

Release 1.4.1 (version 0.9)

Скала[^]р Спектр

Руководство администратора

Листов 57

СОДЕРЖАНИЕ

ГЛОССАРИЙ.....	4
1 ВВЕДЕНИЕ	6
1.1 Область применения программы.....	6
1.2 Уровень подготовки администраторов	6
2 УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ.....	7
2.1 Условия, необходимые для выполнения программы	7
2.2 Используемые сетевые порты	7
3 УСТАНОВКА ПРОГРАММЫ	8
3.1 Установка основного ПО "Спектр"	8
3.2 Установка дополнительного ПО "Спектр"	9
3.2.1 Установка PgAdmin.....	10
3.2.2 Установка Агента "Спектра"	11
3.2.3 Установка Агента уведомлений "Спектра"	13
3.3 Настройка SSL	16
3.3.1 Генерация сертификатов	16
3.3.2 Установка сертификатов	16
3.4 Настройка интеграции с KeyCloak и Avanpost FAM	17
3.4.1 Настройка интеграции с KeyCloak	17
3.4.2 Настройка интеграции с Avanpost FAM	18
4 УСТАНОВКА ОБНОВЛЕНИЙ ПРОГРАММЫ	19
4.1 Обновление ПО "Спектр"	19
4.2 Обновление дополнительного ПО "Спектр"	19
4.2.1 Обновление PgAdmin	20
5 ПРОВЕРКА ПРОГРАММЫ	21
5.1 Отображение веб-интерфейса	21
5.2 Проверка установки модулей.....	21
5.3 Проверка состояния и управление сервисами "Спектра".....	22
5.3.1 Ручная остановка, запуск или перезапуск сервисов.....	22
5.3.2 Проверка статусов сервисов.....	22
5.4 Проверка состояния дополнительного ПО "Спектра".....	24
5.4.1 Проверка состояния агента "Спектра".....	24

5.4.2	Проверка состояния агента уведомлений "Спектра"	24
5.4.3	Проверка состояния PgAdmin	25
5.5	Просмотр состояния системного ПО	26
5.6	Просмотр логов сервисов.....	27
5.6.1	Просмотр логов основного ПО "Спектра".....	27
5.6.2	Просмотр логов дополнительного ПО "Спектра"	28
5.7	Просмотр файлов конфигурации.....	30
6	УСТРАНЕНИЕ НЕИСПРАВНОСТЕЙ "СПЕКТРА"	31
6.1	Не открывается веб-интерфейс "Спектра".....	31
6.1.1	Описание неисправности.....	31
6.1.2	Способ устранения.....	31
6.2	В веб-интерфейсе отсутствуют данные об объектах управления	31
6.2.1	Описание неисправности.....	31
6.2.2	Способ устранения.....	32
6.3	В веб-интерфейсе отображаются сообщения об ошибках при получении данных	32
6.3.1	Описание неисправности.....	32
6.3.2	Способ устранения.....	32
6.4	В веб-интерфейсе отображаются устаревшие данные об объектах управления.....	33
6.4.1	Описание неисправности.....	33
6.4.2	Способ устранения.....	33
6.5	Операция завершается с сообщением об ошибке Агента "Спектра"	34
6.5.1	Описание неисправности.....	34
6.5.2	Способ устранения.....	34
	ПРИЛОЖЕНИЯ	35
	Приложение 1. Примеры файлов конфигураций	35
	Приложение 2. Настройка пользователей в KeyCloak	39
	Приложение 3. Настройка Avapost FAM.....	46

ГЛОССАРИЙ

Термины и сокращения, которые используются для описания внутреннего устройства и функций программы:

Термин	Описание, комментарий
Ansible	ПО с открытым исходным кодом, которое автоматизирует установку программного обеспечения, управление конфигурацией и развёртывание приложений.
API	A pplication P rogramming I nterface, программный интерфейс приложения.
Corosync	Программное обеспечение, позволяющее объединять сервера в кластер.
CPU	C entral P rocessing U nit, центральный процессор.
Dashboard	Графический интерфейс, отображающий ключевые метрики и показатели процессов.
IP-адрес	Уникальный сетевой адрес узла в компьютерной сети, построенной на основе стека протоколов TCP/IP .
JSON	J ava S cript O bject N otation. Текстовый формат обмена данными, который представлен наборами пар "ключ-значение".
NET	Сетевое соединение.
NTP	N etwork T ime P rotocol. Сетевой протокол для синхронизации внутренних часов компьютера с использованием сетей с переменной латентностью.
Resource	Программное обеспечение – менеджер ресурсов кластера.
pgAdmin	Программное обеспечение с открытым исходным кодом для разработки и администрирования баз данных PostgreSQL . Имеет удобный графический интерфейс, который позволяет выполнять задачи мониторинга, обслуживания, конфигурирования сервера PostgreSQL , а также создавать и выполнять SQL-запросы.
Postgres, PostgreSQL	Свободная объектно-реляционная система управления базами данных.
RAM	R andom A ccess M emory, оперативная память.
ROM	R ead O nly M emory, энергонезависимая память (жёсткий диск).
RPM	R ed H at P ackage M anager. Формат пакетов программного обеспечения, используемые в ряде Linux-дистрибутивов.
SSH	S ecure S hell – сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой и туннелирование TCP-соединений.

SSL	Secure Sockets Layer – уровень защищенных сокетов – криптографический протокол, который подразумевает более безопасную связь. Он использует асимметричную криптографию для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений.
web-IPMI	Интерфейс управления ЭВМ, предназначенный для автономного мониторинга и управления функциями, встроенными непосредственно в аппаратное и микропрограммное обеспечения серверных платформ.
Агент Спектра	<p>Специализированное программное обеспечение, которое устанавливается на узлы кластера и используется в работе ПО «Скала^р Спектр» для выполнения следующих задач:</p> <ul style="list-style-type: none"> • получение актуальной конфигурации кластера; • получение и обновление конфигураций СУБД и программы, управляющей пулом соединений PgBouncer; • выполнение операций с кластером, сервисами, узлами; • предоставление метрик о состоянии кластера, узлов, сервисов СУБД, баз данных для системы мониторинга.
База данных	Объект сервиса СУБД, в котором пользователи создают таблицы, представления, хранимые процедуры.
Визион	ПО системы мониторинга "Скала^р Визион".
Инстанс	Экземпляр PostgreSQL , запущенный на одном из серверов сегментов, хранящий и обрабатывающий свою часть данных. Экземпляр класса в объектно-ориентированном программировании.
Кластер	Объект управления верхнего уровня; группа узлов, на которой запускаются сервисы СУБД. Для обеспечения отказоустойчивости сервисов СУБД используется инструмент Pacemaker/Corosync .
ОС	О перационная С истема.
ПО	П рограммное О беспечение.
ПАК	П рограммно- А ппаратный К омплекс – набор технических и программных средств, работающих совместно для выполнения одной или нескольких сходных задач.
Сервис СУБД	Целостная служба СУБД, работающая на конкретном кластере.
СУБД	С истема У правления Б азами Д анных. Набор программ, позволяющий организовывать, контролировать и администрировать базы данных.
Узел	Сервер, входящий в кластер, на который устанавливается кластерное ПО Pacemaker/Corosync и ПО СУБД для запуска экземпляров СУБД и организации из них сервисов СУБД.
ЭВМ	Э лектронно- В ычислительная М ашина.

1 ВВЕДЕНИЕ

1.1 Область применения программы

Основным назначением Модуля управления кластером "Скала^р Спектр" (далее - "Спектр") является предоставление возможности управления кластерами **Pacemaker / Corosync** и СУБД **PostgreSQL** в составе Машины Баз Данных МБД.П ПАК семейства "СКАЛА" персоналом, к квалификации которого не предъявляется серьезных требований, в том числе персоналом сторонних организаций.

Основными задачами, решаемыми "Спектром", являются:

- Регистрация и ведение списков объектов управления;
- Обеспечение мониторинга объектов управления;
- Управление жизненным циклом кластеров СУБД **PostgreSQL**;
- Управление сервисами СУБД в кластере;
- Управление базами данных.

"Спектр" должен обеспечивать реализацию следующих возможностей:

- Регистрация (импорт) существующего кластера;
- Регистрация кластеров **Postgres** из **JSON**, предоставляемого **REST API** Модуля Управления ПАК;
- Управление кластерами для выполнения операций «**старт кластера**», «**остановка кластера**», «**импорт кластера**», «**удаление кластера из Спектра**»;
- Управление узлами кластера для выполнения операций «**старт узла кластера**», «**остановка узла кластера**», «**вывод узла на обслуживание**», «**возвращение узла из обслуживания**»;
- Управление сервисами СУБД для выполнения операций «**старт сервиса**», «**остановка сервиса**», «**перезапуск сервиса**», «**создание отчета pgBadger**», «**остановка экземпляра сервиса**», «**запуск экземпляра сервиса**», «**изменение конфигурации сервиса**», «**применение изменённой конфигурации сервиса**»;
- Управление **Standalone** экземплярами сервиса;
- Управление **pgBouncer**;
- Управление резервным копированием сервисов, в том числе по расписанию;
- Отображение данных из Модуля Мониторинга;
- Добавление сервиса СУБД в кластер / удаление сервиса СУБД из кластера;
- Аудит действий с БД.

1.2 Уровень подготовки администраторов

Администратор должен обладать следующими навыками:

- Установка, настройка системного программного обеспечения в ОС Linux;
- Просмотр и редактирование файлов с использованием команд **ls**, **cat**, **less**, **vim**, **nano**.

2 УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ

2.1 Условия, необходимые для выполнения программы

Работа "Спектра" в основном режиме осуществляется путем отправки управляющих воздействий через **SSH** или при помощи **HTTP**-запросов к Агенту "Спектра". Это означает, что сервер с развернутым "Спектром" должен находиться в одной сети с обслуживаемыми серверами.

Оптимальным решением является выделение под "Спектр" отдельного виртуального сервера в рамках хоста управления ПАК.

Тип реализующей ЭВМ: x86-64.

Минимальная аппаратная конфигурация сервера для установки "Спектра":

- **CPU**: от 2 ядер;
- **RAM**: от 16 Гб;
- **ROM**: от 600 Гб SSD для системных компонент;
- **NET**: от 100 Мбит/с Ethernet.

Для установки "Спектра" требуется наличие ПО ОС "**RedOS 7.3**", "**АльтЛинукс 8 СП релиз 9**" или "**АльтЛинукс 8 СП релиз 10**" на хосте "Спектра", а также наличие запущенной в составе Узла управления виртуальной машины с ПО "Геном".

2.2 Используемые сетевые порты

ПОРТ	ПРОТОКОЛ	НАЗНАЧЕНИЕ
8092	HTTPS	На хосте "Спектра", для доступа к веб-интерфейсу "Спектра".
9000	HTTP	На узлах кластера, для доступа к API Агента "Спектра"
22	SSH	На узлах кластера, для установки Агентов "Спектра"
	TCP/IP	На узлах кластера, для доступа к PostgreSQL (порты зависят от настроек сервисов в кластере)
80	HTTP	На хосте с ПО "Геном", для доступа к HTTP репозиториям пакетов для установки "Спектра".
52888	HTTP	На хосте с ПО "Геном", для доступа к API ПО "Геном"

3 УСТАНОВКА ПРОГРАММЫ



Внимание!

Перед началом установки ПО "Скала^р Спектр" необходимо убедиться в доступности хоста с ПО "Скала^р Геном" с хоста "Спектра". Проверить доступность можно при помощи команды **ping**.

IP-адрес хоста с ПО "Скала^р Геном" можно узнать в инсталляционной карте Машины МБД.П.

Если в выводе команды **ping** содержатся ошибки, необходимо убедиться в том, что:

- виртуальная машина с ПО "Скала^р Геном" запущена на Узле управления. IP-адрес узла управления может быть уточнен в инсталляционной карте;
- обеспечена сетевая связность между хостами ПО "Скала^р Геном" и "Спектра";
- фаерволл не блокирует доступ по протоколу **HTTP** и портам 80 и 52888 на хосте с ПО "Скала^р Геном".

В приведенных в данном руководстве примерах команд в качестве интерпретатора команд терминала используется **Bash**.

Примеры команд в ряде случаев приведены вместе с приглашением интерпретатора команд, которое может отличаться у пользователя. В данном руководстве в приглашении интерпретатора команд выводятся имя пользователя, имя узла, на котором запускаются команды, и текущий каталог, в котором находится пользователь. Приглашение интерпретатора команд заканчивается символом "#", что означает, что все команды выполняются под привилегированным пользователем **root**. Если приглашение заканчивается символом "\$", это означает что пользователь, выполняющий установку, не является привилегированным и при запуске команд необходимо повышение привилегий пользователя при помощи команды **sudo**.

При вводе команд пользователю не требуется вводить приглашение интерпретатора. Например, если в руководстве приводится команда

```
[root@spectrum-host ~]# mkdir -p spectrum-dist
```

это означает, что в терминале необходимо вводить только

```
mkdir -p spectrum-dist
```

3.1 Установка основного ПО "Спектр"

Дистрибутив "Спектра" представляет собой **tar.gz** архив, который необходимо распаковать в произвольный каталог на хосте "Спектра", например в **/root/spectrum-dist**.



Примечание

В данном руководстве приводятся примеры команд для версии "Спектра" 1.1.70. Номер версии используется в имени файла с архивом дистрибутива "**spectrum-dist-altlinux8.4-v1.1.70.tar.gz**", а также имени каталога, в который распаковывается архив.

На момент выполнения установки или обновления "Спектра" пользователем версия может отличаться.

Распакованный архив содержит все **RPM**-пакеты, необходимые для работы "Спектра", а также скрипт установки **install.sh**.

Для установки необходимо перейти в директорию, в которую был распакован архив, и выполнить скрипт **install.sh**:

```
[root@spectrum-host ~]# mkdir -p spectrum-dist
[root@spectrum-host ~]# tar -xvf spectrum-dist-altlinux8.4-v1.1.70.tar.gz -C
./spectrum-dist
[root@spectrum-host ~]# cd spectrum-dist/spectrum-rpm-dist-v1.1.70
[root@spectrum-host spectrum-rpm-dist-v1.1.70]# ./install.sh
```

В ходе установки будут созданы все необходимые каталоги, установлено необходимое ПО (**Ansible** и необходимые для работы **python**-библиотеки), установлено ПО СУБД **Postgres**, создана необходимая для работы "Спектра" база данных.

В ходе установки пользователю необходимо ввести IP-адрес, по которому доступно ПО модуля управления ПО "Скала^р Геном-Управление":

```
[root@spectrum-host spectrum-0.1.0]# ./install.sh
Enter host IP for Genome: 1.2.3.4
```

Также в ходе установке необходимо ввести IP-адрес ПО "Скала^р Визион":

```
Enter host IP for Vision:
1.2.3.5
```

3.2 Установка дополнительного ПО "Спектр"

Для корректной работы всех функций "Спектра" требуется установка на узлы кластеров, которыми управляет "Спектр", следующего дополнительного программного обеспечения:

- Агент "Спектра",
- Агент уведомлений "Спектра",

а также установка на хосте "Спектра" ПО **PgAdmin** для администрирования СУБД.

Установка дополнительных компонентов "Спектра" выполняется в два этапа:

- установка Агента "Спектра" на узлы кластера, выполняется до импорта кластера в "Спектр";
- установка Агента уведомлений "Спектра", выполняется после импорта кластера в "Спектр".

Установка дополнительных компонентов "Спектра" выполняется с хоста "Спектра" из терминала, аналогично установке самого "Спектра".

В каталоге дистрибутива "Спектра" при помощи любого текстового редактора необходимо отредактировать файл **inventory_samples/cluster.yml**, введя параметры кластера и узлов, узнать их можно в инсталляционной карте Машины МБД.П (должны быть указаны IP-адреса в сети управления).

Если каталог, в который распакован дистрибутив "Спектра" – это **/root/spectrum-dist**, то полный путь к нужному файлу будет: **/root/spectrum-dist/spectrum-rpm-dist-v1.1.70/inventory_samples/cluster.yml**. Проверить наличие файла можно командой **ls**:

```
ls /root/spectrum-dist/spectrum-rpm-dist-v1.1.70/inventory_samples/cluster.yml
```

Просмотреть содержимое файла **cluster.yml** можно, например, командой **cat**:

```
cat /root/spectrum-dist/spectrum-rpm-dist-v1.1.70/inventory_samples/cluster.yml
```



Примечание

В примере установка дополнительного ПО для "Спектра" выполняется для одного кластера. Если под управлением "Спектра" находятся несколько кластеров, то шаги по установке дополнительного ПО на узлы (Агента "Спектра" и Агента уведомлений "Спектра") необходимо выполнить для каждого кластера.

3.2.1 Установка PgAdmin

Дистрибутив **PgAdmin** представляет собой **tar.gz** архив **pgadmin_dist_alt84_rpm.tar.gz**, который необходимо распаковать в произвольный каталог на хосте "Спектра", например в **/root/spectrum-dist**.



Примечание

В данном руководстве приводятся примеры команд для архива с названием **pgadmin_dist_alt84_rpm.tar.gz**.

Название архива может отличаться, например, может быть другая версия.

Распакованный архив содержит образы **docker**-контейнеров, необходимых для работы **PgAdmin**, а также скрипт установки **deploy.sh**.

Для установки необходимо перейти в директорию, в которую был распакован архив, и выполнить скрипт **deploy.sh**:

```
[root@spectrum-host ~]# mkdir -p spectrum-dist

[root@spectrum-host ~]# tar -xvf pgadmin-dist_altlinux84.tar.gz -C ./spectrum-dist

[root@spectrum-host ~]# cd spectrum-dist/pgadmin-dist

[root@spectrum-host spectrum-rpm-dist-v1.1.70]# ./deploy.sh
```

3.2.2 Установка Агента "Спектра"

Для установки Агента "Спектра" в файле **cluster.yml**, описанном в п. 3.2, необходимо заполнить раздел **hosts**, в котором надо указать IP-адреса узлов кластера. Должны быть указаны IP-адреса в сети управления, узнать их можно в инсталляционной карте Машины МБД.П. Также можно указать пути к файлам сертификатов, подробнее про настройку **SSL** см. п. 3.3.

```
all:
  vars:
    ansible_user: root
    ansible_ssh_extra_args: "-o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null"
    rpm_mode: True

  # Список переменных, используемых для настройки алерт-агента
  recipients:
    # Название алерта и JSON файла, который его описывает
    - name: spectrum
      # UUID кластера, для которого настраиваются алерты. Можно найти в адресной строке, перейдя на страницу обзора кластера
      cluster_id: cc83ac88-e553-44ce-adb4-a30c75a3af2c
      # Хост Спектра, для которого настраиваются алерты. Указать реальный IP адрес Спектра, доступ к которому есть с узла кластера
      spectrum_host: 192.168.1.

# Хосты узлов, на которых будет настроен алерт-агент
hosts:
  node01:
    # Указать реальные IP адреса узлов кластера, к которым есть доступ с узла Спектра
    ansible_host: 192.168.1.
    # Раскомментировать, если у хоста Спектра есть корневой сертификат и указать реальный путь до сертификата на хосте Спектра
    #host_ca_cert: "" # Путь к корневому сертификату на хосте Спектра, откуда тот будет скопирован на узлы
    # Раскомментировать, если на хосте Спектра настроена аутентификация и указать реальный путь до ключа и сертификата на хосте Спектра
    #host_cert: "" # Путь к клиентскому сертификату на хосте Спектра, откуда тот будет скопирован на узлы
    #host_key: "" # Путь к ключу для клиентского сертификата на хосте Спектра, откуда тот будет скопирован на узлы

  node02:
    ansible_host: 192.168.1.
    #host_ca_cert: ""
    #host_cert: ""
    #host_key: ""

  node03:
    ansible_host: 192.168.1.
    #host_ca_cert: ""
    #host_cert: ""
    #host_key: ""
```



Примечание

По умолчанию файл заполнен некорректными данными, которые необходимо исправить.

При установке Агента "Спектра" раздел **recipients** можно оставить без изменений. Этот раздел будет заполнен при установке Агента уведомлений "Спектра" после выполнения импорта кластера, см. п. 3.2.3.

Полученный файл можно сохранить для использования в последующих обновлениях "Спектра". Например, скопировать в домашний каталог пользователя, под которым выполняется установка "Спектра". Для этого перейти в каталог, в который распакован архив с дистрибутивом "Спектра". Например, если архив распакован в **/root/spectrum-dist**, то выполнить команду

```
cd /root/spectrum-dist/spectrum-rpm-dist-v1.1.70
```

и скопировать файл при помощи команды **cp**:

```
[root@spr-test-installer_v1_1 spectrum-rpm-dist-v1.1.70]# cp
inventory_samples/cluster.yml ~
```

После заполнения IP-адресов узлов кластера можно перейти к установке Агента "Спектра" на узлы кластера.

