Sharx Base 6.2

Руководство пользователя. Командная строка

Оглавление

1. Руководство пользователя 7
2. Общие сведения 7
  2.1. Назначение Sharx Base 7
  2.2. Что такое логический кластер? 7
  2.3. Разделение логического кластера на виртуальные единицы 8
3. Требования 9
  3.1. Требования к узлу 9
  3.2. Требования к сетевой инфраструктуре 9
    3.2.1. Общие требования к сетевой инфраструктуре 9
    3.2.2. Требования и ограничения сегментирования сети 10
    3.2.3. Требования к сетевой доступности 11
  3.3. Требования к АРМ и мобильному устройству 15
  3.4. Требования к интерфейсу командной строки 15
    3.4.1. Стандартный sdc-cli 16
    3.4.2. Отчуждаемый sdc-cli 16
4. Быстрый старт 16
5. Аутентификация 17
6. Просмотреть и изменить структуру ВЦОД 18
7. Валидация объектов ВЦОД 19
  7.1. Управление ключами ВЦОД 19
  7.2. Отправка сертификатов пользователям и включение двухфакторной аутентификации 20
  7.3. Настройка клиентской машины для входа во ВЦОД с использованием сертификата 22
  7.4. Настройка валидации объектов ВЦОД 23
  7.5. Оповещение пользователя о событиях КЦ 27

7.6. Рекомендации по действиям Администратора ВЦОД при нарушении контроля целостности 28

8. Уведомления 28

8.1. Создать почтовый сервер 29

8.2. Сформировать список адресов 29

8.3. Задать шаблоны сообщений 30

8.4. Настроить маршрут отправки уведомлений 31

8.5. Настроить уведомления с помощью файла YAML 31

8.6. Контроль отправленных сообщений 32

9. Права доступа и роли 32

9.1. Права доступа 32

9.2. Роли 33

9.2.1. Роли по умолчанию 34

9.2.2. Действия с ролями 35

10. Пользователи 37

10.1. Создать пользователя 37

10.2. Управлять пользователями 38

11. Загрузить пользователей с использованием LDAP 40

11.1. Общий принцип работы 40

11.2. Управлять конфигурациями LDAP 40

11.2.1. Добавить конфигурацию 40

11.2.2. Просмотреть и проверить статус конфигурации LDAP-каталога 43

11.2.3. Обновить конфигурацию 43

11.2.4. Удалить конфигурацию 44

11.3. Активировать синхронизацию с LDAP 44

11.4. Проверить активацию LDAP 45

12. Политика безопасности учетных записей 45

13. Команды управления базовыми параметрами механизмов безопасности 45

14. Параметры безопасности ВЦОД для обычного пользователя 46

15. Параметры ВЦОД для привилегированного пользователя 50
16. Двухфакторная аутентификация 52
16.1. Проверка настройки 53
17. Создать виртуальный кластер 54
17.1. Создать виртуальный кластер 55
17.2. Дополнительные команды 56
18. Управлять виртуальным кластером 57
18.1. Изменить параметры виртуального кластера 57
18.2. Управлять узлами виртуального кластера 58
18.3. Удалить виртуальный кластер 59
19. Лимиты пользователей 59
19.1. Включить режим лимитов для пользователей 59
19.2. Настроить лимиты пользователя 60
19.3. Обновить лимиты пользователя 60
19.4. Просмотреть лимиты пользователей 61
19.5. Удалить лимиты пользователей 62
20. Настроить метки виртуального кластера 62
20.1. Ограничения работы с метками в виртуальном кластере 64
20.2. Операции с метками в виртуальном кластере 64
21. Подготовить хранилище 66
21.1. Libvirt 66
21.1.1. Создать пул 66
21.1.2. Работа с пулами 67
21.2. РСХД 68
21.3. NFS 68
22. Журнал безопасности 68
22.1. Просмотреть события в журнале безопасности 69

22.2. Настроить параметры записи событий 70
22.2.1. Настроить параметры регистрации событий вручную 70
22.2.2. Загрузить файл с параметрами регистрации событий 71
23. Виртуальные машины 72
23.1. Аутентифицироваться во ВЦОД 73
23.2. Создать том 73
23.2.1. Libvirt. Создать том 74
23.2.2. РСХД. Создать том 75
23.2.3. NFS. Создать том 77
23.3. Работа с томом 79
23.3.1. Libvirt. Работа с томами 79
23.3.2. РСХД 80
23.3.3. NFS. Работа с томами 83
23.4. Сетевые интерфейсы и правила доступа 84
23.4.1. VXLAN-подсети 84
23.4.2. Сетевые интерфейсы 86
23.4.3. Статические IP-адреса 87
23.4.4. Правила доступа 87
23.5. Правила размещения виртуальных машин 92
23.5.1. Выбор узла по идентификатору nodeName 92
23.5.2. Использование меток узлов nodeSelector 92
23.5.3. nodeAffinity, vmAffinity и colocation 94
23.6. Оптимизация ресурсов ВМ 100
23.6.1. Настройка переподписки ЦПУ 101
23.6.2. Настройка переподписки ОЗУ 101
23.7. Операции с виртуальными машинами 102
23.7.1. Создать ВМ 102
23.7.2. Просмотреть ВМ 105

23.7.3. Обновить ВМ 106
23.7.4. Изменить статус ВМ 106
23.7.5. Назначить метку СКЗИ виртуальной машине 107
23.7.6. Удалить ВМ 107
23.8. Запустить виртуальные машины 108
23.9. Снимки ВМ и томов 108
23.9.1. Снимки ВМ 109
23.9.2. Снимки томов Libvirt 110
23.9.3. Снимки томов РСХД 111
23.9.4. Снимки томов NFS 112
23.10. Резервные копии виртуальных машин 113
23.11. Восстановить виртуальную машину 115
23.11.1. Подготовить резервную копию 115
23.11.2. Восстановить ВМ 116
24. Перевести узлы в сервисный режим 117
24.1. Перевести узлы в сервисный режим 117
24.2. Отключить сервисный режим 117
25. Мониторинг 118

1. Руководство пользователя

В данном руководстве описана последовательность действий пользователей через **командную строку** для создания и управления виртуальной инфраструктурой: ВЦОД, виртуальными кластерами, пользовательскими ресурсами, виртуальными машинами.

Сначала действия выполняет пользователь с ролью **Администратор ВЦОД**.

Затем описаны действия пользователя с ролью **Разработчик ВМ**.

Некоторые функции доступны другим ролям в пределах их Прав доступа.

2. Общие сведения

2.1. Назначение Sharx Base

Платформа **Sharx Base 6.2** (далее — Платформа, Sharx Base) — средство виртуализации, предназначенное для развертывания и централизованного управления виртуальной инфраструктурой, частными и публичными облаками.

Платформа позволяет централизованно в едином пространстве эффективно использовать аппаратные ресурсы клиента, объединяя их в логический кластер. Наличие API позволяет интегрировать Платформу в инфраструктуру клиента.

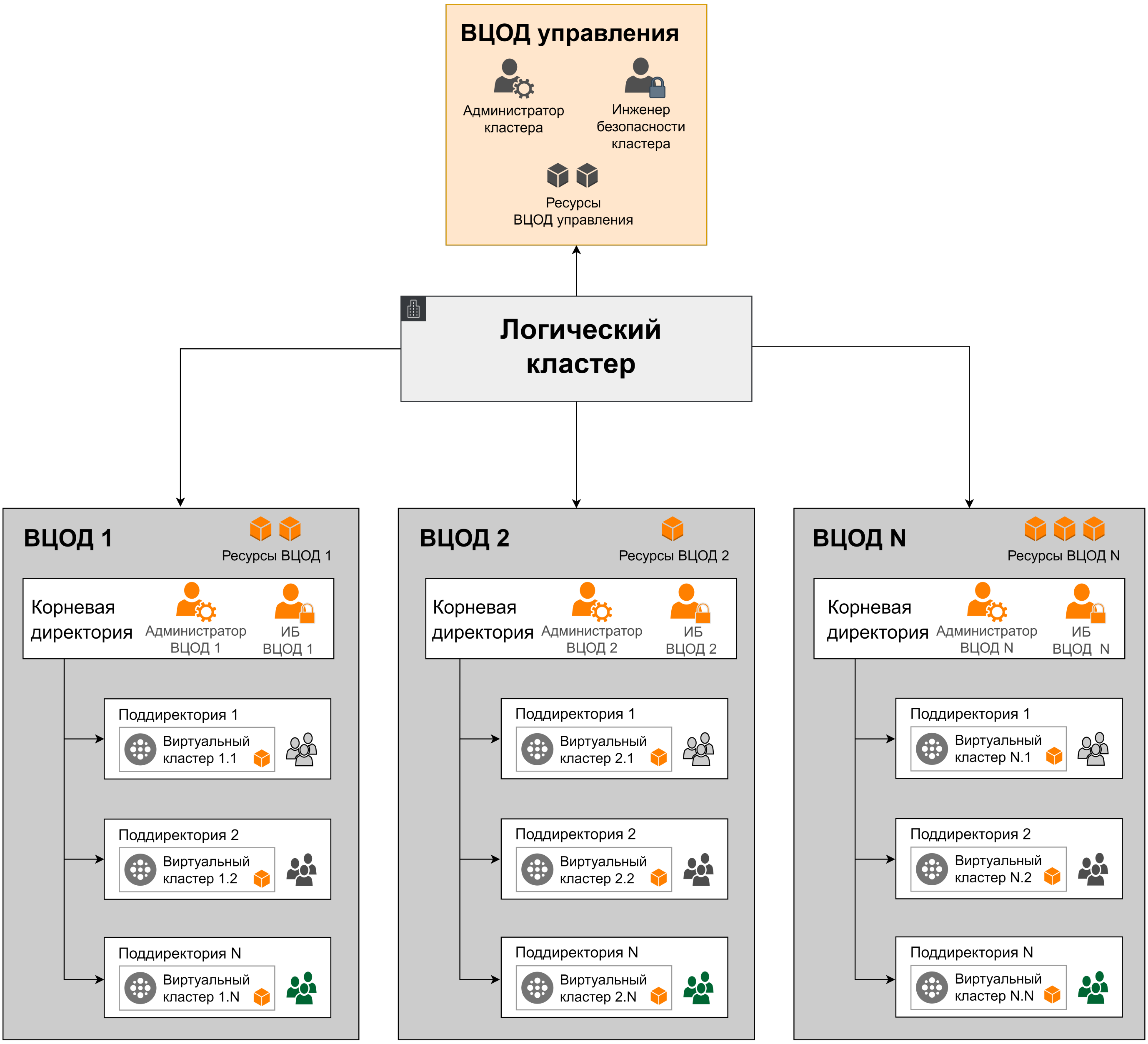
Sharx Base является гипервизором 1 типа, то есть включает все необходимые системные и прикладные компоненты для развертывания на аппаратных средствах без необходимости установки дополнительного ПО. Доступ к прикладным компонентам предоставляется пользователям и администраторам Sharx Base через графический интерфейс, интерфейс командной строки или вызовы API. Доступ к системным компонентам не предоставляется потребителю. Данный доступ использует только техническая поддержка предприятия-изготовителя в рамках запроса потребителя на техническую поддержку платформы Sharx Base.

2.2. Что такое логический кластер?

Sharx Base объединяет все физические ресурсы серверных узлов заказчика в единый логический блок — логический кластер. Логический кластер представляет собой пространство с распределенным хранилищем высокой плотности и вычислительными ресурсами. Взаимодействие узлов кластера происходит через телекоммуникационную инфраструктуру и сетевые интерфейсы каждого узла.

2.3. Разделение логического кластера на виртуальные единицы

Ресурсы логического кластера разделяются на виртуальные единицы: множество ВЦОД — виртуальных центров обработки данных.



Каждый ВЦОД обладает определенными выделенными ресурсами и используется для создания, управления и взаимодействия объектов, принадлежащих исключительно данному пространству. В основе организации виртуальных ресурсов лежит иерархическая структура. По умолчанию у ВЦОД имеется корневая директория, в которой на основе конфигурации имеются управляющие роли: администратор ВЦОД и администратор безопасности ВЦОД, а также другие роли, соответствующие конфигурации.

3. Требования

3.1. Требования к узлу

> ⚠ **Отказоустойчивость**
>
> Для обеспечения отказоустойчивости Платформы необходимо иметь не менее трех узлов с минимальными параметрами, указанными в таблице ниже

Таблица — Минимальные требования к узлу для установки Sharx Base.

Параметр	**Ограничение, не менее**
Процессор	2 (не менее 8 ядер в каждом). Архитектура x86_64, частота 2 ГГц
Оперативная память	128 Гбайт
Системный жесткий диск	SSD. Объем не менее 256 Гбайт
Жесткий диск	3 SSD. Объем не менее 1 Тбайт каждый
Сетевая карта	2 порта. Не менее 10 Гбит/с каждый

3.2. Требования к сетевой инфраструктуре

3.2.1. Общие требования к сетевой инфраструктуре

Сетевая инфраструктура для соединения узлов в кластер должна отвечать требованиям:

1. Пропускная способность портов на сетевом оборудовании для подключения продуктивных интерфейсов не менее 10 Гбит/с.
2. Количество портов должно быть больше количества узлов кластера, умноженное на два.
3. Наличие не менее трех VLAN. Один из VLAN имеет параметр *native*.
4. Наличие одного выделенного порта на каждом из двух сетевых коммутаторов для каждого узла кластера.

5. Наличие выделенного коммутатора сегмента управления для подключения интерфейсов управления узлов кластера.

3.2.2. Требования и ограничения сегментирования сети

Чтобы развернуть кластер Платформы виртуализации, выполните подготовку сети Клиента в соответствии с настоящей статьей.

Важно

В статье указано минимальное количество служебных и продуктивных сетей

Типовой состав подсетей для установки и функционирования Платформы отображает Таблица ниже.

Таблица – Список выделяемых подсетей.

Название подсети	**VLAN**	**Сеть**	**Шлюз**
Подсеть управления ПВ	X	X.X.X.0/24	X.X.X.1
Подсеть внеполосного управления	Y	X.X.У.0/24	X.X.У.1
Подсеть распределенной системы хранения данных (РСХД)	Z	X.X.Z.0/24	шлюз не требуется
Подсеть транспортая VXLAN	V	X.X.V.0/24	шлюз не требуется
Подсеть продуктивная №1	W	X.X.W.0/24	X.X.W.1
…		…	
Подсеть продуктивная №N	N	X.X.N.0/24	X.X.N.1

Подсеть внеполосного управления используется для подключения и дальнейшего управления выделенными портами (access-port) управления аппаратных платформ серверов.

Выделите VLAN и IP-адреса для подключения IPMI/BMC-интерфейсов серверов, укажите шлюз сети, настройте маршрутизацию. Трафик в подсети нетегированный (native).
Подсеть внеполосного управления работает с MTU 1500 без использования RDMA.

Подсеть управления ПВ, **подсеть РСХД** и **продуктивные подсети** подключаются к магистральным портам (trunk-port) узлов платформы виртуализации.

Подсеть управления ПВ используется для подключения гипервизоров (узлов) и функционирования кластера Платформы виртуализации. Выделите VLAN и IP-адреса для подключения интерфейсов узлов Платформы виртуализации, виртуальный IP-адрес кластера, укажите шлюз сети, настройте маршрутизацию. Трафик тегированный.
Подсеть управления работает с MTU 1500 без использования RDMA.

Подсеть РСХД используется для функционирования распределенного хранилища данных. Выделите VLAN и IP-адрес сети. Немаршрутизируемый VLAN. Трафик тегированный.
Для функционирования распределенной системы хранения данных (РСХД) используется RDMA и MTU 9000.
Не рекомендуется размещать в подсети РСХД узлы, не относящиеся к системе, поскольку наличие в подсети РСХД MAC-адресов, не относящихся к системе, может приводить к помехам и задержкам при обработке данных.

Подсеть транспортная VXLAN используется для передачи трафика между клиентскими подсетями при использовании сетей типа VXLAN.
Данная подсеть немаршрутизируемая, внешнего доступа к ней и от нее не требуется. Размер MTU внутри подсети должен быть не менее 1600 байт.

Продуктивные подсети используются для размещения продуктивных ВМ. Число таких подсетей зависит от дизайна информационных систем, размещаемых на Платформе виртуализации. Выделите VLAN и IP-адреса для подключения интерфейсов ВМ, укажите шлюзы сетей, настройте маршрутизацию. Трафик тегированный.

3.2.3. Требования к сетевой доступности

При использовании у Клиента ACL или политик межсетевого экранирования настройте сетевую доступность. Это обеспечит функционирование и обслуживание кластера персоналом и технической поддержкой производителя, а также размещение и эксплуатацию продуктивных информационных систем на Платформе.

