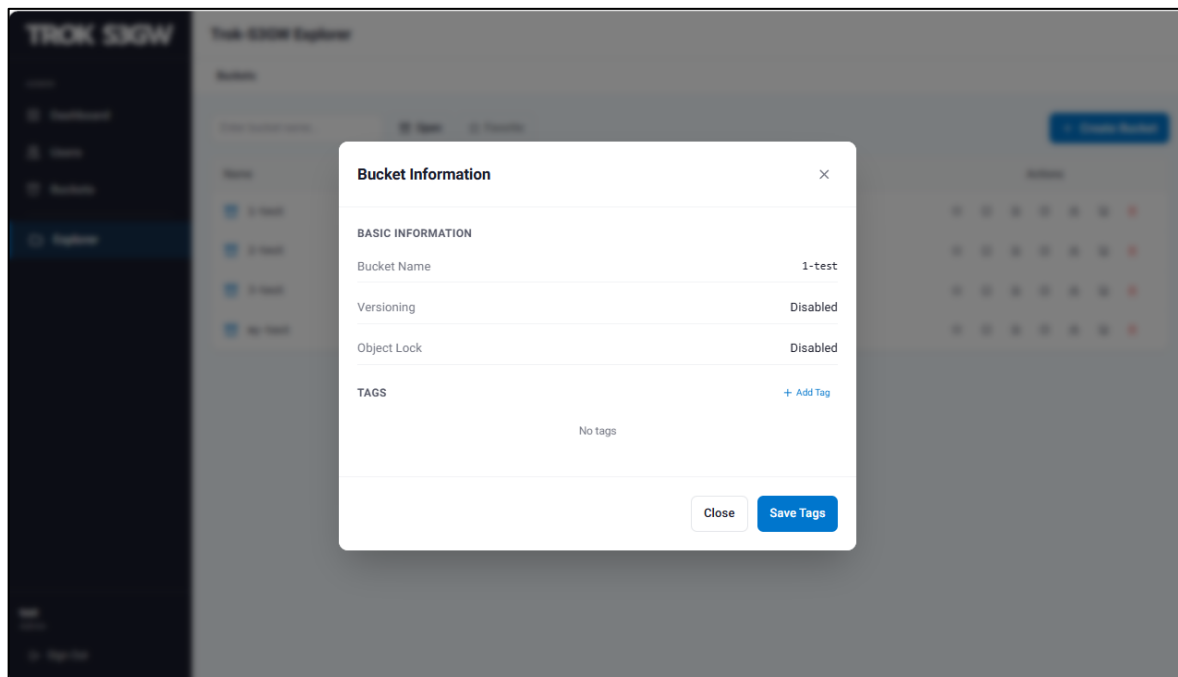


Рисунок 84 – Окно информации о бакете с возможностью добавления тегов

35.3. Настройка политик бакета

Для настройки политик S3 бакета выполните следующие шаги в разделе «Проводник» (Explorer).

Найдите необходимый бакет в списке и перейдите к столбцу «Действия» («Actions»). Нажмите на иконку документа, чтобы открыть интерфейс управления политиками.

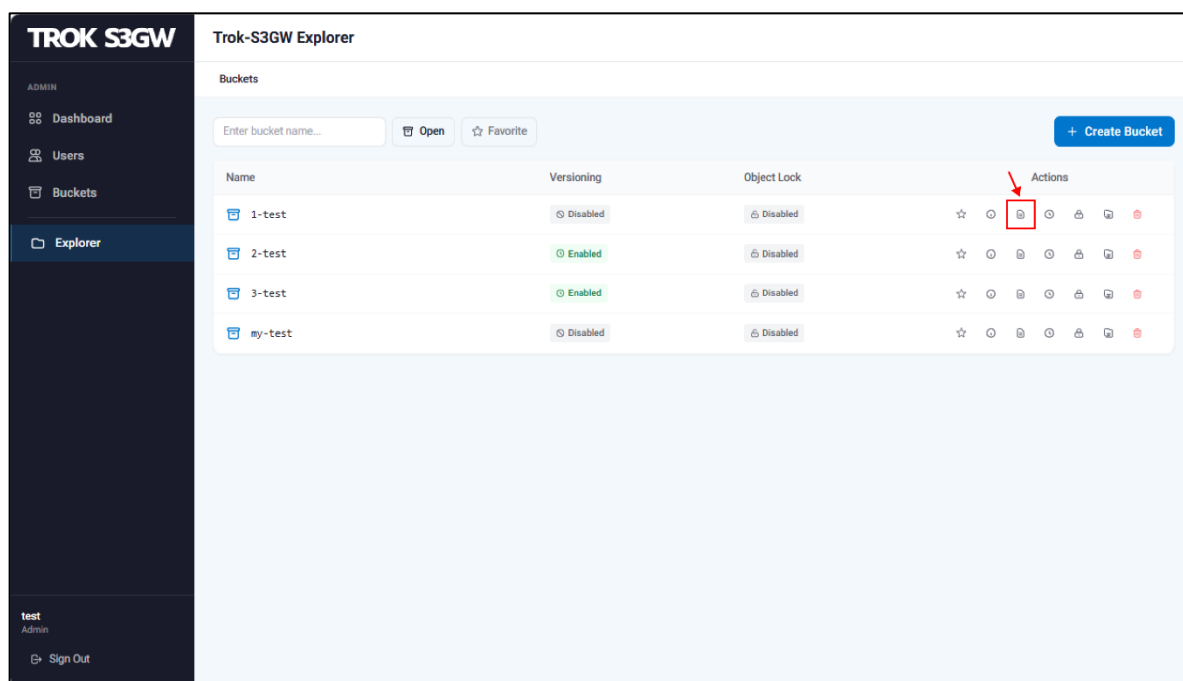


Рисунок 85 – Открытие редактора политики бакета из таблицы бакетов

В открывшемся окне «Bucket Policy Manager» («Управление политиками бакета») настройте политику доступа. В поле «Current Policy» («Текущая политика») укажите необходимый JSON-документ S3-политики. Вы можете:

- Ввести/отредактировать код политики вручную;
- Загрузить пример (Load Example);
- Проверить корректность структуры и синтаксиса с помощью кнопки «Validate» («Валидация»). После внесения изменений нажмите «Save Policy» («Сохранить политику»).

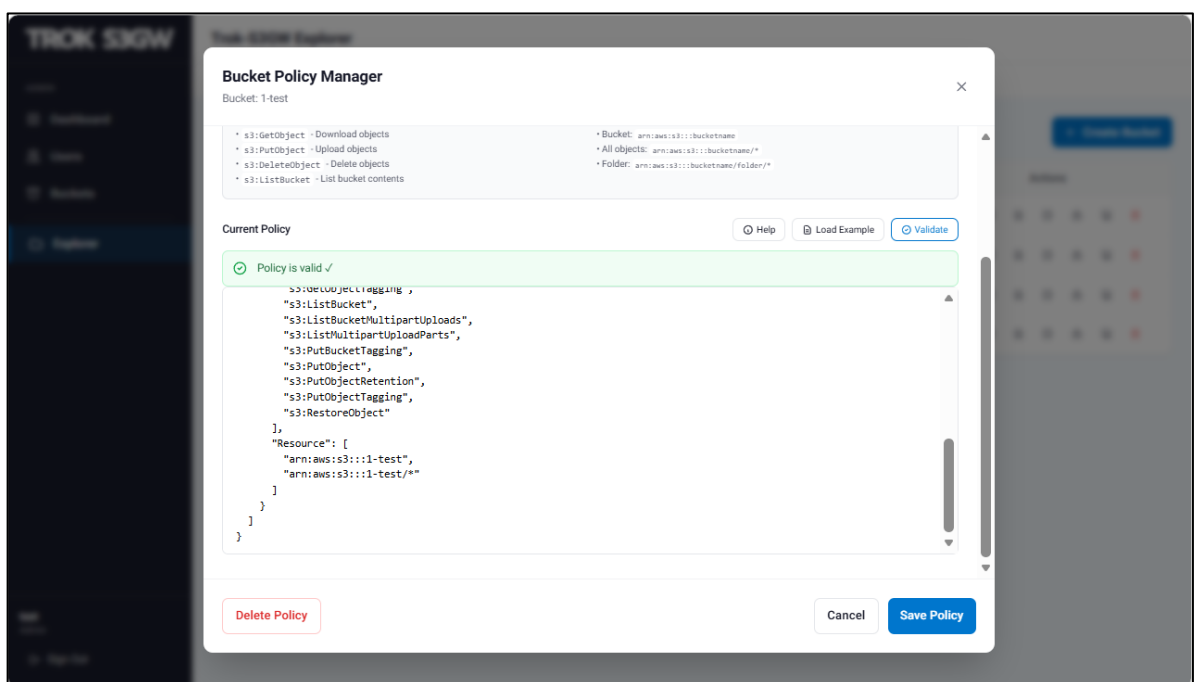


Рисунок 86 – Настройка политики s3, валидация и сохранение

Каждая политика формируется в формате JSON. Команда описывает, КТО (Principal) может ЧТО (Action) делать с КАКИМИ ресурсами (Resource) и с каким эффектом (Effect):

- Version – версия синтаксиса политики. Для современных политик всегда используется "2012-10-17".
- Statement – массив правил (утверждений). Каждое правило – отдельный объект с Effect, Principal, Action, Resource и т. д.
- Sid – человекочитаемый идентификатор одного правила. Опционален; удобно для поиска/аудита.

– Effect – результат правила: «Allow» (разрешить) или «Deny» (запретить). Явное значение «Deny» имеет приоритет над «Allow».

– Principal – субъект, к которому применяется правило (кто выполняет действие). В ресурсных политиках (например, политика бакета) обязателен. Значения: "*" (любой); либо ARN(ы) пользователей/ролей/аккаунтов, например "arn:aws:iam::123456789012:user/alice".

– Action – какие операции разрешены/запрещены. Полный список действий указан в таблицах 1, 2, 3 и 4. Можно указывать массив или одну строку.

– Resource – к каким ресурсам относится правило. Для S3 используются ARN'ы (Бакет: arn:aws:s3:::my-bucket).