Установка производится из каталога, в который распакован архив с дистрибутивом "Спектра". Например, если распаковали архив в **/root/spectrum-dist**, то выполнить команду

```
cd /root/spectrum-dist/spectrum-rpm-dist-v1.1.70
```

И далее выполнить следующие команды:

```
source /opt/skala-r/spectrum/python-modules/bin/activate
ansible-playbook -i inventory_samples/cluster.yml deploy_spectrum_agent.yml -k
```

Если ранее была сделана копия файла **cluster.yml**, то последняя команда будет иметь вид:

```
ansible-playbook -i ~/cluster.yml deploy_spectrum_agent.yml -k
```

После запуска требуется ввести пароль пользователя **root** для доступа к узлам кластера.

При успешном выполнении команды должны отсутствовать сообщения об ошибках, а в итоговом отчете о запуске плейбука должны отсутствовать шаги со статусом **failed**.

```
PLAY RECAP *****
node01      : ok=10   changed=5   unreachable=0    failed=0    skipped=1     rescued=0     ignored=0
node02      : ok=10   changed=5   unreachable=0    failed=0    skipped=1     rescued=0     ignored=0
node03      : ok=10   changed=5   unreachable=0    failed=0    skipped=1     rescued=0     ignored=0
(python-modules) [root@spr-test-installer_v1_1 spectrum-rpm-dist-v1.1.70]#
```

Далее необходимо выполнить импорт кластера в "Спектр" с использованием графического веб-интерфейса. Подробнее об импорте кластеров см. в документе "Руководство пользователя", раздел 7.2 "**Добавление найденного кластера**".

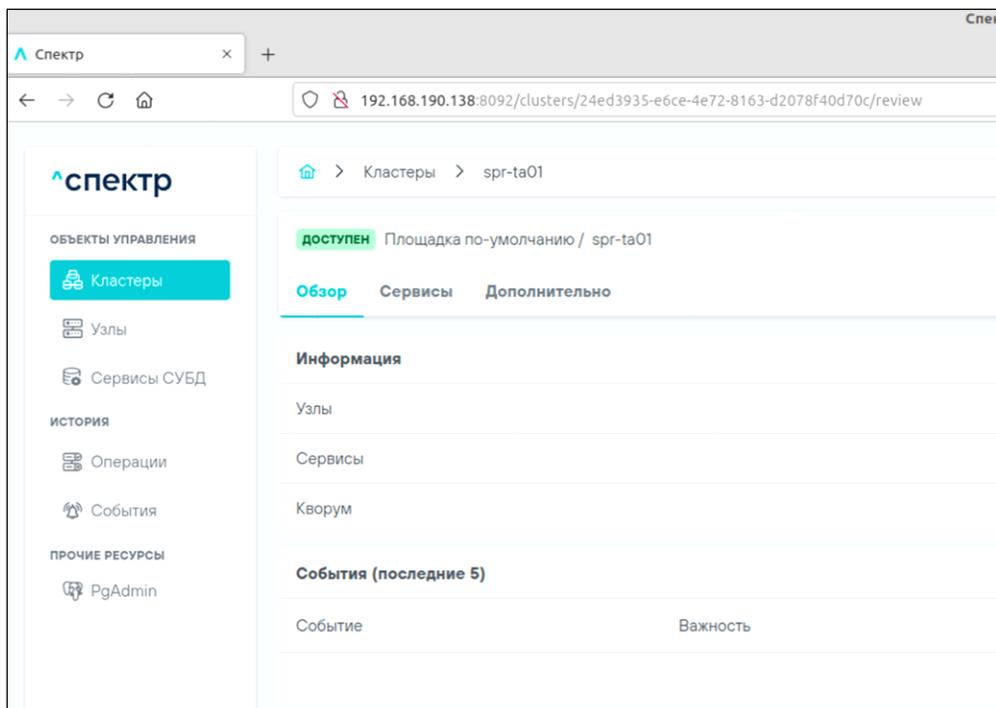
3.2.3 Установка Агента уведомлений "Спектра"

Перед установкой Агента уведомлений "Спектра" кластеры должны быть видны в графическом интерфейсе и иметь статус "Доступен", например:

Кластер	Статус	Площадка	Сервисы	Узлы
spr-dev-ak	Доступен	Стенд_разработки	1	3
spr-ta01	Доступен	Стенд_разработки	1 1	3
sd231108	Доступен	Стенд_разработки	2	3
spr-loadtest	Доступен	Стенд_разработки	3	3
spr-pg14-9-3	Доступен	Стенд_разработки	1	3
spr-da	Доступен	Стенд_разработки	1 1	3
spr-uv01	Доступен	Стенд_разработки	2	3

Для установки Агента уведомлений "Спектра" потребуются IP-адрес хоста "Спектра" и ID импортированного кластера.

IP-адрес отображается в адресной строке браузера, в котором открыт веб-интерфейс "Спектра". ID кластера также отображаются в веб-интерфейсе в адресной строке браузера при переходе на страницу информации о кластере.



В приведенном примере IP-адрес "Спектра" - "192.168.190.138", ID кластера — это значение "24ed3935-e6ce-4e72-8163-d2078f40d70c".

Полученные ID кластера и IP-адрес необходимо вписать в соответствующие строки секции **recipients** того же файла **cluster.yml**, который использовался для установки Агента "Спектра" (см. п. 3.2.2):

```
all:
  vars:
    ansible_user: root
    ansible_ssh_extra_args: "-o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null"
    rpm_mode: True

    # Список переменных, используемых для настройки алерт-агента
  recipients:
    # Название конфигурации уведомлений и JSON файла, который его описывает
    - name: spectrum
    # UUID кластера, для которого настраиваются уведомления. Можно найти в адресной строке, перейдя на страницу обзора кластера
    cluster_id: 24ed3935-e6ce-4e72-8163-d2078f40d70c
    # Хост Спектра, для которого настраиваются уведомления. Указать реальный IP адрес Спектра, доступ к котором есть с узла кластера
    spectrum_host: 192.168.190.138

  # Хосты узлов, на которых будет настроен алерт-агент
  hosts:
    node01:
      # Указать реальные IP адреса узлов кластера, к которым есть доступ с узла Спектра
      ansible_host: 192.168.190.61
      # Раскомментировать, если у хоста Спектра есть корневой сертификат и указать реальный путь до сертификата на хосте Спектра
      host_ca_cert: "/opt/skala-r/spectrum/certs/ca.crt" # Путь к корневому сертификату на хосте Спектра, откуда тот будет скопирован на узлы
      # Раскомментировать, если на хосте Спектра настроена аутентификация и указать реальный путь до ключа и сертификата на хосте Спектра
      host_cert: "/home/root/node01_server.crt" # Путь к клиентскому сертификату на хосте Спектра, откуда тот будет скопирован на узлы
      host_key: "/home/root/node01_server.key" # Путь к ключу для клиентского сертификата на хосте Спектра, откуда тот будет скопирован на узлы

    node02:
      ansible_host: 192.168.190.62
      host_ca_cert: "/opt/skala-r/spectrum/certs/ca.crt"
      host_cert: "/home/root/node02_server.crt"
      host_key: "/home/root/node02_server.key"

    node03:
      ansible_host: 192.168.190.63
      host_ca_cert: "/opt/skala-r/spectrum/certs/ca.crt"
      host_cert: "/home/root/node03_server.crt"
      host_key: "/home/root/node03_server.key"
```

Для кластера, приведенного в примере выше, итоговое содержимое файла **inventory_samples/cluster.yml** будет иметь вид:

```
all:
  vars:
    ansible_user: root
    ansible_ssh_extra_args: "-o StrictHostKeyChecking=no -o
UserKnownHostsFile=/dev/null"
    rpm_mode: True

    # Список переменных, используемых для настройки алерт-агента
  recipients:
    # Название конфигурации уведомлений и JSON файла, который его описывает
    - name: spectrum
    # UUID кластера, для которого настраиваются уведомления. Можно найти в
адресной строке, перейдя на страницу обзора кластера
    cluster_id: 24ed3935-e6ce-4e72-8163-d2078f40d70c
    # Хост Спектра, для которого настраиваются уведомления. Указать реальный
IP адрес Спектра, доступ к котором есть с узла кластера
    spectrum_host: 192.168.190.138

  # Хосты узлов, на которых будет настроен алерт-агент
  hosts:
    node01:
      # указать реальные IP адреса узлов кластера, к которым есть доступ с узла
Спектра
      ansible_host: 192.168.190.61
      # Раскомментировать, если на хосте Спектра настроена аутентификация и
указать реальный путь до ключа и сертификата на хосте Спектра
```

```

host_ca_cert: "/opt/skala-r/spectrum/certs/ca.crt" # Путь к корневому
сертификату на хосте Спектра, откуда тот будет скопирован на узлы
# Раскомментировать, если на хосте Спектра настроена аутентификация и
указать реальный путь до ключа и сертификата на хосте Спектра
host_cert: "/home/root/node01_server.crt" # Путь к клиентскому
сертификату на хосте Спектра, откуда тот будет скопирован на узлы
host_key: "/home/root/node01_server.key" # Путь к ключу для клиентского
сертификата на хосте Спектра, откуда тот будет скопирован на узлы

node02:
  ansible_host: 192.168.190.62
  host_ca_cert: "/opt/skala-r/spectrum/certs/ca.crt"
  host_cert: "/home/root/node02_server.crt"
  host_key: "/home/root/node02_server.key"

node03:
  ansible host: 192.168.190.63
  host_ca_cert: "/opt/skala-r/spectrum/certs/ca.crt"
  host_cert: "/home/root/node03_server.crt"
  host_key: "/home/root/node03_server.key"

```

Далее для установки Агента уведомлений перейти в каталог, в который распакован архив с дистрибутивом "Спектра". Например, если архив распакован в `/root/spectrum-dist`, то выполнить команду

```
cd /root/spectrum-dist/spectrum-rpm-dist-v1.1.70
```

и далее следующие команды:

```
source /opt/skala-r/spectrum/python-modules/bin/activate
ansible-playbook -i inventory_samples/cluster.yml
deploy_spectrum_alert_agent.yml -k
```

Если ранее была сделана копия файла `cluster.yml`, то последняя команда будет иметь вид:

```
ansible-playbook -i ~/cluster.yml deploy_spectrum_alert_agent.yml -k
```

После запуска требуется ввести пароль пользователя `root` для доступа к узлам кластера.

При успешном выполнении команды в отчете о запуске плейбука должны отсутствовать шаги со статусом `failed`, при первой установке агента уведомлений будут присутствовать шаги со статусом `ignored` и сообщение об ошибке с пометкой `"...ignoring"`:

```

TASK [remove existing alerts for spectrum] *****
fatal: [node01]: FAILED! => ("changed": true, "cmd": "pcs alert delete spectrum alert", "delta": "0:00:00.452087", "end": "2023-06-01 16:12:14.057743",
": "Error: alert 'spectrum alert' does not exist\nError: Errors have occurred, therefore pcs is unable to continue", "stderr_lines": ["Error: alert 'sp
", "continue"], "stdout": "", "stdout_lines": []]
..ignoring

TASK [debug] *****
ok: [node01] => {
  "msg": ""
}

TASK [create spectrum alerts] *****
changed: [node01]

TASK [add recipients to spectrum alert] *****
changed: [node01] => (item={'name': 'spectrum', 'cluster_id': '24ed3935-e6ce-4e72-8163-d2078f40d70c', 'spectrum_host': '192.168.190.138'})

PLAY RECAP *****
node01      : ok=16   changed=9   unreachable=0    failed=0    skipped=2     rescued=0     ignored=1
node02      : ok=11   changed=6   unreachable=0    failed=0    skipped=2     rescued=0     ignored=0
node03      : ok=11   changed=6   unreachable=0    failed=0    skipped=2     rescued=0     ignored=0

```

3.3 Настройка SSL

"Спектр" поддерживает возможность использования **SSL** для шифрования данных, передаваемых между хостом "Спектра" и узлами кластеров, а также хостами "Генома" и "Визиона". Для корректной работы **SSL** необходимо предоставить подписанные корневой, серверный и клиентские сертификаты с ключами, а также внести изменения в конфигурационные файлы модулей "Спектра", Агента "Спектра" и Агента уведомлений "Спектра". При использовании **SSL** с "Геномом" или "Визионом" потребуется также настроить соответствующее ПО на соответствующих хостах, инструкция по настройке которых не входит в данное руководство.

3.3.1 Генерация сертификатов

При первой установке "Спектра" генерируются ключ и самоподписанный серверный сертификат, но для корректной работы необходимо иметь также и корневой сертификат, которым будут подписаны все остальные сертификаты. Сертификаты могут иметь цепочки любой длины, должны иметь формат **PEM** с расширением файла **.pem** или **.crt** и могут быть самоподписанными. Ключ сертификата должен иметь расширение файла **.key**.

3.3.2 Установка сертификатов



Примечание

Название файлов сертификатов и ключей сертификатов, а также их расположение может отличаться от приведенных в данном руководстве.

После генерации сертификатов необходимо расположить их на соответствующих хостах. На хосте "Спектра" должны присутствовать:

- Корневой сертификат **ca.crt**;
- Серверный сертификат **spectrum_server.crt**;
- Ключ серверного сертификата **spectrum_server.key**;
- Клиентский сертификат **spectrum_client.crt**;
- Ключ клиентского сертификата **spectrum_client.key**.

На узлах кластеров должны присутствовать:

- Корневой сертификат **ca.crt**;
- Серверный сертификат **node_server.crt**;
- Ключ серверного сертификата **node_server.key**.

В данном руководстве сертификаты и ключи будут расположены в директории **/opt/skala-r/spectrum/certs**.

В файлах конфигурации на хосте "Спектра" **spectrum-api**, **spectrum-tasks**, **spectrum-scheduler** необходимо указать пути к файлам сертификатов и ключей. Каждая из секций с параметрами сертификатов отвечает за настройку соединения с отдельным компонентом, т. е., если необходимо включить **SSL** для Агента "Спектра", нужно указать пути к файлам в секции **spectrum-agent**.

Необходимо удостовериться, что в конфигурации **Nginx** присутствуют следующие директивы с корректно указанными путями:

```
ssl_certificate      /opt/skala-r/spectrum/certs/spectrum_server.crt;
ssl_certificate_key  /opt/skala-r/spectrum/certs/spectrum_server.key;

ssl_verify_client   optional;
ssl_client_certificate /opt/skala-r/spectrum/certs/ca.crt;
```

В файлах конфигурации **spectrum-agent** и Агента уведомлений "Спектра" на узлах кластера должны быть указаны пути к сертификатам.

Если при установке Спектра производилась установка Агента "Спектра" и Агента уведомлений "Спектра" (см. п. 3.2.2 и п. 3.2.3), то в соответствующих файлах конфигурации уже будут указаны пути к файлам сертификатов и ключей, а сами файлы находятся на узлах кластера.

Примеры файлов конфигураций приведены в Приложении 1 данного руководства.

3.4 Настройка интеграции с KeyCloak и Avanpost FAM

"Спектр" поддерживает возможность интеграции с системами аутентификации **KeyCloak** и **Avanpost FAM**. Перед этим необходимо настроить соответствующие платформы, инструкции по настройке изложены в Приложении 2 (для **KeyCloak**) и Приложении 3 (для **Avanpost FAM**) соответственно.

3.4.1 Настройка интеграции с KeyCloak

В файле конфигурации **spectrum-api** нужно заполнить секцию аутентификации, согласно примеру, приведенному ниже:

```
[auth]
auth_server=https://192.168.1.2:8080 # Адрес хоста, на котором расположен identity
provider
realm=spectrum # Название realm, который был выбран в процессе настройки
client_id=spectrum-users # ID клиента, который был создан в процессе настройки
cert=/opt/skala-r/spectrum/certs/keycloak-chain.pem # Путь к файлу сертификата.
Опциональный параметр
enabled=1 # Указать 1 для включения аутентификации, 0 для отключения
```

Также необходимо сохранить значение поля **Client secret**, полученное после создания клиента, в файл зашифрованного хранилища **.vault_store**.

Для этого выполнить следующие команды:

```
source /opt/skala-r/spectrum/python-modules/bin/activate
ansible-vault edit /opt/skala-r/spectrum/.vault_store
```

После этого необходимо ввести пароль хранилища, после чего будет открыт текстовый редактор. Пример содержания файла можно найти в Приложении 1.

Добавить следующую строку, где **secret** - значение поля **Client secret** из **KeyCloak**:

```
client_secret: secret
```

и сохранить файл.

После этого нужно перезапустить сервис **API**, введя команду:

```
systemctl restart spectrum-api
```

3.4.2 Настройка интеграции с Avanpost FAM

Шаги по настройке аналогичны шагам из п. 3.4.1, за исключением содержания секции аутентификации в файле конфигурации **spectrum-api**:

```
[auth]
auth_server=http://192.168.1.3:8080 # Адрес хоста, на котором расположен identity
provider
client_id=spectrum-users # ID клиента, который был создан в процессе настройки
enabled=1 # Указать 1 для включения аутентификации, 0 для отключения
```

В зашифрованном хранилище в поле **client_secret** нужно указать значение поля **secret**, совпадающее с заданным при создании приложения в **Avanpost FAM**.

4 УСТАНОВКА ОБНОВЛЕНИЙ ПРОГРАММЫ

Установка обновлений ПО "Спектр" и дополнительного ПО выполняется в случае, если на хосте "Спектра" уже существуют ранее установленные версии ПО. Для установки и для обновления "Спектра" используется один и тот же пакет установочных файлов, поставляемый в виде архива с именем **spectrum-dist-altlinux<номер версии>-v<номер версии Спектра>.tar.gz**, например **spectrum-dist-altlinux8.4-v1.1.70.tar.gz**.

4.1 Обновление ПО "Спектр"

Новый архив с дистрибутивом "Спектра" необходимо распаковать в произвольный каталог на хосте "Спектра", например в **/root/spectrum-dist**.

Распакованный архив содержит все **RPM**-пакеты, необходимые для работы "Спектра", а также скрипт установки **install.sh** и скрипт обновления **update.sh**.

Для обновления необходимо перейти в директорию, в которую был распакован архив, и выполнить скрипт **update.sh**:

```
[root@spectrum-host ~]# mkdir -p spectrum-dist
[root@spectrum-host ~]# tar -xvf spectrum-dist-altlinux8.4-v1.2.0.tar.gz -C
./spectrum-dist
[root@spectrum-host ~]# cd spectrum-dist/spectrum-rpm-dist-v1.2.0
[root@spectrum-host ~]# rpm -Uvh rpm/spectrum-venv-*.rpm
[root@spectrum-host spectrum-rpm-dist-v1.2.0]# ./update.sh
```

4.2 Обновление дополнительного ПО "Спектр"

Шаги, необходимые для обновления дополнительного ПО "Спектр", за исключением **PgAdmin**, полностью повторяют шаги первичной установки. Подробно установка дополнительного ПО описана в разделе 3.2 "Установка дополнительного ПО "Скала^р Спектр".

При выполнении обновления "Спектра" для кластеров, импортированных в "Спектр" до выполнения обновления, необходимо выполнить процедуру перевода в режим использования Агента.

Для этого требуется выполнить HTTP запрос к **API** "Спектра", например, при помощи команды **curl**. В адресе, на который выполняется запрос, используется **ID** кластера, который может быть получен из веб-интерфейса "Спектра":

```
curl http://localhost:8092/api/spectrum/001/cluster/24ed3935-e6ce-4e72-8163-
d2078f40d70c/agent -d '{"use_agent": true}' -X PUT -H "Content-Type:
application/json"
```

Если команда введена корректно, то в выводе будут отсутствовать сообщения об ошибках:

```
{"result": {"update_cluster_agent_use_mode": 0}, "errors": []}
```

4.2.1 Обновление PgAdmin

Дистрибутив **PgAdmin** представляет собой **tar.gz** архив **pgadmin_dist_alt84_rpm.tar.gz**, который необходимо распаковать в произвольный каталог на хосте "Спектра", например в **/root/pgadmin-dist**.



Примечание

В данном Руководстве приводятся примеры команд для архива с названием **pgadmin_dist_alt84_rpm.tar.gz**.

Название архива может отличаться, например, может быть другая версия.

Распакованный архив содержит образы **docker**-контейнеров, необходимые для работы **PgAdmin**, а также скрипт обновления **update.sh**.