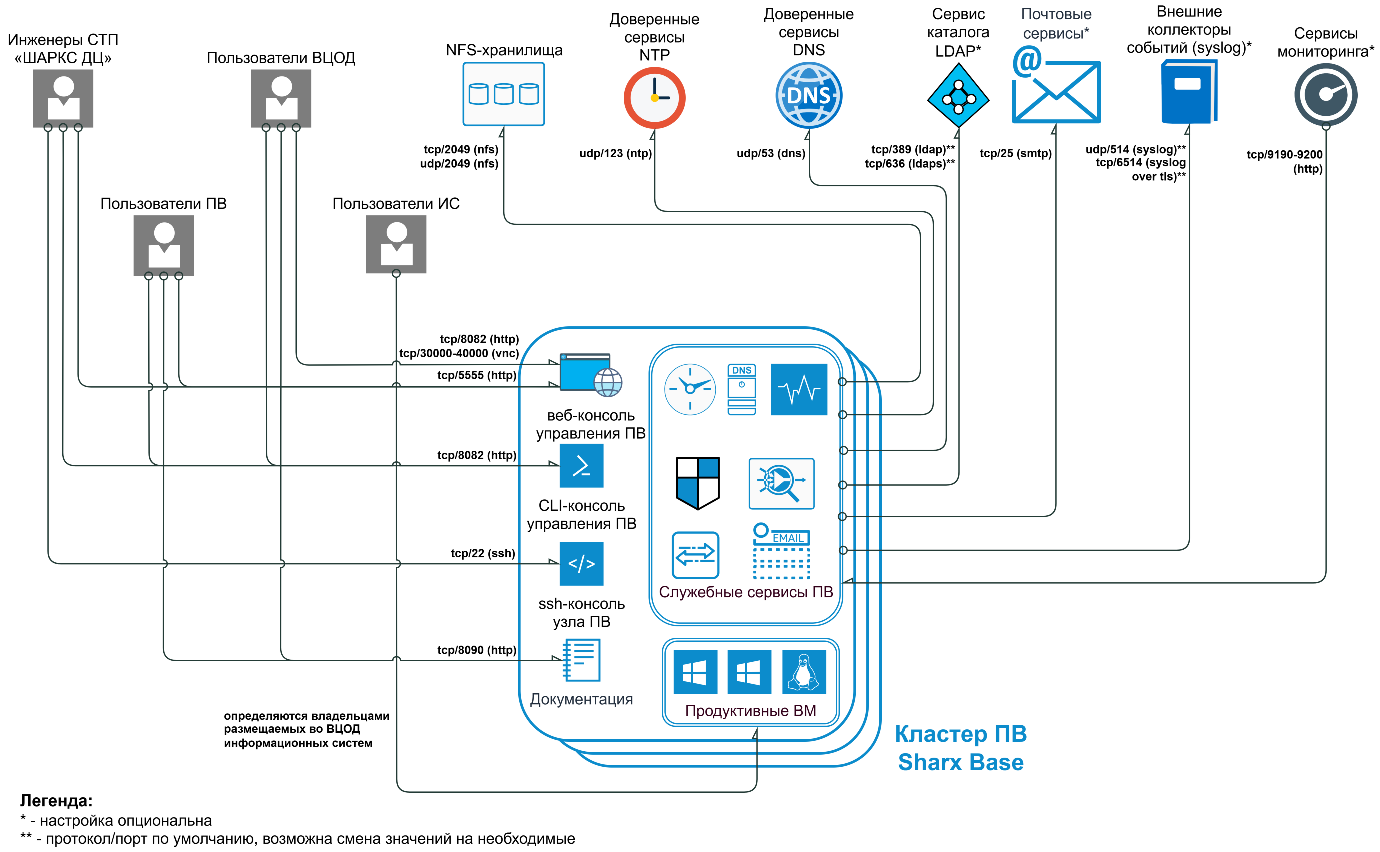


Таблица — Требования к сетевой доступности Платформы.

Описание	Источник	Приемник	Порт и протокол
Доступ персонала клиентов к веб-консоли управления ПВ	АРМ пользователей ПВ	VIP кластера ПВ в подсети управления ПВ	tcp/5555 (HTTP)
Доступ персонала клиентов к веб-консоли управления ВЦОД	АРМ пользователей ВЦОД	VIP кластера ПВ в подсети управления ПВ	tcp/80 (HTTP)
Доступ персонала клиентов к CLI-консоли	АРМ пользователей ПВ/ВЦОД	VIP кластера ПВ в подсети управления ПВ	tcp/8082

Описание	Источник	Приемник	Порт и протокол
управления ПВ/ ВЦОД			
Сеть службы технической поддержки ООО «Шаркс ДЦ»	Доступ службы ТП вендора к консолям управления ПВ	VIP кластера ПВ в подсети управления ПВ	tcp/5555 (HTTP) tcp/ 8082
Доступ службы ТП вендора к SSH-консоли узлов ПВ	Сеть службы технической поддержки ООО «Шаркс ДЦ»	VIP кластера ПВ, IP-адреса узлов ПВ в подсети управления ПВ	tcp/22 (SSH)
Доступ персонала клиентов к консолям управления ВМ	АРМ администраторов ВМ/инженеров DevOps	VIP кластера ПВ в подсети управления ПВ	tcp/30000-40000 (VNC)
Синхронизация точного времени узлов ПВ	IP-адреса узлов ПВ в подсети управления ПВ	IP-адреса или FQDN доверенных NTP-служб	udp/123 (NTP)
Разрешение имен узлами ПВ	IP-адреса узлов ПВ в подсети управления ПВ	IP-адреса доверенных сервисов DNS-клиентов	udp/53 (DNS)
Обращение к внешним хранилищам NFS	IP-адреса узлов ПВ в подсети управления ПВ	IP-адреса внешних хранилищ	tcp/2049 (NFS) udp/ 2049 (NFS)
Интеграция с каталогами LDAP клиентов	VIP кластера ПВ в подсети управления ПВ	IP-адреса сервисов LDAP клиентов	

Описание	Источник	Приемник	Порт и протокол
			tcp/389 (LDAP) tcp/636 (LDAPS)
Отправка почтовых уведомлений персоналу клиентов	VIP кластера ПВ в подсети управления ПВ	IP-адреса SMTP-серверов клиентов	tcp/25 (SMTP)
Отправка событий на внешнее хранилище событий/SIEM	VIP кластера ПВ в подсети управления ПВ	IP-адреса внешних syslog-коллекторов клиентов	udp/514 (SYSLOG) tcp/6514 (SYSLOG OVER TLS)
Доступ персонала клиента ПВ/ВЦОД к документации производителя	АРМ персонала ПВ/ВЦОД, администраторов безопасности, администраторов ВМ/инженеров DevOps	VIP кластера ПВ в подсети управления ПВ	tcp/8090 (HTTP)

Указаны протоколы и порты по умолчанию. При необходимости их можно изменить в настройках в соответствии с конфигурацией внешнего ресурса

Примечание

Сообщите IP-адреса доверенных NTP, DNS и LDAP-сервисов клиента технической поддержке Платформы

3.3. Требования к АРМ и мобильному устройству

Таблица - Минимальные требования к АРМ клиента и мобильному устройству для установки Sharx Base.

Параметр	Ограничение, не менее
АРМ	
Процессор	Архитектура x86_64, частота 2 ГГц
Оперативная память	512 Мбайт
Свободное пространство на жестком диске	500 Мбайт
Операционная система	Linux (.deb-based или .rpm-based дистрибутив)
Мобильное устройство	
Операционная система	Android 12 и выше
Дополнительные приложения	SharxOTP
Дополнительные характеристики	Наличие фотокамеры

Примечание

Дистрибутив мобильного приложения SharxOTP в формате .apk недоступен для скачивания и предоставляется технической поддержкой ООО «Шаркс ДЦ». С инструкцией по установке можно ознакомиться в статье Установить SharxOTP

3.4. Требования к интерфейсу командной строки

sdc-cli — интерфейс командной строки. Реализует пользовательский интерфейс механизма управления **Sharx Base**, позволяющий управлять кластером.

> **Примечание**
>
> Не все функции реализованы в **веб-интерфейсе** Sharx Base. Поэтому `sdc-cli` необходим для работы с кластером на удаленном рабочем месте

Существуют два вида sdc-cli: стандартный и отчуждаемый.

3.4.1. Стандартный sdc-cli

1. На клиентском АРМ проверьте наличие **python** версии 3.11 или выше.
2. Установите пакет sdc-cli — интерфейс командной строки Sharx Base.
3. Установите пакет **sdc-pyenv3** — набор библиотек, необходимых для работы кластера.

3.4.2. Отчуждаемый sdc-cli

Установите пакет sdc-cli — интерфейс командной строки Sharx Base.

Отчуждаемый sdc-cli содержит sdc-pyenv3, поэтому дополнительная установка не требуется.

4. Быстрый старт

Чтобы настроить ВЦОД, выполните следующие действия:

1. Аутентифицируйтесь во ВЦОД.
2. Просмотрите и при необходимости измените структуру ВЦОД.
3. Настройте валидацию объектов ВЦОД.
4. Настройте уведомления.
5. Создайте пользователей или измените их параметры.
6. Включите двухфакторную аутентификацию(опционально){target="_blank"}.
7. Создайте виртуальный кластер.
8. Подготовьте хранилище.
9. Настройте журнал безопасности.
10. Создайте виртуальную машину.

5. Аутентификация

Важно

Для первичной аутентификации необходимо наличие:

- логина и пароля;
- управляющего IP-адреса кластера;
- поддерживаемой ОС;
- наличие пакета sdc-cli на АРМ пользователя.

Действия для аутентификации:

1. Введите в командной строке АРМ пользователя или со стороннего аппаратного средства (в случае удаленного подключения)

   ```
   sdc-cli --host <vip_cluster_addr> [--tls_verify <path_to_cert_key>] [--tls yes]
   ```

   где

   - `host` — управляющий IP-адрес кластера. IP-адрес имеет вид `a.b.c.d`;
   - `tls-verify` — путь к сертификату TLS. Обязателен при включенном TLS-соединении;
   - `tls` — флаг использования TLS. Обязателен при включенном TLS-соединении.

2. Аутентифицируйтесь во ВЦОД

   ```
   login <user_login> --cluster <cluster_name> --ns <namespace_name> [--cluster_to <cluster_na
   me>] [--ns_to <vcod_name>]
   ```

   где

   - `login` — логин пользователя;
   - `cluster` — имя логического кластера;
   - `ns` — имя ВЦОД управления;
   - `cluster_to` — имя логического кластера. Необязательный параметр. При наличии одного логического кластера значения параметров `cluster` и `cluster_to` одинаковы;
   - `ns_to` — имя ВЦОД для подключения из ВЦОД управления. Необязательный параметр.

6. Просмотреть и изменить структуру ВЦОД

По умолчанию у ВЦОД имеется корневая директория, в которой размещены **Администратор ВЦОД** и **Инженер безопасности ВЦОД**. Созданный ВЦОД обладает определенными выделенными ресурсами и используется для создания, управления и взаимодействия объектов, принадлежащих исключительно данному пространству.

Директории во ВЦОД — это средство управления доступом пользователей к объектам, находящимся внутри этих директорий: виртуальных кластеров, ВМ и иных объектов виртуализации.

> **Примечание**
>
> Пользователи, находящиеся в определенной директории, имеют доступ к этой директории и всем директориям, расположенным ниже по иерархии.
> Подробнее о расположении пользователей во ВЦОД и их доступе читайте в статье Пользователи

Далее в пределах ВЦОД создаются поддиректории, изолированные друг от друга. В каждой из них инициализируются пользователи, имеющие доступ только к объектам данной поддиректории. При этом Администратор ВЦОД, находясь в корневой директории ВЦОД, имеет доступ ко всем ресурсам пользователей, находящихся в поддиректориях.

> **Примечание**
>
> Действия выполняются пользователем с ролью **Администратор ВЦОД**

1. Чтобы посмотреть структуру ВЦОД, введите

   ```
   aaa namespace path show
   ```

   > **Внимание**
   >
   > В текущей версии Sharx Base некорректно обрабатывается путь до директории, в имени которой имеется знак `.` Поэтому название директории не должно содержать в себе знак `.`

2. Внести изменения в структуру директорий ВЦОД

```
aaa namespace path update --paths <PATHS>
```

где `paths` — имя корневой директории ВЦОД в формате JSON. По умолчанию имеет значение "/".

Пример `paths`

```
"{\"/\": {\"folder\": null, \"folder2\": null}}"
```

где

- `/` — имя корневой директории ВЦОД;
- `folder`, `folder2` — имя поддиректории ВЦОД.

7. Валидация объектов ВЦОД

Примечание

Действия выполняются пользователем с ролью **Администратор ВЦОД**

Подпись и валидация объектов используются для обеспечения контроля целостности (КЦ) объектов виртуальной инфраструктуры в **Sharx Base**.

Механизм работы основан на применении цифровой подписи.

При создании любого **ВЦОД** Sharx Base автоматически генерирует ключ цифровой подписи ВЦОД, который используется для контроля целостности объектов виртуальной инфраструктуры.

При создании любого **пользователя** — ключи и сертификат пользователя, которые используются для двухфакторной аутентификации пользователей.

7.1. Управление ключами ВЦОД

1. Чтобы посмотреть список ключей в созданном ВЦОД, введите команду

   ```
   signer key list
   ```

2. Посмотреть подробную информацию о конкретном ключе можно с помощью команды

```
signer key show --keyid <KEYID>
```

где `keyid` — идентификатор ключа. Обязательный аргумент.

7.2. Отправка сертификатов пользователям и включение двухфакторной аутентификации

При создании пользователя внутри ВЦОД ему автоматически генерируется сертификат, который в будущем пользователь сможет использовать для двухфакторной аутентификации.

> **Важно**
>
> Перед включением двухфакторной аутентификации **Администратор ВЦОД** обязан выслать сертификаты пользователям на почту

Чтобы включить двухфакторную аутентификацию, Администратору необходимо выполнить следующие шаги:

1. Создать маршрут, в котором указать шаблон сообщения (пример signer_template.yaml) и почтовые адреса пользователей.
   Для корректной отправки сертификатов почтовые адреса в маршруте должны совпадать с почтовыми адресами пользователей, указанными при их создании.
2. Добавить маршрут

   ```
   aaa param update --notif_route <NOTIF_ROUTE>
   ```

   где `notif_route` — имя маршрута отправки уведомлений о событии.
3. Отправить пользователям их сертификаты

   ```
   aaa user cert notify --login <LOGIN>
   ```

   где `login` — логин пользователя. Обязательный аргумент.
4. Включить вход во ВЦОД по сертификату

   ```
   aaa param update --cert yes
   ```

   где `cert` — включение или выключение механизма входа пользователя по сертификату. По умолчанию — `no`. Возможные значения:

- `yes` — механизм входа пользователя по сертификату включен,
- `no` — механизм входа пользователя по сертификату выключен;

> **Примечание**
>
> После включения данного параметра при аутентификации пользователя необходимо будет дополнительно указывать параметр `--cert` — идентификатор ключа сертификата пользователя, который получен пользователем в письме от Администратора.
> Пример
>
> ```
> login vcod_admin --cluster test --ns test --cert <CERT_KEY>
> ```

5. Чтобы обновить сертификат пользователя, Администратор должен ввести следующую команду

```
aaa user update --login <LOGIN> --cert yes
```

где

- `login` — логин пользователя. Обязательный аргумент.
- `cert` — включение или выключение механизма обновления сертификата пользователя. По умолчанию — `no`. Возможные значения:
  - `yes` — механизм обновления сертификата пользователя включен, старый сертификат будет удален,
  - `no` — механизм обновления сертификата пользователя выключен, сертификат пользователя не изменится;

> **Важно**
>
> После обновления сертификата пользователя необходимо:
>
> - выполнить повторную отправку сертификата на почту пользователя;
> - пользователю обновить на локальной машине свои сертификаты

7.3. Настройка клиентской машины для входа во ВЦОД с использованием сертификата

> **Примечание**
>
> Действия выполняет Клиент на своем АРМ

Перед началом работы с сертификатами пользователь должен убедиться, что у него установлены следующие пакеты:

- **libksba8_1.6.3**;
- **gpgsm_2.2.40**;
- **dirmngr_2.2.40**.

Письмо от Администратора, полученное на почтовый адрес, содержит идентификатор ключа сертификата пользователя и 3 файла:

- **ns_cert** — сертификат ВЦОД;
- **user_cert** — сертификат пользователя;
- **secret** — закрытый ключ пользователя.

> **Примечание**
>
> Для аутентификации пользователя во ВЦОД можно использовать идентификатор ключа `<CERT_KEY>`. По данному ключу сверяется сертификат на машине клиента с сертификатом, который есть во ВЦОД. Если они совпадают, то вход разрешается. Если нет, то выдается ошибка

После копирования приведенных выше файлов на свою машину необходимо выполнить следующие действия:

1. Импортировать сертификат ВЦОД

   ```
   gpgsm --import ns_cert
   ```

   Посмотреть список загруженных сертификатов можно, используя команду

   ```
   gpgsm --list-keys
   ```

2. Создать файл `~/.gnupg/trustlist.txt`, в который занести `fingerprint` сертификата ВЦОД. В конце через пробел добавить букву `S`.

> **Пример**
>
> A5:0D:83:06:4A:E8:8E:D1:C9:EB:27:B4:5C:F6:A0:43:6F:8F:0C:59 S

3. Импортировать сертификат пользователя

```
gpgsm --import user_cert
```

4. Импортировать закрытый ключ пользователя

```
gpgsm --import secret
```

После выполнения вышеперечисленных шагов пользователю можно перейти к авторизации во ВЦОД.

7.4. Настройка валидации объектов ВЦОД

Для обеспечения контроля целостности на уровне виртуальной инфраструктуры ВЦОД необходимо настроить и включить валидацию объектов соответствующего ВЦОД.