Варианты стандартных S3-политик и примеры кода:

Перечень действий S3 и примеры кода:

1) Полный доступ для определённых пользователей

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "FullS3Access",
      "Effect": "Allow",
      "Principal": "user001,user002",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::foo-bar",
        "arn:aws:s3:::foo-bar/*"
      ]
    }
  ]
}
```

2) Только на чтение для всех пользователей (public read)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PublicReadGetObject",
      "Effect": "Allow",
```

```

    "Principal": "*",
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::foo-bar/*"
  }
]
}

```

3) Ограничение доступа только на загрузку (upload only)

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "UploadOnly",
      "Effect": "Allow",
      "Principal": "user001",
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::foo-bar/*"
    }
  ]
}

```

4) Разрешение на работу с тегами

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowTagging",
      "Effect": "Allow",
      "Principal": "user003",
      "Action": [
        "s3:GetBucketTagging",
        "s3:PutBucketTagging"
      ],
      "Resource": "arn:aws:s3:::foo-bar"
    }
  ]
}

```

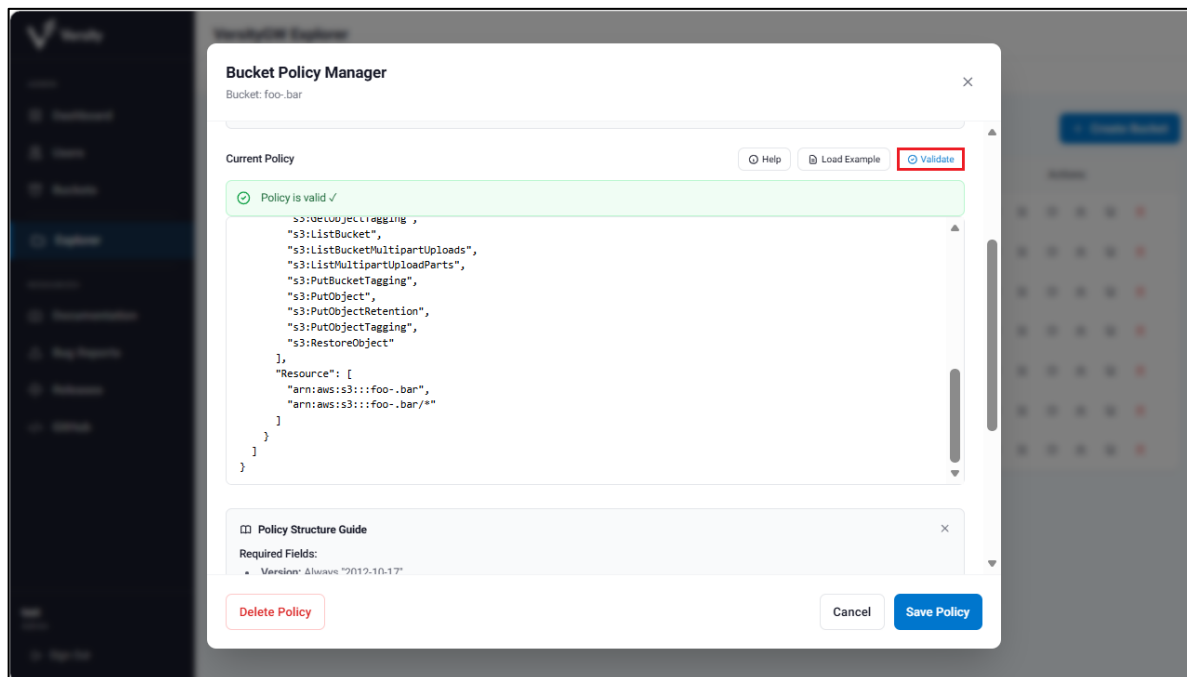


Рисунок 87 – Настройка, валидирование и сохранение политики S3

Полный перечень действий для S3. Все действия сгруппированы по уровням доступа:

Таблица 3 – Действия уровня List (получение списков)

Действие	Описание
s3:ListBucket	Разрешает операцию ListObjects – получение списка объектов в бакете
s3:ListBucketMultipartUploads	Разрешает операцию ListMultipartUploads – список незавершённых мультипартидных загрузок
s3:ListBucketVersions	Разрешает операцию ListObjectVersions – список версий объектов (действие на бакет)
s3:ListAllMyBuckets	Разрешает операцию ListBuckets – список всех бакетов (глобальное)
s3:ListMultipartUploadParts	Разрешает операцию ListParts – список загруженных частей мультипартидной загрузки

Таблица 4 – Действия уровня Read (чтение конфигураций и метаданных бакета)

Действие	Описание
----------	----------

s3:GetBucketAcl	Получение ACL бакета
s3:GetBucketCORS	Получение CORS-конфигурации бакета
s3:GetBucketLocation	Получение региона бакета
s3:GetBucketLogging	Получение настроек логирования бакета
s3:GetBucketNotification	Получение конфигурации уведомлений бакета
s3:GetBucketObjectLockConfiguration	Получение конфигурации Object Lock бакета
s3:GetBucketPolicy	Получение политики бакета
s3:GetBucketPolicyStatus	Получение статуса политики бакета
s3:GetBucketPublicAccessBlock	Получение настроек Public Access Block бакета
s3:GetBucketReplication	Получение конфигурации репликации бакета
s3:GetBucketRequestPayment	Получение настроек плательщика запросов
s3:GetBucketTagging	Получение тегов бакета
s3:GetBucketVersioning	Получение статуса версионирования бакета
s3:GetBucketWebsite	Получение конфигурации веб-сайта бакета
s3:GetEncryptionConfiguration	Получение конфигурации шифрования бакета
s3:GetLifecycleConfiguration	Получение конфигурации жизненного цикла бакета
s3:GetMetricsConfiguration	Получение метрик бакета
s3:GetReplicationConfiguration	Получение конфигурации репликации
s3:HeadBucket	Проверка существования бакета и прав доступа к нему

Таблица 5 – Действия уровня Write (создание, изменение, удаление бакетов и их конфигураций)

Действие	Описание
s3:AbortMultipartUpload	Отмена мультипартидной загрузки (процесс, не объект)
s3:CreateBucket	Создание нового бакета
s3>DeleteBucket	Удаление пустого бакета
s3>DeleteBucketWebsite	Удаление конфигурации веб-сайта бакета
s3:PutBucketCORS	Установка CORS-конфигурации бакета
s3:PutBucketLogging	Установка настроек логирования бакета
s3:PutBucketNotification	Установка конфигурации уведомлений бакета
s3:PutBucketObjectLockConfiguration	Установка конфигурации Object Lock бакета
s3:PutBucketReplication	Установка конфигурации репликации бакета
s3:PutBucketRequestPayment	Установка настроек плательщика запросов
s3:PutBucketVersioning	Включение/отключение версионирования бакета
s3:PutBucketWebsite	Установка конфигурации веб-сайта бакета
s3:PutEncryptionConfiguration	Установка конфигурации шифрования бакета
s3:PutLifecycleConfiguration	Установка конфигурации жизненного цикла бакета
s3:PutMetricsConfiguration	Установка метрик бакета
s3:PutReplicationConfiguration	Установка конфигурации репликации
s3:UpdateBucketPolicy	Обновление политики бакета

Таблица 6 – Действия уровня Permissions management (управление разрешениями на уровне бакета)

Действие	Описание
s3:PutBucketAcl	Установка ACL бакета
s3:PutBucketPolicy	Установка политики бакета
s3:PutBucketPublicAccessBlock	Установка Public Access Block бакета
s3:PutBucketTagging	Установка тегов бакета
s3>DeleteBucketPolicy	Удаление политики бакета
s3>DeleteBucketPublicAccessBlock	Удаление Public Access Block бакета
s3>DeleteBucketTagging	Удаление тегов бакета

35.4. Управление версионированием бакета

Для включения или приостановки версионирования выполните следующие действия в разделе «Проводник» («Explorer»). Найдите нужный бакет в списке и перейдите к столбцу «Действия» («Actions»). Нажмите на иконку с символом часов/круговой стрелки (иконка версионирования), расположенную справа от названия бакета в соответствующей строке.

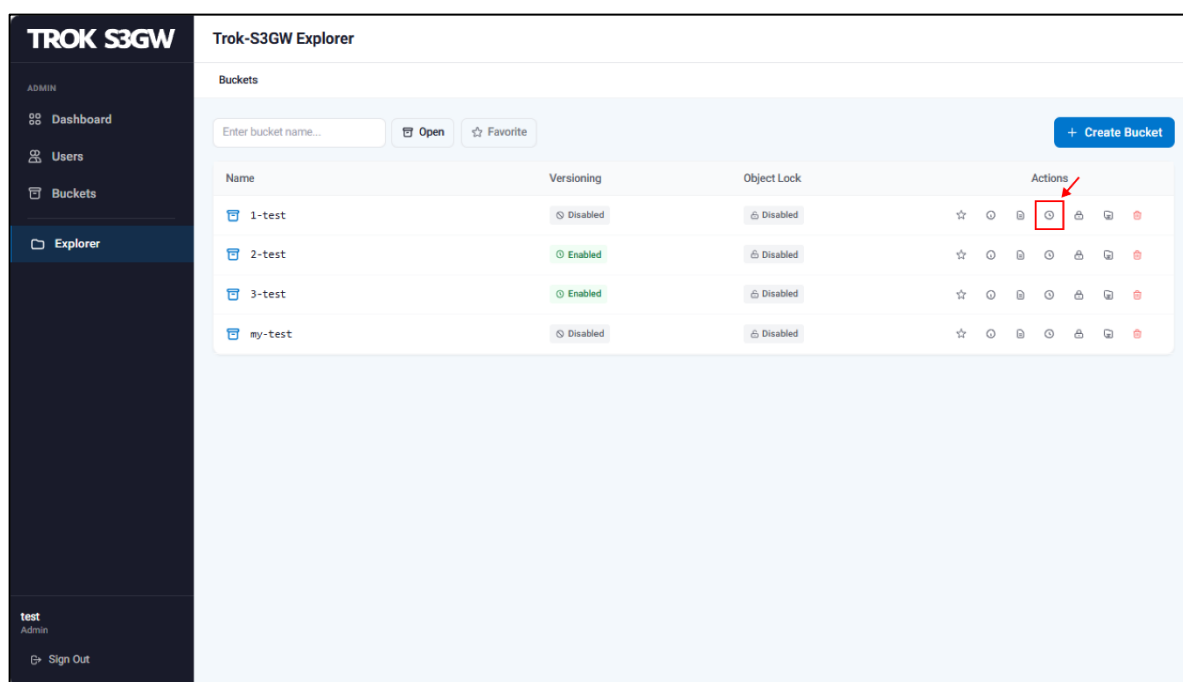


Рисунок 88 – Вызов окна управления версионированием через меню действий

В открывшемся окне «Bucket Versioning» («Версионирование бакета») ознакомьтесь с параметрами:

- Имя бакета (Bucket Name) – идентификатор контейнера данных, к которому применяется настройка;
- Текущий статус (Current Status) – состояние версионирования (Enabled – Включено, Disabled – Отключено, Suspended – Приостановлено);
- Примечание (Note) – после включения версионирования нельзя полностью отключить; доступна только приостановка (Suspend).