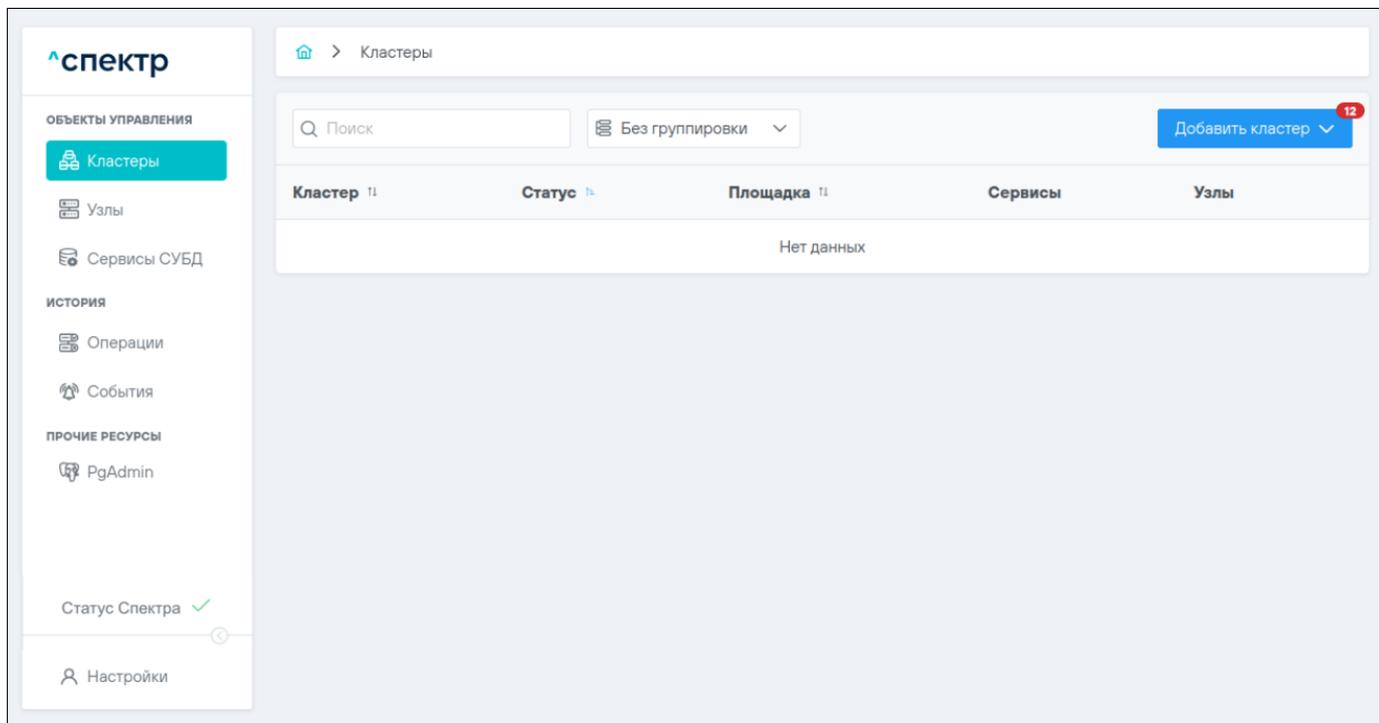
Для обновления необходимо перейти в директорию, в которую был распакован архив, и выполнить скрипт **update.sh**. Для этого нужно выполнить следующие команды:

```
[root@spectrum-host ~]# mkdir -p spectrum-dist  
  
[root@spectrum-host ~]# tar -xvf pgadmin_dist_alt84_rpm.tar.gz -C ./spectrum-dist  
  
[root@spectrum-host ~]# cd spectrum-dist/pgadmin-dist  
  
[root@spectrum-host pgadmin-dist]# ./update.sh
```

5 ПРОВЕРКА ПРОГРАММЫ

5.1 Отображение веб-интерфейса

По завершении процесса установки, должен быть доступен веб-интерфейс "Спектра" через браузер по адресу [https://<хост "Спектра">:8092](https://<хост):



5.2 Проверка установки модулей

Модули системы установлены в каталог `/opt/skala-r/spectrum`:

- **dashboards** - конфигурация мониторинга "Спектра";
- **spectrum-api** — REST API "Спектра";
- **spectrum-scheduler** — модуль выполнения задач по расписанию;
- **spectrum-task-daemon** — модуль выполнения задач по запросу;
- **spectrum-ui** — файлы веб интерфейса;
- **modules/spectrum_pcs_pgsq1** — модуль управления кластером **Postgres**;
- **python-modules** - дополнительные **python**-библиотеки, необходимые для работы "Спектра".

Проверить наличие модулей командой `ls -l`:

```
ls -l /opt/skala-r/spectrum
```

Пример вывода команды:

```
root@spr-test-installer-min-alt84 ~]# ls -l /opt/skala-r/spectrum
total 28
-rwxr-xr-x 3 root root 4096 Jun  3 09:53 dashboards
-rwxr-xr-x 3 root root 4096 Jun  3 09:51 modules
-rwxr-xr-x 6 root root 4096 Jun  3 09:50 python-modules
-rwxr-xr-x 5 root root 4096 Jun  3 12:05 spectrum-api
-rwxr-xr-x 4 root root 4096 Jun  3 09:51 spectrum-scheduler
-rwxr-xr-x 3 root root 4096 Jun  3 09:51 spectrum-tasks-daemon
-rwxr-xr-x 3 root root 4096 Jun  3 09:51 spectrum-ui
root@spr-test-installer-min-alt84 ~]#
```

5.3 Проверка состояния и управление сервисами "Спектра"

По завершении процессов установки в системе будут запущены сервисы:

- **spectrum-api.service** — сервис **http API**, реализующего взаимодействие пользовательского интерфейса с системой;
- **spectrum-tasks.service** — демон, предназначенный для выполнения задач (выполняет запрос к **spectrum-api** и получает список невыполненных заданий);
- **spectrum-scheduler.service** — демон, предназначенный для выполнения задач по расписанию.

Логи работы **spectrum-api**, **spectrum-tasks**, **spectrum-scheduler** сохраняются в файлы **/opt/skala-r/var/log/spectrum/spectrum-api-<дата записи лога>.log**, **spectrum_task_daemon-<дата записи лога>.log**, **spectrum_scheduler-<дата записи лога>.log** соответственно.

5.3.1 Ручная остановка, запуск или перезапуск сервисов

Для ручной остановки, запуска или перезапуска сервисов используются стандартные механизмы **systemctl**, например:

```
systemctl stop spectrum-tasks
systemctl start spectrum-tasks
systemctl restart spectrum-tasks
```

5.3.2 Проверка статусов сервисов

Для проверки статуса сервисов используется стандартный механизм **systemctl**. Проверить состояние сервисов можно посредством выполнения следующих команд:

```
systemctl status spectrum-tasks
```

Пример активного сервиса **spectrum-tasks**:

```
[root@spectrum-stage02 ~]# systemctl status spectrum-tasks
● spectrum-tasks.service - spectrum-tasks-daemon
   Loaded: loaded (/etc/systemd/system/spectrum-tasks.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2022-10-27 07:25:15 UTC; 55min ago
   Main PID: 2026315 (python3)
     Tasks: 1 (limit: 24752)
    Memory: 26.2M
    CGroup: /system.slice/spectrum-tasks.service
            └─2026315 /usr/bin/python3 /opt/spectrum/spectrum-tasks-daemon/app.py

окт 27 08:20:19 spectrum-stage02 python3[2026315]: 2022-10-27 08:20:19,456 - urllib3.connectionpool - DEBUG - http://localhost:8000 "GET /api/spectrum/v001/tasks/unprocessed?size=10 HTTP/1.1" 200 86
окт 27 08:20:19 spectrum-stage02 python3[2026315]: 2022-10-27 08:20:19,456 - runners.task_runner - INFO - Found 0 new tasks
окт 27 08:20:19 spectrum-stage02 python3[2026315]: 2022-10-27 08:20:19,456 - _main_ - INFO - Finished processing tasks
окт 27 08:20:24 spectrum-stage02 python3[2026315]: 2022-10-27 08:20:24,461 - _main_ - INFO - Begin processing tasks
окт 27 08:20:24 spectrum-stage02 python3[2026315]: 2022-10-27 08:20:24,461 - skalar.spectrum.client.apiclient - DEBUG - config: {'app': '<Section: app>', 'database_config': '<Section: database_config>', 'spectrum_api':
окт 27 08:20:24 spectrum-stage02 python3[2026315]: 2022-10-27 08:20:24,462 - skalar.spectrum.client.apiclient - DEBUG - spectrum api localhost 8000
окт 27 08:20:24 spectrum-stage02 python3[2026315]: 2022-10-27 08:20:24,463 - urllib3.connectionpool - DEBUG - Starting new HTTP connection (1): localhost:8000
окт 27 08:20:24 spectrum-stage02 python3[2026315]: 2022-10-27 08:20:24,470 - urllib3.connectionpool - DEBUG - http://localhost:8000 "GET /api/spectrum/v001/tasks/unprocessed?size=10 HTTP/1.1" 200 86
окт 27 08:20:24 spectrum-stage02 python3[2026315]: 2022-10-27 08:20:24,471 - runners.task_runner - INFO - Found 0 new tasks
окт 27 08:20:24 spectrum-stage02 python3[2026315]: 2022-10-27 08:20:24,471 - _main_ - INFO - Finished processing tasks
```

```
systemctl status spectrum-api
```

Пример активного сервиса **spectrum-api**:

```
[root@spectrum-stage02 ~]# systemctl status spectrum-api
● spectrum-api.service - gunicorn daemon
   Loaded: loaded (/etc/systemd/system/spectrum-api.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2022-10-27 07:25:15 UTC; 48min ago
   Main PID: 2026297 (gunicorn)
     Tasks: 2 (limit: 24752)
    Memory: 40.0M
    CGroup: /system.slice/spectrum-api.service
            └─2026297 /usr/bin/python3.8 /usr/local/bin/gunicorn app:create_app() --chdir /opt/spectrum/spectrum-api/
            └─2026314 /usr/bin/python3.8 /usr/local/bin/gunicorn app:create_app() --chdir /opt/spectrum/spectrum-api/

окт 27 08:14:05 spectrum-stage02 gunicorn[2026297]: 2022-10-27 08:14:05,110 - opers - INFO - Fetched oper list.
окт 27 08:14:05 spectrum-stage02 gunicorn[2026297]: 2022-10-27 08:14:05,110 - oper - INFO - Ended api get_opsers.
окт 27 08:14:08 spectrum-stage02 gunicorn[2026297]: 2022-10-27 08:14:08,303 - tasks - INFO - Started api get_unprocessed_tasks.
окт 27 08:14:08 spectrum-stage02 gunicorn[2026297]: 2022-10-27 08:14:08,304 - tasks - INFO - Started fetching unprocessed tasks.
окт 27 08:14:08 spectrum-stage02 gunicorn[2026297]: 2022-10-27 08:14:08,309 - tasks - INFO - Fetched unprocessed tasks list.
окт 27 08:14:08 spectrum-stage02 gunicorn[2026297]: 2022-10-27 08:14:08,309 - tasks - INFO - Ended api get_unprocessed_tasks.
окт 27 08:14:09 spectrum-stage02 gunicorn[2026297]: 2022-10-27 08:14:09,458 - oper - INFO - Started api get_opsers.
окт 27 08:14:09 spectrum-stage02 gunicorn[2026297]: 2022-10-27 08:14:09,459 - opers - INFO - Started fetching opsers.
окт 27 08:14:09 spectrum-stage02 gunicorn[2026297]: 2022-10-27 08:14:09,827 - opers - INFO - Fetched oper list.
окт 27 08:14:09 spectrum-stage02 gunicorn[2026297]: 2022-10-27 08:14:09,828 - oper - INFO - Ended api get_opsers.
```

```
systemctl status spectrum-scheduler
```

Пример активного сервиса **spectrum-scheduler**:

```
[root@spectrum-stage02 spectrum]# systemctl status spectrum-scheduler
● spectrum-scheduler.service - spectrum-scheduler
   Loaded: loaded (/etc/systemd/system/spectrum-scheduler.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2022-10-27 07:25:16 UTC; 5h 1min ago
   Main PID: 2026334 (python3)
     Tasks: 2 (limit: 24752)
    Memory: 28.4M
    CGroup: /system.slice/spectrum-scheduler.service
            └─2026334 /usr/bin/python3 /opt/spectrum/spectrum-scheduler/app.py

окт 27 12:23:19 spectrum-stage02 python3[2026334]: 2022-10-27 12:23:19,899 - scheduler - INFO - Новых операций не найдено
окт 27 12:24:19 spectrum-stage02 python3[2026334]: 2022-10-27 12:24:19,900 - scheduler - INFO - Отправлен запрос на получение операций
окт 27 12:24:19 spectrum-stage02 python3[2026334]: 2022-10-27 12:24:19,915 - scheduler - INFO - Получен список операций
окт 27 12:24:19 spectrum-stage02 python3[2026334]: 2022-10-27 12:24:19,915 - scheduler - INFO - Новых операций не найдено
окт 27 12:25:19 spectrum-stage02 python3[2026334]: 2022-10-27 12:25:19,915 - scheduler - INFO - Отправлен запрос на получение операций
окт 27 12:25:19 spectrum-stage02 python3[2026334]: 2022-10-27 12:25:19,925 - scheduler - INFO - Получен список операций
окт 27 12:25:19 spectrum-stage02 python3[2026334]: 2022-10-27 12:25:19,925 - scheduler - INFO - Новых операций не найдено
окт 27 12:26:19 spectrum-stage02 python3[2026334]: 2022-10-27 12:26:19,926 - scheduler - INFO - Отправлен запрос на получение операций
окт 27 12:26:19 spectrum-stage02 python3[2026334]: 2022-10-27 12:26:19,933 - scheduler - INFO - Получен список операций
окт 27 12:26:19 spectrum-stage02 python3[2026334]: 2022-10-27 12:26:19,933 - scheduler - INFO - Новых операций не найдено
root@spectrum-stage02 spectrum]# systemctl status spectrum-tasks
```

5.4 Проверка состояния дополнительного ПО "Спектра"

Проверка состояния агента "Спектра" и агента уведомлений "Спектра" выполняется на узлах кластера.

5.4.1 Проверка состояния агента "Спектра"

Проверить состояние сервиса можно, выполнив команду:

```
systemctl status spectrum-agent
```

Пример активного сервиса **spectrum-agent**:

```
[root@spr-ta01-node01 ~]# systemctl status spectrum-agent
● spectrum-agent.service - spectrum-agent
   Loaded: loaded (/etc/systemd/system/spectrum-agent.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2023-05-31 16:44:40 MSK; 23h ago
     Main PID: 1864154 (spectrum-agent)
        Tasks: 11 (limit: 4659)
       Memory: 12.2M
      CGroup: /system.slice/spectrum-agent.service
             └─1864154 /opt/skala-r/spectrum/spectrum-agent/spectrum-agent

May 31 16:44:40 spr-ta01-node01 systemd[1]: Started spectrum-agent.
[root@spr-ta01-node01 ~]#
```

5.4.2 Проверка состояния агента уведомлений "Спектра"

Проверить состояние сервиса можно, выполнив команду:

```
pcs alert
```

Пример вывода команды:

```
[root@spr-ta01-node01 ~]# pcs alert
Alerts:
Alert: spectrum_alert (path=/opt/skala-r/spectrum/alert-agent/cluster_events.py)
Description: Send cluster events to Spectrum
Meta options: timestamp-format="%Y-%m-%d %H:%M:%S.%03N %z"
Recipients:
Recipient: spectrum (value=/opt/skala-r/spectrum/alert-agent/spectrum.json)
[root@spr-ta01-node01 ~]#
```

В выводе команды используются имена файлов, наличие которых можно проверить при помощи команды **ls**:

```
[root@spr-ta01-node01 ~]# ls /opt/skala-r/spectrum/alert-agent/cluster_events.py
/opt/skala-r/spectrum/alert-agent/cluster_events.py
[root@spr-ta01-node01 ~]#
[root@spr-ta01-node01 ~]# ls /opt/skala-r/spectrum/alert-agent/spectrum.json
/opt/skala-r/spectrum/alert-agent/spectrum.json
[root@spr-ta01-node01 ~]#
```

Файл, указанный в блоке **Recipient** команды **pcs alert**, должен содержать IP-адрес "Спектра" и **ID** кластера, который был настроен на этапе установки дополнительного ПО "Спектр", см. раздел 3.2 данного руководства. Посмотреть содержимое файла можно при помощи команды **cat**:

```
[root@spr-ta01-node01 ~]#
[root@spr-ta01-node01 ~]# cat /opt/skala-r/spectrum/alert-agent/spectrum.json
{"spectrum_host": "192.168.190.138", "spectrum_port": "8092", "cluster_id": "24ed3935-e6ce-4e72-8163-d2078f40d70c"}
```

5.4.3 Проверка состояния PgAdmin

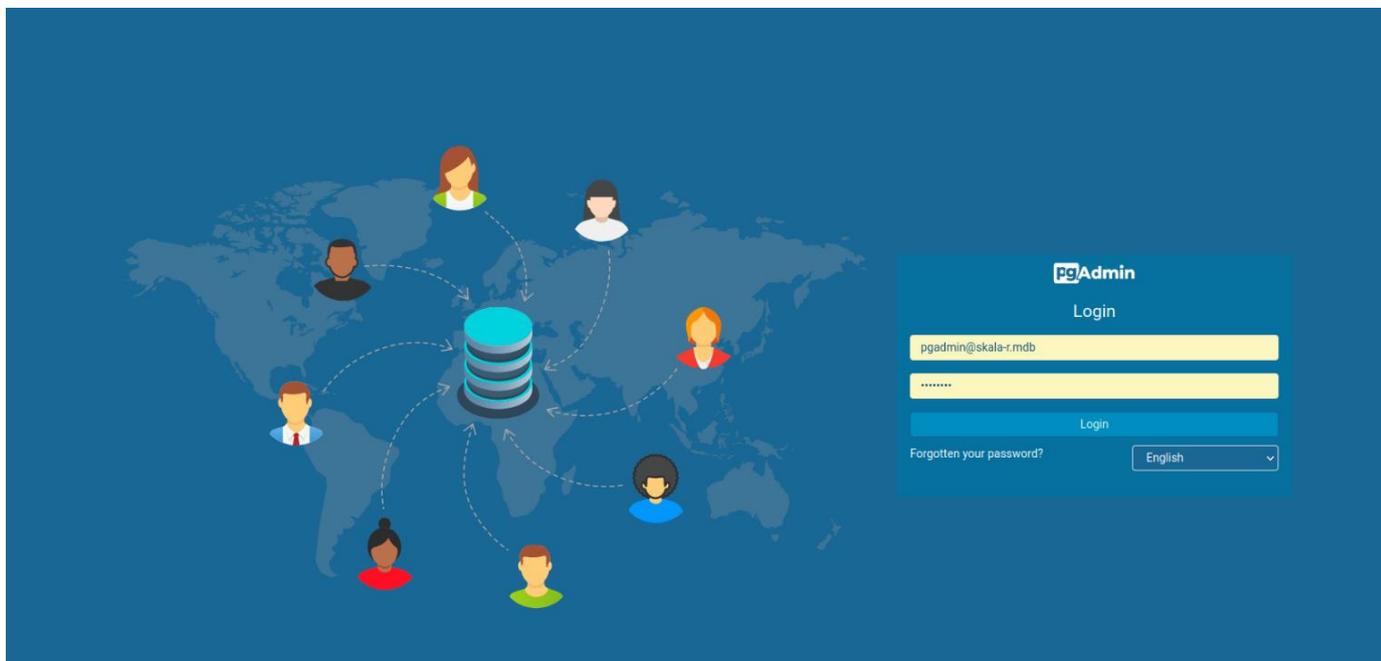
На хосте "Спектра" выполнить команду:

```
docker ps
```

При корректной установке **PgAdmin** должен отобразиться список контейнеров со статусом **Up**. Пример вывода команды:

```
[root@spr-test-installer-min-alt84 ~]# docker ps
CONTAINER ID   IMAGE                                COMMAND                                CREATED        STATUS                    PORTS
189995f81cfc   dpage/pgadmin4:6.7                 "/entrypoint.sh"                     2 days ago    Up 2 days (healthy)    80/tcp, 0
fc24bab73189   nginx:1.20.2                       "/docker-entrypoint..."            2 days ago    Up 2 days (healthy)    0.0.0.0:8
[root@spr-test-installer-min-alt84 ~]#
[root@spr-test-installer-min-alt84 ~]#
```

Также, в веб-интерфейсе при нажатии на пункт меню "**PgAdmin**" в новой вкладке браузера должна открываться форма авторизации **PgAdmin**:



5.5 Просмотр состояния системного ПО

При работе "Спектр" использует следующее системное ПО, существенно влияющее на функционирование "Спектра":

- Веб-сервер **Nginx**;
- СУБД **PostgreSQL**.