1. Задайте параметры для настройки валидации ВЦОД

```
signer param add [--sign_enable <SIGN_ENABLE>]
                 [--validation_action <VALIDATION_ACTION>]
                 [--sign_models <SIGN_MODELS>]
                 [--sign_objects <SIGN_OBJECTS>]
                 [--sign_objects_exclude <SIGN_OBJECTS_EXCLUDE>]
```

где

- `sign_enable` — включение или выключение механизма валидации для объектов в рамках ВЦОД при обращении к ним. По умолчанию — `no`. Возможные значения:
  - `yes` — механизм валидации включен,
  - `no` — механизм валидации выключен;
- `validation_action` — тип действия при обнаружении невалидного объекта. По умолчанию — `ERROR`. Возможные значения:
  - `ERROR` — запрет на совершение операции и вывод сообщения об ошибке,

- `WARN` — отправка уведомления с возможностью дальнейшей работы;
- `sign_models` — набор моделей, с которыми будет работать плагин sdc-plgn-signer;
- `sign_objects` — набор объектов, которые существуют в рамках ВЦОД и валидацию которых нужно осуществлять;
- `sign_objects_exclude` — набор объектов, которые нужно исключить из подписываемой модели и валидацию которых осуществлять не нужно.

> **Примечание**
>
> Если ввести команду без дополнительных аргументов — `signer param add`, будут созданы пустые параметры со значениями аргументов по умолчанию

2. Сгенерируйте ключ, которым будет осуществляться подпись объектов внутри ВЦОД. Для этого включите механизм валидации

```
signer param update --sign_enable yes
```

где `sign_enable` — включение или выключение механизма валидации для объектов в рамках ВЦОД при обращении к ним. По умолчанию — `no`. Возможные значения:

- `yes` — механизм валидации включен,
- `no` — механизм валидации выключен.

> **Важно**
>
> В случае необходимости **изменения параметров валидации** вы можете это сделать с помощью команды `signer param update`, указав изменяемые параметры. Эти параметры такие же, как и для команды `signer param add`.
>
> Например, для **выключения валидации**, в том числе уже подписанных объектов, необходимо ввести команду
>
> ```
> signer param update --sign_enable no
> ```
>
> При **повторном включении валидации** будет сгенерирован новый ключ. После этого повторно добавьте объекты в отслеживаемые для обновления их подписи

3. Добавьте модели в список отслеживаемых

   a. Посмотрите список доступных моделей

   ```
   signer sign model list
   ```

b. Добавьте модели в список отслеживаемых

```
signer sign model add --models <MODELS>
                      [--force_sign <FORCE_SIGN>]
```

где

- `models` — список моделей, которые будут отслеживаться. Обязательный аргумент. При `models`, равном `*`, все модели автоматически будут добавлены в отслеживаемые;
- `force_sign` — принудительная валидация уже существующих и создаваемых позже объектов. По умолчанию — `no`. Возможные значения:
  - `yes` — принудительная валидация включена,
  - `no` — принудительная валидация выключена.

> **Важно**
>
> Если объекты в модели существуют до того, как включается валидация, то дополнительно в команде укажите флаг `force_sign` со значением `yes`. Это принудительно подпишет уже существующие объекты.
> Если внутри модели объектов нет, данный флаг указывать не нужно, все новые объекты будут отслеживаться автоматически.
>
> При обновлении ключа подписи (повторном включении механизма валидации) все модели, которые входят в параметры, необходимо повторно добавить с использованием флага `force_sign`

c. Чтобы удалить модели из отслеживаемых, введите команду

```
signer sign model del --models <MODELS>
```

где `models` — список моделей, которые больше не будут отслеживаться. Обязательный аргумент. При `models`, равном `*`, все модели автоматически не будут отслеживаться.

4. Добавьте объекты внутри моделей в список отслеживаемых.
   Плагин позволяет добавлять в список не только модели целиком, но и их отдельные части — объекты.

   a. Посмотрите список всех доступных объектов внутри модели

```
signer sign entity list --model <MODEL>
```

где `model` — имя модели. Обязательный аргумент.

b. Добавьте объекты в список отслеживаемых

```
signer sign entity enable --pk <PK> --force_sign yes
```

где

- `pk` — информация об объекте в формате JSON. Обязательный аргумент;
- `force_sign` — принудительная валидация уже существующих и создаваемых позже объектов. По умолчанию — `no`. Возможные значения:
  - `yes` — принудительная валидация включена,
  - `no` — принудительная валидация выключена.

Пример команды для добавления тома Libvirt в список отслеживаемых

```
signer sign entity enable --pk '{"classpath": "storage_libvirt_vols","pk":
{"ctx_cluster": "sdcci6569","ctx_ns": "vdc1","ctx_path": "/","name": "2.qcow2","pool":
"test","uuid": "7206ed44-3d52-4b30-9224-cee888e8a719"}' --force_sign yes
```

c. Чтобы удалить объект из списка отслеживаемых, введите

```
signer sign entity del --pk <PK>
```

где `pk` — информация об объекте в формате JSON. Обязательный аргумент.

5. Добавьте объекты в список исключений.
Плагин позволяет внести объект в список исключений для подписи.
Это означает, что все объекты модели при ее добавлении будут подписаны, а указанные в исключении объекты подписаны не будут и не будут отслеживаться.

a. Чтобы добавить объект в список исключений, введите команду

```
signer sign entity disable --pk <PK>
```

где `pk` — информация об объекте в формате JSON. Обязательный аргумент.

b. Чтобы удалить объект из списка исключений, введите команду

```
signer sign entity del --pk <PK>
```

где `pk` — информация об объекте в формате JSON. Обязательный аргумент.

6. Информирование о нарушении валидации.
   При нарушении валидации отслеживаемого объекта автоматически в журнал безопасности будет занесено соответствующее событие.

   > **Пример**
   >
   > ```
   >     {
   >     "ctx_cluster": "sdcci6569",
   >     "ctx_ns": "vdc1",
   >     "data": "{\"pk\": {\"classpath\": \"scheduler_request\", \"pk\": {\"ctx_cluster\":
   > \"sdcci6569\", \"ctx_ns\": \"vdc1\", \"ctx_path\": \"/\", \"vcluster\": \"vcl1\", \"kind\": \
   > "vm\", \"name\": \"test1\", \"uuid\": \"db8ef9f0-035c-4162-8aba-79026618a0fe\"}}}",
   >     "error": "Object was illegally modified. Signature is not valid",
   >     "event_time": "2024-09-09T08:52:04.050000+00:00",
   >     "group": "default",
   >     "level": "INFO",
   >     "login": null,
   >     "operation": "Object was illegally modified. Signature is not valid",
   >     "prop": null,
   >     "result": null,
   >     "session": null,
   >     "uuid": "9063f5ff-1194-4f26-8f40-3f222d70e48a"
   > },
   > ```

   Повторная автоматическая проверка произойдет через 24 часа после предыдущего обнаружения ошибки.
   Каждое обращение пользователя к объекту будет приводить к дополнительной записи в журнал безопасности с информацией о том, что было обращение к объекту, у которого нарушена валидация.

7. Чтобы удалить параметры валидации ВЦОД, введите команду

   ```
   signer param del
   ```

   > **Важно**
   >
   > При удалении параметров, также как и при отключении валидации, ключ подписи удаляется.
   > При повторном создании параметров будет сгенерирован новый ключ

7.5. Оповещение пользователя о событиях КЦ

Платформа позволяет настроить оповещение Администратора ВЦОД о регистрации заданных типов событий, в том числе о событиях нарушения контроля целостности. Оповещение осуществляется по электронной почте, подробнее настройка оповещений описана в разделе Уведомления Руководства пользователя Sharx Base.

7.6. Рекомендации по действиям Администратора ВЦОД при нарушении контроля целостности

При получении сообщения о нарушении целостности объекта виртуальной инфраструктуры Администратору рекомендуется:

1. Перевести контроль целостности на уровне виртуальной инфраструктуры в уведомительный режим (WARN).
2. Удалить объект из перечня отслеживаемых.
3. Провести анализ причин срабатывания контроля целостности на нарушение целостности объекта.
4. В случае необходимости — пересоздать объект. Рекомендуется восстановить объект из резервной копии, если имеется актуальная резервная копия объекта.

> **Важно**
>
> Восстановить виртуальную машину из резервной копии пользователь Sharx Base с соответствующими правами может по инструкции в разделе Восстановить виртуальную машину Руководства пользователя.
> Другие объекты можно восстановить из резервных копий, обратившись в Техническую поддержку ООО «Шаркс ДЦ»

8. Уведомления

> **Примечание**
>
> Действия выполняются пользователем с ролью **Администратор ВЦОД**

Уведомления в **Sharx Base** используются для информирования пользователей о важных событиях в системе: отправки сертификатов при включенной двухфакторной аутентификации, настройке журнала безопасности, наличии ошибок при выполнении процессов и т. д.

Внимание

Предварительно настройте сторонний SMTP-сервер, по которому будет осуществляться подключение

Чтобы настроить уведомления, необходимо сформировать **маршрут отправки**.
Создание маршрута состоит из следующих шагов:

1. Создать **Почтовый сервер**.
   Возможно существование нескольких типов почтовых серверов. Самый распространенный — SMTP-сервер для отправки почты.
2. Сформировать **Адреса** для отправки сообщения. Возможно задать набор адресов для получения сообщения.
3. Задать **Шаблоны**. Данные для шаблона передаются как часть сообщения, которое приходит в маршрутизатор.
4. Создать **Маршрут отправки**.

8.1. Создать почтовый сервер

1. Чтобы добавить почтовый сервер, введите в командной строке

```
notif endpoint add --descr <endpoint_description> --name <endpoint_name> --passwd <service_server_passwd> --tls <yes_or_no> --type SMTP --url <server_url> --user <service_server_user>
```

2. Запрос информации о всех почтовых серверах

```
notif endpoint list
```

3. Запрос информации об определенном почтовом сервере

```
notif endpoint show --name <endpoint_name>
```

8.2. Сформировать список адресов

1. Чтобы добавить адреса, введите в командной строке

```
notif mailing add --addresses <list_of_addresses> --descr <description> --name <mailing_name>
```

где `addresses` — адреса, существующие на SMTP-сервере и на кластере **Sharx Base**.

2. Запрос информации о всех адресах

```
notif mailing list
```

3. Запрос информации об определенном адресе

```
notif mailing show --name <mailing_name>
```

8.3. Задать шаблоны сообщений

> **Примечание**
>
> Добавляйте шаблон с помощью файла YAML, так как часто возникают ошибки при внесении данных.
> Файл с примером шаблона notif.yaml

1. Чтобы добавить шаблон, введите в командной строке

```
notif template add --data <template_data> --descr <template_description> --name <template_name>
```

2. Запрос информации о всех шаблонах сообщений

```
notif template list
```

3. Запрос информации об определенном шаблоне

```
notif template show --name <template_name>
```

8.4. Настроить маршрут отправки уведомлений

Маршрут объединяет в себе ранее созданные сущности `endpoint`, `mailing`, `template` и является точкой входа и отправки сообщения по конкретному пути.

1. Чтобы настроить маршрут, введите в командной строке

```
notif route add --name <route_name> --descr <route_description> --endpoint <endpoint_name>
--mailing <mailing_name> --sender <message_sender_name> --subject <message_subject> --
template <template_name>
```

2. Чтобы запросить информацию о всех маршрутах, введите

```
notif route list
```

3. Запросить информацию об определенном маршруте

```
notif route show --name <route_name>
```

8.5. Настроить уведомления с помощью файла YAML

1. Создайте файл формата YAML.

   В файле опишите все необходимые параметры и команды для настройки отправки уведомлений. Пример файла автоматической настройки уведомлений notif.yaml

   > **Примечание**
   >
   > Пример файла содержит шаблон для уведомления о невалидном объекте (неправомерном изменении объекта)

2. Сохраните файл настройки уведомлений.
3. Загрузите файл на Платформу.

   Чтобы загрузить файлы в Sharx Base, введите команду

```
resource --spec <path_to_yaml_file>
```

   где `spec` — расположение YAML-файла.

   При локальном расположении на АРМ пользователя введите путь до файла.

   При расположении в стороннем репозитории укажите ссылку на данный файл.

4. Проверьте загрузку файла

```
notif <arg_notif> list
```

где `arg_notif` — аргумент для проверки конкретного параметра настройки уведомлений.

Возможные значения:

- `endpoint` — конечный сервис или сервер для отправки сообщений;
- `mailing` — список адресантов или рассылки для отправки сообщения;
- `template` — шаблон сообщения;
- `route` — маршрут отправки уведомлений.

Проверяемые параметры уведомлений должны отобразиться в общем списке.

8.6. Контроль отправленных сообщений

1. Чтобы проверить перечень отправленных сообщений, введите команду

```
notif message list
```

2. Отобразить подробную информацию об отправленном сообщении

```
notif message show --uuid <message_uuid>
```

где `uuid` — идентификатор сообщения.

9. Права доступа и роли

9.1. Права доступа

Права доступа — набор действий, разрешенных для выполнения субъектам над объектами данных. Права доступа определяют правила, которые затем объединяются в роли. Создание, изменение и удаление прав доступа невозможно. Список прав доступа единый для всей системы и может меняться только при ее обновлении.

Существуют следующие виды прав доступа:

- **Командные** — определяют право доступа пользователя к определенной команде. Например, `cluster_list`, `aaa_user_show`.

- **Функциональные** — определяют право доступа к определенным действиям в команде. Например, `login_namespace` — позволяет выполнять авторизацию через атрибуты `cluster_to`, `ns_to`.

Права доступа имеют следующие параметры:

- `name` — имя права доступа;
- `uuid` — уникальный идентификатор в системе;
- `privileged` — флаг привилегированности. `1` — привилегированное, `0` — обычное;
- `descr` — описание права доступа.

Команды управления правами доступа:

1. Просмотр всех прав доступа во ВЦОД

```
aaa permissions list
```

2. Просмотр подробной информации о конкретном праве доступа.

```
aaa permissions show --name <NAME>
```

где `name` — имя права доступа.

9.2. Роли

Роль — набор прав доступа, которые необходимы пользователю или группе пользователей для выполнения определенных задач. Права доступа объединяются в роли. Затем роли присваиваются пользователю или группе пользователей. Каждый пользователь может быть связан от 1 до N ролей.

Разные роли создаются в системе на определенных уровнях объектов виртуализации. Роли и пользователей по умолчанию удалить нельзя. Новые роли и пользователи могут быть созданы администраторами в пределах их ВЦОД. Права доступа выбираются из стандартного набора без права изменения и удаления.

Роли уровня ВЦОД автоматически появляются в каждом новом создаваемом ВЦОД из стандартных конфигураций:

- администратор ВЦОД.
- администратор безопасности ВЦОД.
- администратор ВМ.

- разработчик ВМ.

Ниже рассмотрены подробные права каждой роли, созданной по умолчанию.

9.2.1. Роли по умолчанию

АДМИНИСТРАТОР ВЦОД

Функции **Администратора ВЦОД** ограничены рамками ВЦОД и включают следующее:

- настройка структуры каталогов ВЦОД;
- настройка параметров безопасности ВЦОД и интеграции ВЦОД с каталогом LDAP;
- просмотр прав доступа, создание, удаление и настройка ролей ВЦОД;
- управление сессиями доступа ВЦОД;
- создание, удаление, настройка параметров учетных записей пользователей ВЦОД;
- просмотр журналов и конфигурационных параметров журналирования событий ВЦОД;
- просмотр выделенных для ВЦОД ресурсов виртуальных сетей, настройка параметров и операции с сетевым стеком, виртуальными сетями и ACL ВЦОД;
- конфигурационные настройки и операции с сетевым стеком и ACL ВМ ВЦОД;
- настройка параметров механизма уведомлений и рассылок ВЦОД;
- настройка параметров и выполнение операций с виртуальными кластерами, ресурсными ограничениями виртуальных кластеров, пользовательскими ресурсами в виртуальных кластерах ВЦОД;
- создание, удаление, настройка параметров и выполнение операций по управлению состоянием ВМ (включение, выключение, перезагрузка);
- выполнение операций со снимками ВМ ВЦОД;
- выполнение операций с узлами виртуальных кластеров во ВЦОД;
- выполнение операций с метками виртуальных кластеров во ВЦОД;
- настройка параметров и выполнение операций с ключами и сертификатами ВЦОД;
- настройка параметров и выполнение операций с СХД (пулы хранения, тома для ВМ, снимки ВМ, копии дисков);
- выполнение операций мониторинга ВЦОД;
- доступ к VNC-консоли ВМ.

АДМИНИСТРАТОР БЕЗОПАСНОСТИ ВЦОД

Функции **Администратора безопасности ВЦОД**:

- просмотр всех конфигурационных параметров ВЦОД;

- настройка параметров и выполнение операций контроля целостности ВЦОД;
- настройка параметров журналирования событий ВЦОД и просмотр событий ВЦОД;
- выполнение операций мониторинга во ВЦОД.

АДМИНИСТРАТОР ВМ

Функции **Администратора ВМ**:

- отображение списка пользовательских ресурсов виртуальных кластеров ВЦОД;
- выполнение операций по управлению состоянием ВМ (включение, выключение, перезагрузка);
- доступ к VNC-консоли ВМ.