Для активации версионирования нажмите кнопку «Enable Versioning» («Включить версионирование»). После применения настройки статус изменится на Enabled.

Для закрытия окна используйте кнопку «Close» («Закрыть»).

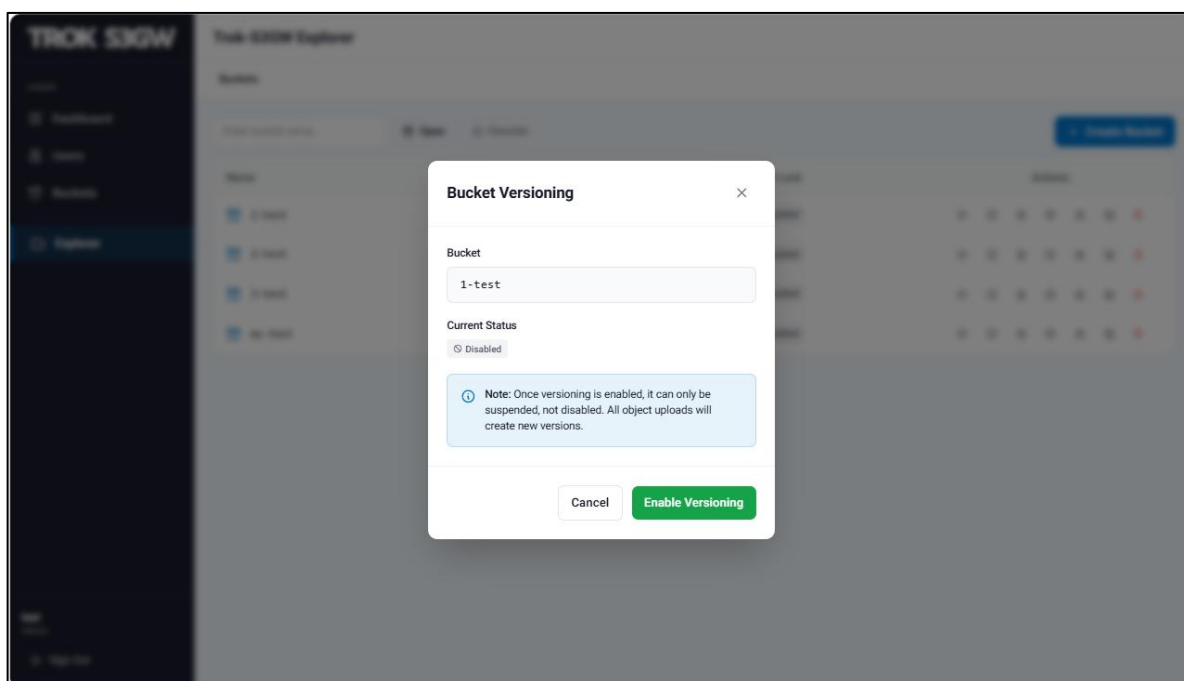


Рисунок 89 – Окно «Bucket Versioning» («Версионирование бакета»)

Примечания:

- Новые версии создаются для всех объектов, загруженных после включения функции; удаление объекта создаёт маркер удаления (delete marker), не удаляя данные физически.
- Доступ к операциям с версиями должен быть разрешён вашей учётной записи согласно политике доступа.

– Хранение версий увеличивает объём занимаемого пространства; при необходимости настройте политику жизненного цикла для автоматического управления старыми версиями.

35.5. Включение и настройка блокировки объектов

Для настройки блокировки объектов откройте раздел «Проводник» («Explorer»). Найдите нужный бакет в списке и в столбце «Действия» («Actions») нажмите на иконку с изображением замка, расположенную справа от названия бакета в соответствующей строке.

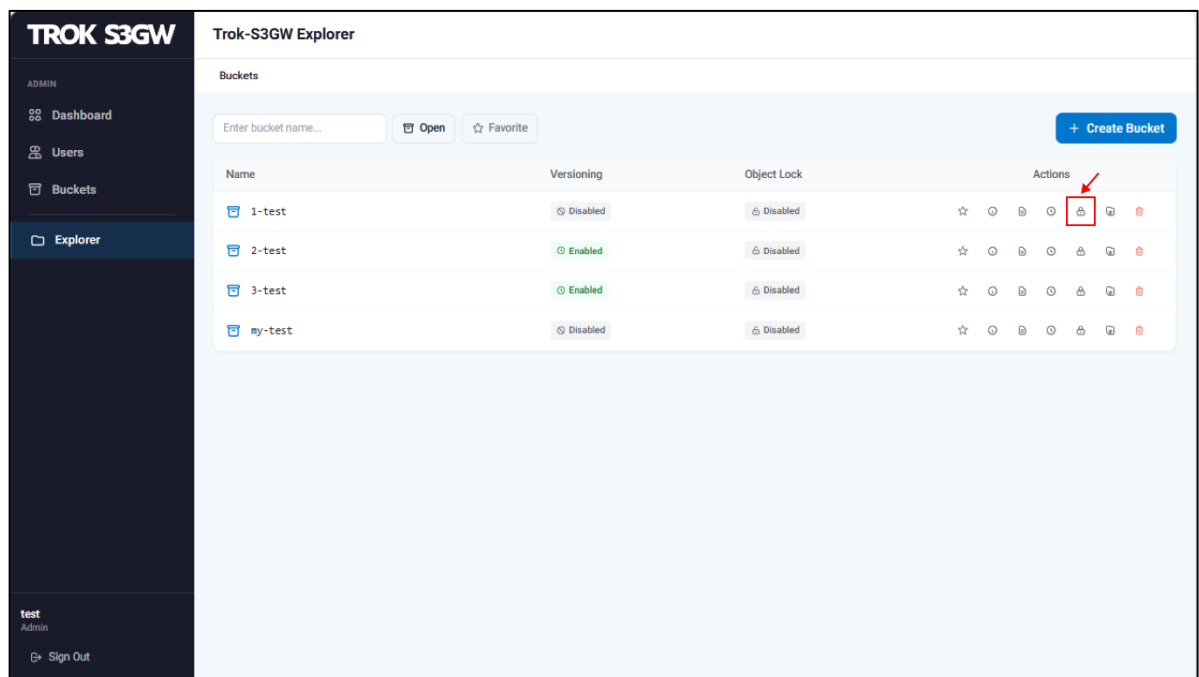


Рисунок 90 – Переход к настройкам Object Lock из списка бакетов

В открывшемся окне «Object Lock Configuration» («Конфигурация блокировки объектов») ознакомьтесь с параметрами:

- Bucket (Имя бакета) – контейнер данных, к которому применяется политика удержания;
- Current Status (Текущий статус) – состояние Object Lock (Enabled – Включено, Disabled – Отключено);
- Retention Mode (Режим удержания) – вариант политики блокировки;
- Retention Period (Период удержания) – срок сохранности (указывается в днях или годах, только одно поле).

Перед включением Object Lock необходимо активировать версионирование.

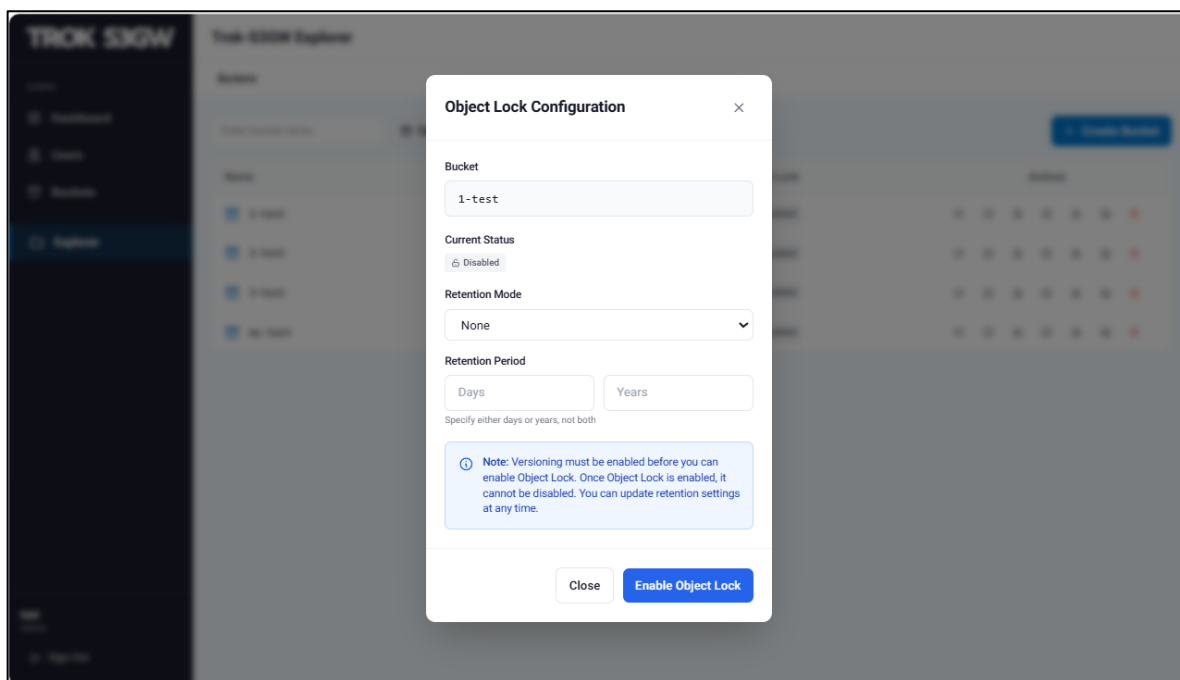


Рисунок 91 – Окно Object Lock с текущим статусом и параметрами удержания
Откройте список «Retention Mode» и выберите режим:

- GOVERNANCE – защита от удаления и изменения версий для обычных пользователей; администраторы с соответствующим правом могут обойти политику;
- COMPLIANCE – строгая защита до окончания срока удержания; обойти или сократить срок нельзя.