Состояние веб-сервера **Nginx** можно проверить при помощи команды:

```
systemctl status nginx
```

Пример вывода команды:

```
[root@spectrum-136 ~]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: disabled)
   Active: active (running) since Tue 2023-09-12 14:23:32 MSK; 2 days ago
     Main PID: 228351 (nginx)
        Tasks: 11 (limit: 4699)
       Memory: 8.9M
      CGroup: /system.slice/nginx.service
             └─228351 nginx: master process /usr/sbin/nginx -g daemon off;
                └─228352 nginx: worker process
                └─228353 nginx: worker process
                └─228354 nginx: worker process
                └─228355 nginx: worker process
                └─228356 nginx: worker process
                └─228357 nginx: worker process
                └─228358 nginx: worker process
                └─228359 nginx: worker process
                └─228360 nginx: worker process
                └─228361 nginx: worker process
```

Состояние СУБД **PostgreSQL** можно проверить при помощи команды:

```
systemctl status postgresql
```

Пример вывода команды:

```
[root@spectrum-136 ~]# systemctl status postgresql
● postgresql.service - PostgreSQL database server
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2023-08-07 16:24:25 MSK; 1 months 7 days ago
     Main PID: 3245275 (postgres)
        Tasks: 54 (limit: 4699)
       Memory: 290.1M
      CGroup: /system.slice/postgresql.service
             └─202254 postgres: admin spectrum 127.0.0.1(40114) idle
                └─202255 postgres: admin spectrum 127.0.0.1(40124) idle
                └─202256 postgres: admin spectrum 127.0.0.1(40134) idle
                └─202257 postgres: admin spectrum 127.0.0.1(40148) idle
                └─202258 postgres: admin spectrum 127.0.0.1(40158) idle
                └─202259 postgres: admin spectrum 127.0.0.1(40164) idle
                └─202260 postgres: admin spectrum 127.0.0.1(40168) idle
                └─202261 postgres: admin spectrum 127.0.0.1(40176) idle
                └─202262 postgres: admin spectrum 127.0.0.1(40182) idle
                └─202263 postgres: admin spectrum 127.0.0.1(40190) idle
                └─202264 postgres: admin spectrum 127.0.0.1(40196) idle
                └─202265 postgres: admin spectrum 127.0.0.1(40210) idle
                └─202266 postgres: admin spectrum 127.0.0.1(40214) idle
                └─202267 postgres: admin spectrum 127.0.0.1(40230) idle
                └─202268 postgres: admin spectrum 127.0.0.1(40244) idle
                └─202269 postgres: admin spectrum 127.0.0.1(40254) idle
```

5.6 Просмотр логов сервисов

5.6.1 Просмотр логов основного ПО "Спектра"

Просмотреть логи работы сервисов за интересующий период можно, используя службу журналирования логов **journalctl**, выполнив команду:

```
journalctl -u <имя сервиса> --since "начало интервала YYYY-MM-DD hh:mm:ss" --until "конец интервала YYYY-MM-DD hh:mm:ss"
```

Просмотреть текущие логи в режиме онлайн можно, используя службу журналирования логов **journalctl**, выполнив команду:

```
journalctl -fu <имя сервиса>
```

где <имя сервиса> - название одного из сервисов "Спектра":

- **spectrum-api**;
- **spectrum-tasks**;
- **spectrum-scheduler**;
- **spectrum-agent**.



Примечание

Просмотр логов работы сервиса **spectrum-agent** выполняется на узлах кластера.

Пример запуска команды:

```
journalctl -u spectrum-api --since "2024-08-08 11:05:00" --until "2024-08-08 11:10:00"
```

Пример вывода команды:

```
[root@spectrum-test-stand-0 ~]# journalctl -u spectrum-api --since "2024-08-08 11:05:00" --until "2024-08-08 11:10:00"
-- Logs begin at Wed 2024-08-07 17:51:08 MSK, end at Thu 2024-08-08 13:52:23 MSK. --
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: objects - Started api get_object_full.
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009178]: SPR_MAIN: objects - Started api get_object_full.
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009178]: SPR_MAIN: objects - Grouping children objects by type is set to True
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009178]: SPR_MAIN: objects - Started get object with childred. id = ff535e51-bb39-4373-a26b-314df370f50d.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: objects - Grouping children objects by type is set to True
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: objects - Started get object with childred. id = ff535e51-bb39-4373-a26b-314df370f50d.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: objects - getting children objects by root
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009178]: SPR_MAIN: objects - getting children objects by root
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: objects - Ended fetched object with id = ff535e51-bb39-4373-a26b-314df370f50d.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: objects - Ended api get_object_full.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: objects - Started api get_object_full.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: objects - Grouping children objects by type is set to True
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009178]: SPR_MAIN: objects - Started api get_object_full.
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009178]: SPR_MAIN: objects - Grouping children objects by type is set to True
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009178]: SPR_MAIN: objects - Started get object with childred. id = ff535e51-bb39-4373-a26b-314df370f50d.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: objects - Started get object with childred. id = ff535e51-bb39-4373-a26b-314df370f50d.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: objects - getting children objects by root
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009178]: SPR_MAIN: objects - getting children objects by root
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: objects - Ended fetched object with id = ff535e51-bb39-4373-a26b-314df370f50d.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: objects - Ended api get_object_full.
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009178]: SPR_MAIN: objects - Ended fetched object with id = ff535e51-bb39-4373-a26b-314df370f50d.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: SPR_MAIN: objects - Ended api get_object_full.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: clusters - Started api cluster_use_agent.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: nodes - Getting nodes ip by cluster ID ff535e51-bb39-4373-a26b-314df370f50d.
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009178]: SPR_MAIN: clusters - Started api cluster_use_agent.
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009178]: SPR_MAIN: nodes - Getting nodes ip by cluster ID ff535e51-bb39-4373-a26b-314df370f50d.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: nodes - Getting nodes ip by cluster ID ff535e51-bb39-4373-a26b-314df370f50d
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009178]: SPR_MAIN: nodes - Getting nodes ip by cluster ID ff535e51-bb39-4373-a26b-314df370f50d
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: SPR_MAIN: clusters - {'spr-da-node01': '192.168.190.35', 'spr-da-node00': '192.168.190.36', 'spr-da-node02': '192.168.190.34'}
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009178]: SPR_MAIN: clusters - Ended api cluster_use_agent.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009178]: clusters - {'spr-da-node01': '192.168.190.35', 'spr-da-node00': '192.168.190.36', 'spr-da-node02': '192.168.190.34'}
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009183]: oper - Started api get_oper.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009182]: oper - Started api get_oper.
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009183]: SPR_MAIN: oper - Started get oper with id = ccd5edc2-4276-11ed-b878-0242ac120002
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009183]: SPR_MAIN: oper - Started get oper with id = ccd5edc2-4276-11ed-b878-0242ac120002
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009182]: SPR_MAIN: oper - Started api get_oper.
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009182]: SPR_MAIN: oper - Started get oper with id = ccd5edc2-4276-11ed-b878-0242ac120002
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009182]: SPR_MAIN: oper - Started get oper with id = ccd5edc2-4276-11ed-b878-0242ac120002
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009182]: SPR_MAIN: oper - Attempt to get oper with id = ccd5edc2-4276-11ed-b878-0242ac120002.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009183]: SPR_MAIN: oper - Attempt to get oper with id = ccd5edc2-4276-11ed-b878-0242ac120002.
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009182]: SPR_MAIN: oper - Attempt to get oper with id = ccd5edc2-4276-11ed-b878-0242ac120002.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009182]: oper - Ended api get_oper.
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009183]: SPR_MAIN: oper - Attempt to get oper with id = ccd5edc2-4276-11ed-b878-0242ac120002.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009183]: oper - Ended api get_oper.
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009182]: SPR_MAIN: oper - Ended api get_oper.
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009183]: SPR_MAIN: oper - Ended api get_oper.
авр 08 11:05:01 spectrum-test-stand-0 SPR_MAIN[1009187]: oper - Started api get_oper.
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009187]: SPR_MAIN: oper - Started get oper with id = ccd5edc2-4276-11ed-b878-0242ac120002
авр 08 11:05:01 spectrum-test-stand-0 gunicorn[1009187]: SPR_MAIN: oper - Started get oper with id = ccd5edc2-4276-11ed-b878-0242ac120002
```

Возможно также просмотреть лог-файлы сервисов "Спектра":

- `/opt/skala-r/var/log/spectrum/spectrum-api.log`
- `/opt/skala-r/var/log/spectrum/spectrum-task-daemon.log`
- `/opt/skala-r/var/log/spectrum/spectrum-scheduler.log`

Для Агентов "Спектра" файлы расположены на узлах кластера:

- `/opt/skala-r/var/log/spectrum/alert_manager/alert_manager.log`
- `/opt/skala-r/var/log/spectrum/spectrum-agent.log`

Для просмотра логов в режиме онлайн можно использовать команду `tail`:

```
tail -f /opt/skala-r/var/log/spectrum/spectrum-api.log
```

5.6.2 Просмотр логов дополнительного ПО "Спектра"

5.6.2.1 Просмотр логов Nginx

Просмотреть логи работы веб-сервера **Nginx** за интересующий период можно, используя службу журналирования логов **journalctl**, например:

```
journalctl -u nginx --since "2024-06-01 16:00:00" --until "2024-06-01 16:05:00"
```

Просмотреть текущие логи в режиме онлайн можно, используя службу журналирования логов **journalctl**, выполнив команду:

```
journalctl -fu nginx
```

Веб-сервер **Nginx** также записывает логи запросов и ошибок в файлы **/var/log/nginx/access.log** и **/var/log/nginx/error.log** соответственно.

Для просмотра логов в режиме онлайн можно использовать команду **tail**:

```
tail /var/log/nginx/access.log  
tail /var/log/nginx/error.log
```



Примечание

При просмотре журнала логов **Nginx** с использованием службы **journalctl** записи в логах могут отсутствовать, если никаких запросов к "Спектру" не выполняется.

Также могут отсутствовать файлы логов **Nginx** **/var/log/nginx/access.log** и **/var/log/nginx/error.log**

5.6.2.2 Просмотр логов PostgreSQL

Просмотреть логи СУБД **PostgreSQL** за интересующий период можно, используя службу журналирования логов **journalctl**, например:

```
journalctl -u postgresql --since "2024-06-01 16:00:00" --until "2024-06-01  
16:05:00"
```

Просмотреть текущие логи в режиме онлайн можно, используя службу журналирования логов **journalctl**, выполнив команду:

```
journalctl -fu postgresql
```



Примечание

Для **PostgreSQL** на хосте "Спектра" не используется логирование в файл.

5.7 Просмотр файлов конфигурации

При установке "Спектра" для каждого сервиса создаются файлы конфигурации. Файлы конфигураций могут быть изменены администратором в процессе эксплуатации.

Пути расположения файлов конфигурации:

Сервис Спектра	Расположение	Название systemd-юнита	Путь к файлу конфигурации
API "Спектра"	Хост "Спектра"	spectrum-api	/opt/skala-r/etc/spectrum/spectrum-api/config.conf
Служба выполнения фоновых задач	Хост "Спектра"	spectrum-tasks	/opt/skala-r/etc/spectrum/spectrum-tasks-daemon/config.conf
Служба выполнения задач по расписанию	Хост "Спектра"	spectrum-scheduler	/opt/skala-r/etc/spectrum/spectrum-scheduler/config.conf
Агент "Спектра"	Узлы кластера	spectrum-agent	/opt/skala-r/etc/spectrum/spectrum-agent/config.conf
Агент уведомлений "Спектра"	Узлы кластера	-	/opt/skala-r/etc/spectrum/alert-agent/spectrum.json

Для просмотра файла конфигурации можно использовать команду **cat**, например:

```
cat /opt/skala-r/etc/spectrum/spectrum-api/config.conf
```

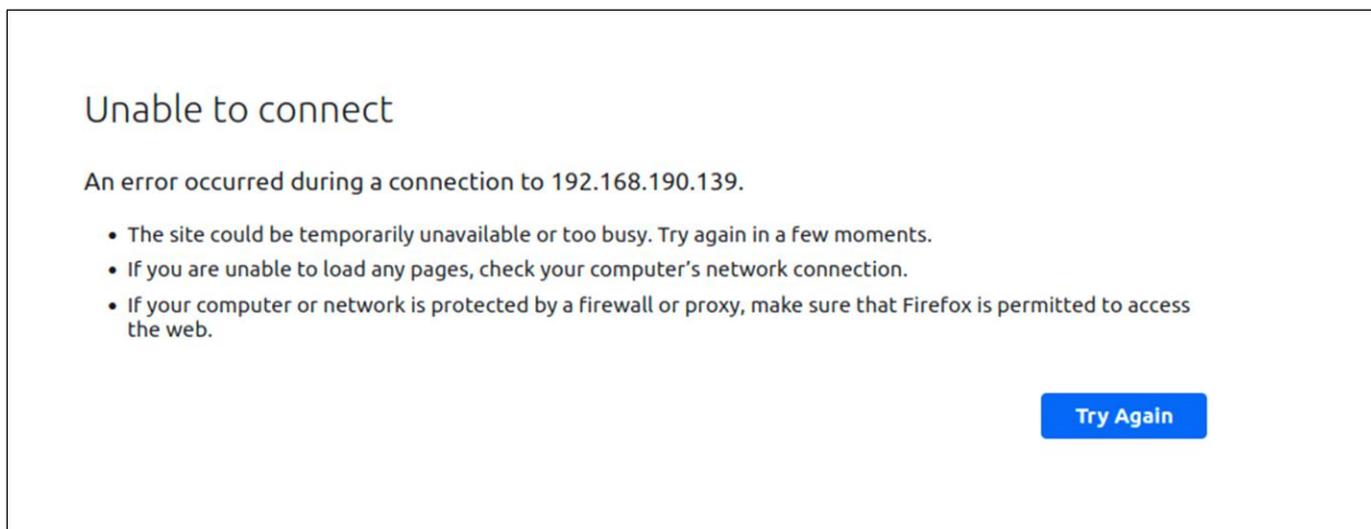
Примеры файлов конфигураций приведены в Приложении 1 данного руководства.

6 УСТРАНЕНИЕ НЕИСПРАВНОСТЕЙ "СПЕКТРА"

6.1 Не открывается веб-интерфейс "Спектра"

6.1.1 Описание неисправности

В браузере отображается сообщение об ошибке браузера, например **"Unable to connect"**



6.1.2 Способ устранения

1) Проверить корректность указания порта и протокола в адресной строке браузера. При использовании протоколов **HTTP** и **HTTPS** порт "Спектра" по умолчанию 8092.

Ниже приведены корректные способы указания адреса "Спектра" в браузере:



2) Если адрес введен корректно, проверить статус веб-сервера **Nginx** на хосте "Спектра", выполнив команду:

```
systemctl status nginx
```

Если статус веб-сервера отличается от **"active (running)"**, запустить веб-сервер командой:

```
systemctl start nginx
```

6.2 В веб-интерфейсе отсутствуют данные об объектах управления

6.2.1 Описание неисправности

Веб-интерфейс "Спектра" открывается, но в списке кластеров, узлов, сервисов отображается сообщение **"Нет данных"**, хотя ранее данные были.

6.2.2 Способ устранения

1) На хосте "Спектра" проверить статус СУБД **PostgreSQL** командой:

```
systemctl status postgresql
```

Если статус отличается от **"active (running)"**, запустить веб-сервер командой:

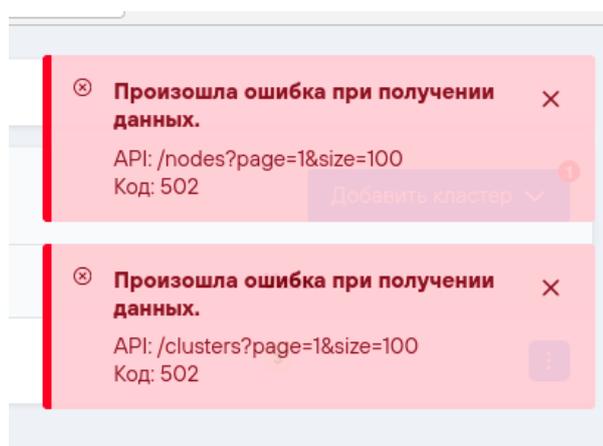
```
systemctl start postgresql
```

2) Если в выводе статуса присутствуют сообщения о том, что закончилось свободное место на диске, необходимо освободить место на диске. Например, могут быть удалены или перемещены на другой диск старые файлы логов "Спектра", расположенные в папке **/var/log/spectrum**. После очистки места на диске повторить запуск **PostgreSQL**, см. п. 1.

6.3 В веб-интерфейсе отображаются сообщения об ошибках при получении данных

6.3.1 Описание неисправности

При открытии страницы спектра появляются сообщения об ошибках:



6.3.2 Способ устранения

1) На хосте "Спектра" проверить статус сервиса **spectrum-api**:

```
systemctl status spectrum-api
```

Если статус отличается от **"active (running)"**, запустить сервис командой:

```
systemctl start spectrum-api
```

2) Проверить работоспособность веб-сервера **Nginx**. См. описание неисправности **"Не открывается веб-интерфейс "Спектра"**.

3) Проверить содержание конфигурационных файлов "Спектра".

6.4 В веб-интерфейсе отображаются устаревшие данные об объектах управления

6.4.1 Описание неисправности

В веб-интерфейсе "Спектра" отображаются статусы объектов управления, не соответствующие реальным статусам объектов. Например, в "Спектре" отображается статус узла "Резерв", а в выводе команды управления кластерным ПО Pacemaker/Corosync "pcs status" статус узла "Online".

6.4.2 Способ устранения

1) На хосте "Спектра" проверить статус сервиса **spectrum-scheduler**, отвечающего за обновление данных об объектах управления в "Спектре":

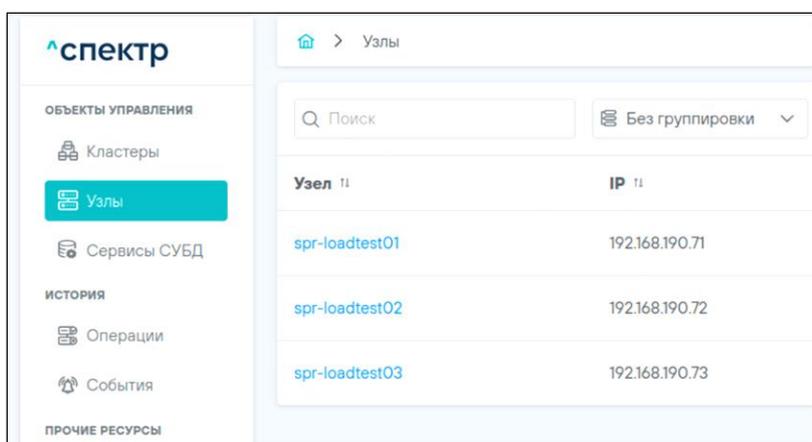
```
systemctl status spectrum-scheduler
```

Если статус отличается от "active (running)", запустить сервис командой:

```
systemctl start spectrum-scheduler
```

2) Если сервис **spectrum-scheduler** работает корректно, то необходимо проверить доступность узлов кластера с хоста "Спектра".

Для этого через веб-интерфейс получить IP-адреса узлов кластера, через которые "Спектр" осуществляет управление кластером, они отображаются на странице, доступной через пункт левого меню "Узлы":



С хоста "Спектра" при помощи команды **ping** проверить доступность узла:

```
ping 192.168.190.71
```

Если в выводе команды **ping** присутствуют сообщения о недоступности узла – устранить проблему сетевой связности между хостом "Спектра" и узлом кластера.

3) Если узел доступен с хоста "Спектра", на узле необходимо проверить статус Агента "Спектра":

```
systemctl status spectrum-agent
```

Если статус отличается от **"active (running)"**, запустить сервис командой:

```
systemctl start spectrum-agent
```

6.5 Операция завершается с сообщением об ошибке Агента "Спектра"

6.5.1 Описание неисправности

При запуске операций над объектами управления одна из задач завершается с сообщениями об ошибках выполнения запросов к Агенту "Спектра".