РАЗРАБОТЧИК ВМ

Функции **Разработчика ВМ**:

- просмотр журналов собственных событий ВЦОД;
- просмотр сведений и параметров виртуальных сетей ВЦОД;
- настройка и выполнение операций с сетевым стеком ВМ ВЦОД;
- просмотр сведений о пользовательских ресурсах и ресурсных ограничениях виртуальных кластеров ВЦОД;
- создание, удаление, настройка параметров и выполнение операций по управлению состоянием ВМ (включение, выключение, перезагрузка);
- выполнение операций по управлению снимками ВМ;
- просмотр сведений о виртуальных кластеров ВЦОД, метках виртуальных кластеров ВЦОД;
- просмотр сведений о пулах хранения СХД, томах для ВМ, снимках ВМ, шаблонах дисков ВМ;
- выполнение операций по созданию и удалению резервной копии ВМ;
- доступ к VNC-консоли ВМ.

9.2.2. Действия с ролями

Администратор ВЦОД может управлять ролями в рамках своего ВЦОД.

Роль имеет следующие параметры:

- `name` — имя роли;
- `uuid` — уникальный идентификатор роли;
- `builtin` — флаг встроенной (`true`) или пользовательской (`false`) роли;

- `privileged` — флаг привилегированности роли. Зависит от наличия в роли привилегированных прав доступа. Равен `1`, если роль содержит привилегированные права, и равен `0`, если роль не содержит привилегированные права;
- `permissions` — список прав доступа, входящих в роль;
- `descr` — описание роли.

Команды для управления ролями:

1. Чтобы добавить роли (создать пользовательскую роль), в командной строке введите

   ```
   aaa role add --role <ROLE>
                --permissions <PERMISSIONS>
                --descr <DESCR>
   ```

   где

   - `role` — имя роли;
   - `permissions` — имя или список имен прав доступа без разделительных знаков;
   - `descr` — описание роли.

   Параметры `builtin` и `privileged` проставляются автоматически. `Privileged` зависит от наличия привилегированных прав доступа в списке.

2. Просмотреть список всех ролей в пределах ВЦОД

   ```
   aaa role list
   ```

3. Просмотреть информацию о роли

   ```
   aaa role show --role <ROLE>
   ```

4. Изменить пользовательскую роль, добавить или удалить право доступа

   ```
   aaa role update --role <ROLE>
                   --permissions <PERMISSIONS>
   ```

   > **Примечание**
   >
   > В параметре `permissions` перечислите список прав доступа, исключив удаляемое право. Или выведите список действующих прав доступа, затем добавьте к нему новые права

5. Удалить пользовательскую роль.

   Перед удалением роли проверьте, что она не присвоена ни одному из пользователей.

После проверки для удаления роли введите команду

```
aaa role del --role <ROLE>
```

> **Примечание**
>
> Если значение параметра `role` равно `*`, будут удалены все пользовательские роли в пределах ВЦОД.
> Если роль привязана к пользователю, то она не будет удалена.
> Роли по умолчанию удалить нельзя

10. Пользователи

10.1. Создать пользователя

> **Примечание**
>
> В статье описано ручное создание пользователя.
>
> Также Sharx Base поддерживает создание пользователей с использованием LDAP

1. Чтобы создать пользователя в командной строке **Sharx Base**, введите

```
aaa user add --login <LOGIN>
             [--passwd <PASSWD>]
             [--path <PATH> ]
             [--email <EMAIL> ]
             [--roles <ROLES>]
             [--whitelist_networks <WHITELIST_NETWORKS>]
```

где

- `login` — логин пользователя;
- `passwd` — пароль пользователя. Если пароль не был указан при создании, то он сформируется автоматически по следующим правилам: длина пароля — 8 символов, обязательно должны присутствовать цифры, заглавные и строчные буквы латиницы и специальные символы `!@#$%^&*`;
- `path` — расположение пользователя в пределах ВЦОД. Администратор ВЦОД находится в корне `/`. Пользователи в вышестоящей директории при наличии необходимых ролей могут просматривать и изменять параметры пользователей, находящихся уровнями

ниже. Пользователи, находящиеся на одном уровне, но в разных поддиректориях, не могут видеть друг друга.

- `email` — электронная почта пользователя;
- `roles` — имя или список имен ролей;
- `whitelist_networks` — список белых адресов, с которых можно подключаться пользователю.

Пример `path`

```
folder1: --path /folder1
```

После создания пользователя у него появляется служебный атрибут `passwd_is_generated`, доступный только для чтения. Атрибут показывает происхождение пароля пользователя:

- `true` — пароль установлен администратором;
- `false` — пароль установлен самим пользователем.

Этот флаг в связке с глобальной политикой ВЦОД `require_generated_password_change` определяет, может ли пользователь с административно заданным паролем работать в системе до его обязательной смены.

2. Чтобы посмотреть список пользователей в рамках ВЦОД, введите

```
aaa user list
```

3. Просмотр данных о пользователе

```
aaa user show --login <LOGIN>
```

10.2. Управлять пользователями

1. **Администратор** может изменить список ролей, пароль, почту и расположение пользователя. Для этого введите логин пользователя и изменяемые параметры

```
aaa user update --login <LOGIN>
               [--passwd <PASSWD>]
               [--prev_passwd <PREV_PASSWD>]
               [--path <PATH>]
```

```
                 [--email <EMAIL> ]
                 [--roles <ROLES>]
```

2. **Пользователь** может изменить только свой пароль или почту

```
aaa user param update --email <EMAIL>
                     [--passwd <PASSWD>]
                     [--prev_passwd <PREV_PASSWD>]
```

3. Изменить статус пользователя

```
aaa user status update --login <LOGIN>
                       --status <STATUS>
```

где `status` — статус пользователя. Возможные значения `ACTIVE` — активный, `BLOCKED` — заблокирован.

4. Удалить собственный OTP-токен

```
aaa user otp del
```

5. Удалить токен пользователя в пределах ВЦОД

```
aaa user otp user del --login <LOGIN>
```

6. Изменить расположение пользователя, параметр `path`

```
aaa user path update --login <LOGIN>
                     --path <PATH>
```

7. Удалить пользователя

```
aaa user del --login <LOGIN>
```

После данного действия статус пользователя изменится на *REMOVED*.

Удалите пользователя из перечня вводом команды

```
aaa user clear --login <LOGIN>
```

> ❌ **Внимание**
>
> Невозможно изменить или удалить системных пользователей

11. Загрузить пользователей с использованием LDAP

Примечание

Действия выполняются пользователем с ролью **Администратор ВЦОД**

Sharx Base позволяет аутентифицировать пользователей с помощью внешнего LDAP-каталога, автоматически создавая для них учетные записи во ВЦОД и назначая роли.

Платформа совместима с тремя типами LDAP:

1. **OpenLDAP**.
2. **FreeIPA**.
3. **ActiveDirectory**.

11.1. Общий принцип работы

1. Добавьте во ВЦОД конфигурацию подключения к LDAP-каталогу.
2. Активируйте конфигурацию и включите синхронизацию с ней.
3. Проверьте активацию LDAP-каталога.

Внимание

Одновременно во ВЦОД может быть активирована связь только с одним LDAP.

Если были синхронизированы несколько LDAP, то активными во ВЦОД будут только те пользователи, чья конфигурация LDAP синхронизируется в данный момент.

Если синхронизация выключена, то пользователь из LDAP не сможет аутентифицироваться во ВЦОД

11.2. Управлять конфигурациями LDAP

11.2.1. Добавить конфигурацию

Конфигурацию можно добавить с помощью команды в CLI или путем загрузки YAML-файла.

ДОБАВИТЬ КОНФИГУРАЦИЮ КОМАНДОЙ В CLI

Чтобы добавить конфигурацию, введите команду

```
aaa param ldap add --name <NAME>
                  [--url <URL>]
                  [--ca_data <CA_DATA>]
                  [--ca_file <CA_FILE>]
                  [--user <USER>]
                  [--passwd <PASSWD>]
                  [--base_dn <BASE_DN>]
                  [--query_group <QUERY_GROUP>]
                  [--query_user <QUERY_USER>]
                  [--query_active_users <QUERY_ACTIVE_USERS>]
                  [--user_map <USER_MAP>]
                  [--group_map <GROUP_MAP>]
                  [--default_path <DEFAULT_PATH>]
                  [--notif_route <NOTIF_ROUTE>]
                  [--error_limit <ERROR_LIMIT>]
```

где

- `name` — имя конфигурации LDAP в рамках текущего ВЦОД. Обязательный параметр;
- `url` — адрес сервера LDAP, к которому осуществляется подключение. Укажите IP-адрес или доменное имя сервера;
- `ca_data` — данные сертификата. Не может быть использован одновременно с `ca_file`;
- `ca_file` — путь к файлу сертификата. Не может быть использован одновременно с `ca_data`;
- `user` — пользователь на сервере LDAP, от имени которого будет осуществляться подключение;
- `passwd` — пароль пользователя на сервере LDAP, от имени которого будет осуществляться подключение;
- `base_dn` — корневой узел дерева LDAP, относительно которого система будет искать объекты;
- `query_group` — условия поиска, по которым система ищет групповые объекты в каталоге LDAP;
- `query_user` — определяет критерии поиска пользователей в каталоге LDAP;
- `query_active_users` — проверка статуса пользователя в каталоге LDAP;
- `user_map` — присвоение атрибутам пользователя ВЦОД характеристик из LDAP. Например, для `login` LDAP устанавливается соответствие `uid`, для `email` LDAP устанавливается соответствие `email`, для `status` LDAP устанавливается соответствие `accountStatus`. Указывается в формате JSON;
- `group_map` — присвоение группе в каталоге LDAP роли пользователя ВЦОД. Указывается в формате JSON;
- `default_path` — путь к директории в рамках ВЦОД, куда будут размещены пользователи из LDAP;

- `notif_route` — имя заранее созданного маршрута для уведомлений об ошибках;
- `error_limit` — количество повторений одинаковой ошибки. При превышении указанного количества Sharx Base отправляет администратору сообщение о неполадках с LDAP. По умолчанию равно `3`.

Пример

```
aaa param ldap add --name <config_name>
                   --url "ldap:<ldap_addr>"
                   --user "cn=<admin_login>,dc=<dc>,dc=<dc>"
                   --passwd <ldap_admin_user_passwd>
                   --base_dn "dc=<dc>,dc=<dc>"
                   --query_group "(objectClass=<ldap_query_group>)"
                   --query_user "(objectClass=<ldap_query_user>)"
                   --query_active_users "<ldap_query_active_users>"
                   --user_map "{\"login\": \"uid\", \"email\": \"email\", \"status\":
\"accountStatus\"}
                   --group_map "{\"<ldap_group>\": \"<sdc_core_role>\"}"
                   --default_path "<path_on_sdc_core_cluster>"
                   --notif_route <route_for_notification>
                   --error_limit <count_error>
```

ДОБАВИТЬ КОНФИГУРАЦИЮ YAML-ФАЙЛОМ

1. Создайте файл конфигурации в формате YAML. Общий шаблон конфигурации приведен ниже

```yaml
apiVersion: v1
kind: api
metadata:
  plugin:
    aaa:
      param:
        ldap: add
spec:
  name: "<config_name>"
  url: "ldap:<ldap_addr>"
  user: "cn=<admin_login>,dc=<dc>,dc=<dc>"
  passwd: "<passwd_to_LDAP>"
  base_dn: "dc=<dc>,dc=<dc>"
  query_group: "(objectClass=<ldap_query_group>)"
  query_user: "(objectClass=<ldap_query_user>)"
  query_active_users: "<ldap_query_active_users>"
  user_map: {"login": "uid", "email": "email", "status": "accountStatus"}
  group_map: {"<ldap_group>": "<sdc_core_role>"}
  default_path: "/"
  notif_route: "<route_name>"
  error_limit: "<count_error>"
```

Описание полей приведено выше.

2. Загрузите файл конфигурации на Платформу.

   Чтобы загрузить файлы в Sharx Base, введите команду

   ```
   resource --spec <path_to_yaml_file>
   ```

   где `spec` — расположение YAML-файла.

   При локальном расположении на АРМ пользователя введите путь до файла.

   При расположении в стороннем репозитории укажите ссылку на данный файл.

После загрузки рекомендуется проверить загрузку файла конфигурации и статус соединения согласно инструкции ниже.

11.2.2. Просмотреть и проверить статус конфигурации LDAP-каталога

1. Просмотреть список всех конфигураций

   ```
   aaa param ldap list
   ```

   Обратите внимание, что указанное имя файла конфигурации, добавленное выше, должно отобразиться в общем списке LDAP-конфигураций ВЦОД.

2. Посмотрите информацию о подгруженной конфигурации

   ```
   aaa param ldap show --name <NAME>
   ```

   где `name` — имя конфигурации LDAP в рамках текущего ВЦОД.

   В информации отобразятся два дополнительных параметра:

   - `status` — статус соединения. Возможные значения:

     CONNECTED — соединение установлено,
     UNCONNECTED — соединение отключено,
     ERROR — ошибка соединения;

   - `last_send_ldap_error_text` — запись о последней ошибке.

11.2.3. Обновить конфигурацию

Чтобы изменить параметры существующей конфигурации, введите

```
aaa param ldap update --name <NAME>
                      [--url <URL>]
```

```
                        [--user <USER>]
                        [--passwd <PASSWD>]
                        [--base_dn <BASE_DN>]
                        [--query_group <QUERY_GROUP>]
                        [--query_user <QUERY_USER>]
                        [--query_active_users <QUERY_ACTIVE_USERS>]
                        [--user_map <USER_MAP>]
                        [--group_map <GROUP_MAP>]
                        [--default_path <DEFAULT_PATH>]
                        [--notif_route <NOTIF_ROUTE>]
                        [--error_limit <ERROR_LIMIT>]
```

Описание полей приведено выше.

11.2.4. Удалить конфигурацию

Чтобы удалить конфигурацию LDAP, введите

```
aaa param ldap del --name <NAME>
```

где `name` — имя конфигурации LDAP в рамках текущего ВЦОД.

11.3. Активировать синхронизацию с LDAP

Чтобы активировать подключение с определенным LDAP-каталогом, выполните следующие действия:

1. Укажите, какая конфигурация будет использоваться

   ```
   aaa param update --ldap <LDAP>
   ```

   где `ldap` — имя LDAP-каталога.

2. Включите синхронизацию с LDAP

   ```
   aaa param update --ldap_sync yes
   ```

   где `ldap_sync` — синхронизация с LDAP. Возможные значения:

   - `yes` — синхронизация включена;
   - `no` — синхронизация выключена.

11.4. Проверить активацию LDAP

После синхронизации пользователи из LDAP должны появиться в списке пользователей ВЦОД с закрепленными ролями согласно конфигурации.

Для проверки просмотрите пользователей ВЦОД

```
aaa user list
```

Если пользователи отображаются в списке, настройка проведена корректно.
Роли пользователям назначаются через параметр `group_map` при настройке конфигурации.
Если `group_map` не был указан, пользователи будут созданы во ВЦОД без назначенных ролей.

Пользователи могут аутентифицироваться во ВЦОД, используя свой логин и пароль из LDAP. Дополнительных настроек не требуется.

> **Примечание**
>
> Если синхронизация **выключена**, пользователь из LDAP не сможет аутентифицироваться во ВЦОД даже при наличии его учетной записи в списке

12. Политика безопасности учетных записей

ВЦОД по умолчанию обладает базовыми параметрами механизмов безопасности платформы.
Настройки безопасности разделены между обычными и привилегированными пользователями.

> **Примечание**
>
> Параметры политики безопасности учетных записей устанавливаются в рамках текущего ВЦОД

13. Команды управления базовыми параметрами механизмов безопасности

1. Чтобы посмотреть текущие параметры политики безопасности учетных записей ВЦОД, введите в командной строке

   ```
   aaa param show
   ```

Описание параметров приведено в разделах Параметры безопасности ВЦОД для обычного пользователя и Параметры безопасности ВЦОД для привилегированного пользователя.

2. Чтобы изменить параметры, введите

```
aaa param update --<param_name1> <value> --<param_name2> <value>
```

14. Параметры безопасности ВЦОД для обычного пользователя

Таблица — Описание параметров ВЦОД для обычного пользователя.

Параметр ВЦОД	Описание
`acc_block_try_cnt`	Количество попыток ввода некорректного логина и пароля. По умолчанию — *4*. Допустимые значения — целочисленные. Диапазон от 0 до 30, где 0 — отключение проверки. Установка параметра = 0 отключает проверку счетчика попыток аутентификации, а также отключает проверки параметров `acc_block_try_suspend_sec` и `acc_block_try_timeout_sec`
`acc_block_try_suspend_sec`	Время блокировки аккаунта при превышении лимита неверных попыток ввода логина и пароля. По умолчанию — *3600* (1 ч). Допустимые значения — целочисленные. Диапазон 0, далее от 60 с (1 мин) до 86 400 с (1 сутки)
`acc_block_try_timeout_sec`	Интервал времени, в течение которого учитываются неудачные попытки входа для блокировки аккаунта. По умолчанию — *300* (5 мин)*. Допустимые значения — целочисленные
`acc_block_unused_days`	Количество неактивных дней после которого пользователь будет заблокирован.