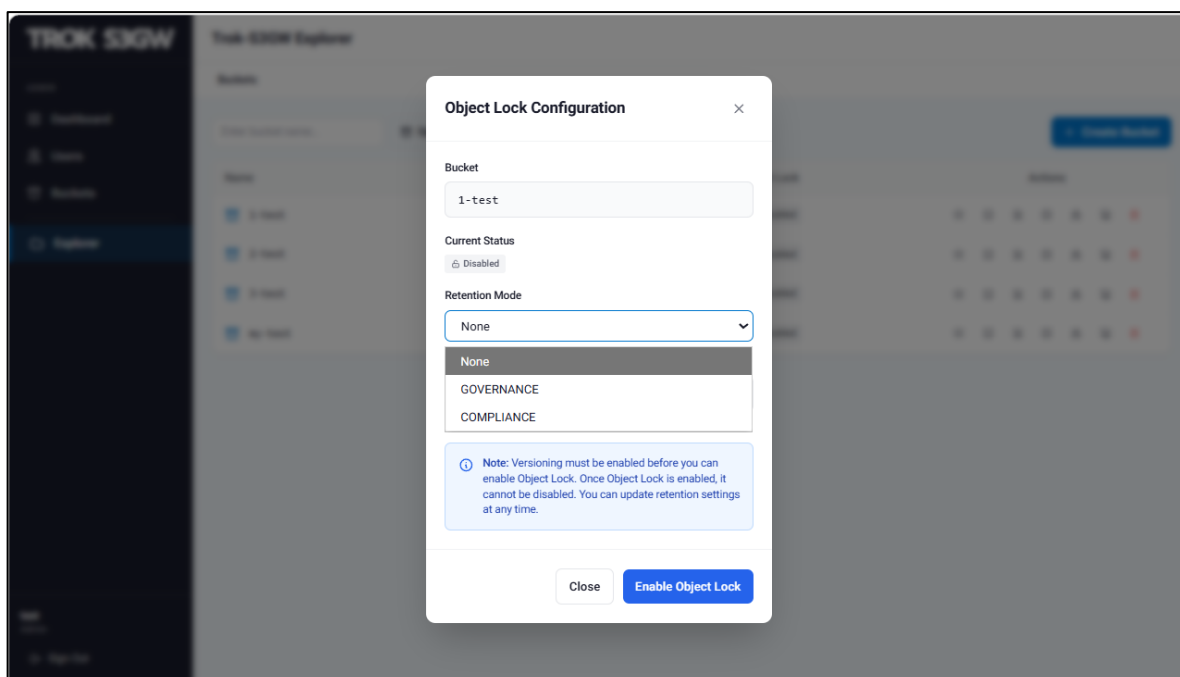


Рисунок 92 – Выбор режима удержания (Governance/Compliance) перед включением

Укажите «Retention Period» (в днях или годах) и нажмите «Enable Object Lock» («Включить Object Lock»). После применения настройки в списке бакетов статус в столбце «Object Lock» изменится на Enabled.

35.6. Просмотр и управление незавершёнными multipart-загрузками

Для проверки и очистки незавершённых загрузок откройте раздел «Проводник» («Explorer»). Найдите нужный бакет в списке и перейдите к столбцу «Действия» («Actions»). Нажмите на иконку с изображением двух перекрывающихся квадратов (иконка «Multipart Uploads»), расположенную справа от названия бакета в соответствующей строке.

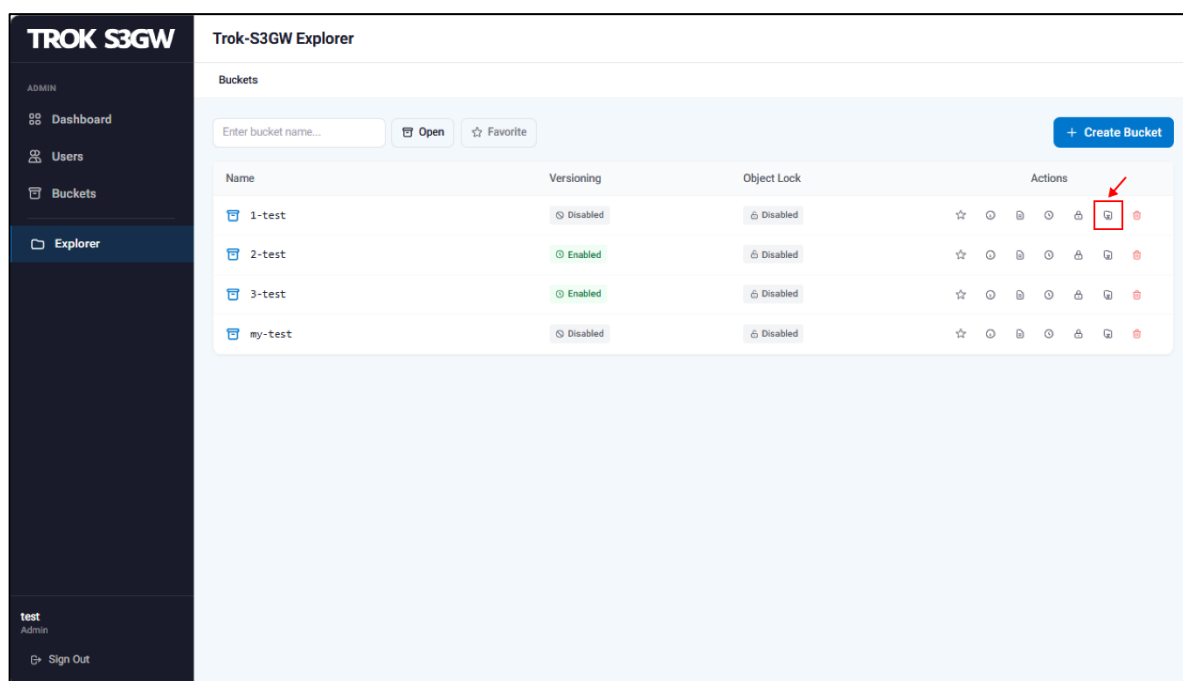


Рисунок 93 – Вызов окна «Multipart Uploads» через меню действий бакета

В открывшемся окне «Multipart Uploads» («Многосоставные загрузки») в верхней части отображается имя выбранного бакета. Ниже выводится список активных незавершённых загрузок. Если таких загрузок нет, отображается сообщение «No Multipart Uploads».

Для обновления списка нажмите кнопку «Refresh» («Обновить»). Для закрытия окна используйте кнопку «Close» («Заккрыть»).



Для удаления бакета выполните действия в разделе «Проводник» («Explorer»).

Найдите нужный бакет в списке и перейдите к столбцу «Действия» («Actions»).

Нажмите на иконку с изображением корзины («Delete»), расположенную справа от названия бакета в соответствующей строке.



В открывшемся окне «Delete Bucket» («Удалить бакет») проверьте имя бакета, которое указано в тексте подтверждения. Если вы уверены, нажмите «Delete Bucket» («Удалить бакет») для подтверждения операции. Чтобы отменить удаление и вернуться к списку, нажмите «Cancel» («Отмена»).

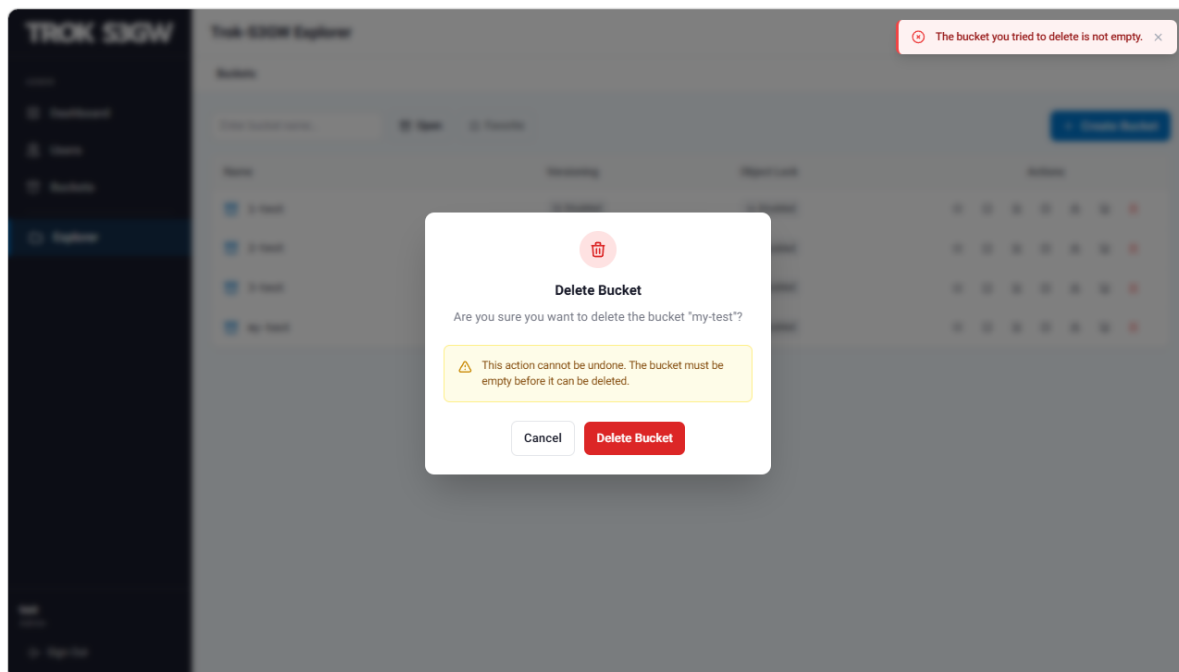


Рисунок 96 – Окно подтверждения удаления бакета

Примечания:

- Перед удалением убедитесь, что в бакете нет объектов и версий объектов, а также маркеров удаления и незавершённых многосоставных загрузок.
- Если включены «Object Lock» или действует период удержания, удаление будет недоступно до снятия ограничений или истечения срока.
- Операция удаления необратима; восстановить бакет после подтверждения нельзя.

36. ОПЕРАЦИИ С ОБЪЕКТАМИ / ФАЙЛАМИ

При входе в бакет в разделе «Проводник» («Explorer») открывается интерфейс управления объектами и папками выбранного контейнера.

Навигационная цепочка (breadcrumbs) позволяет вернуться к списку бакетов или на уровень выше внутри текущего пути. Отображает имя активного бакета.