6.5.2 Способ устранения

На всех узлах кластера, над которым выполнялась операция, проверить статус Агента "Спектра" и при необходимости запустить сервис **spectrum-agent**, см. шаг 3 в разделе "В веб-интерфейсе отображаются устаревшие данные об объектах управления".

ПРИЛОЖЕНИЯ

Приложение 1. Примеры файлов конфигураций

Файл конфигурации spectrum-api

```
# Параметры подключения к БД Спектра
[database_config]
host_db=127.0.0.1
port_db=5432
dbname=spectrum
username_db=admin

[bottle]
# Путь к модулям управления кластером
modules_dir=/opt/skala-r/spectrum/modules
# Путь к файлам дашбордов
dashboards_dir=/opt/skala-r/spectrum/dashboards

# Параметры API Генома-У
[genome_config]
host=192.168.190.90
port=50888
base_url=/api/genome/v001/spectrum
# Параметры подключения к БД Сервисов кластера
username_db=genome
# Пути к сертификатам для обращения к Геному-У, по умолчанию закомментированы
символом ';' или '#' в начале строки
cert=/opt/skala-r/spectrum/certs/ca.crt
client_cert=/opt/skala-r/spectrum/certs/client.crt
client_key=/opt/skala-r/spectrum/certs/client.key

# Параметры Визион
[vision_config]
host=192.168.190.133
port=8428
base_url=/vision/victoriametrics/api/v1
# Пути к сертификатам для обращения к Визиону, по умолчанию закомментированы
символом ';' или '#' в начале строки
cert=/opt/skala-r/spectrum/certs/ca.crt
client_cert=/opt/skala-r/spectrum/certs/client.crt
client_key=/opt/skala-r/spectrum/certs/client.key

# Параметры API Агента Спектра
[agent_config]
port=9000
# Пути к сертификатам для обращения к агенту Спектра, по умолчанию
закомментированы символом ';' или '#' в начале строки
cert=/opt/skala-r/spectrum/certs/ca.crt
client_cert=/opt/skala-r/spectrum/certs/client.crt
client_key=/opt/skala-r/spectrum/certs/client.key

# Параметры конфигурации для аутентификации, по умолчанию закомментированы
символом ';' или '#' в начале строки
[auth]
auth_server=https://192.168.1.2:8080
realm=spectrum
client_id=spectrum-users
```

```
client_secret=xxx
cert=/opt/skala-r/spectrum/certs/keycloak-chain.pem
enabled=1
```

Файл конфигурации spectrum-tasks

```
[app]
# Путь к модулям управления кластером
modules_dir=/opt/skala-r/spectrum/modules
# Уровень логирования
log_level=DEBUG

# Параметры подключения к БД сервисов в кластере
[database_config]
database_config.dbname=postgres
database_config.username_db=genome

# Параметры доступа к API Спектра
[spectrum_api]
host=localhost
port=8000
# Пути к сертификатам для обращения к API Спектра, по умолчанию закомментированы
символом ';' или '#' в начале строки
cert=/opt/skala-r/spectrum/certs/ca.crt
client_cert=/opt/skala-r/spectrum/certs/client.crt
client_key=/opt/skala-r/spectrum/certs/client.key

# Параметры доступа к Агенту Спектра
[agent_config]
# Пути к сертификатам для обращения к Агенту Спектра, по умолчанию
закомментированы символом ';' или '#' в начале строки
cert=/opt/skala-r/spectrum/certs/ca.crt
client_cert=/opt/skala-r/spectrum/certs/client.crt
client_key=/opt/skala-r/spectrum/certs/client.key
```

Файл конфигурации spectrum-scheduler

```
[app]
# Путь к модулям управления кластером
modules_dir=/opt/skala-r/spectrum/modules
# Уровень логирования
log_level=DEBUG

# Параметры подключения к БД сервисов в кластере
[database_config]
database_config.dbname=postgres
database_config.username_db=genome

# Параметры доступа к API Спектра
[spectrum_api]
host=localhost
port=8000
# Пути к сертификатам для обращения к API Спектра, по умолчанию закомментированы
символом ';' или '#' в начале строки
```

```
cert=/opt/skala-r/spectrum/certs/ca.crt
client_cert=/opt/skala-r/spectrum/certs/client.crt
client_key=/opt/skala-r/spectrum/certs/client.key

# Параметры доступа к Агенту Спектра
[spectrum_agent]
port=9000
# Пути к сертификатам для обращения к Агенту Спектра, по умолчанию
закомментированы символом ';' или '#' в начале строки
cert=/opt/skala-r/spectrum/certs/ca.crt
client_cert=/opt/skala-r/spectrum/certs/client.crt
client_key=/opt/skala-r/spectrum/certs/client.key
```

Файл конфигурации spectrum-agent

```
{
  "log": {
    "level": "debug",
    "type": "file",
    "path": "/var/log/spectrum/"
  },
  "host": {
    "port": "9000"
  },
  "status": {
    "frequency": 5
  },
  "cert": {
    "server_cert": "",
    "server_key": "",
    "ca_cert": ""
  }
}
```

Описание полей файла конфигурации:

Поле файла конфигурации	Описание поля
log	Параметры логирования
log.level	Уровень логирования
log.type	Тип логирования. В примере используется логирование в файл
log.path	Путь к каталогу, в котором располагаются файлы логов Агента "Спектра"
host.port	Порт, на котором запущен Агент "Спектра"
status.frequency	Частота расчета конфигурации кластера в секундах
cert	Параметры сертификатов Агента Спектра

Поле файла конфигурации	Описание поля
cert.server_cert	Путь к серверному сертификату. Может быть пустым
cert.server_key	Путь к ключу серверного сертификата. Может быть пустым
cert.ca_cert	Путь к корневому сертификату. Может быть пустым

Файл .vault_store с зашифрованными секретами

```
spectrum_db_pass: xx
password_db: x
client_secret: xxx
variable: value
```

Файл конфигурации Агента уведомлений "Спектра"

```
{
  "spectrum_host": "192.168.186.136",
  "spectrum_port": "8092",
  "cluster_id": "cc83acb8-e553-44ce-adb4-a30c75a3af2c",
  "proto": "https"
  , "ca_cert": "/opt/skala-r/spectrum/certs/ca.crt"
  , "cert": "/opt/skala-r/spectrum/certs/client.crt",
  "key": "/opt/skala-r/spectrum/certs/client.key"
}
```

Описание полей файла конфигурации:

Поле файла конфигурации	Описание поля
spectrum_host	IP-адрес Спектра, на который будут отправляться уведомления о событиях кластера
spectrum_port	Порт, на котором запущен API Спектра
cluster_id	ID кластера в Спектре, для которого настраиваются уведомления
proto	Протокол, с помощью которого будут отправляться уведомления. Может принимать значения https или http
ca_cert	Путь к корневому сертификату. Поле может отсутствовать в файле конфигурации
cert	Путь к серверному сертификату. Поле может отсутствовать в файле конфигурации
key	Путь к ключу серверного сертификата. Поле может отсутствовать в файле конфигурации

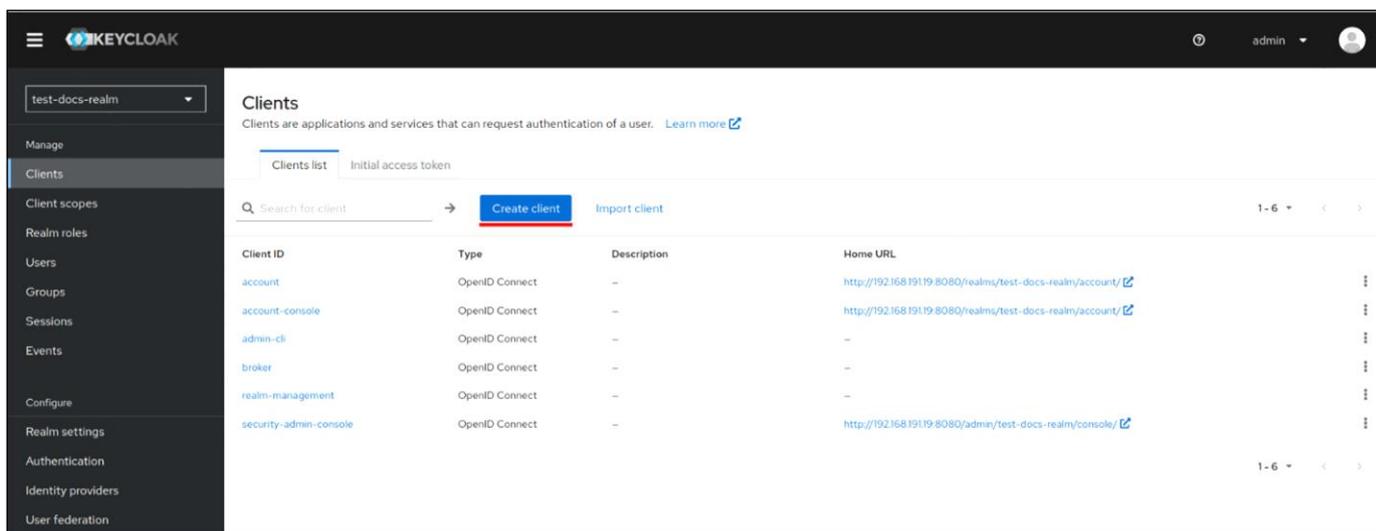
Приложение 2. Настройка пользователей в KeyCloak



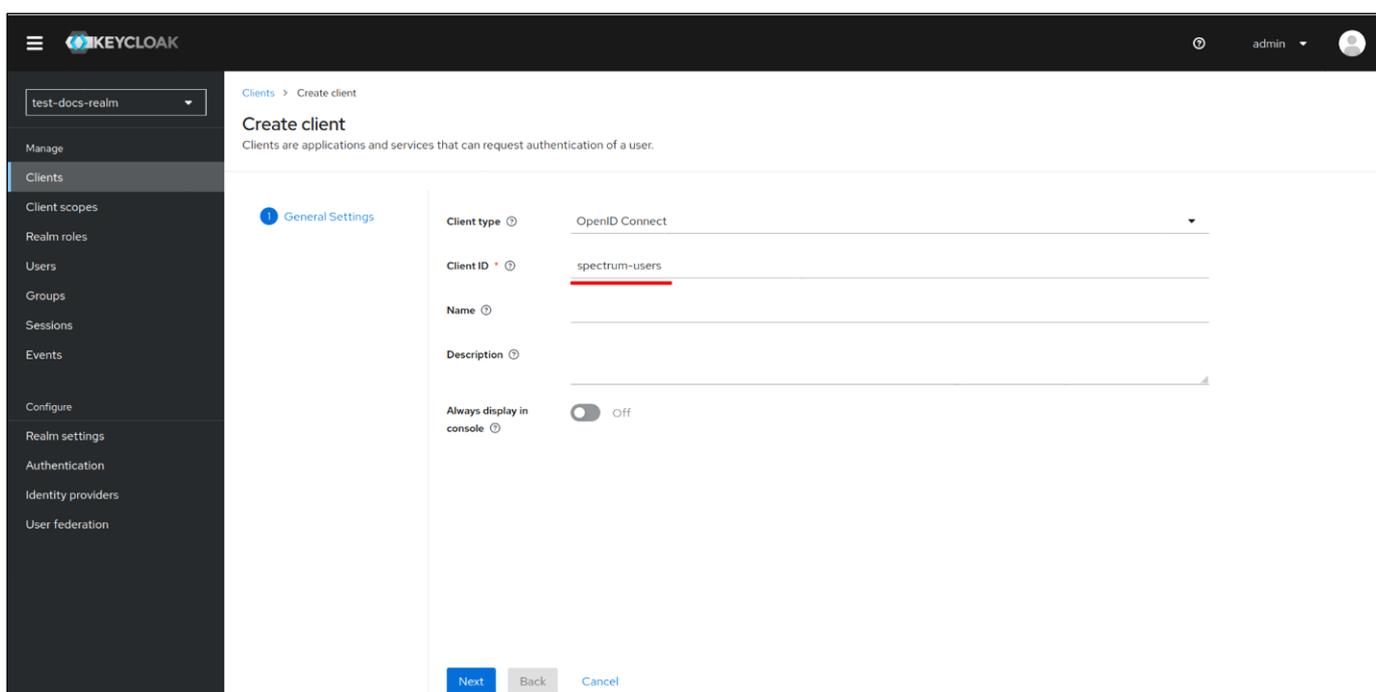
Пользовательский интерфейс консоли администратора **KeyCloak** может отличаться от приведенного в данном руководстве.

Для работы со "Спектром" при включенной авторизации необходимо настроить пользователей в **KeyCloak**.

Сначала необходимо выбрать нужный **Realm** (рабочее пространство) в выпадающем списке над меню или создать новый (на скриншоте ниже выбрано значение "**test-docs-realm**"). После задания нужного **Realm** надо перейти в раздел "**Клиенты**" (**Clients**) и нажать на кнопку "**Создать клиента**" (**Create client**):



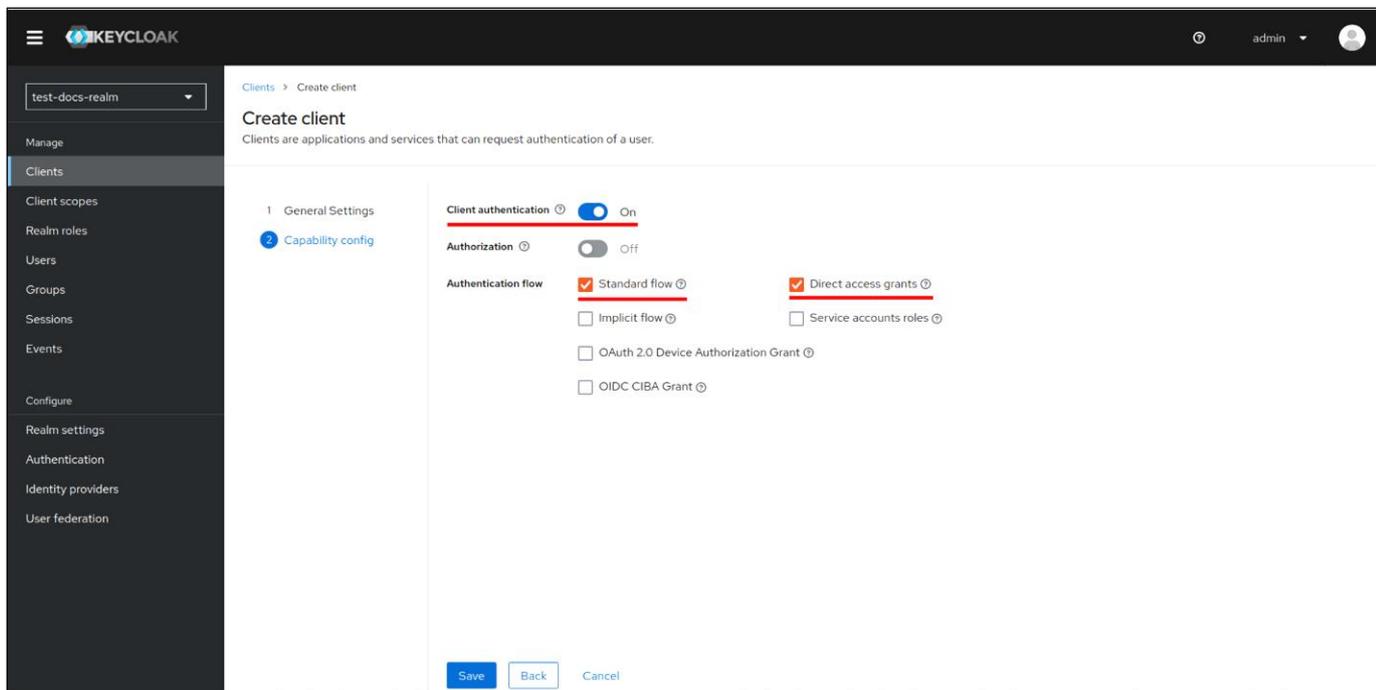
В поле "**Client ID**" нужно ввести идентификатор клиента и нажать кнопку "**Далее**" (**Next**). Заданный **ID** клиента будет в дальнейшем использоваться в параметрах конфигурации "Спектра":



На следующей странице:

- включить переключатель "**Аутентификация клиента**" (**Client authentication**),
- выставить флаг "**Standart Flow**",
- выставить флаг "**Direct access grants**",

после чего нажать кнопку "**Сохранить**" (**Save**).



Эти параметры позднее можно изменить в настройках клиента.

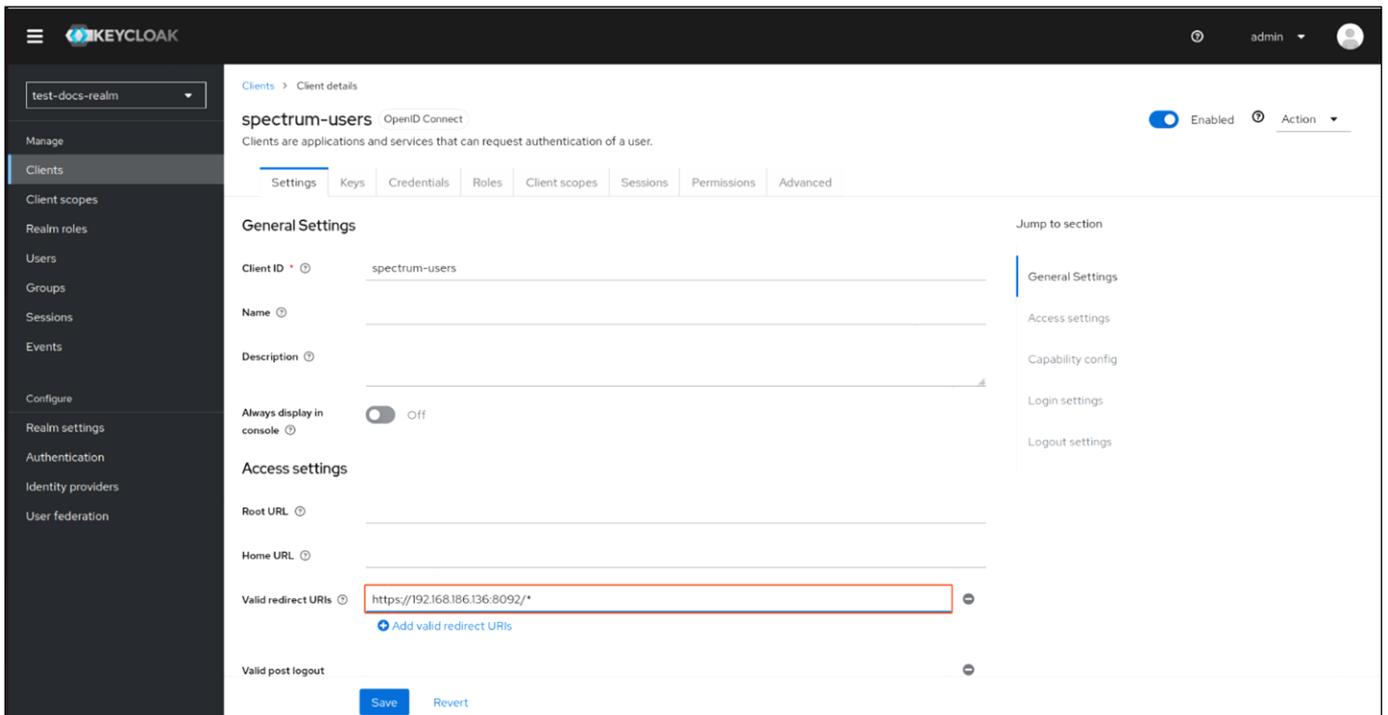
Перейти во вкладку "**Настройки**" (**Settings**), в поле "**Валидация URI перенаправления**" (**Valid redirect URIs**) добавить IP-адрес "Спектра" в формате:

```
PROTO://IP:XXXX/*
```

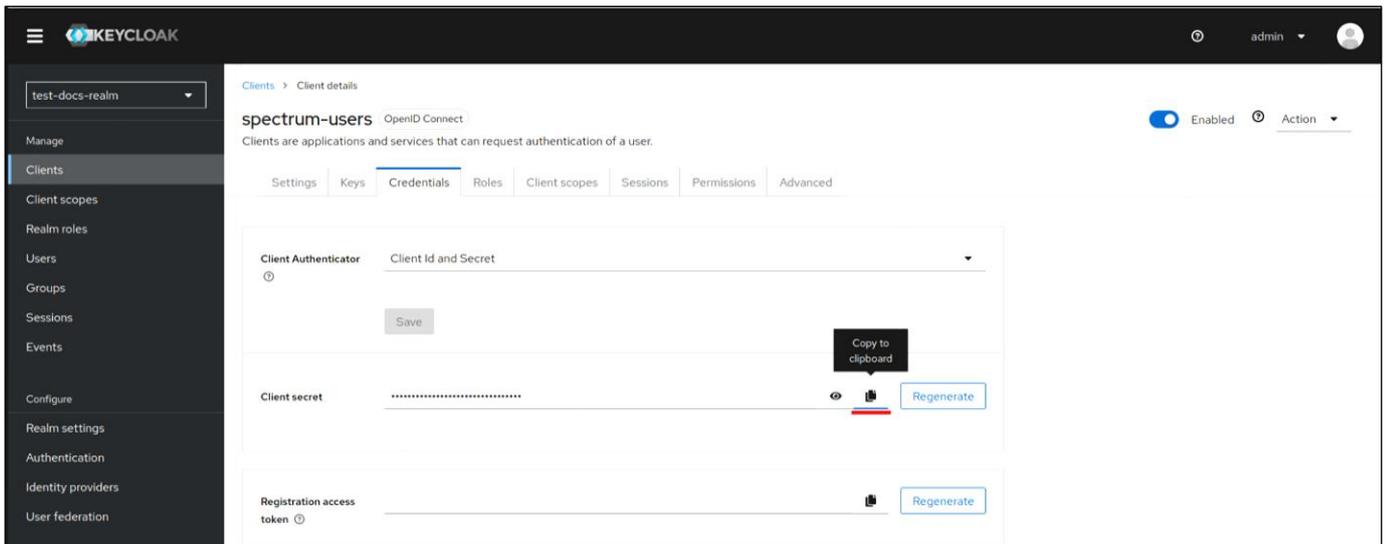
где

- **PROTO** - **http** или **https**, в зависимости от настроек **SSL** "Спектра" (см. п. 3.3);
- **IP** - IP адрес "Спектра";
- **XXXX** - порт "Спектра" для доступа к веб-интерфейсу.