Параметр ВЦОД	Описание
	По умолчанию — *45 дней*. Допустимые значения — целочисленные. Диапазон 0, далее от 10 до 365, где 0 – отключение проверки
`acc_delete_days`	Количество дней после удаления пользователя, по истечении которых вся информация о нем будет окончательно удалена из системы. По умолчанию — *1095 дней*
`auth_type`	Тип аутентификации. Возможные значения: • *BASIC* — логин и пароль (значение по умолчанию), • *TFA* — двухфакторная аутентификация
`cert`	Использование сертификата при аутентификации. Возможные значения: • *no* — не использовать (значение по умолчанию), • *yes* — использовать
`ldap`	Имя конфигурации **LDAP**, подключаемого к Sharx Base
`ldap_sync`	Включение синхронизации с **LDAP**-сервером. Возможные значения: • *no* — синхронизация выключена (значение по умолчанию), • *yes* — синхронизация включена
`nesting_path_limit`	Лимит вложенности ВЦОД. По умолчанию — *3*
`notif_route`	Установленный маршрут для уведомлений. По умолчанию — `null`. См. Маршрут отправки уведомлений
`ns_owner_access`	Разрешение входа во ВЦОД Администратору кластера. Возможные значения:

Параметр ВЦОД	**Описание**
	• *yes* — разрешен (значение по умолчанию), • *no* — не разрешен
`otp_code_live_period_years`	Срок действия OTP-ключа, выраженный в годах. Возможные значения — любое положительное целое число. По умолчанию — *3 года*
`password_diff_cnt`	Минимальное количество различных (неповторяющихся) символов, которое должно содержаться в пароле пользователя. По умолчанию — *4*. Допустимые значения — целочисленные. Диапазон от 0 до 20, где 0 — отключение проверки. При установке значения 0 также отключается проверка `password_min_exp_days`
`password_exp_days`	Срок действия пароля. По умолчанию — *60 дней*. 0 — отключение проверки
`password_min_change`	Минимальное количество символов, которое надо изменить при замене пароля. По умолчанию — *4*
`password_min_exp_days`	Минимальный срок, который должен пройти после установки пароля, прежде чем его можно будет сменить. По умолчанию — *1 день*
`password_pattern`	Сложность пароля Минимум 8 символов, включающих прописные буквы, строчные буквы, цифры, специальные знаки
`require_generated_password_change`	Определяет, требуется ли пользователю сменить автоматически сгенерированный пароль при первом входе во ВЦОД. Возможные значения: • *no* — не требуется (значение по умолчанию), • *yes* — требуется сменить пароль

Параметр ВЦОД	**Описание**
`sessions_max_cnt`	Максимальное количество сессий пользователя. По умолчанию — *2*. Допустимые значения — целочисленные. Диапазон от 0 до 10, где 0 — отключение проверки
`sessions_timeout_sec`	Время отключения сессии при бездействии. По умолчанию — *180* с (3 мин). Допустимые значения — целочисленные. Диапазон 0, далее от 180 с (3 мин) до 43 200 с (12 ч), где 0 — отключение проверки
`sessions_multi_origin`	Определяет, разрешено ли создание нескольких параллельных сессий для одного пользователя с разных IP-адресов. Возможные значения: • *yes* — разрешено (значение по умолчанию), • *no* — запрещено. **Примечание**: При смене значения с *yes* на *no* уже существующие сессии с разными IP-адресами не будут автоматически разорваны
`tfa_client`	Клиент для двухфакторной аутентификации. По умолчанию — *OTP*
`tfa_wait_sec`	Таймаут для двухфакторной аутентификации По умолчанию — *60* с
`validation_ip`	Проверка IP-адреса пользователя. Возможные значения: • *no* — выключена (значение по умолчанию), • *yes* — включена
`whitelist_networks`	Список белых адресов, с которых можно подключаться к Sharx Base

15. Параметры ВЦОД для привилегированного пользователя

Чтобы разделить настройки безопасности между обычными и привилегированными пользователями, в параметры ВЦОД входят аналогичные поля с постфиксом `_privileged`.

Таблица — Описание параметров ВЦОД для привилегированного пользователя.

Параметр ВЦОД	Описание
`acc_block_try_cnt_privileged`	Количество попыток ввода некорректного логина и пароля. По умолчанию — *4*. Допустимые значения — целочисленные. Диапазон от 0 до 30, где 0 — отключение проверки. Установка параметра = 0 отключает проверку счетчика попыток аутентификации, а также отключает проверки параметров `acc_block_try_suspend_sec_privileged` и `acc_block_try_timeout_sec_privileged`
`acc_block_try_suspend_sec_privileged`	Время блокировки аккаунта при превышении лимита неверных попыток ввода логина и пароля. По умолчанию — *3600* (1 ч). Допустимые значения — целочисленные. Диапазон 0, далее от 60 с (1 мин) до 86 400 с (1 сутки)
`acc_block_try_timeout_sec_privileged`	Интервал времени, в течение которого учитываются неудачные попытки входа для блокировки аккаунта. По умолчанию — *300* (5 мин)*. Допустимые значения — целочисленные
`acc_block_unused_days_privileged`	Количество неактивных дней после которого пользователь будет заблокирован. По умолчанию — *45 дней*. Допустимые значения — целочисленные. Диапазон 0, далее от 10 до 365, где 0 – отключение проверки

Параметр ВЦОД	Описание
`auth_type_privileged`	Тип аутентификации. Возможные значения: • *BASIC* — логин и пароль (значение по умолчанию), • *TFA* — двухфакторная аутентификация
`cert_privileged`	Использование сертификата при аутентификации. Возможные значения: • *no* — не использовать (значение по умолчанию), • *yes* — использовать
`password_diff_cnt_privileged`	Минимальное количество различных (неповторяющихся) символов, которое должно содержаться в пароле пользователя. По умолчанию — *4*. Допустимые значения — целочисленные. Диапазон от 0 до 20, где 0 — отключение проверки. При установке значения 0 также отключается проверка `password_min_exp_days_privileged`
`password_exp_days_privileged`	Срок действия пароля. По умолчанию — *60 дней*. 0 — отключение проверки
`password_min_change_privileged`	Минимальное количество символов, которое надо изменить при замене пароля. По умолчанию — *4*
`password_min_exp_days_privileged`	Минимальный срок, который должен пройти после установки пароля, прежде чем его можно будет сменить. По умолчанию — *1 день*
`password_pattern_privileged`	Сложность пароля Минимум 8 символов, включающих прописные буквы, строчные буквы, цифры, специальные знаки

Параметр ВЦОД	Описание
`sessions_max_cnt_privileged`	Максимальное количество сессий пользователя. По умолчанию — *2*. Допустимые значения — целочисленные. Диапазон от 0 до 10, где 0 — отключение проверки
`sessions_timeout_sec_privileged`	Время отключения сессии при бездействии. По умолчанию — *180* с (3 мин). Допустимые значения — целочисленные. Диапазон 0, далее от 180 с (3 мин) до 43 200 с (12 ч), где 0 — отключение проверки

16. Двухфакторная аутентификация

> **Примечание**
>
> Действия выполняются пользователем с ролью **Администратор ВЦОД**

Двухфакторная аутентификация — опциональная функция.
При необходимости ее включения выполните настройки из данной статьи.

Шаги для включения двухфакторной аутентификации:

1. Посмотрите текущие настройки механизмов безопасности платформы

   ```
   aaa param show
   ```

   Sharx Base выведет текущие настройки.

   Обратите внимание на параметр `auth_type`. Если его значение не равно *TFA*, то выполните действия ниже.

2. Чтобы включить двухфакторную аутентификацию введите команду

   ```
   aaa param update --auth_type TFA
   ```

3. Проверьте включение двухфакторной аутентификации в настройках механизмов безопасности платформы

```
aaa param show
```

Параметр `auth_type` должен быть равен *TFA*.

> **Внимание**
>
> Для использования двухфакторной аутентификации у пользователей на мобильном устройстве должно быть установлено приложение **SharxOTP**

16.1. Проверка настройки

1. Выйдите из ВЦОД.
2. Аутентифицируйтесь во ВЦОД.
3. После включения двухфакторной аутентификации при первом входе пользователя генерируется OTP-токен, на экране Sharx Base отобразится QR-код. Запустите на мобильном устройстве приложение SharxOTP, введите пароль для входа в приложение.

> **Примечание**
>
> Дистрибутив мобильного приложения SharxOTP в виде файла с расширением .apk недоступен для скачивания и предоставляется технической поддержкой ООО «Шаркс ДЦ».
> С инструкцией по установке можно ознакомиться в статье Установить SharxOTP

4. Нажмите в приложении кнопку .
5. Нажмите кнопку считывания QR-кода.
6. При необходимости в настройках мобильного устройства разрешите приложению SharxOTP снимать фото и видео.
7. Считайте сгенерированный QR-код.
8. После считывания в мобильном приложении появится шестизначный код. В течение 15 секунд после его появления введите код в командную строку Sharx Base

```
Enter OTP code: ******
```

9. Если QR-код не считывается, в приложении нажмите кнопку , затем кнопку, похожую на карандаш.

В открывшемся окне в полях введите:

- в поле **jdoe@example.com** введите свою *электронную почту* или *логин пользователя* в Sharx Base;
- в поле **Example Inc** — *имя кластера* или *адрес выделенного узла*;
- в поле **Secret** — значение *OTP secret* из консоли Sharx Base;
- в поле **Type** выберите флаг *TOTP*;
- в поле **Digits** — *6*;
- в поле **Algorithm** — *SHA1*;
- в поле **Interval** — *15*.

10. Нажмите кнопку **ADD TOKEN**.

    Появится шестизначный код.

    В течение 15 секунд после его появления введите код в командную строку Sharx Base

```
Enter OTP code: ******
```

> **Внимание**
>
> Код аутентификации действует 15 с

11. Если Вы не успели ввести код в течение 15 с:

- Останьтесь в приложении SharxOTP.
- Нажмите на экран мобильного устройства.
- Сгенерируется новый код.
- Введите его в Sharx Base в течение 15 с.

17. Создать виртуальный кластер

> **Примечание**
>
> Действия выполняются пользователем с ролью **Администратор ВЦОД**

17.1. Создать виртуальный кластер

Примечание

При создании виртуального кластера узлам, входящим в него, автоматически назначаются метки, отданные во ВЦОД

1. Проверьте количество доступных ресурсов

```
scheduler vcluster resource show
```

2. Если ресурсов достаточно, создайте виртуальный кластер

```
scheduler vcluster add --name NAME
                      [--uuids <UUIDS>]
                      [--descr <DESCR>]
                      --labels <LABELS>
                      [--rf <RF>]
                      [--isolation_strategy <ISOLATION_STRATEGY>]
                      [--drain_strategy <DRAIN_STRATEGY>]
                      [--ulimits <ULIMITS>]
                      [--cpu <CPU>]
                      [--ram <RAM>]
                      [--path_to <PATH_TO>]
                      [vram_scaling <FALSE|TRUE>]
                      [vcpu_scaling <FALSE|TRUE>]
```

где

- `name` — имя создаваемого виртуального кластера;
- `uuids` — перечень узлов, входящих в виртуальный кластер;
- `descr` — описание виртуального кластера;
- `labels` — метки, назначаемые виртуальному кластеру;
- `rf` — количество узлов, возможных к выходу и строя. По умолчанию — *n/2*, где n — количество узлов в виртуальном кластере;
- `isolation_strategy` — логика планирования ресурсов в случае изоляции узла. По умолчанию — *RECREATE*. Возможные значения:
    - `NOTHING` — не производить никаких действий,
    - `STOP` — остановить все ресурсы на текущем узле,
    - `RECREATE` — остановить все ресурсы на изолированном узле, пересоздать на доступных;

- `drain_strategy` — логика планирования ресурсов в случае выведения узла из эксплуатации. По умолчанию — `MIGRATE`. Возможные значения:
  - `MIGRATE` — перенести ресурсы на другие доступные узлы,
  - `NOTHING` — не производить никаких действий,
  - `STOP` — остановить все ресурсы на текущем узле;
- `ulimits` — использование пользовательских лимитов на уровне виртуального кластера. Значение по умолчанию — `yes` Возможные значения:
  - `yes` — лимиты включены,
  - `no` — лимиты выключены;
- `cpu` — лимит ЦПУ. При отсутствии параметра система вычислит его автоматически с учетом доступных на данный момент ресурсов;
- `ram` — лимит ОЗУ. При отсутствии параметра система вычислит его автоматически с учетом доступных на данный момент ресурсов. Например, единственному виртуальному кластеру в системе будут отданы все свободные ресурсы;
- `path_to` — размещение виртуального кластера в иерархии директорий;
- `vram_scaling` — возможность увеличивать ОЗУ ВМ, когда она включена. Возможные значения:
  - `false` — возможность выключена. Значение по умолчанию,
  - `true` — возможность включена. Максимальное увеличение может в 2 раза превышать начальное значение ОЗУ при включении ВМ.
- `vcpu_scaling` — возможность увеличивать ВЦПУ ВМ, когда она включена. Возможные значения:
  - `false` — возможность выключена. Значение по умолчанию,
  - `true` — возможность включена. Максимальное увеличение может в 2 раза превышать начальное значение ВЦПУ при включении ВМ.

> **Внимание**
>
> Параметры `vram_scaling` и `vcpu_scaling` нельзя изменить после создания виртуального кластера

17.2. Дополнительные команды

1. Просмотреть список всех виртуальных кластеров в пределах ВЦОД

```
scheduler vcluster list
```

2. Просмотреть информацию о конкретном виртуальном кластере

```
scheduler vcluster show --name <NAME>
```

где `name` — имя виртуального кластера.

3. Просмотреть логины и роли пользователей, имеющих доступ к виртуальному кластеру

```
scheduler vcluster rights --name <NAME>
```

18. Управлять виртуальным кластером

Примечание

Действия выполняются пользователем с ролью **Администратор ВЦОД**

Администратор ВЦОД может настраивать различные параметры виртуального кластера после его создания. Возможно изменить ресурсы, стратегии поведения, состав узлов и лимиты.

18.1. Изменить параметры виртуального кластера

Обновить параметры кластера в части ресурсов, стратегий или пользовательских лимитов

```
scheduler vcluster update --name <NAME>
                          [--descr <DESCR>]
                          [--rf <RF>]
                          [--cpu <CPU>]
                          [--ram <RAM>]
                          [--isolation_strategy <ISOLATION_STRATEGY>]
                          [--drain_strategy <DRAIN_STRATEGY>]
                          [--ulimits <ULIMITS>]
```

где

- `name` — имя изменяемого виртуального кластера;
- `descr` — описание виртуального кластера;
- `rf` — количество узлов, возможных к выходу и строя;
- `ram` — лимит ОЗУ;
- `cpu` — лимит ЦПУ;

- `isolation_strategy` — логика планирования ресурсов в случае изоляции узла. По умолчанию — *RECREATE*. Возможные значения:
  - `NOTHING` — не производить никаких действий,
  - `STOP` — остановить все ресурсы на текущем узле,
  - `RECREATE` — остановить все ресурсы на изолированном узле, пересоздать на доступных;
- `drain_strategy` — логика планирования ресурсов в случае выведения узла из эксплуатации. По умолчанию — *MIGRATE*. Возможные значения:
  - `MIGRATE` — перенести ресурсы на другие доступные узлы,
  - `NOTHING` — не производить никаких действий,
  - `STOP` — остановить все ресурсы на текущем узле;
- `ulimits` — использование пользовательских лимитов на уровне виртуального кластера. Значение по умолчанию — *yes*. Возможные значения:
  - `yes` — лимиты включены,
  - `no` — лимиты выключены.
  Подробная информация о настройке описана в статье Лимиты пользователей.

18.2. Управлять узлами виртуального кластера

1. Добавить узлы в виртуальный кластер

```
scheduler vcluster nodes add --name <NAME>
                             --uuids <UUIDS>
```

где

- `name` — имя изменяемого виртуального кластера;
- `uuids` — список идентификаторов добавляемых узлов.

2. Удалить узлы из виртуального кластера

```
scheduler vcluster nodes del --name <NAME>
                             --uuids <UUIDS>
```

3. Просмотреть текущие узлы в составе виртуального кластера

```
scheduler vcluster nodes list
```

18.3. Удалить виртуальный кластер

1. Удалить виртуальный кластер

```
scheduler vcluster del --name <NAME>
```

где `name` — имя виртуального кластера.

2. Очистить список виртуальных кластеров в БД

```
scheduler vcluster clear --name <NAME>
```

19. Лимиты пользователей

> **Примечание**
>
> Действия выполняются пользователем с ролью **Администратор ВЦОД**

Лимиты пользователей используются для контроля объема доступных ресурсов, выделенных каждому пользователю. Чтобы настроить систему пользовательских ограничений, необходимо активировать и правильно настроить соответствующие лимиты в конфигурации виртуального кластера.

Эта функция позволяет:

- Установить максимальные пределы использования ресурсов пользователями.
- Контролировать потребление вычислительных мощностей.
- Предотвратить чрезмерное использование ресурсов пользователем.
- Обеспечить распределение ресурсов между всеми пользователями кластера.