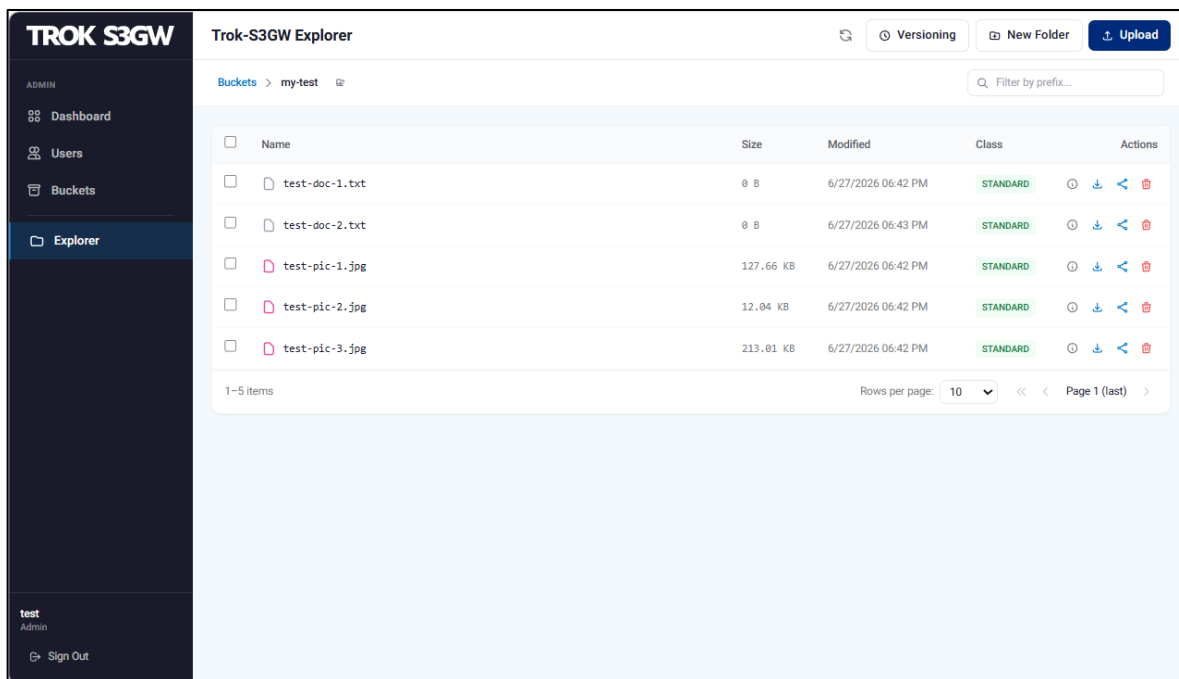


Рисунок 97 – Общий вид интерфейса управления объектами

В правой верхней части экрана расположены элементы:

- Иконка «Обновить» – перезагрузка списка объектов.
- Кнопка «Versioning» («Версионирование») – вызов окна управления версионированием бакета (включение/приостановка).
- Кнопка «New Folder» («Новая папка») – создание новой папки внутри бакета.
- Кнопка «Upload» («Загрузить») – загрузка одного или нескольких объектов в текущую директорию.

Поиск и фильтрация Поле «Filter by prefix...» («Фильтрация по префиксу...») предоставляет фильтрацию списка объектов по префиксу имени (быстрый отбор файлов и папок по началу пути/имени).

В центральной части отображаются элементы хранилища с колонками:

– Name – имя объекта/папки; слева от имени – чекбокс для множественного выбора.

– Size – размер объекта.

– Modified – дата и время последнего изменения.

– Class – класс хранения (например, STANDARD).

– Actions – набор иконок для операций над объектом.

Элементы «Rows per page» («Количество строк на странице») и переключатель страниц позволяют управлять объемом выдачи и переходить по страницам списка.

Доступные операции над объектами

– В столбце «Actions» («Действия») нажмите иконку «i», чтобы открыть окно информации (метаданные, полезные атрибуты).

– В столбце «Actions» («Действия») нажмите иконку «Download» (стрелка вниз) для скачивания файла на локальное устройство.

– Нажмите иконку «Share/Link» («Поделиться/Ссылка») для копирования URL-адреса объекта или вызова механизма формирования ссылки для распространения (при наличии соответствующих прав и настроек политики).

– Нажмите иконку корзины «Delete» («Удалить») и подтвердите действие в открывшемся диалоговом окне.

– Отметьте чекбоксами несколько элементов для одновременной работы с ними (например, пакетное удаление); доступность и состав массовых действий зависят от текущей реализации интерфейса и прав доступа.

Примечание. При включенном версионировании удаление создает маркер удаления; физические данные могут сохраняться до явного удаления версии или по политике жизненного цикла.

36.1. Загрузка файлов

Загрузка файлов в выбранный бакет/папку выполняется двумя эквивалентными способами: через кнопку «Upload» («Загрузить») или посредством перетаскивания (Drag & Drop) на специальную область на странице.

Для выполнения операции загрузки файлов нажмите кнопку «Upload» («Загрузить»), расположенную в правой верхней части экрана выбранного бакета.

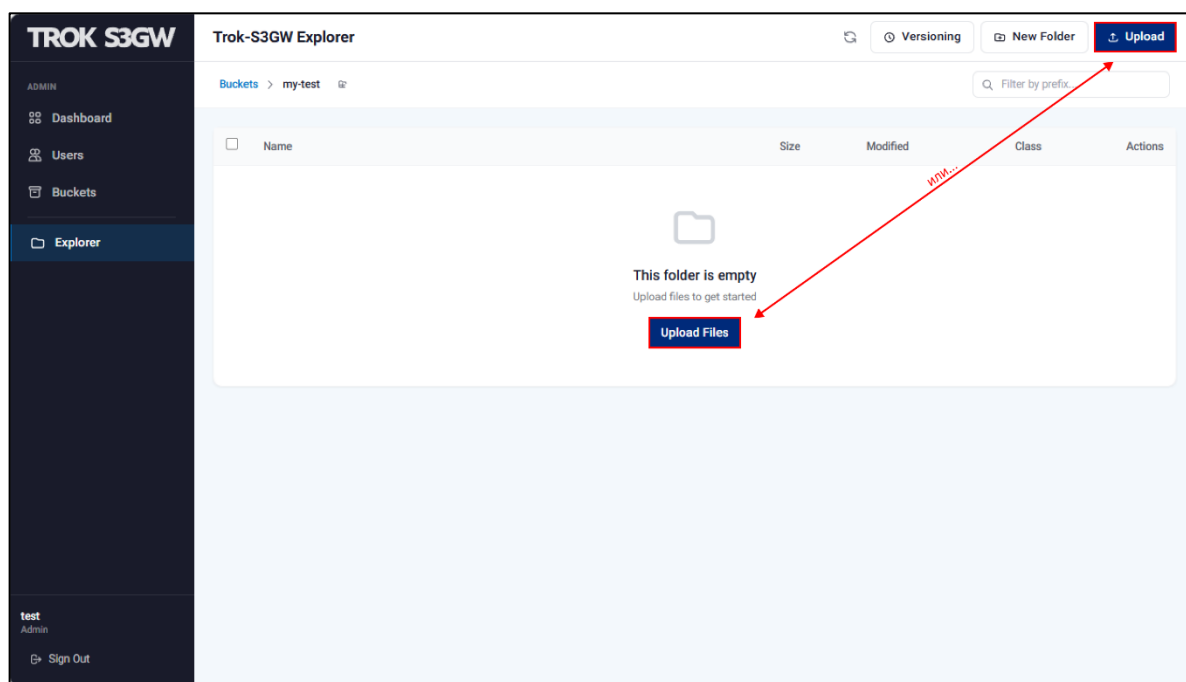


Рисунок 98 – Вызов диалога выбора файлов для загрузки в текущий бакет

После выбора команды «Upload» («Загрузить») откроется стандартное системное окно для выбора одного или нескольких файлов. Подтвердите выбор кнопкой «Open» («Открыть») – загрузка начнется автоматически.

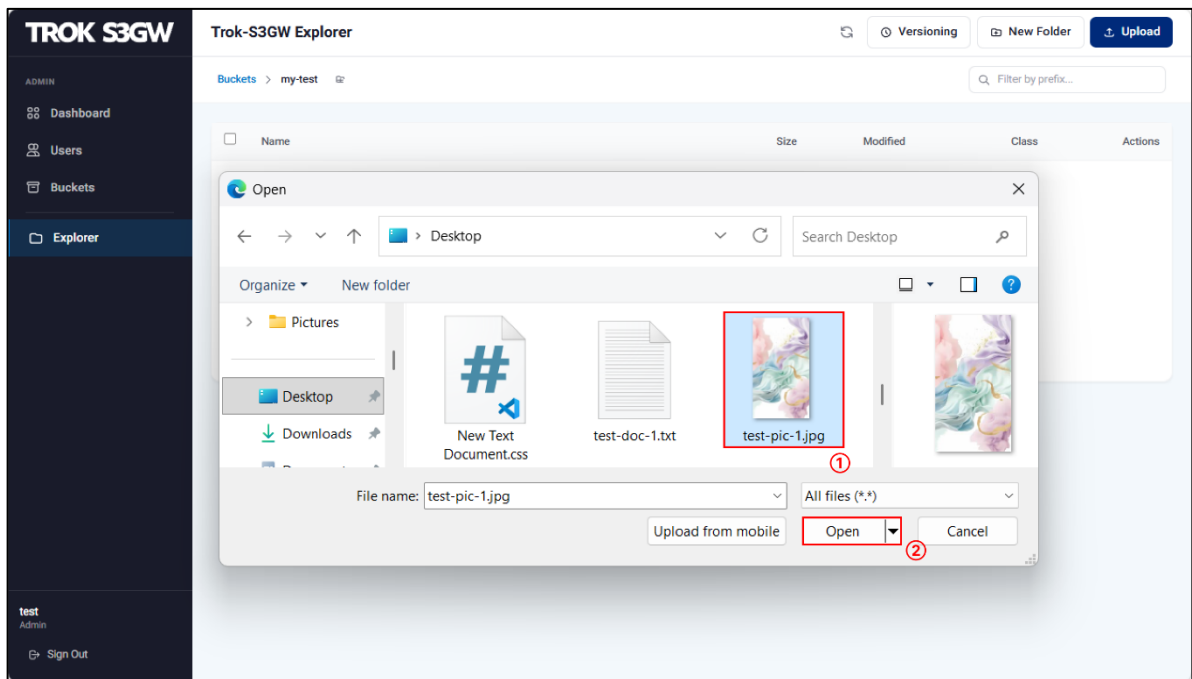


Рисунок 99 – Выбор файла для загрузки

В качестве альтернативы используйте перетаскивание (Drag & Drop). Для этого выделите файлы в проводнике вашей операционной системы и перетащите их в область «Drop files to upload», отображаемую в центре страницы. После отпускания кнопки мыши файлы будут загружены в текущую папку бакета без дополнительных подтверждений.