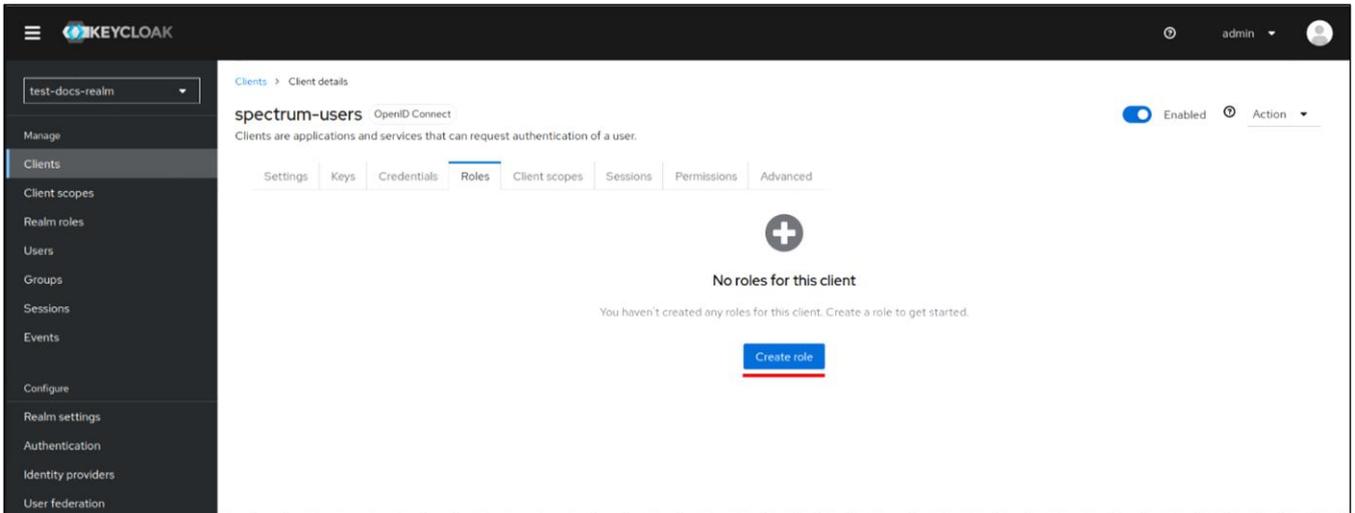
Нажать кнопку "**Сохранить**" (**Save**).



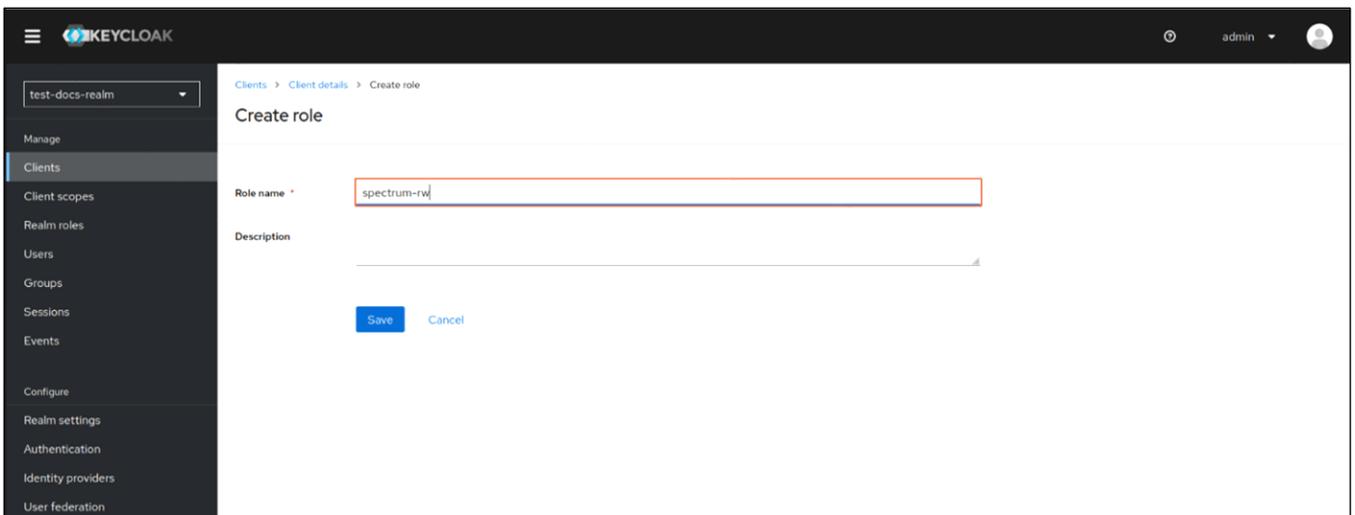
Во вкладке "Учетные данные" ("Credentials") есть поле "Секрет" ("Client secret"), значение которого будет в дальнейшем использоваться в параметрах конфигурации "Спектра". Значение можно скопировать в буфер обмена, нажав на соответствующую кнопку возле поля:



Перейти во вкладку **"Роли" (Roles)**, нажать кнопку **"Создать роль" (Create role)**:

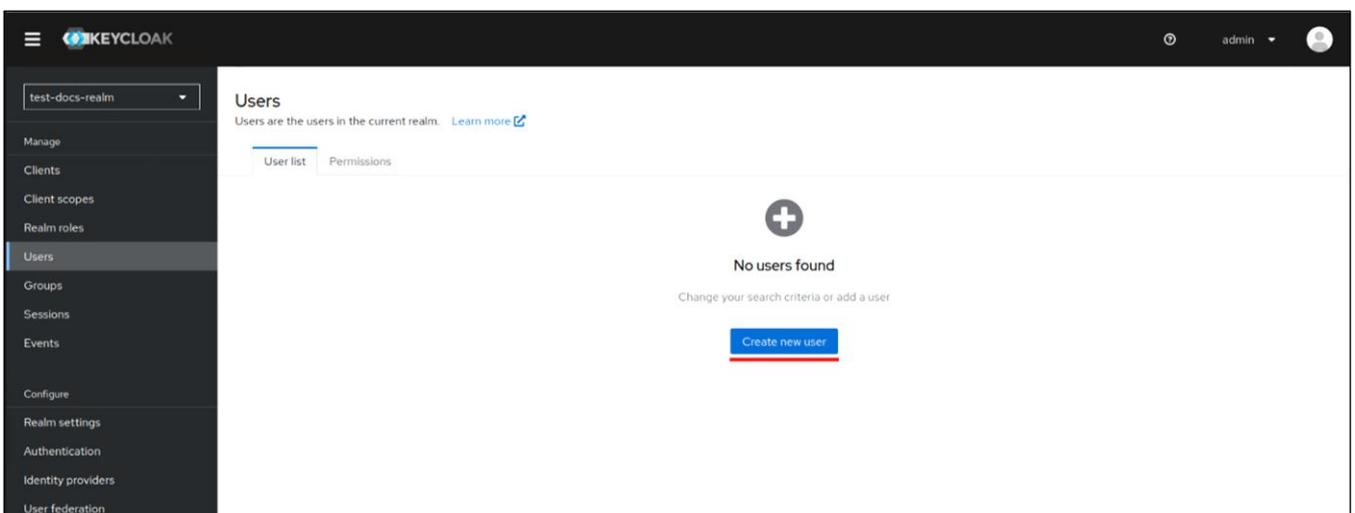


На экране создания роли ввести имя роли и нажать **"Сохранить" (Save)**:



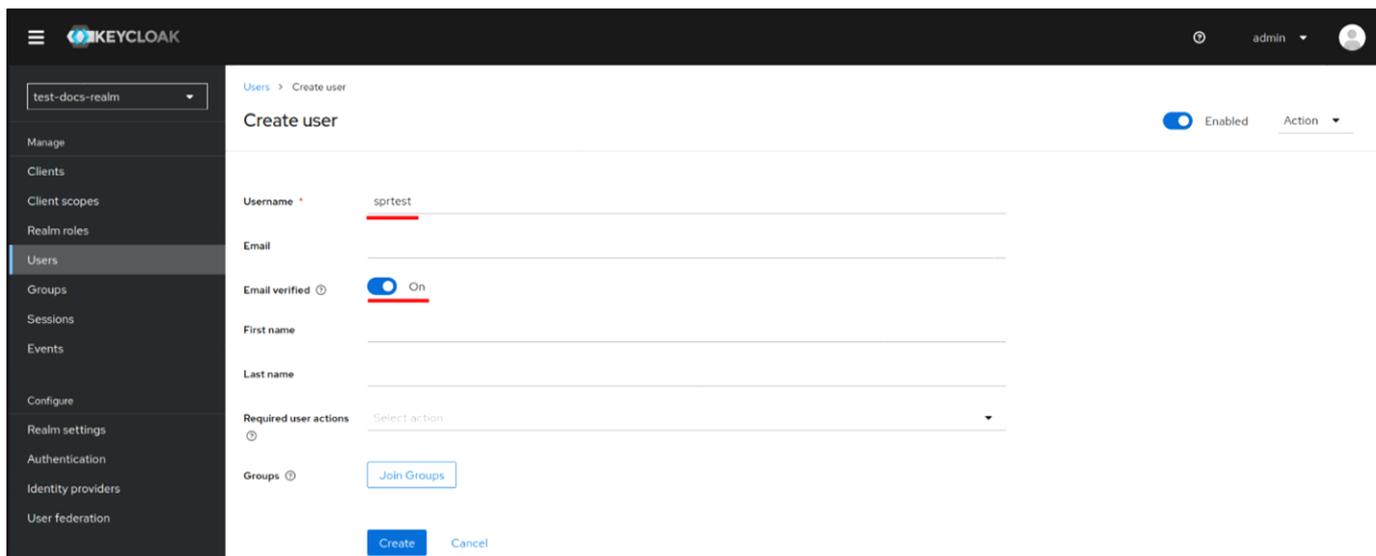
В примере, приведенном в данном руководстве, должны быть созданы две роли: **"spectrum-ro"** и **"spectrum-rw"**.

Перейти в пункт меню **"Пользователи" (Users)** и на странице **"Пользователи" (User list)** нажать кнопку **"Создать пользователя" (Create new user)**:



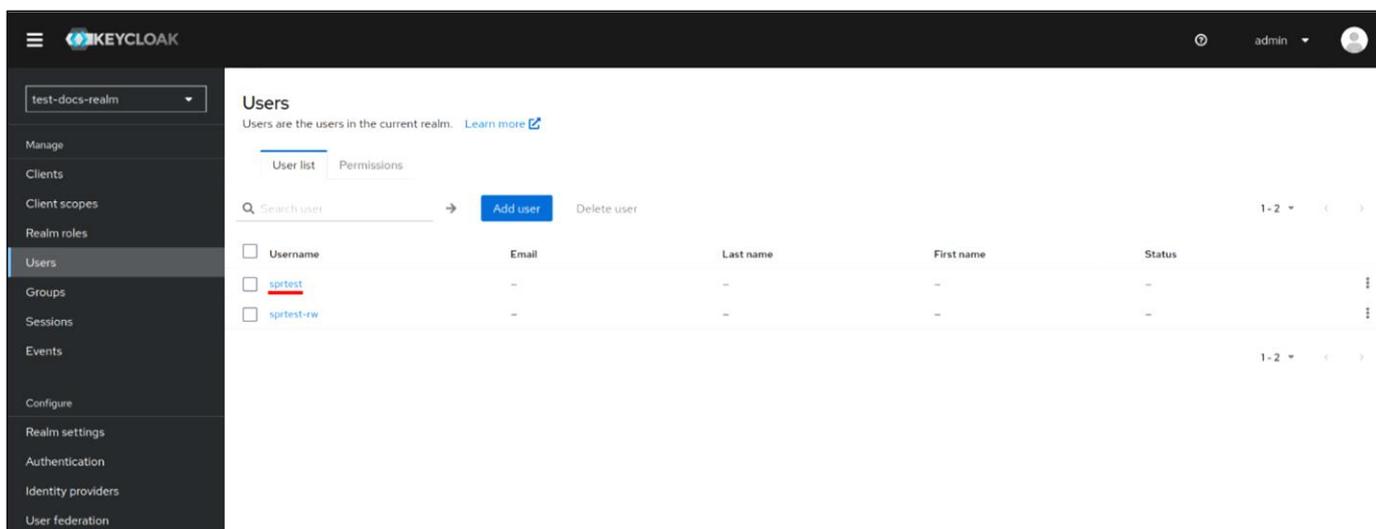
На экране создания пользователя:

- ввести желаемое имя пользователя,
- включить "E-mail подтвержден" (Email verified),
- нажать кнопку "Создать" (Create).

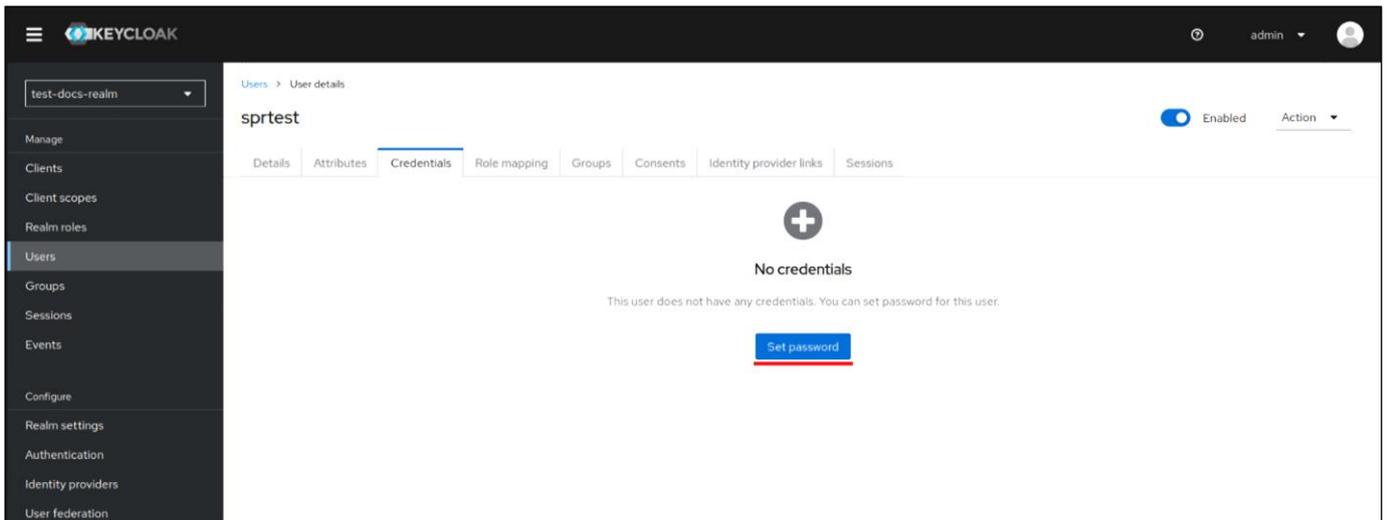


В примере, приведенном в данном руководстве, будет создано два пользователя: "sptest" и "sptest-rw".

Перейти в пункт меню "Пользователи" и выбрать созданного пользователя, кликнув по его имени:



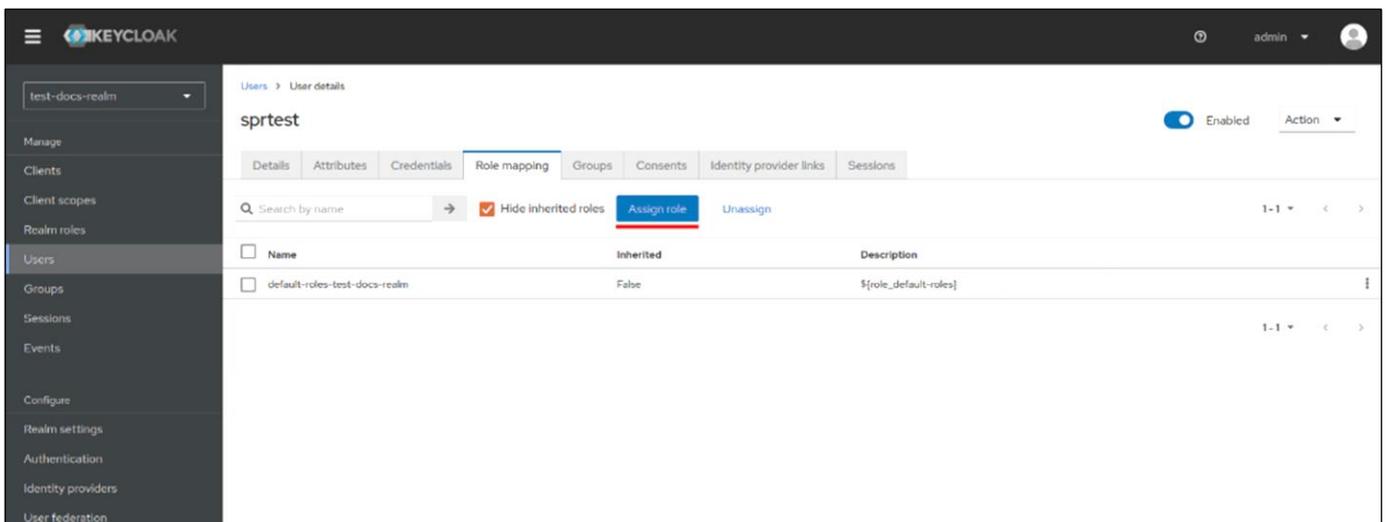
Перейти во вкладку "Учетные данные" (Credentials), нажать на кнопку "Установить пароль" (Set password):



В появившемся окне задать пароль, повторить пароль и выключить признак "Временный" (Temporary), если не требуется смена пароля пользователем при первом входе. Нажать кнопку "Сохранить" (Save):

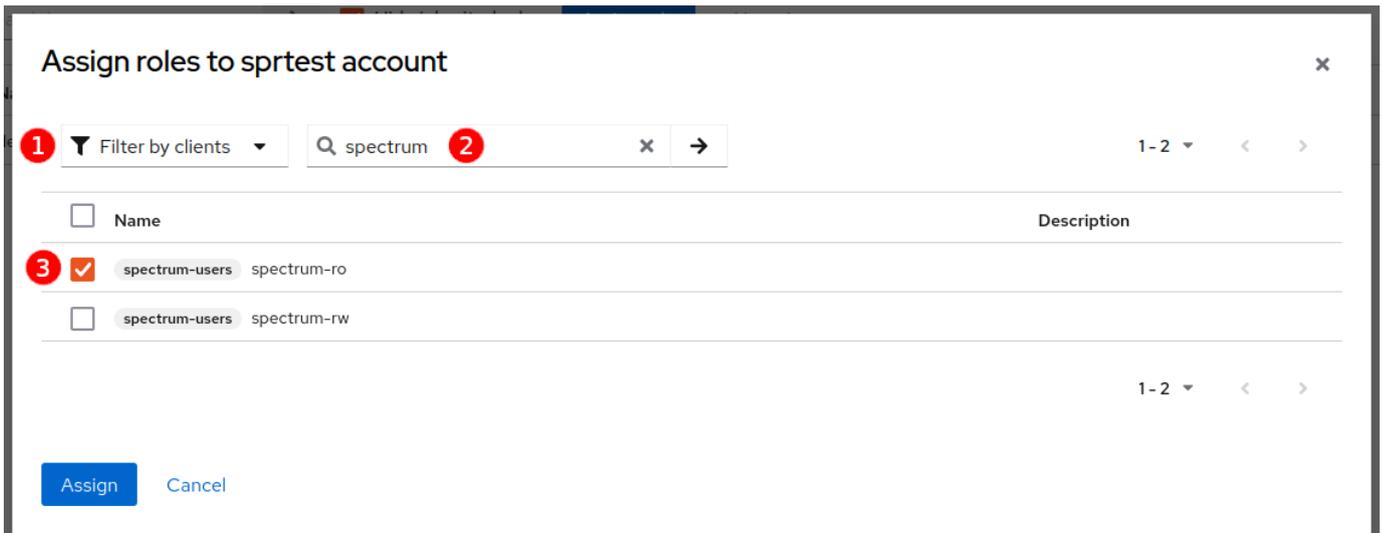
The screenshot shows a modal dialog titled 'Set password for sprtest'. It contains two password input fields: 'Password' and 'Password confirmation', both with eye icons to toggle visibility. Below the fields is a 'Temporary' toggle switch, which is currently turned 'Off'. At the bottom of the dialog are two buttons: 'Save' and 'Cancel'.