19.1. Включить режим лимитов для пользователей

> **Важно**
>
> Ограничьте доступ к командам изменения лимитов в правах пользователей.
> Настройка должна производиться только администраторами

Чтобы включить режим лимита пользователей, введите команду

```
scheduler vcluster update --ulimits yes
```

19.2. Настроить лимиты пользователя

Укажите объемы ресурсов и параметры для каждого пользователя

```
scheduler user param add --vcluster <VCLUSTER>
                         --login <LOGIN>
                        [--rf <RF>]
                        [--isolation_strategy <ISOLATION_STRATEGY>]
                        [--drain_strategy <DRAIN_STRATEGY>]
                        [--cpu <CPU>]
                        [--ram <RAM>]
```

где

- `vcluster` — имя виртуального кластера;
- `login` — логин пользователя;
- `rf` — фактор репликации;
- `isolation_strategy` — логика планирования ресурсов в случае изоляции узла. По умолчанию — *NOTHING*. Возможные значения:
  - `NOTHING` — не производить никаких действий,
  - `STOP` — остановить все ресурсы на текущем узле,
  - `RECREATE` — остановить все ресурсы на изолированном узле, пересоздать на доступных;
- `drain_strategy` — логика планирования ресурсов в случае выведения узла из эксплуатации. По умолчанию — *NOTHING*. Возможные значения:
  - `MIGRATE` — перенести ресурсы на другие доступные узлы,
  - `NOTHING` — не производить никаких действий,
  - `STOP` — остановить все ресурсы на текущем узле;
- `cpu` — лимит ЦПУ;
- `ram` — лимит ОЗУ.

19.3. Обновить лимиты пользователя

```
scheduler user param update --vcluster <VCLUSTER>
                            --login <LOGIN>
                            [--rf <RF>]
                            [--isolation_strategy <ISOLATION_STRATEGY>]
```

```
                                [--drain_strategy <DRAIN_STRATEGY>]
                                [--cpu <CPU>]
                                [--ram <RAM>]
```

где

- `vcluster` — имя виртуального кластера;
- `login` — логин пользователя;
- `rf` — фактор репликации;
- `isolation_strategy` — логика планирования ресурсов в случае изоляции узла. По умолчанию — *NOTHING*. Возможные значения:
  - `NOTHING` — не производить никаких действий,
  - `STOP` — остановить все ресурсы на текущем узле,
  - `RECREATE` — остановить все ресурсы на изолированном узле, пересоздать на доступных;
- `drain_strategy` — логика планирования ресурсов в случае выведения узла из эксплуатации. По умолчанию — *NOTHING*. Возможные значения:
  - `MIGRATE` — перенести ресурсы на другие доступные узлы,
  - `NOTHING` — не производить никаких действий,
  - `STOP` — остановить все ресурсы на текущем узле;
- `cpu` — лимит ЦПУ;
- `ram` — лимит ОЗУ.

19.4. Просмотреть лимиты пользователей

1. Чтобы посмотреть список всех пользователей, у которых установлены пользовательские лимиты в виртуальном кластере, введите

   ```
   scheduler user param list --vcluster <VCLUSTER>
   ```

   где `vcluster` — имя виртуального кластера.

2. Просмотр информации о лимитах конкретного пользователя

   ```
   scheduler user param admin --vcluster <VCLUSTER>
                              --login <LOGIN>
   ```

где

- `vcluster` — имя виртуального кластера;
- `login` — логин пользователя.

3. Пользователь может посмотреть свои лимиты в пределах виртуального кластера командой

```
scheduler user param show --vcluster <VCLUSTER>
```

19.5. Удалить лимиты пользователей

Чтобы удалить лимиты пользователей, введите

```
scheduler user param del --vcluster <VCLUSTER>
                         --login <LOGIN>
```

где

- `vcluster` — имя виртуального кластера;
- `login` — логин пользователя.

20. Настроить метки виртуального кластера

Метка (label) — пара ключ-значение формата `key=value`, которая описывает атрибуты объектов кластера. Играет ключевую роль в механизме планирования размещения пользовательских ресурсов (ВМ) на узлах. Метки могут быть назначены в кластере, ВЦОД, виртуальном кластере и ВМ.
В Sharx Base используются следующие типы меток: обычная, делегированная, игнорируемая, метка-ограничение, системная и криптографическая метка.

Основные понятия:

- **Обычная метка** — метка, создаваемая и назначаемая пользователями вручную. Не имеет специальных системных свойств, таких как `taint` или игнорирование.
- **Системная метка** — используется системой, недоступна для создания и назначения вручную. Например, метка активации сервисного обслуживания `system_drain_node=yes`. Подробная информация о данной метке описана в статье CLI.Руководство администратора/Сервисное обслуживание.

- **Игнорируемая метка** — метка, существующая в системе, но не учитываемая при планировании. Метка может игнорироваться в контексте ВЦОД и виртуального кластера. Планировщик не учитывает игнорируемые метки во время поиска узлов по ним.
- **Делегирование метки** — предоставление возможности использовать метку объектам: ВЦОД или виртуальному кластеру.
- **Криптографическая метка** — метка, применяемая к ВМ для обозначения ее принадлежности к контуру криптографической защиты (СКЗИ). Имеет ключ-значение `vm=crypto`. Метка определяет политику управления ВМ, блокируя функции миграции и создания снимков для исключения риска утечки защищаемых данных. Криптографическую метку может назначить или снять только определенная роль специальной командой.

Метки могут быть назначены следующим объектам виртуальной инфраструктуры:

- кластеру (см. Руководство администратора);
- ВЦОД (см. Руководство администратора);
- виртуальному кластеру.
- виртуальной машине.

Администратор ВЦОД управляет метками в виртуальном кластере.

Чтобы просмотреть метки, отданные во ВЦОД, введите команду

```
scheduler labels ns show [--filter <FILTER>]
```

где `filter` — фильтрация меток. Возможные значения:

- `a` — all, все метки;
- `i` — ignored, игнорируемые метки;
- `u` — unignored, все метки, кроме игнорируемых.

Внимание

Пользователи могут видеть метки только ВЦОД, в котором они находятся.
При попытке запроса информации о других ВЦОД будет выдаваться ошибка.
Исключение — ВЦОД управления

20.1. Ограничения работы с метками в виртуальном кластере

- При создании виртуального кластера в него автоматически добавляется метка сервисного обслуживания `system_drain_node=yes` при ее наличии в родительском ВЦОД.
- В виртуальный кластер можно добавить только метки данного ВЦОД.
- В виртуальный кластер нельзя добавить метки, которые игнорируются в данном ВЦОД.
- Нельзя удалить **все** метки из виртуального кластера.
- При удалении виртуального кластера все записи о делегированных ему метках удаляются.
- Нельзя открепить метки, если по ним размещены ресурсы.
- При указании в `nodeSelector` меток, игнорируемых в кластере, ВЦОД или виртуальном кластере, будет выведена ОШИБКА.
- Системная метка `system_drain_node=yes` запрещает размещение ресурсов на соответствующих узлах.
- При обновлении ресурса изменение `nodeSelector` невозможно.

20.2. Операции с метками в виртуальном кластере

> **Примечание**
>
> Далее в тексте понятие **Текущий ВЦОД** — это ВЦОД, в котором выполняется команда

1. Добавить метки в заданные виртуальные кластеры текущего ВЦОД

   ```
   scheduler labels vcluster add --vcluster <VCLUSTER>
                                 --labels <LABELS>
   ```

   где

   - `vcluster` — список имен виртуальных кластеров. При значении `*` метка устанавливается на все виртуальные кластеры текущего ВЦОД;
   - `labels` — список меток для добавления в виртуальные кластеры. Задается в формате `key=value`.

2. Просмотреть метки виртуальных кластеров

   ```
   scheduler labels vcluster show [--vcluster <VCLUSTER>]
                                  [--filter <FILTER>]
   ```

где

- `vcluster` — список имен виртуальных кластеров для просмотра их меток. При значении `*` отобразятся все метки на всех виртуальных кластерах;
- `filter` — фильтрация меток. Возможные значения:
  - `a` — all, все метки;
  - `i` — ignored, игнорируемые метки;
  - `u` — unignored, все метки, кроме игнорируемых.

3. Просмотреть метки со списками виртуальных кластеров, в которые они делегированы

```
scheduler labels vcluster list [--vcluster <VCLUSTER>]
                               [--filter <FILTER>]
```

где

- `vcluster` — список имен виртуальных кластеров для просмотра делегированных им меток. При значении `*` отобразятся все делегированные метки на всех виртуальных кластерах;
- `filter` — фильтрация меток. Возможные значения:
  - `a` — all, все метки;
  - `i` — ignored, игнорируемые метки;
  - `u` — unignored, все метки, кроме игнорируемых.

4. Просмотреть группы совместимых меток в виртуальном кластере

```
scheduler labels vcluster sharing --vcluster <VCLUSTER>
```

В результате выполнения команды будут показаны группы меток, которые можно использовать вместе для успешного планирования пользовательских ресурсов на узлы кластера.

5. Удалить метки из виртуальных кластеров текущего ВЦОД

> ❌ **Внимание**
>
> - Из виртуального кластера можно удалить только метки, которые делегированы в текущий ВЦОД.
> - Нельзя удалить метки, если их используют ВМ

```
scheduler labels vcluster del --vcluster <VCLUSTER>
                              --labels <LABELS>
```

где

- `vcluster` — список имен виртуальных кластеров для просмотра делегированных им меток. При значении `*` отобразятся все делегированные метки на всех виртуальных кластерах;
- `labels` — список меток для удаления. Задается в формате `key=value`.

21. Подготовить хранилище

Sharx Base поддерживает работу с тремя типами хранилища:

1. Libvirt. Локальное хранилище узла.
2. РСХД. Распределенная система хранения данных.
3. NFS-хранилище. Сетевое файловое хранилище.

Последовательность подготовки объектов хранилища зависит от типа хранилища и описана в разделах ниже.

21.1. Libvirt

Libvirt позволяет управлять хранилищем на физическом узле через пулы и тома.

Пул — это объем хранилища, выделенный администратором для использования виртуальными машинами. Пулы хранилища делятся на тома. Тома назначаются виртуальным машинам (далее — ВМ) в качестве дисков.

Администратор ВЦОД создает пулы, **Разработчик ВМ** — тома.
Действия с пулами описаны ниже.
Создание томов — в статье Создать том. Работа с томами описана в статье Работа с томами.

21.1.1. Создать пул

> **Внимание**
>
> Пулы, созданные в рамках одного ВЦОД, не доступны из другого

Чтобы создать пул, введите в командной строке

```
storage libvirt pool add --name <pool_name> --descr <pool_description> --labels <pool_label>
[--sc_node <node_uuid_where_pool_create>]
```

где `sc_node` — идентификатор узла для создания пула.

При `sc_node`, равном `*`, или отсутствии `sc_node`, пул будет создан на всех узлах кластера.

На физическом узле пул появится в директории `<dir_name>/<vcod_name>/<pool_name>`, где `dir_name` — директория, заданная по умолчанию при настройке Sharx Base технической поддержкой.

21.1.2. Работа с пулами

1. Просмотреть информацию и статус определенного пула

   ```
   storage libvirt pool show --name <pool_name> [--sc_node <node_uuid>]
   ```

   при отсутствии параметра `sc_node` отобразится информация о всех пулах с именем `name` на всех узлах кластера.

2. Просмотреть список всех пулов

   ```
   storage libvirt pool list
   ```

3. Удалить неиспользуемый пул

   ```
   storage libvirt pool del --name <pool_name> --sc_node <node_uuid>
   ```

4. Пул может зависнуть в статусе *Pending (Ожидание)*.

   В таких случаях удалите его из базы данных командой

   ```
   storage libvirt pool clear --name <pool_name> --sc_node <node_uuid>
   ```

   После этого создайте пул заново, указав корректные параметры.

> **Примечание**
>
> Команда `clear` используется для удаления из списка объектов, зависших в процессе создания и фактически несозданных.
> Команда `del` применяется для удаления уже созданных объектов

21.2. РСХД

РСХД — распределенная система хранения данных. Объединяет напрямую подключенные жесткие диски или твердотельные накопители с серверов и создает единое пространство общего хранилища.

Объекты РСХД:

- **Группы размещения** — наборы дисков, на которые будут реплицироваться объекты. Например, наборы по типу SSD.
- **Шаблон** — заранее заданный набор параметров, который является основой для создания томов. Шаблоны позволяют упростить процесс создания новых томов, обеспечивая быстрое и единообразное развертывание томов с заданными характеристиками.
- **Том** — единица деления шаблона. Тома назначаются виртуальным машинам в качестве дисков, на которых хранится определенная информация.
- **Снимок** — моментальный снимок тома. Копия тома на уровне блоков физических или виртуальных систем, выполненная без остановки системных служб. Снимок включает в себя структуру директорий, файлов и информацию о состоянии системы на фиксированный момент времени. При необходимости из снимка можно восстановить том.

Администратор кластера предварительно должен подготовить группы размещения и шаблоны.
Разработчик ВМ на основе подготовленного шаблона создает тома. Создание томов описано в статье Создать том. Работа с томами — в статье Работа с томами.

21.3. NFS

NFS-хранилище обеспечивает доступ к файлам, расположенным на удаленных серверах, и позволяет работать с этими файлами точно так же, как и с локальными.

Администратор кластера создает и настраивает подключение NFS-хранилища. Администратор ВЦОД или Разработчик ВМ — тома. Создание томов описано в статье Создать том. Работа с томами — в статье Работа с томами.

22. Журнал безопасности

Примечание

Действия выполняются пользователем с ролью **Администратор безопасности ВЦОД**

Журнал безопасности — хронологическая запись действий пользователей в рамках ВЦОД. Параметры событий определяются политикой безопасности Клиента и иными руководящими документами.

> **Внимание**
>
> Настройка и отслеживание событий происходит в рамках ВЦОД

В рамках журнала безопасности можно выполнить следующие действия:

1. Просмотреть зарегистрированные события.
2. Настроить параметры записи событий:
   - вводом параметров регистрации событий через командную строку;
   - загрузкой файла с параметрами регистрации событий.

22.1. Просмотреть события в журнале безопасности

1. Чтобы отобразить отслеживаемые события, введите в командной строке

```
aaaevents event list
```

2. Чтобы настроить фильтр отображаемых событий, используйте параметры, указанные в Таблице ниже.

   Таблица — Параметры настройки фильтра отображаемых событий.

Параметр	**Функции параметра**
`begin`, `end`	Временной диапазон отображаемых событий
`full`	Подробная информация о событии. Включает результат завершения операции: успех или ошибку
`group <event_group>`	Отображение событий в пределах указанной группы
`level <INFO, WARNING, CRITICAL>`	Уровень событий
`limit <number_events>`	Отображение заданного количества событий

Параметр	Функции параметра
`login <user_login>`	Список всех действий пользователя за последнее время
`obj_uuid`	Отобразить события, связанные только с определенным виртуальным кластером или ВМ
`operation`	Тип событий

3. Чтобы отобразить конкретное событие, введите

```
aaaevents event show --uuid <event_uuid>
```

где `uuid` — идентификатор события.

4. Просмотр всех типов событий в пределах ВЦОД

```
aaaevents event types
```

22.2. Настроить параметры записи событий

22.2.1. Настроить параметры регистрации событий вручную

1. Настроить уровень регистрируемых событий

```
aaaevents event loglevel --setlevel {INFO, WARNING, CRITICAL}
```

2. Задать параметры хранения событий

```
aaaevents param add --events_params <group or level>  --ttl_days <days_number>
```

где

- `events_params` — указывает группы или уровни события;
- `ttl_days` — период хранения событий в днях.

3. Обновить параметры

```
aaaevents param update --events_params <group or level>  --ttl_days <days_number>
```

4. Удалить параметры

```
aaaevents param del
```

22.2.2. Загрузить файл с параметрами регистрации событий

Параметры записи событий можно загрузить с помощью файла YAML.

1. Создайте и сохраните файл с параметрами записи событий.

   Пример файла

```yaml
apiVersion: v1
kind: api
metadata:
   plugin:
      aaaevents:
         param: add

spec:
   ttl_days: 14
   events_params:
      "User logout":
         group: "Authentication and authorization"
         level: "WARN"
         notif_route: test
      "Auth to cluster":
         group: "Authentication and authorization"
         level: "CRITICAL"
         notif_route: test
      "Add new user":
         group: "User account"
         level: "INFO"
         notif_route: test
```

   где

   - `"User logout"`, `"Auth to cluster"`, `"Add new user"` — типы событий;
   - `group` — группа, к которой относится данный тип события;
   - `level` — уровень критичности события. Доступны 3 уровня критичности: *INFO*, *WARNING*, *CRITICAL*;
   - `notif_route` — имя маршрута для отправки уведомлений о событии.

> **Примечание**
>
> Заранее настройте маршрут отправки уведомлений о событиях.
> Для каждого типа событий можно использовать отдельные маршруты записи

2. Загрузите файл на Платформу.

   Чтобы загрузить файлы в Sharx Base, введите команду

   ```
   resource --spec <path_to_yaml_file>
   ```

   где `spec` — расположение YAML-файла.

   При локальном расположении на АРМ пользователя введите путь до файла.

   При расположении в стороннем репозитории укажите ссылку на данный файл.

3. Проверьте загрузку файла

   ```
   aaaevents param show
   ```

   Параметры событий должны отобразиться в общем списке.

23. Виртуальные машины

Создание ВМ состоит из следующих шагов:

1. Аутентифицируйтесь во ВЦОД.
2. Создайте том.
3. Создайте виртуальную машину.
4. Запустите виртуальные машины.