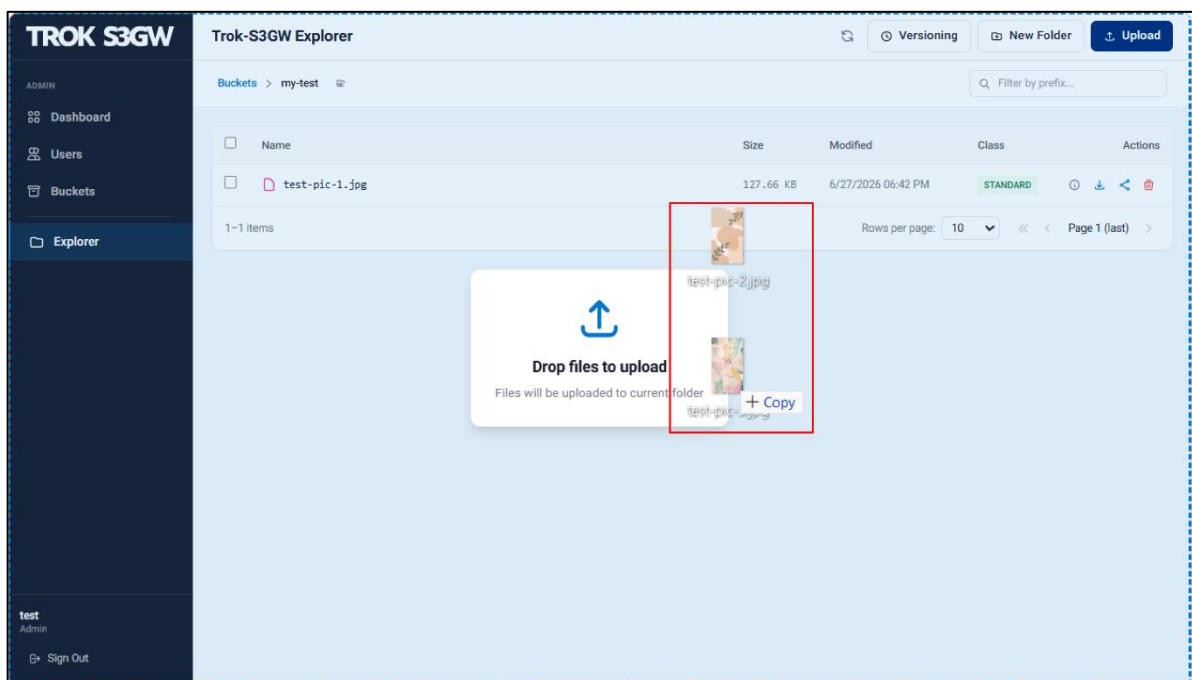


Рисунок 100 – Перетаскивание файлов в зону «Drop files to upload» для мгновенной загрузки

По завершении операции загруженные объекты появляются в таблице с указанием имени, размера, времени изменения и класса хранения. В столбце «Actions» («Действия») становятся доступны операции: просмотр сведений об объекте, скачивание, копирование ссылки и удаление.

36.2. Просмотр информации об объекте

Для выполнения операции просмотра сведений об объекте откройте нужный бакет в разделе «Explorer» «Проводник» и найдите строку с требуемым файлом в таблице объектов. Нажмите значок «i» в столбце «Actions» («Действия»), расположенный справа от имени объекта.

После выбора команды «Просмотреть информацию» открывается модальное окно «Object Info» («Информация об объекте») с детализированными параметрами выбранного объекта.

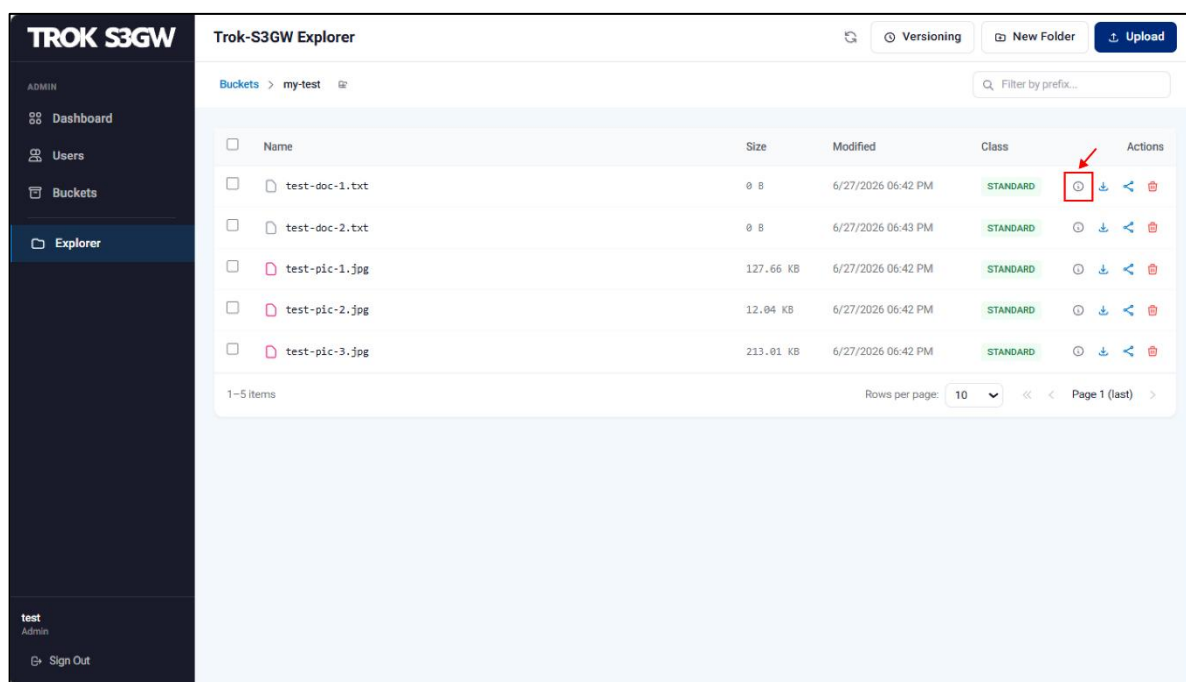


Рисунок 101 – Вызов окна «Object Info» («Информация об объекте») из списка объектов

В появившемся окне «Object Info» («Информация об объекте») отображаются основные атрибуты: ключ объекта (Key), размер (Size), дата и время последнего изменения (Last Modified), тип содержимого (Content Type), класс хранения (Storage Class), ETag, а при необходимости – версия объекта (Version ID) и иные

служебные параметры. Эти сведения доступны только для просмотра и помогают оценить состояние и свойства файла.

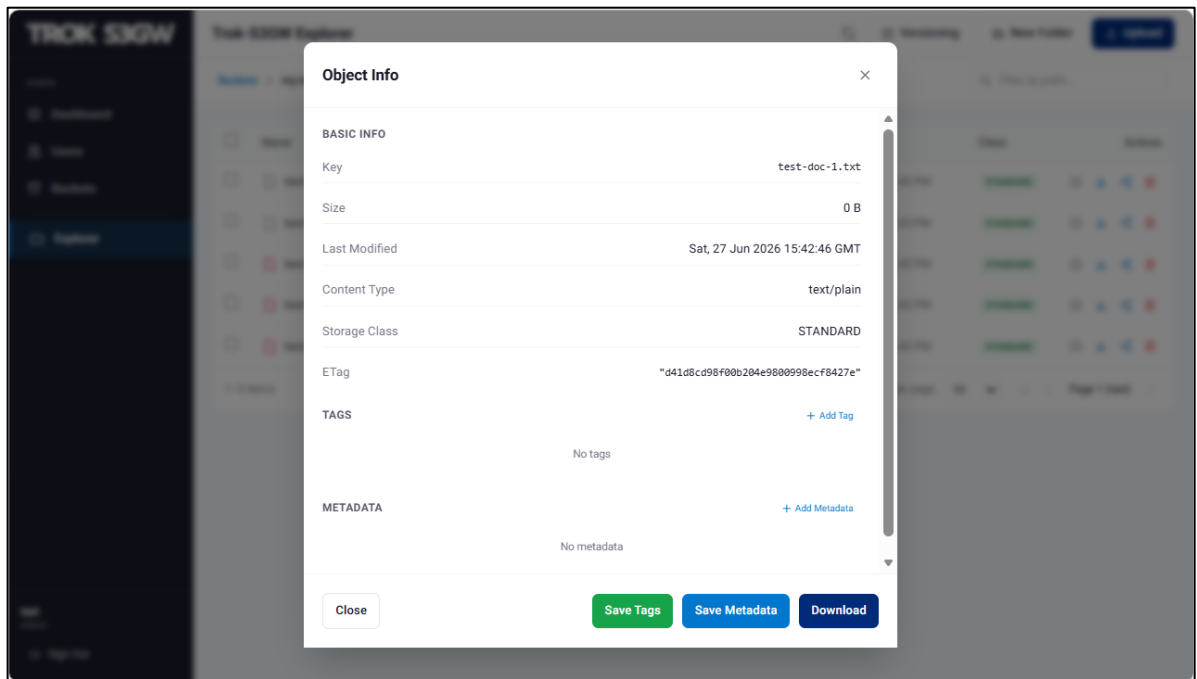


Рисунок 102 – Просмотр базовой информации и атрибутов объекта

Для добавления тегов перейдите в раздел «Tags» («Теги») внутри окна «Object Info» («Информация об объекте») и нажмите кнопку «+ Add Tag» («+ Добавить Тег»). В открывшейся строке укажите пару «Key» и «Value» (ключ и значение тега). При необходимости добавьте несколько строк. Сохраните изменения кнопкой «Save Tags» («Сохранить теги»).

Для добавления пользовательских метаданных откройте раздел «Metadata» («Метаданные») и нажмите кнопку «+ Add Metadata» («+ Добавить метаданные»). Заполните поля «Key» и «Value» для каждой пары метаданных. Подтвердите внесенные значения кнопкой «Save Metadata» («Сохранить метаданные»).

Для завершения работы с окном «Object Info» используйте кнопку «Close» («Заккрыть»). При необходимости вы можете скачать файл из этого же окна с помощью кнопки «Download» («Скачать»). После сохранения изменений теги и метаданные немедленно применяются и будут отображаться при повторном открытии сведений об объекте.

36.3. Скачивание объекта

Для выполнения операции скачивания объекта откройте нужный бакет в разделе «Explorer» и найдите строку с требуемым файлом в таблице объектов. В столбце «Actions» («Действия») нажмите значок со стрелкой вниз «Download» («Скачать»), расположенный справа от имени объекта.

После выбора команды «Скачать» браузер инициирует загрузку файла в папку по умолчанию (обычно «Загрузки»). Если в настройках браузера включен запрос пути сохранения, укажите каталог и подтвердите действие.