Перейти во вкладку "Сопоставление ролей" (Role mapping), нажать кнопку "Назначить роль" (Assign role):



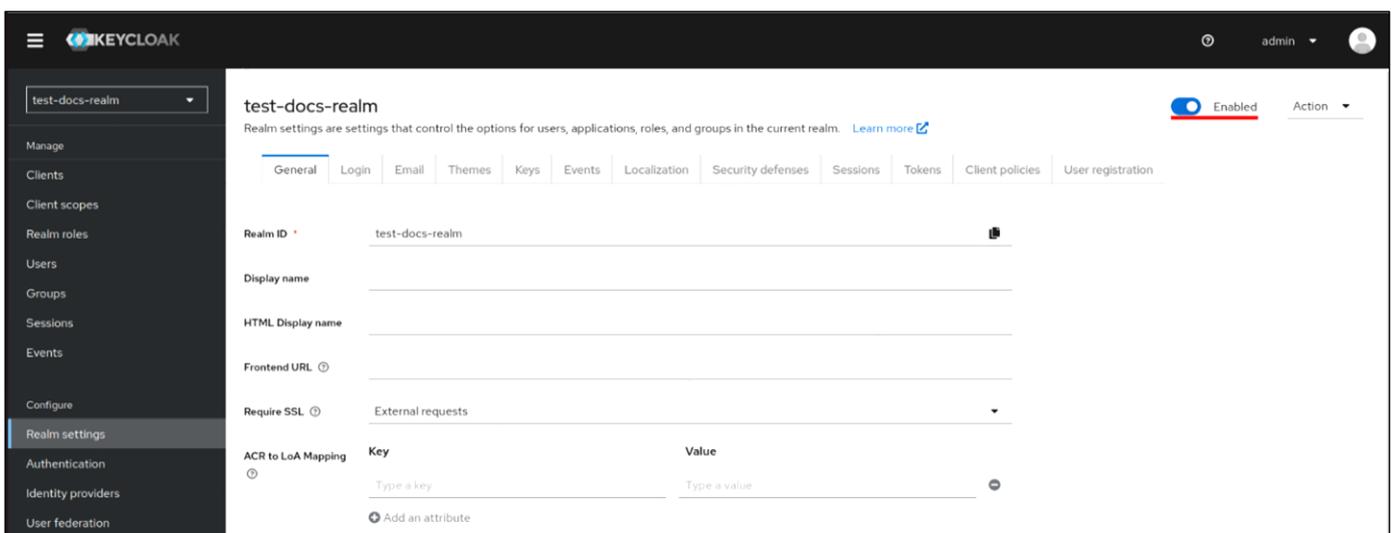
На экране назначения ролей:

- 1) в выпадающем списке фильтра выбрать "Отсортировать по клиентам" (**Filter by client**),
- 2) ввести полностью или частично имя необходимой роли / общей части имени ролей, созданной ранее,
- 3) выбрать роли для назначения данному пользователю, выставив соответствующие флаги,
- 4) нажать кнопку "Назначить" (**Assign**).



В примере, приведенном в данном руководстве, пользователю **sprtest** назначается роль **spectrum-ro**, а пользователю **sprtest-rw** – **spectrum-ro** и **spectrum-rw**.

После настройки необходимо включить настроенный **Realm**, если этого не было сделано ранее. Для этого нужно перейти в пункт меню "Настройки реалма" ("**Realm settings**") и включить переключатель "Включен" (**Enabled**):



Приложение 3. Настройка Avanpost FAM



Пользовательский интерфейс консоли администратора **Avanpost** может отличаться от приведенного в данном руководстве.

1. Создать приложение.

Перейти во вкладку "Приложения" и нажать кнопку "+ Добавить приложение":

Наименование	Тип	Статус
adminconsole	↔	Active
lk-client	↔	Active
my-spectrum	↔	Active
spectrum	↔	Active
spectrum-2	↔	Active
test	↔	Active

2. В окнах 1–3 выбрать / указать следующие данные:

*Новое приложение

Наименование:

Тип:

- SAML
- Open ID
- Reverse Proxy
- Desktop
- RADIUS

Показывать приложение пользователям

Отмена

- Выбрать **Open ID**;

- Указать **secret**;
- Указать необходимые **Redirect URLs** (их можно будет изменить в дальнейшем);
- В разделе "**Allowed grant types**" активировать пункты "**Authorization code**", "**Refresh token**".
- В разделе "**Scope**" (default) указать "**openid profile email permissions groups**".

В результате будет создано приложение, которое отобразится в списке на странице "**Приложения**".

Пример настроенного приложения **Spectrum**, с возможностью запускать веб-приложение на **localhost**:

The screenshot displays the 'Настройки интеграции' (Integration Settings) page in the Avanpost admin console. The interface includes a top navigation bar with the Avanpost logo and menu items for 'Пользователи', 'Группы', 'Приложения', 'Отчеты', and 'Сервис'. The user 'admin2' is logged in. The configuration fields are as follows:

- Client ID:** 7aa80354-66be-4115-874e-2fb92f0e3b27
- Public:**
- ID synonym:** spectrum-users
- Base URL:** http://localhost:3000/
- Redirect URIs:** http://localhost:3000/clusters
- Backchannel-logout URI:** Backchannel-logout URI
- Post logout redirect URIs:** Post logout redirect URIs
- Audience:** Audience claim value
- Audience type:** Строка Массив
- Client assertion type:** client_secret_basic
- Allowed grant types:** Authorization code, Implicit, Hybrid, Refresh token, Password, Client credentials
- Access token lifetime (in seconds):** 1800
- Access token type:** Случайное значение JSON web token
- Authorization code lifetime (in seconds):** 300
- Refresh token lifetime (in seconds):** 3600
- Refresh token strategy:** Переиспользовать refresh token
- ID token lifetime (in seconds):** 3600
- End session strategy:** Clear None
- Scope (default):** openid profile email permissions groups

3. Нажать на имя приложения и перейти на вкладку "Модель доступа":

Изменить

Основное Настройки MFA Scopes **Модель доступа** Сертификаты Тех.поддержка

Код spectrum_permissions

Описание Права пользователя

Права (Scopes)

	Код	Описание
1	spectrum-ro	Чтение
2	spectrum-rw	Чтение и запись

Сохранить Отмена

Задать следующие параметры:

- Код;
- Описание;
- Права.

Нажать на кнопку "Сохранить".

4. Перейти на вкладку "Сервис":

Avanpost Пользователи Группы Приложения Отчеты **Сервис**

- Менеджер сессий
- Настройки ролей и прав
- Настройки консоли администратора
- Настройки личного кабинета
- Настройки инициализации пользователей
- Настройки OpenID провайдеров
- Настройки SAML-провайдеров
- Настройки LDAP провайдеров
- Настройки политики паролей
- Настройки методов аутентификации
- Настройки процессов аутентификации
- Настройка оповещений пользователей
- Разделяемые секреты RADIUS
- Словари RADIUS VSA
- Редактор скриптов
- OTP токены
- Корневые сертификаты
- Лицензии
- О программе

5. Перейти в раздел "Настройка ролей и прав" и нажать кнопку "+ Добавить":

Avianpost Пользователи Группы Приложения Отчеты Сервис admin3

***Новая роль**

Наименование:

Описание:

Объекты доступа и права:

OAuth Scopes

- Все права
- Чтение
- Создание
- Обновление
- Удаление

qa_team qa

- Все права
- Только чтение
- Полный доступ

Группы

- Все права
- Чтение

ОТР токены

- Все права
- Чтение
- Создание
- Удаление
- Обновление

Radius

- Все права
- Чтение
- Создание
- Удаление
- Обновление

Журнал безопасности

- Все права

6. Указать наименование и описание:

Avianpost Пользователи Группы Приложения Отчеты Сервис admin3

***Новая роль**

Наименование:

Описание:

Объекты доступа и права:

OAuth Scopes

- Все права
- Чтение
- Создание
- Обновление
- Удаление

qa_team qa

- Все права
- Только чтение
- Полный доступ

Группы

- Все права
- Чтение
- Создание

ОТР токены

- Все права
- Чтение
- Создание
- Удаление
- Обновление

Radius

- Все права
- Чтение
- Создание
- Удаление
- Обновление

Журнал безопасности

- Все права

7. Для создаваемой роли выбрать необходимые права:

***Новая роль**

Наименование:

Описание:

Объекты доступа и права:

Интерфейсы	Права пользователя
<input type="checkbox"/> Все права <input type="checkbox"/> Использование интерфейса поддержки <input type="checkbox"/> Использование интерфейса администратора <input type="checkbox"/> Использование интерфейса ИБ <input type="checkbox"/> Администратор группы adminconsole <input type="checkbox"/> Администратор группы lk-client <input type="checkbox"/> Администратор приложения s3gw <input type="checkbox"/> Администратор группы test <input type="checkbox"/> Администратор группы LDAP-USERS	<input type="checkbox"/> Все права <input type="checkbox"/> Чтение <input checked="" type="checkbox"/> Чтение и запись

8. Создать группы для добавления пользователей.

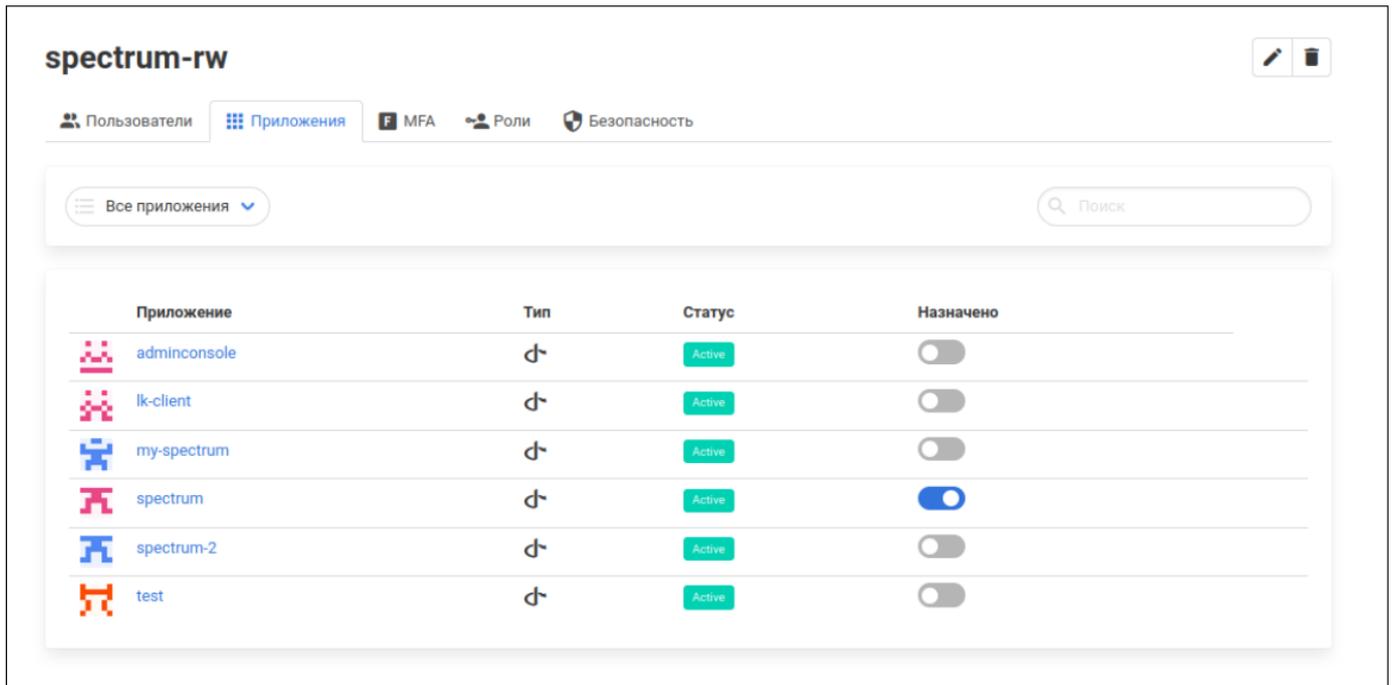
Перейти на вкладку "Группы" и нажать на кнопку "+ Добавить группу":

Avanpost Пользователи Группы Приложения Отчеты Сервис admin2

[+ Добавить группу](#) Поиск

Наименование	Пользователи	Приложения	
adminconsole	3	1	Сделать по умолчанию
GrafanaAdmin	1	1	Сделать по умолчанию
Grafana Admins	1	1	Сделать по умолчанию
Grafana Editor	1	1	Сделать по умолчанию
Grafana Viewer	1	1	Сделать по умолчанию
LDAP-USERS	0	1	Сделать по умолчанию
lk-client	1	1	Сделать по умолчанию
spectrum-rw	3	1	Сделать по умолчанию
spectrum-test	3	1	Сделать по умолчанию
test	1	1	Сделать по умолчанию

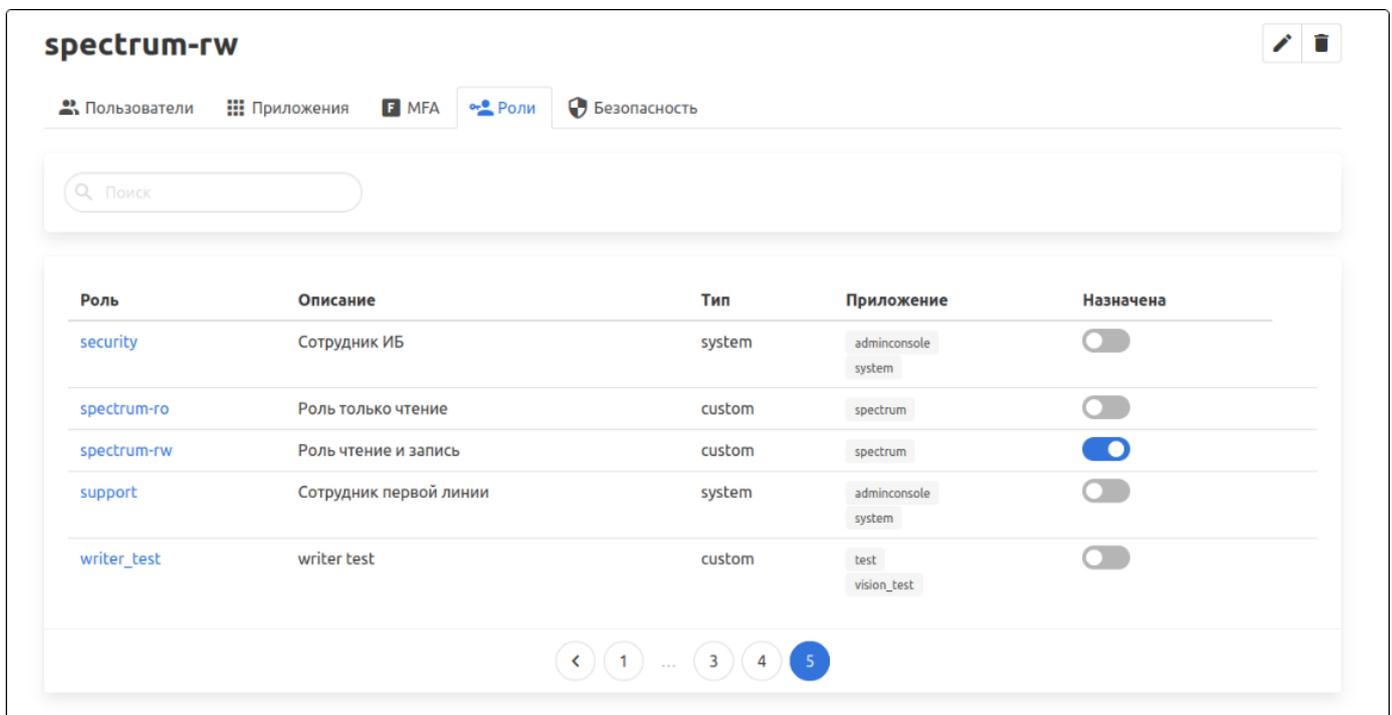
9. На вкладке "Приложения" указать приложение, для которого будет активна создаваемая группа:



The screenshot shows the 'spectrum-rw' interface with the 'Приложения' tab selected. The navigation menu includes 'Пользователи', 'Приложения', 'MFA', 'Роли', and 'Безопасность'. A search bar is present with the text 'Поиск'. Below the search bar is a table of applications:

Приложение	Тип	Статус	Назначено
adminconsole	🔗	Active	<input type="checkbox"/>
lk-client	🔗	Active	<input type="checkbox"/>
my-spectrum	🔗	Active	<input type="checkbox"/>
spectrum	🔗	Active	<input checked="" type="checkbox"/>
spectrum-2	🔗	Active	<input type="checkbox"/>
test	🔗	Active	<input type="checkbox"/>

10. На вкладке "Роли" выбрать роли, которыми будут обладать пользователи, входящие в создаваемую группу:



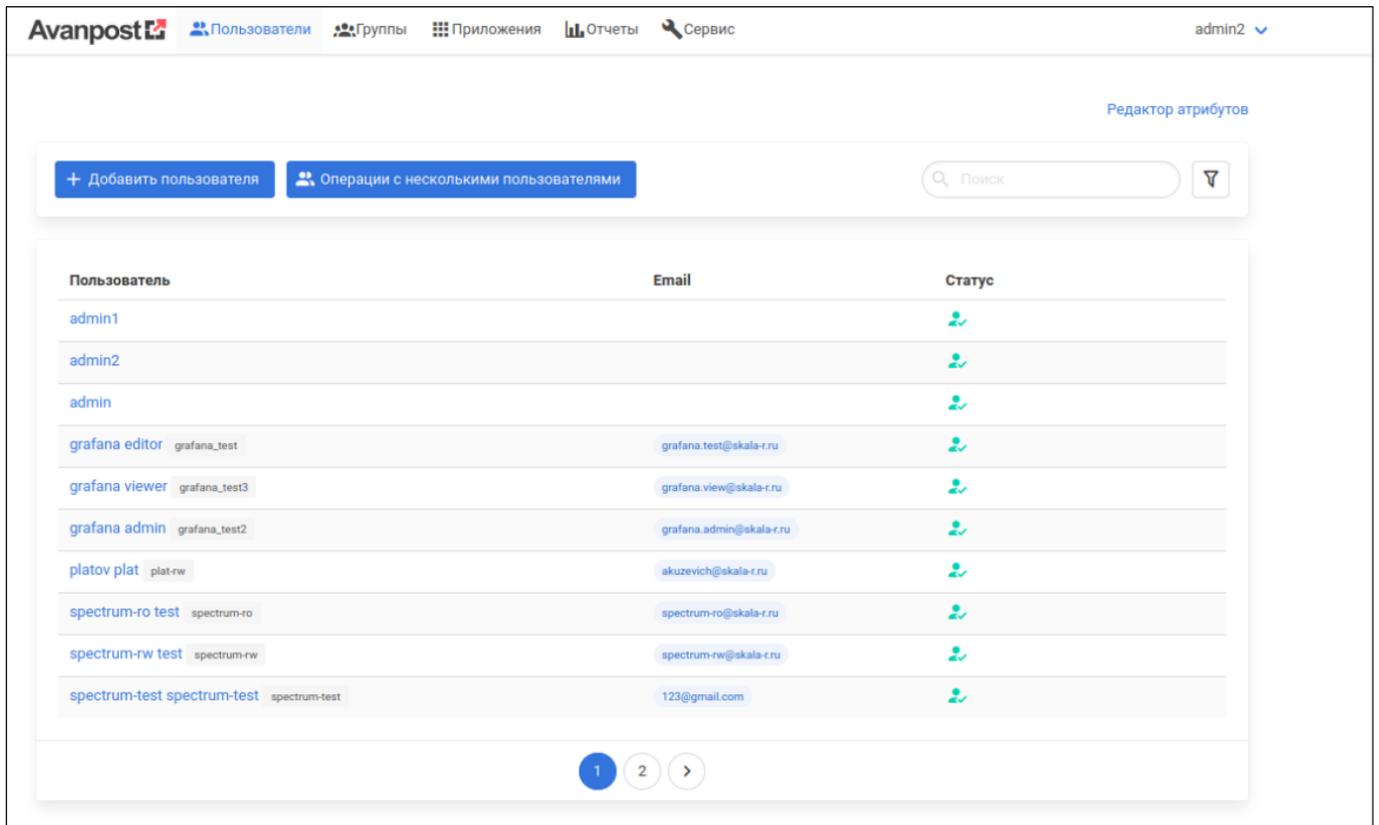
The screenshot shows the 'spectrum-rw' interface with the 'Роли' tab selected. The navigation menu includes 'Пользователи', 'Приложения', 'MFA', 'Роли', and 'Безопасность'. A search bar is present with the text 'Поиск'. Below the search bar is a table of roles:

Роль	Описание	Тип	Приложение	Назначена
security	Сотрудник ИБ	system	adminconsole system	<input type="checkbox"/>
spectrum-ro	Роль только чтение	custom	spectrum	<input type="checkbox"/>
spectrum-rw	Роль чтение и запись	custom	spectrum	<input checked="" type="checkbox"/>
support	Сотрудник первой линии	system	adminconsole system	<input type="checkbox"/>
writer_test	writer test	custom	test vision_test	<input type="checkbox"/>

At the bottom of the table, there is a pagination control showing a sequence of numbers: < 1 ... 3 4 5, where the number 5 is highlighted in a blue circle.