> **Примечание**
>
> Далее действия выполняются пользователем с ролью **Разработчик ВМ**.
> **Администратор ВЦОД** также может выполнять данные действия

23.1. Аутентифицироваться во ВЦОД

> **Примечание**
>
> Действия выполняются пользователем с ролью **Администратор ВЦОД** или **Разработчик ВМ**

Действия для аутентификации:

1. Введите в командной строке АРМ пользователя или со стороннего аппаратного средства (в случае удаленного подключения)

   ```
   sdc-cli --host <vip_cluster_addr> [--tls_verify <path_to_cert_key>] [--tls yes]
   ```

   где

   - `host` — управляющий IP-адрес кластера. IP-адрес имеет вид `a.b.c.d`;
   - `tls-verify` — путь к сертификату TLS. Обязателен при включенном TLS-соединении;
   - `tls` — флаг использования TLS. Обязателен при включенном TLS-соединении.

2. Аутентифицируйтесь во ВЦОД

   ```
   login <user_login> --cluster <cluster_name> --ns <namespace_name> [--cluster_to <cluster_na
   me>] [--ns_to <vcod_name>]
   ```

   где

   - `login` — логин пользователя;
   - `cluster` — имя логического кластера;
   - `ns` — имя ВЦОД управления;
   - `cluster_to` — имя логического кластера. Необязательный параметр. При наличии одного логического кластера значения параметров `cluster` и `cluster_to` одинаковы;
   - `ns_to` — имя ВЦОД. Необязательный параметр.

23.2. Создать том

> **Примечание**
>
> Действия выполняются пользователем с ролью **Администратор ВЦОД** или **Разработчик ВМ**

Том — объект хранилища, который назначается виртуальным машинам в качестве диска, на котором хранится информация.

Возможно создать тома для хранилищ Libvirt, РСХД и NFS.

Дополнительная информация описана в статье Подготовить хранилище.

23.2.1. Libvirt. Создать том

> **Важно**
>
> **Администратор ВЦОД** заранее должен создать пулы Libvirt

В Libvirt том — единица деления пула.

Чтобы создать том в хранилище Libvirt:

1. Просмотрите список пулов

   ```
   storage libvirt pool list
   ```

2. Чтобы создать том, введите в командной строке

   ```
   storage libvirt volume add --capacity <volume_capacity> --descr <volume_description> --frmt <volume_format> --labels <volume_label> --name <volume_name> --pool <pool_name> --sc_node <node_uuid_where_create_volume>  --source <source>
   ```

   где `sc_node` — идентификатор узла для создания тома.

   При `sc_node`, равном `*`, том будет создан на всех узлах кластера, если пул также существует на всех узлах.

   При отсутствии параметра `sc_node` том будет создан на текущем узле.

3. Чтобы использовать тома Libvirt для виртуальных машин, необходимо загрузить их в пул командой

   ```
   storage libvirt image upload --name <volume_name> --pool <pool_name> --path <path_to_vm_image> [--sc_node SC_NODE]
   ```

   где

   - `name` - имя тома;
   - `pool` - имя пула;

- `path` - расположение тома. При локальном расположении на узле кластера введите путь на узле до тома. При расположении в стороннем репозитории укажите ссылку на том.

> **Примечание**
>
> Команды для работы с томами Libvirt описаны в статье Работа с томом

23.2.2. РСХД. Создать том

> **Важно**
>
> **Администратор кластера** заранее должен создать группы размещения и шаблоны РСХД

В РСХД том — единица деления шаблона.

Чтобы создать том в хранилище РСХД:

1. Просмотрите список созданных шаблонов

   ```
   storage sp template list
   ```

2. Просмотрите подробную информацию о шаблоне

   ```
   storage sp template show --name <name_template>
   ```

   где `name` — имя шаблона.

3. Просмотрите лимиты шаблона

   ```
   storage sp template limit show --name <name_template>
   ```

   где `name` — имя шаблона.

4. Чтобы создать том, введите в командной строке

   ```
   storage sp volume add [--base_on BASE_ON] [--bw BW] [--descr DESCR] [--iops IOPS] --name <v
   olume_name> [--parent PARENT] --persistent <yes_or_no> [--reuse_server REUSE_SERVER] --
   size <volume_size> --template <template_name> --type <cdrom,system,datablock>
   ```

где

- `base_on` — имя тома, на основе которого создается новый том. Новый том будет скопирован с существующего тома-источника со всеми существующими данными.

> **Примечание**
>
> Использование параметра `base_on` увеличивает время выполнения команды `add`

- `bw` — ограничение пропускной способности в Кбайт;
- `descr` — описание тома;
- `iops` — ограничение операций ввода-вывода;
- `name` — имя тома;
- `parent` — имя снимка, на основе которого можно создать том;
- `persistent` — возможность изменения тома;
- `reuse_server` — разрешение на размещение реплик на одном сервере;
- `size` — размер тома;
- `template` — имя шаблона;
- `type` — тип тома. Возможные значения: *cdrom, system, datablock*.

Также создание тома РСХД может осуществляться следующими способами:

1. Импортировать том в указанный шаблон

```
storage sp volume import [--descr <DESCR>] --name <volume_name> --path <where_parent_volume
> --persistent <yes_or_no> --template <template_name> --type <cdrom,system,datablock>
```

где

- `descr` — описание тома;
- `name` — имя тома;
- `path` — расположение тома.

  При локальном расположении на узле кластера введите путь на узле до файла.

  При расположении в стороннем репозитории укажите ссылку на данный файл;
- `persistent` — возможность изменения тома;
- `reuse_server` — разрешение на размещение реплик на одном сервере;
- `size` — размер тома;

- `template` — имя шаблона;
- `type` — тип тома. Возможные значения: *cdrom, system, datablock*.

2. Создать том из снимка

```
storage sp volume fromparent --name <volume_name> --persistent <yes_or_no> --snapshot_uuid
<snap_uuid> --type <cdrom,system,datablock>
```

где `snapshot_uuid` — идентификатор снимка.

3. Экспортировать том в локальное хранилище

```
storage sp volume export --uuid <volume_uuid>
                         --export_name <EXPORT_NAME>
                         [--export_format <EXPORT_FORMAT>]
```

где

- `uuid` — идентификатор тома;
- `export_name` — имя экспорта;
- `export_format` — формат экспорта. Возможные значения 'qcow2', 'iso'.

> **Примечание**
>
> Команды для работы с томами РСХД описаны в статье Работа с томом

23.2.3. NFS. Создать том

> **Важно**
>
> **Администратор кластера** заранее должен Настроить NFS-хранилище

Тома — образы дисков, которые назначаются виртуальным машинам для хранения и использования информации.

> **Внимание**
>
> Операции с ВМ, которые используют NFS-том, имеют ряд ограничений:
>
> - нельзя делать снимки ВМ;
> - невозможно изменить размер подключенного тома к ВМ;
> - недоступна корректировка характеристик созданного тома;
> - не работает миграция ВМ с подключенным томом NFS.

Создание тома NFS возможно двумя способами:

1. Создать том

   ```
   storage nfs volume add --name <NAME>
                          --mount_uuid <MOUNT_UUID>
                          [--size <SIZE>]
                          --type <TYPE>
                          --persistent <PERSISTENT>
                          [--base_on <BASE_ON>]
                          [--descr <DESCR>]
                          [--path_to <PATH_TO>]
   ```

   где

   - `name` — имя создаваемого диска в виде `test.qcow2`. Формат диска `.qcow2` определяется через заданное имя. В настоящее время поддерживаются 2 формата: `.qcow2` и `.iso`;
   - `mount_uuid` — идентификатор подключения. По нему определяется путь создания диска;
   - `size` — размер создаваемого диска. Не требуется при указании параметра `base_on`;
   - `type` — тип хранилища. Возможные значения `cdrom`, `system`, `datablock`;
   - `persistent` — возможность изменения тома. Возможные значения `yes` или `no`;
   - `base_on` — идентификатор тома, на основе которого создается новый том;
   - `descr` — описание тома.

2. Импорт NFS-тома

   ```
   storage nfs volume import --name <NAME>
                             --mount_uuid <MOUNT_UUID>
                             --type <TYPE>
                             --persistent <PERSISTENT>
                             [--descr <DESCR>]
                             --path <PATH>
                             [--path_to <PATH_TO>]
   ```

где

- `name` — имя импортируемого диска в виде виде `test.qcow2`. Формат диска `.qcow2` определяется через заданное имя. В настоящее время поддерживаются 2 формата: `.qcow2` и `.iso`;
- `mount_uuid` — идентификатор подключения. По нему определяется путь импорта диска;
- `type` — тип хранилища. Возможные значения `cdrom`, `system`, `datablock`;
- `persistent` — возможность изменения тома. Возможные значения *yes* или *no*;
- `descr` — описание тома;
- `path` — путь загрузки диска: локальный путь или URL;
- `path_to` — путь для размещения объекта в структуре каталогов. Позволяет указать, где именно должен быть размещен том.

> **Примечание**
>
> Команды для работы с томами РСХД описаны в статье Работа с томом

23.3. Работа с томом

> **Примечание**
>
> Действия выполняются пользователем с ролью **Администратор ВЦОД** или **Разработчик ВМ**

В статье описана работа с томами Libvirt, РСХД и NFS.

Создание томов описано в статье Создать том.

23.3.1. Libvirt. Работа с томами

1. Просмотреть список всех томов ВЦОД в рамках указанного пула

   ```
   storage libvirt volume list --pool <pool_name>
   ```

   В качестве дополнительного параметра можно указать `sc_node`, чтобы отобразить список томов на конкретном узле.

2. Просмотр подробной информации о конкретном томе

   ```
   storage libvirt volume show --name <volume_name> --pool <pool_name>
   ```

Введите параметр `sc_node`, если запрос производится не с того узла, на котором находится запрашиваемый том.

3. Обновить параметры тома

```
storage libvirt volume update --capacity <new_capacity> --name <volume_name>  --pool <pool_name> --sc_node <node_uuid_where_volume_place>
```

4. Удалить том

```
storage libvirt volume del --name <volume_name> --pool <pool_name> --sc_node <node_uuid_where_volume_place>
```

5. Том может зависнуть в статусе *Pending (Ожидание)*.

Внимание

Данная команда доступна только Администратору ВЦОД

В таких случаях удалите его из базы данных командой

```
storage libvirt volume clear --name <volume_name> --pool <pool_name> --sc_node <node_uuid_where_volume_place>
```

После этого создайте том заново, указав корректные параметры.

Примечание

Команда `clear` используется для удаления из списка объектов, зависших в процессе создания и фактически несозданных.
Команда `del` применяется для удаления уже созданных объектов

23.3.2. РСХД

РАБОТА С ТОМАМИ

1. Просмотреть список всех томов в рамках указанного шаблона

```
storage sp volume list  [--all {yes,no}] [--template <template_name>]
```

где

- `all` — список всех томов.

Возможные значения:

- по умолчанию — *yes*, список всех томов,
- *no* — список томов, принадлежащих определенному шаблону `template`;

- `template` — имя шаблона при флаге `all no`.

2. Просмотр подробной информации о конкретном томе

```
storage sp volume show --uuid <volume_uuid>
```

где `uuid` — идентификатор тома.

3. Обновить параметры тома

```
storage sp volume update [--bw <BW>] [--descr <DESCR>] [--iops <IOPS>] --uuid
<volume_uuid> [--persistent <yes_or_no>] [--reuse_server <REUSE_SERVER>] [--size <volume_si
ze>] [--type <cdrom,system,datablock>]
```

где

- `bw` — ограничение пропускной способности в Кбайт;
- `descr` — описание тома;
- `iops` — ограничение операций ввода-вывода;
- `uuid` — идентификатор тома;
- `persistent` — возможность изменения тома;
- `reuse_server` — разрешение на размещение реплик на одном сервере;
- `size` — размер тома;
- `type` — тип тома. Возможные значения: *cdrom, system, datablock*.

Примечание

Чтобы увеличить размер тома, подключенного к ВМ, выполните действия подраздела Увеличить размер тома, подключенного к виртуальной машине

4. Конвертировать том в снимок.

Важно

Том после команды удаляется

```
storage sp volume freeze --uuid <volume_uuid>
```

где `uuid` — идентификатор тома.

5. Том может зависнуть в статусе *Pending (Ожидание)*.

> **Внимание**
>
> Данная команда доступна только Администратору ВЦОД

В таких случаях удалите его из базы данных командой

```
storage sp clear --sc_uuid <node_uuid> --volume_uuid <volume_uuid>
```

После этого создайте том заново, указав корректные параметры.

6. Удалить том

```
storage sp volume del --uuid <volume_uuid>
```

> **Примечание**
>
> Команда `clear` используется для удаления из списка объектов, зависших в процессе создания и фактически несозданных.
> Команда `del` применяется для удаления уже созданных объектов

УВЕЛИЧИТЬ РАЗМЕР ТОМА, ПОДКЛЮЧЕННОГО К ВИРТУАЛЬНОЙ МАШИНЕ

> **Важно**
>
> Возможно только увеличить размера тома, подключенного к ВМ

Чтобы увеличить размер тома, подключенного к виртуальной машине:

1. Просмотрите список всех томов

```
storage sp volume list --all yes
```

2. Выберите из списка `base_on` том, размер которого нужно увеличить. Запишите его идентификатор.
3. Введите команду

```
storage sp volume update --uuid <base_on_volume_uuid> --size <new_size>
```

где

- `uuid` — идентификатор `base_on` тома;
- `size` — новый размер тома. Может быть только больше предыдущего размера.

4. Если виртуальная машина, к которой подключен увеличенный диск, была включена, выключите ВМ командой

```
scheduler request state --name <vm_name> --vcluster <vcluster_name> --state stop
```

где

- `name` — имя ВМ;
- `vcluster` — имя виртуального кластера.

5. Включите виртуальную машину

```
scheduler request state --name <vm_name> --vcluster <vcluster_name> --state start
```

23.3.3. NFS. Работа с томами

> **Примечание**
>
> Ограничения работы с NFS-томами описаны в статье NFS. Создать том

1. Просмотреть информацию о томе

```
storage nfs volume show --uuid <UUID>
```

где `uuid` — идентификатор тома.

2. Просмотреть все существующие тома

```
storage nfs volume list [--mount_uuid <MOUNT_UUID>]
```

где `mount_uuid` — идентификатор подключения.

3. Удалить том

```
storage nfs volume del --uuid <UUID>
```

где `uuid` — идентификатор тома.

4. Том может зависнуть в статусе *Pending (Ожидание)*. В таких случаях удалите его из базы данных командой

```
storage nfs volume clear --uuid <UUID>
```

где `uuid` — идентификатор тома.

После этого создайте том заново, указав корректные параметры.

23.4. Сетевые интерфейсы и правила доступа

Статья описывает операции с сетевыми ресурсами, доступные **Администратору ВЦОД** после настройки базовой инфраструктуры Администратором кластера.

Основные операции управления сетями:

1. VXLAN-подсети. Создание и управление IP-сегментами в overlay-сетях.
2. Сетевые интерфейсы. Просмотр и изменение параметров интерфейсов ВМ.
3. Статические IP-адреса. Резервирование фиксированных адресов.
4. Правила доступа (ACL). Настройка политик безопасности для сетей и портов ВМ.

> **Примечание**
>
> Базовая настройка сетевой инфраструктуры должна быть выполнена Администратором кластера в соответствии с Руководством администратора

23.4.1. VXLAN-подсети

VXLAN-подсеть — это IP-сегмент внутри VXLAN-сети, который предоставляет услуги маршрутизации для виртуальных машин.

Особенности VXLAN-подсетей:

- каждая подсеть создается внутри родительской VXLAN-сети и наследует ее ВЦОД;
- при создании подсети автоматически назначается VID из диапазона `vxlan_vid_range`;
- не поддерживают применение правил списков контроля доступа.

Действия с VXLAN-подсетями:

1. Чтобы создать VXLAN-подсеть, введите в командной строке

```
ipam vxlan subnet add --name <NAME>
                      --vxlan_uuid <VXLAN_UUID>
                      --network <NETWORK>
                      --ip_first <IP_FIRST>
                      --ip_last <IP_LAST>
                      [--gateway <GATEWAY>]
                      [--dns_1 <DNS_1>]
                      [--dns_2 <DNS_2>]
                      [--desc <DESC>]
```

где

- `name` — имя подсети;
- `vxlan_uuid` — идентификатор родительской VXLAN-сети;
- `network` — IP-адрес подсети в формате CIDR;
- `ip_first` — начало диапазона DHCP IP-адресов;
- `ip_last` — конец диапазона DHCP IP-адресов;
- `gateway` — адрес шлюза по умолчанию;
- `dns_1` — первичный адрес DNS-сервера;
- `dns_2` — вторичный адрес DNS-сервера;
- `desc` — описание VXLAN-подсети.

Пример

```
ipam vxlan subnet add --name vx-sub-1
                      --vxlan_uuid <UUID parent VXLAN>
                      --network 10.1.2.0/24
                      --gateway 10.1.2.1
                      --ip_first 10.1.2.2
                      --ip_last 10.1.2.254
```

2. Просмотреть список подсетей

```
ipam vxlan subnet list [--vxlan_uuid <VXLAN_UUID>]
```

Из текущего ВЦОД видны только подсети, находящиеся в нем. При указании `vxlan_uuid` отобразятся подсети конкретной VXLAN-сети.

3. Просмотреть подробную информацию о VXLAN-подсети

```
ipam vxlan subnet show --subnet_uuid <SUBNET_UUID>
```

где `subnet_uuid` — идентификатор VXLAN-подсети.