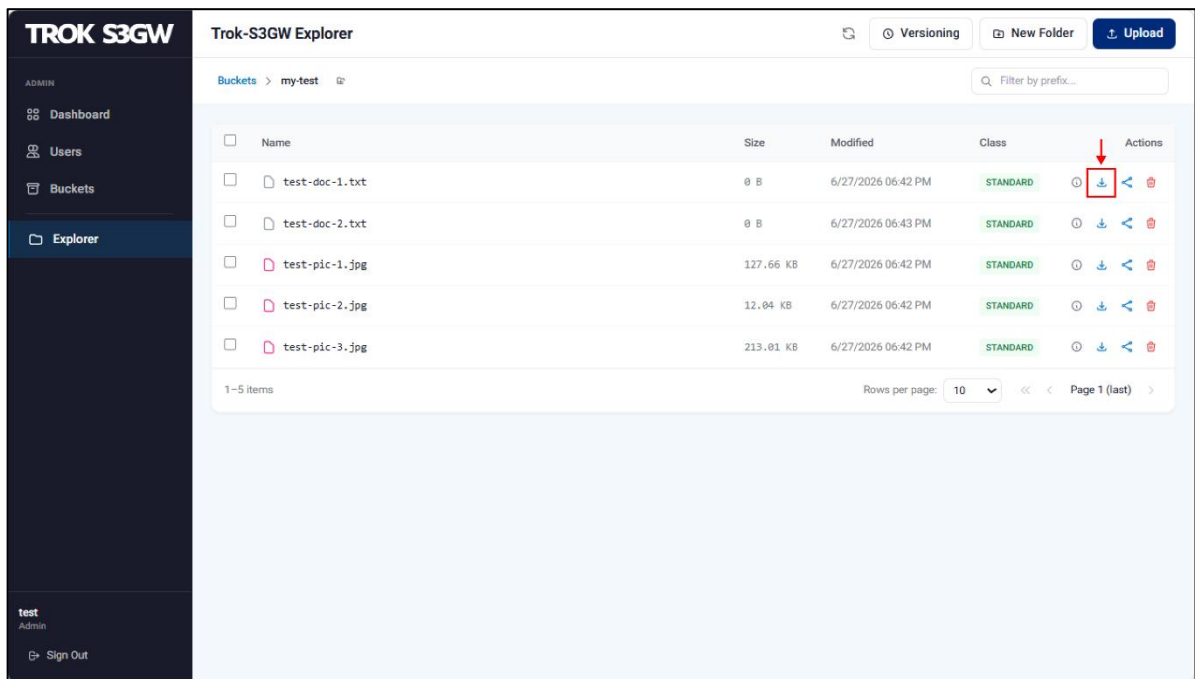


Рисунок 103 – Вызов скачивания объекта из списка: нажатие значка «Download» в столбце «Actions»

После начала загрузки в правом верхнем углу отобразится панель «Downloads» («Загрузки») браузера со статусом текущей операции. В панели будет показано имя скачиваемого файла и доступна ссылка «Open file» («Открыть файл») для немедленного просмотра содержимого. При необходимости разверните панель и убедитесь, что загрузка завершилась успешно.

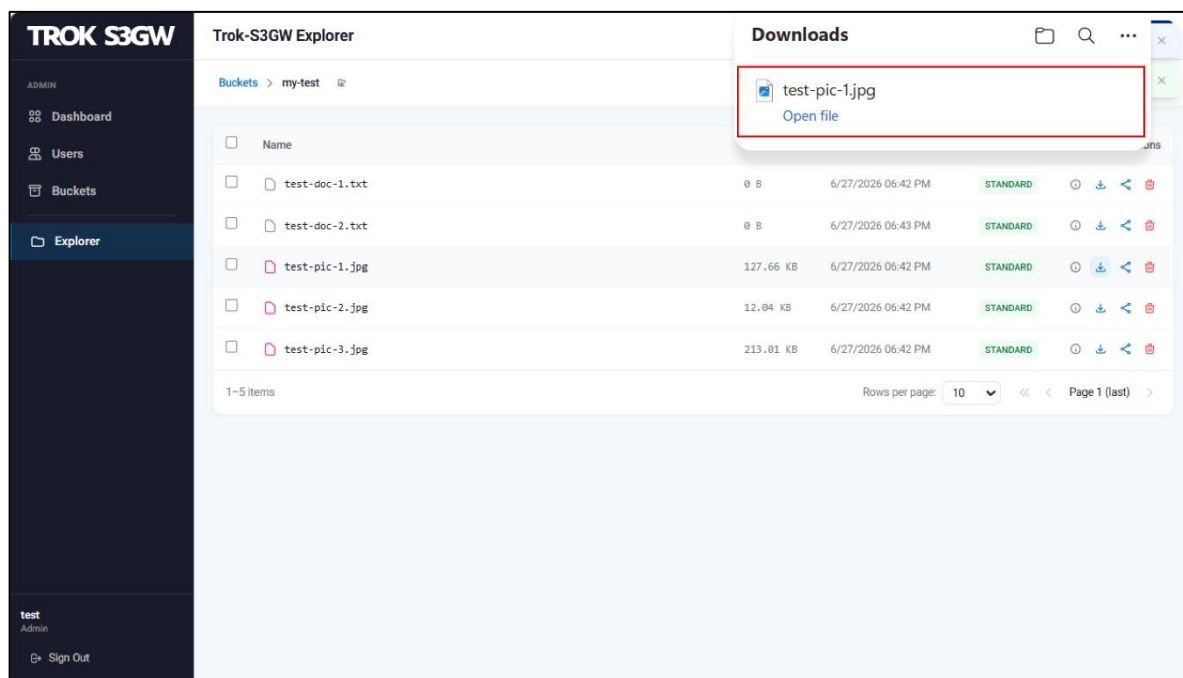


Рисунок 104 – Панель загрузок браузера: подтверждение успешного скачивания и переход по ссылке «Open file»

36.4. Генерация предварительно подписанной ссылки (Presigned URL)

Для выполнения операции генерации предварительно подписанной ссылки откройте нужный бакет в разделе «Explorer» и найдите строку с требуемым объектом в таблице. В столбце «Actions» («Действия») нажмите на значок «Generate presigned URL» («Сгенерировать предварительно подписанный URL»), расположенный справа от имени объекта. После выбора команды открывается модальное окно настройки параметров ссылки.

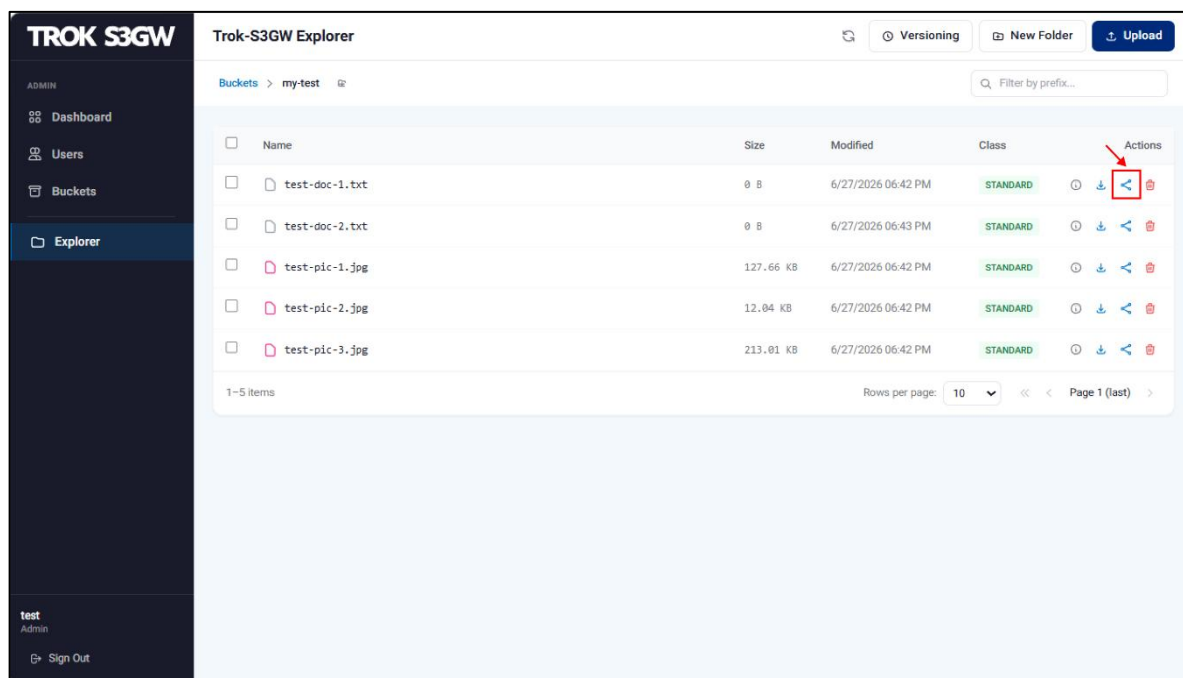


Рисунок 105 – Вызов окна генерации пресайнд-ссылки из списка объектов

В появившейся форме «Generate presigned URL» («Сгенерировать предварительно подписанный URL») автоматически отображаются параметры «Bucket» (имя бакета) и «Key» (ключ объекта). Следующий обязательный шаг – задание времени жизни ссылки. В поле «Expiration» («Истечение срока действия») укажите числовое значение и выберите единицы измерения из раскрывающегося списка (например, минуты, часы).