11. Создать пользователя.

Перейти на вкладку "Пользователи" и нажать кнопку "+ Добавить пользователя":

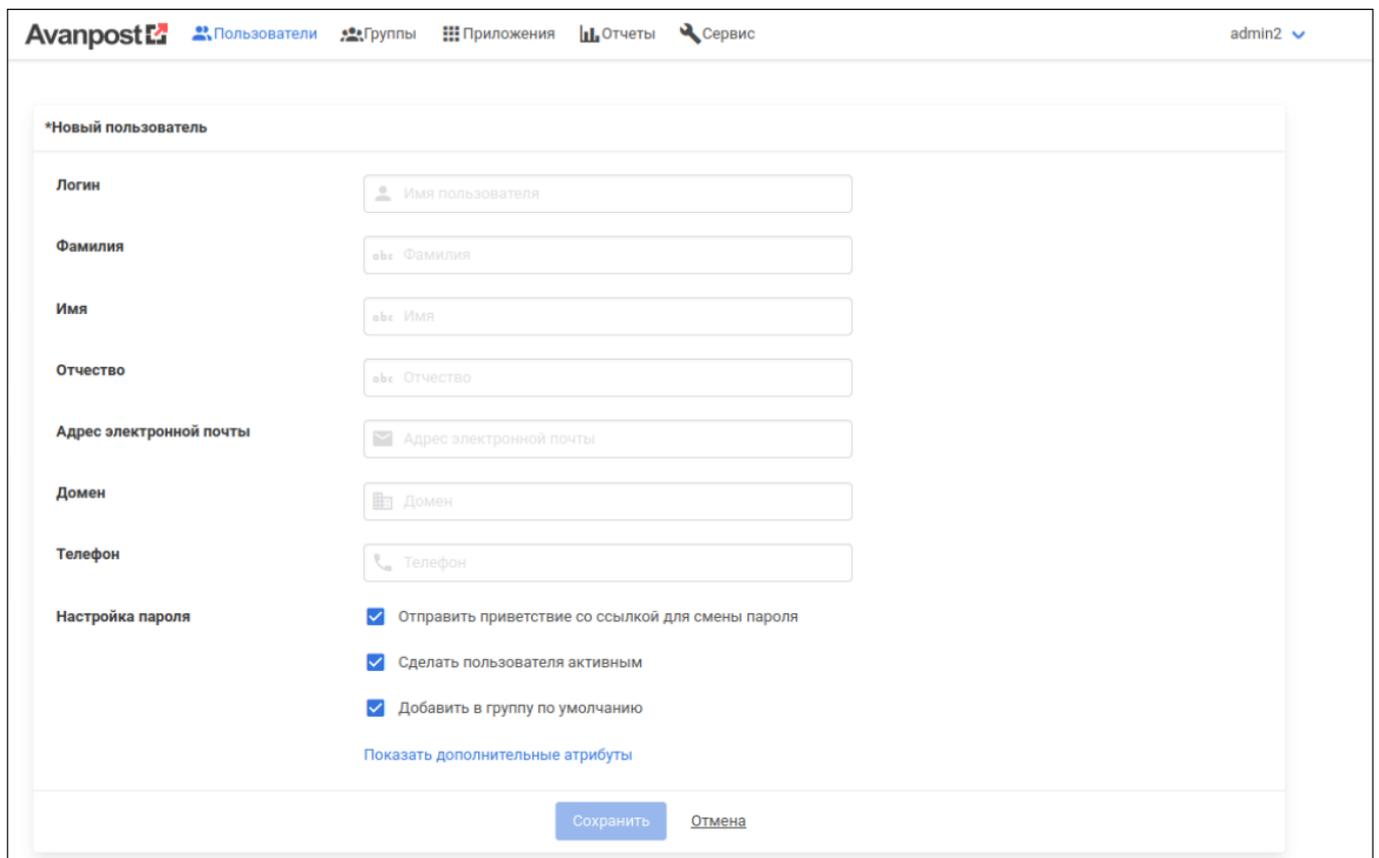


The screenshot shows the Avanspost user management interface. At the top, there is a navigation bar with the Avanspost logo and menu items: Пользователи, Группы, Приложения, Отчеты, and Сервис. The user 'admin2' is logged in. Below the navigation bar, there is a search bar and a filter icon. The main content area contains a table of users with the following columns: Пользователь, Email, and Статус.

Пользователь	Email	Статус
admin1		
admin2		
admin		
grafana editor grafana_test	grafana.test@skala-r.ru	
grafana viewer grafana_test3	grafana.view@skala-r.ru	
grafana admin grafana_test2	grafana.admin@skala-r.ru	
platov plat plat-rw	akuzevich@skala-r.ru	
spectrum-ro test spectrum-ro	spectrum-ro@skala-r.ru	
spectrum-rw test spectrum-rw	spectrum-rw@skala-r.ru	
spectrum-test spectrum-test spectrum-test	123@gmail.com	

At the bottom of the table, there are pagination controls showing page 1 of 2.

Указать следующие данные:



The screenshot shows the Avanspost user creation form. At the top, there is a navigation bar with the Avanspost logo and menu items: Пользователи, Группы, Приложения, Отчеты, and Сервис. The user 'admin2' is logged in. Below the navigation bar, there is a search bar and a filter icon. The main content area contains a form titled "*Новый пользователь" with the following fields:

- Логин:
- Фамилия:
- Имя:
- Отчество:
- Адрес электронной почты:
- Домен:
- Телефон:
- Настройка пароля:
 - Отправить приветствие со ссылкой для смены пароля
 - Сделать пользователя активным
 - Добавить в группу по умолчанию

At the bottom of the form, there is a link "Показать дополнительные атрибуты" and two buttons: "Сохранить" and "Отмена".

- Логин
- Фамилия
- Имя
- Адрес электронной почты

Нажать на кнопку **"Сохранить"**.

12. Настроить пароль.

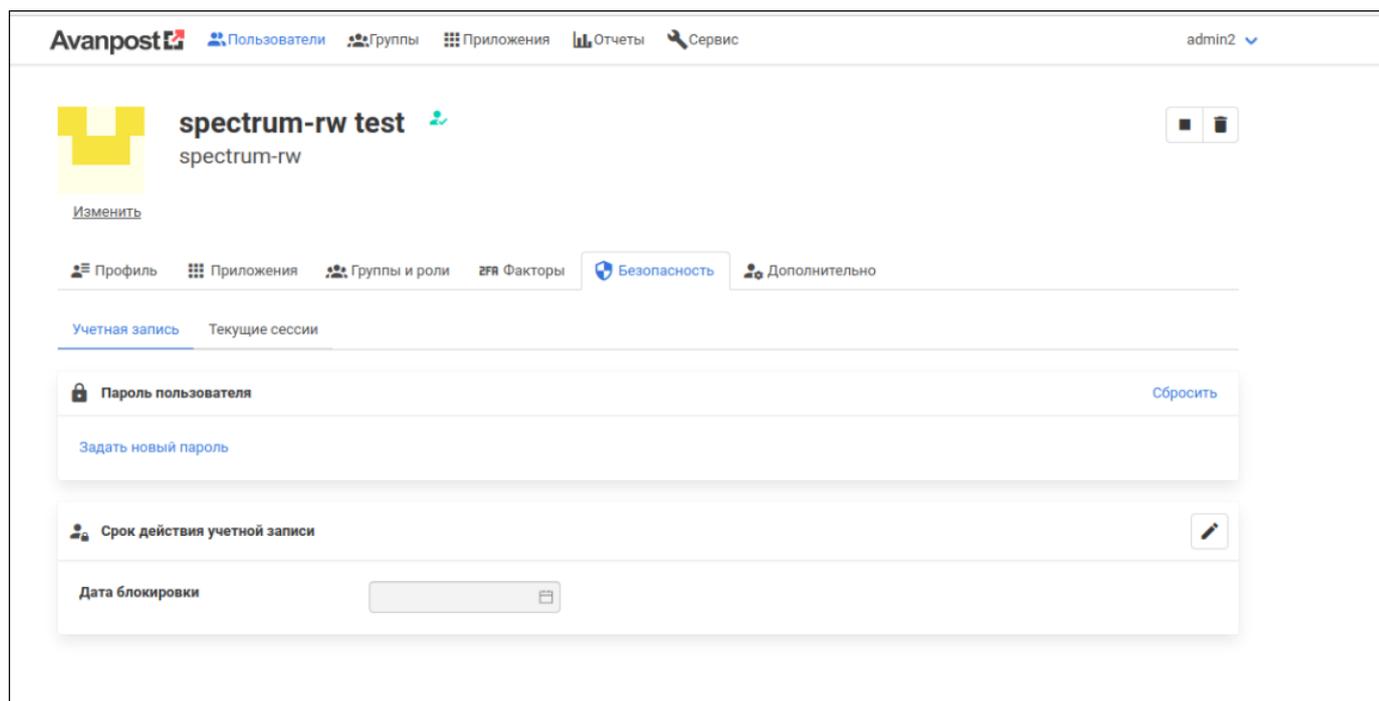
Перейти на вкладку **"Пользователи"** и нажать на имя требуемого пользователя:

The screenshot displays the Avianpost user management interface. The top navigation bar includes the Avianpost logo and tabs for 'Пользователи', 'Группы', 'Приложения', 'Отчеты', and 'Сервис'. The user 'admin2' is currently selected. Below the navigation, there are buttons for '+ Добавить пользователя' and 'Операции с несколькими пользователями', along with a search bar and a 'Редактор атрибутов' link. The main content area shows a table of users:

Пользователь	Email	Статус
admin1		✓
admin2		✓
admin		✓
grafana editor grafana_test	grafana.test@skala-r.ru	✓
grafana viewer grafana_test3	grafana.view@skala-r.ru	✓
grafana admin grafana_test2	grafana.admin@skala-r.ru	✓
platov plat plat-rw	akuzevich@skala-r.ru	✓
spectrum-ro test spectrum-ro	spectrum-ro@skala-r.ru	✓
spectrum-rw test spectrum-rw	spectrum-rw@skala-r.ru	✓
spectrum-test spectrum-test spectrum-test	123@gmail.com	✓

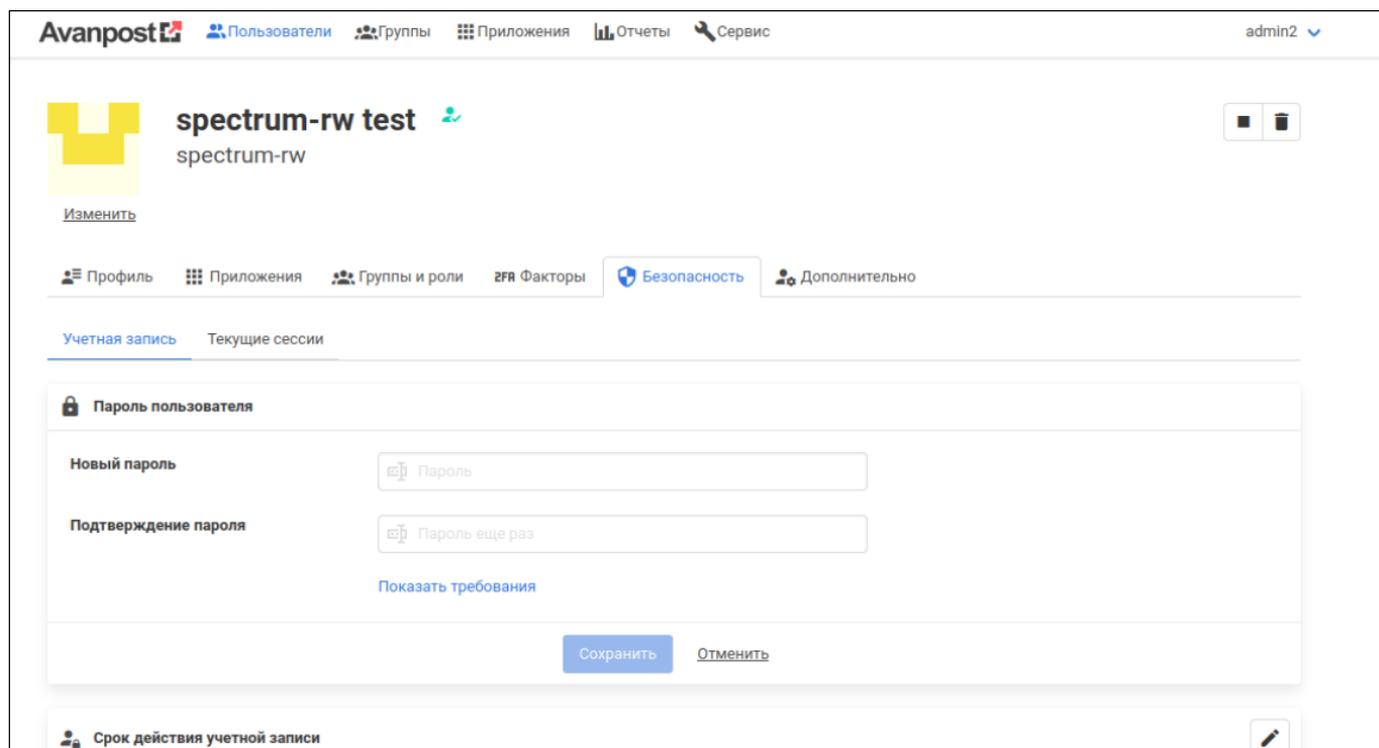
At the bottom of the table, there are pagination controls showing '1', '2', and a right arrow.

Перейти на вкладку "**Безопасность**" и нажать на ссылку "**Задать новый пароль**":



The screenshot shows the Avast user management interface. At the top, there is a navigation bar with the Avast logo and menu items: Пользователи, Группы, Приложения, Отчеты, and Сервис. The user profile for 'spectrum-rw test' is displayed, with a yellow profile picture and the name 'spectrum-rw'. Below the profile information, there are tabs for 'Профиль', 'Приложения', 'Группы и роли', '2FA Факторы', 'Безопасность' (selected), and 'Дополнительно'. Under the 'Безопасность' tab, there are sub-tabs for 'Учетная запись' and 'Текущие сессии'. The 'Учетная запись' sub-tab is active, showing the 'Пароль пользователя' section with a 'Сбросить' link and a 'Задать новый пароль' button. Below this, there is a 'Срок действия учетной записи' section with an edit icon, and a 'Дата блокировки' section with a date picker.

Ввести новый пароль, подтверждение пароля и нажать на кнопку "**Сохранить**":



The screenshot shows the same Avast user management interface as above, but with the password change form open. The 'Пароль пользователя' section is expanded, showing two input fields: 'Новый пароль' (New password) and 'Подтверждение пароля' (Confirm password). Both fields have a placeholder text 'Пароль' and 'Пароль еще раз' respectively. Below the input fields, there is a 'Показать требования' link. At the bottom of the form, there are two buttons: 'Сохранить' (Save) and 'Отменить' (Cancel). The 'Срок действия учетной записи' section is partially visible at the bottom of the screen.

13. Добавить пользователя в созданную группу.

Перейти на вкладку "Группы" и нажать на имя созданной группы:

Avanpost Пользователи Группы Приложения Отчеты Сервис admin2

+ Добавить группу Поиск

Наименование	Пользователи	Приложения	
adminconsole	3	1	Сделать по умолчанию
GrafanaAdmin	1	1	Сделать по умолчанию
Grafana Admins	1	1	Сделать по умолчанию
Grafana Editor	1	1	Сделать по умолчанию
Grafana Viewer	1	1	Сделать по умолчанию
LDAP-USERS	0	1	Сделать по умолчанию
lk-client	1	1	Сделать по умолчанию
spectrum-rw	3	1	Сделать по умолчанию
spectrum-test	3	1	Сделать по умолчанию
test	1	1	Сделать по умолчанию

Отобразится вкладка "Пользователи" созданной группы.

Нажать на кнопку "Добавить пользователей":

spectrum-rw

Пользователи Приложения MFA Роли Безопасность

+ Добавить пользователей Поиск

Пользователь	Email	Статус	Удалить
admin		✓	✕
spectrum-rw test spectrum-rw	spectrum-rw@skala-r.ru	✓	✕
Афти Александр AAfti@skala-r.ru	AAfti@skala-r.ru	✓	✕

Выбрать в списке требуемого пользователя и нажать на кнопку "Сохранить":

The screenshot shows the Avast user management interface. At the top, there is a navigation bar with the Avast logo and menu items: Пользователи, Группы, Приложения, Отчеты, and Сервис. The main heading is "Группа - spectrum-rw". Below this is a modal window titled "Добавить пользователей". Inside the modal, there is a search bar labeled "Поиск". A list of users is displayed, each with a checkbox and a label: "Все", "admin1", "admin2", "grafana viewer grafana_test3", "grafana admin grafana_test2", "grafana editor grafana_test", "platov plat plat-rw", "spectrum-ro test spectrum-ro", "spectrum-test spectrum-test spectrum-test", "testetedd testetedd testetedd", and "testing testing testing". At the bottom of the modal, there are pagination controls (1, 2, >) and two buttons: "Сохранить (0)" and "Отмена".

14. Проверка.

Перейти во вкладки "Приложения" и "Группы и роли" и убедиться, что добавленный пользователь присутствует в созданных приложениях и группе.

The screenshot shows the user interface for a user named 'spectrum-rw test'. The top navigation bar includes 'Пользователи', 'Группы', 'Приложения', 'Отчеты', and 'Сервис'. The user's profile is visible, and the 'Приложения' tab is selected. Below the profile, there are tabs for 'Профиль', 'Приложения', 'Группы и роли', 'Факторы', 'Безопасность', and 'Дополнительно'. The 'Приложения' section shows a table of available applications:

Приложение	Тип	Статус
spectrum		Активен

The screenshot shows the user interface for the same user, but with the 'Группы и роли' tab selected. The navigation bar and user profile are the same. The 'Группы и роли' section shows a search bar and a table of user groups:

Наименование
spectrum-rw