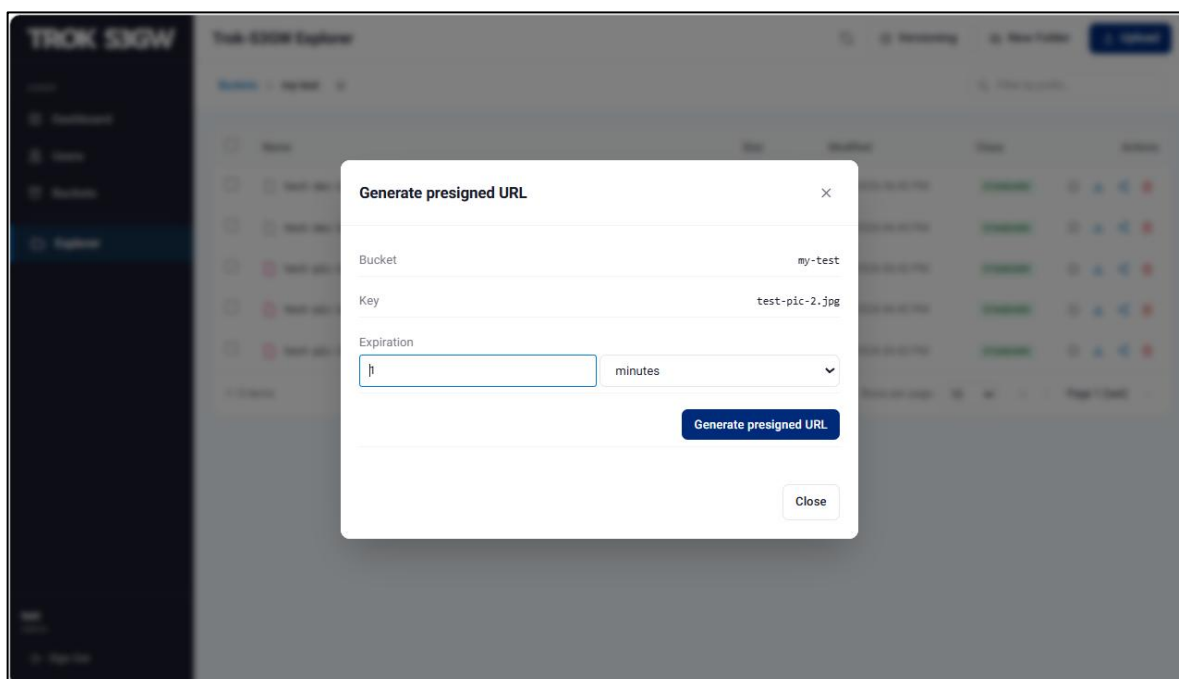


Рисунок 106 – Заполнение параметров истечения срока действия пресайнд-ссылки

Для завершения процесса нажмите кнопку «Generate presigned URL» («Сгенерировать ссылку»). После успешного выполнения под формой появится поле «URL» со сгенерированной ссылкой и значком копирования. Скопируйте ссылку и передайте получателю. Для закрытия окна используйте кнопку «Close» («Закрыть»). Ссылка действует до истечения заданного времени.

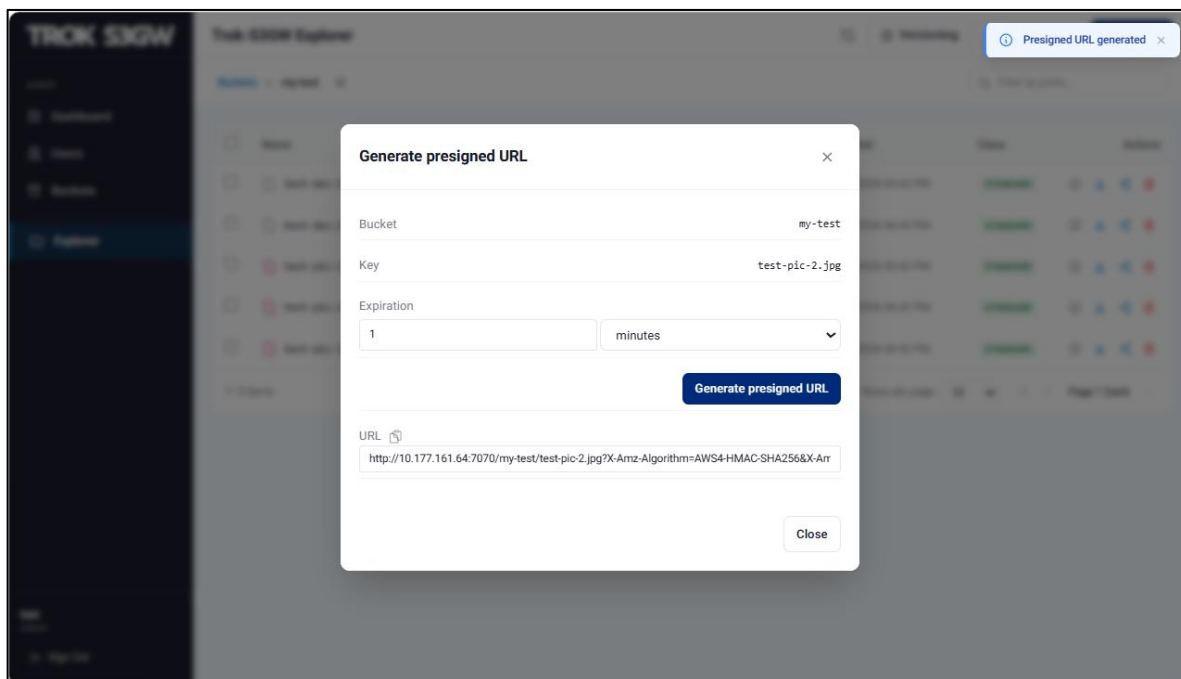


Рисунок 107 – Получение и копирование сгенерированной пресайнд-ссылки

36.5. Удаление объекта

Для выполнения операции удаления объекта откройте раздел «Explorer» и перейдите в нужный бакет (если включено версионирование, то необходимо удалить все версии файла, если включена блокировка объекта, то для удаления необходимо отключить блокировку объекта). В таблице объектов найдите требуемый файл и в столбце «Actions» («Действия») нажмите значок корзины «Delete» («Удалить»), расположенный справа от имени объекта.

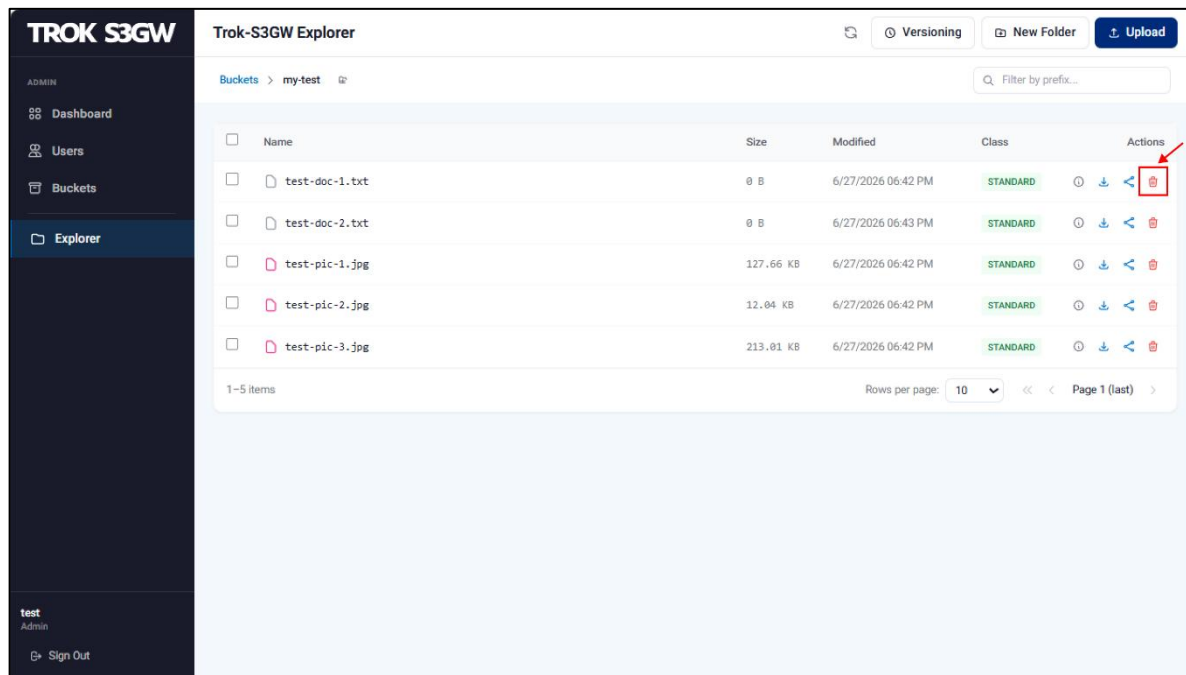


Рисунок 108 – Вызов удаления объекта из списка бакета

После выбора команды «Удалить» открывается модальное окно «Delete Objects» с предупреждением о необратимости операции. В тексте окна отображается ключ (имя) удаляемого объекта, например «test-pic-3.jpg».

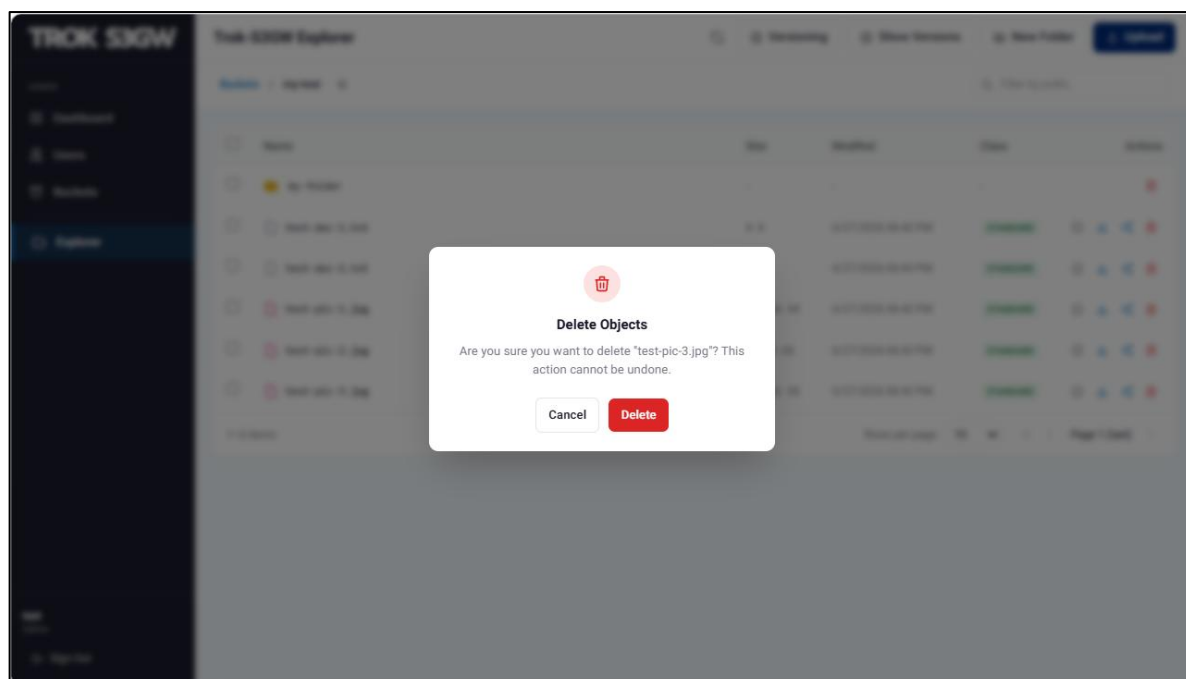


Рисунок 109 – Подтверждение удаления выбранного объекта

Для завершения процедуры нажмите кнопку «Delete» («Удалить»). Для отмены операции используйте кнопку «Cancel» («Отмена»). После

подтверждения объект безвозвратно удаляется и исчезает из общего списка файлов выбранного бакета.