4. Просмотреть интерфейсы ВМ в VXLAN-подсети

```
ipam iface list --net_uuid <SUBNET_UUID>
```

где `net_uuid` — идентификатор VXLAN-подсети.

5. Удалить VXLAN-подсеть

Внимание

Нельзя удалить VXLAN-подсеть, если в ней есть интерфейсы ВМ

```
ipam vxlan subnet del --subnet_uuid <SUBNET_UUID>
```

где `subnet_uuid` — идентификатор VXLAN-подсети.

23.4.2. Сетевые интерфейсы

Sharx Base автоматически управляет сетевыми интерфейсами:

- При создании ВМ в VXLAN-подсети система автоматически создает сетевой интерфейс.
- Для каждого интерфейса генерируются уникальные MAC и IP-адреса.
- При удалении ВМ все связанные сетевые интерфейсы освобождаются автоматически.
- Освобожденные IP-адреса могут быть назначены новым ВМ.

Также сетевыми интерфейсами можно управлять вручную.

1. Посмотреть список сетевых интерфейсов

```
ipam iface list [--net_uuid <NET_UUID>]
                [--vxlan_uuid <VXLAN_UUID>]
                [--show_acl <SHOW_ACL>]
```

где

- `net_uuid` — идентификатор сети. Обязателен;
- `vxlan_uuid` — идентификатор VXLAN. Обязателен;

Примечание

Для выполнения команды необходимо указать один из параметров: `net_uuid` или `vxlan_uuid`

- `show_acl` — показать правила списков контроля доступа для каждого интерфейса. Возможные значения:
    - `y` — да;
    - `n` — нет.

2. Обновить IP-адрес сетевого интерфейса виртуальной машины

```
ipam iface update --iface_id <IFACE_ID>
                  --ip_address <IP_ADDRESS>
```

где

- `iface_id` — идентификатор сетевого интерфейса, который нужно изменить;
- `ip_address` — новый IP-адрес для указанного сетевого интерфейса.

23.4.3. Статические IP-адреса

1. Чтобы создать статический IP-адрес, введите

```
ipam addr virtual add --ip <IP>
                      --net_uuid <NET_UUID>
```

где

- `ip` — новый виртуальный IP-адрес для указанной сети;
- `net_uuid` — идентификатор сети.

2. Посмотреть список созданных адресов

```
ipam addr virtual list --net_uuid <NET_UUID>
```

3. Удалить статический IP-адрес

```
ipam addr virtual del --ip <IP>
                      --net_uuid <NET_UUID>
```

23.4.4. Правила доступа

Списки контроля доступа ACL можно применять как к конкретному порту ВМ, так и на всю виртуальную локальную сеть.

> **Примечание**
>
> Добавляйте ACL с помощью файла YAML, так как часто возникают ошибки при внесении данных

ПРАВИЛА ДОСТУПА ДЛЯ ПОРТА ВМ

> **Важно**
>
> Сначала загружайте ACL с разрешениями, затем с запретом. Неправильный порядок загрузки ACL приводит к блокировке трафика ВМ.
> ACL работают только для VLAN

1. Пример для настройки ACL для ВМ acl_vm.yaml.

   Файл описывает ACL `allow-rules`. Перечень списков контроля доступа включает:

   - разрешение доступа к репозиторию 10.2.2.11;
   - TCP-трафик до ya.ru;
   - TCP-трафик до конкретного адреса или с конкретного адреса.

2. ACL для блокировки остального вида трафика содержатся в файле deny_all.yaml.
3. Загрузите файл на Платформу.

   Чтобы загрузить файлы в Sharx Base, введите команду

   ```
   resource --spec <path_to_yaml_file>
   ```

   где `spec` — расположение YAML-файла.

   При локальном расположении на АРМ пользователя введите путь до файла.

   При расположении в стороннем репозитории укажите ссылку на данный файл.

4. Чтобы посмотреть список всех созданных ACL, введите

   ```
   ipam acl list
   ```

5. Посмотреть подробную информацию о конкретном списке контроля доступа

   ```
   ipam acl show --name <NAME>
   ```

   где `name` — имя списка контроля доступа.

6. Обновить список контроля доступа

```
ipam acl update --name <NAME>
                --data <DATA>
```

где `data` — обновленный перечень правил, входящих в файл **acl**.

7. Добавить списки контроля доступа для сетевого интерфейса

```
ipam iface acls add --acls <[ACLS ...]>
                    --mac <MAC>
                    --net_uuid <NET_UUID>
```

где

- `acls` — перечень имен списков контроля доступа для указанной сети;
- `mac` — MAC-адрес сетевого интерфейса;
- `net_uuid` — идентификатор сети.

8. Удалить список контроля доступа с порта ВМ можно командой

```
ipam iface acls del --acls <[ACLS ...]>
                    --mac <MAC>
                    --net_uuid <NET_UUID>
```

СПИСКИ КОНТРОЛЯ ДОСТУПА ACL ДЛЯ ВИРТУАЛЬНОЙ ЛОКАЛЬНОЙ СЕТИ

> **Важно**
>
> Сначала загружайте ACL с разрешениями, затем с запретом. Неправильный порядок загрузки ACL приводит к блокировке трафика ВМ.
> ACL работают только для VLAN

1. Пример для настройки ACL для виртуальной сети aclvlan.yaml.

   Файл описывает ACL `allow-rules`. Они совпадают с ACL для порта ВМ, но в aclvlan.yaml заданы списки контроля доступа для черных списков, а для порта ВМ — только для белых.

   В перечень ACL входит:

   - разрешение доступа к репозиторию 10.2.2.11;
   - TCP-трафик до ya.ru;
   - TCP-трафик до конкретного адреса или с конкретного адреса;
   - **ACL для черных списков**.

2. Списки контроля доступа для блокировки остального вида трафика содержатся в файле denyall.yaml.
3. Загрузите файл на Платформу.

   Чтобы загрузить файлы в Sharx Base, введите команду

   ```
   resource --spec <path_to_yaml_file>
   ```

   где `spec` — расположение YAML-файла.

   При локальном расположении на АРМ пользователя введите путь до файла.

   При расположении в стороннем репозитории укажите ссылку на данный файл.
4. Чтобы добавить ACL к виртуальной сети, введите

   ```
   ipam network acls add --acls <[ACLS ...]>
                         --net_uuid <NET_UUID>
   ```

   где

   - `acls` — перечень имен списков контроля доступа для указанной сети;
   - `net_uuid` — идентификатор сети.
5. Удалить списки контроля доступа у виртуальной сети

   ```
   ipam network acls del --acls <[ACLS ...]>
                         --net_uuid <NET_UUID>
   ```
6. Удалить списки контроля доступа с кластера **Sharx Base**

   ```
   ipam acl del --name <NAME>
   ```

   где `name` — имя списка контроля доступа.

ПЕРЕЧЕНЬ ДОПОЛНИТЕЛЬНЫХ СПИСКОВ КОНТРОЛЯ ДОСТУПА ACL

1. ACL, позволяющий использовать ICMP

   ```
   allow-ping:
     - rule:
         # Разрешить ICMP (ip protocol number 1)
         eth_type: 0x800
         ip_proto: 1
         actions:
           allow: True
     - rule:
         # Блокировать весь остальной трафик
   ```

```
      eth_type: 0x800
      actions:
        allow: False
```

2. ACL с блокировкой ICMP по критерию запрос/ответ

```
block-icmp-by-type:
  - rule:
      # Блокировать ICMP Echo Reply
      eth_type: 0x800
      ip_proto: 1
      icmpv4_type: 0
      actions:
        allow: False
  - rule:
      # Блокировать ICMP Echo Request
      eth_type: 0x800
      ip_proto: 1
      icmpv4_type: 8
      actions:
        allow: False
```

3. ACL для защиты от спуфинга (antispoofing)

```
antispoof_acl:
  - rule:
      # Разрешить ARP с IP-адресом ВМ (IP antispoofing)
      dl_type: 0x806
      arp_spa: $VM_interface_IP
      actions:
        allow: True
  - rule:
      # Разрешить ARP с MAC-адресом ВМ (MAC antispoofing)
      dl_type: 0x806
      arp_sha: $VM_interface_MAC
      actions:
        allow: True
  - rule:
      # Блокировать все остальные ARP-пакеты
      dl_type: 0x806
      actions:
        allow: False
```

4. ACL для разрешения всего трафика

```
permit_all_acl:
  - rule:
     # Разрешить весь трафик
     actions:
       allow: True
```

23.5. Правила размещения виртуальных машин

Sharx Base поддерживает гибкие правила выбора узлов при создании ВМ. Основные способы:

- указание конкретного узла `nodeName`;
- использование меток узлов `nodeSelector`;
- совместное размещение `affinity`, раздельное совмещение `antiAffinity` и предпочтительное размещение `colocation`.

> **Важно**
>
> Создавайте ВМ с помощью файлов YAML, так как часто возникают ошибки при внесении данных

23.5.1. Выбор узла по идентификатору `nodeName`

В YAML-файле создания ВМ в разделе `data` задайте параметр `nodeName` — идентификатор узла, на который нужно спланировать ресурс

```
nodeName: <uuid_node_where_place_vms>
```

ВМ будет размещена на узле с указанным идентификатором.

23.5.2. Использование меток узлов `nodeSelector`

> **Примечание**
>
> Подробная информация о метках приведена в статье Работа с метками

Позволяет указать пары `ключ-значение` для выбора узлов. Количество данных пар может быть не ограничено, но чаще всего используется только одна пара. ВМ будет размещена на узле, удовлетворяющем **всем** условиям.
Ниже рассмотрены случаи применения меток по умолчанию и пользовательских меток.

МЕТКА ПО УМОЛЧАНИЮ

По умолчанию каждый узел уже имеет метку `base`, которую можно применить в разделе `nodeSelector`.

Пример определения `nodeSelector` на основе метки `base`

```
nodeSelector:
  key: base
```

В данном случае ВМ будет размещена на менее загруженном узле виртуального кластера.

Пример файла с заданным `nodeSelector` с меткой по умолчанию:

- для **Libvirt** scheduler_request_nodeselector_libvirt.yaml.
- для **РСХД** scheduler_request_nodeselector_sp.yaml.

МЕТКИ ПОЛЬЗОВАТЕЛЯ

При вводе пользовательских меток ВМ будет размещена на узле, содержащем **все** метки.

Чтобы ввести правило выбора узла на основе меток, выполните следующие действия:

1. Просмотрите список меток, отданных во ВЦОД

   ```
   scheduler labels ns show [--filter <FILTER>]
   ```

   где `filter` — фильтрация меток. Возможные значения:

   - `a` — all, все метки;
   - `i` — ignored, игнорируемые метки;
   - `u` — unignored, все метки, кроме игнорируемых.

   > **Внимание**
   >
   > Пользователи могут видеть метки только ВЦОД, в котором они находятся.
   > При попытке запроса информации о других ВЦОД будет выдаваться ошибка.
   > Исключение — ВЦОД управления

2. Просмотрите группы совместимых меток в виртуальном кластере

   ```
   scheduler labels vcluster sharing --vcluster <VCLUSTER>
   ```

   В результате выполнения команды будут показаны группы меток, которые можно использовать вместе для успешного планирования пользовательских ресурсов на узлы кластера.

3. В YAML-файл создания ВМ в раздел `spec` добавьте параметр `nodeSelector`, содержащий комбинацию совместимых меток, например

```
nodeSelector:
  disktype: ssd
  vcluster: test
```

ВМ будет размещена на узле, содержащем метки `disktype=ssd` и `vcluster=test`.

23.5.3. `nodeAffinity`, `vmAffinity` и `colocation`

Планировщик **Scheduler** позволяет управлять размещением виртуальных машин в кластере с помощью правил **совместного** `affinity`, **раздельного** `antiAffinity` и предпочтительного `colocation` размещения. Эти правила задаются в YAML-файлах создания ВМ и включают:

- `nodeAffinity` — ограничение по меткам узлов.
- `vmAffinity`, `antiAffinity` — совместное или раздельное размещение ВМ
- `colocation` — совместное размещение связанных компонентов. Позволяет задавать более сложные условия выбора узлов с операторами и весами.

ОБЩИЕ ТИПЫ СТРАТЕГИЙ ПЛАНИРОВАНИЯ

Для `nodeAffinity` и `vmAffinity` применяются две стратегии:

Стратегия	Описание
`requiredDuringSchedulingIgnoredDuringExecution`	**Жесткое правило**. Если ни один узел не соответствует, ВМ не будет размещена
`preferredDuringSchedulingIgnoredDuringExecution`	**Мягкое правило**. Если узлы не найдены, ВМ будет размещена на более подходящих узлах

NODEAFFINITY

Правило `nodeAffinity` позволяет ограничить, на каких узлах может выполняться виртуальная машина. `nodeAffinity` — аналог `nodeSelector`, но более гибкий за счет применения стратегий.

При использовании `requiredDuringSchedulingIgnoredDuringExecution` планировщик будет выбирать только узлы, строго удовлетворяющие условиям. В режиме `preferredDuringSchedulingIgnoredDuringExecution` задается предпочтение: узлы с нужными метками получают бОльший вес при оценке, но при отсутствии таких узлов ВМ все равно создастся.

Для селекторов меток узлов `nodeAffinity` поддерживает операторы:

- `In` — значение присутствует;
- `NotIn` — значение отсутсвует;
- `Exist` — ключ существует;
- `DoesNotExist` — ключ отсутствует.

Например, можно указать

```
...
matchExpressions:
  - key: disktype
    operator: In
    values:
      - ssd
...
```

и выбор будет выполняться **только** среди узлов с меткой `disktype=ssd`.

Пример описания `nodeAffinity` из YAML-файла

```
...
    affinity:
      nodeAffinity:
        requiredDuringSchedulingIgnoredDuringExecution:
          nodeSelectorTerms:
            - matchExpressions:
              - key: az-name
                operator: NotIn
                values:
                  - az1
              - key: stage
                operator: DoesNotExist

              - key: disktype
                operator: In
                values:
                  - ssd

            - matchExpressions:
              - key: vcluster
                operator: In
                values:
                  - test
                  - test2

        preferredDuringSchedulingIgnoredDuringExecution:
          - weight: 100
            preference:
              matchExpressions:
                - key: az-name
                  operator: In
                  values:
                    - az2

                - key: vcluster
                  operator: In
                  values:
                    - test

                - key: vcluster
                  operator: Exist

          - weight: 70
            preference:
              matchExpressions:
                - key: security
                  operator: In
                  values:
                    - hi_secure

          - weight: 10
            preference:
              matchExpressions:
                - key: vcluster
                  operator: In
                  values:
```

```
                    - test
  ...
```

где возможные значения `operator`:

- `In` — значение присутствует;
- `NotIn` — значение отсутсвует;
- `Exist` — ключ существует;
- `DoesNotExist` — ключ отсутствует.

`vmAffinity`, `antiAffinity`, `colocation`

Правила `vmAffinity`, `antiAffinity` и `colocation` позволяют учитывать **расположение других ВМ** при планировании:

- совместное размещение ВМ `vmAffinity`;
- раздельное размещение ВМ `antiAffinity`;
- размещение ВМ с учетом уже размещенных ресурсов `colocation`.

Типы стратегий описаны в разделе выше.

СОВМЕСТНОЕ РАЗМЕЩЕНИЕ `vmAffinity`

`vmAffinity` определяет, что ВМ должна размещаться на том же узле, что и другие ВМ с нужными метками.

Пример применения правила представлен в файле YAML. Выберите файл в зависимости от типа хранилища.

Libvirt

scheduler_request_colocation_libvirt.yaml

в файле возможные значения `operator`:

- `In` — значение присутствует;
- `NotIn` — значение отсутствует;
- `Exist` — ключ существует;
- `DoesNotExist` — ключ отсутствует.

РСХД

scheduler_request_colocation_sp.yaml

в файле возможные значения `operator`:

- `In` — значение присутствует;
- `NotIn` — значение отсутствует;
- `Exist` — ключ существует;
- `DoesNotExist` — ключ отсутствует.

Разберем, как сработает планировщик на основе данного примера.

1. `nodeSelector`

    ```
    nodeSelector:
      key: base
      disktype: ssd
      vcluster: test2
    ```

    Это правило жестко ограничивает выбор узлов. Узел должен иметь все перечисленные метки: `key=base`, `disktype=ssd` и `vcluster=test2`.
    Если ни один узел в кластере не соответствует этим условиям, ВМ не будет размещена.

2. `vmAffinity`

    ```
    vmAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      - labelSelector:
          matchExpressions:
            - key: role
              operator: NotIn
              values:
                - web-db2
    ```

    Жесткое правило. ВМ не может быть размещена на том же узле, где уже есть другие ВМ с меткой `role=web-db2`.

    Далее

    ```
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 100
      vmAffinityTerm:
        labelSelector:
          matchExpressions:
            - key: authtor
              operator: In
              values:
                - dev
    ```

Мягкое предпочтение. ВМ желательно размещать рядом с другими ВМ, у которых метка `authtor=dev`. Если таких ВМ нет, это правило не мешает размещению.

3. Вывод

   Учитывая правила, ВМ будет размещена следующим образом:

   - планировщик сначала жестко отфильтрует узлы по `nodeSelector`. Узел должен иметь метки `key=base` и `disktype=ssd`, и `vcluster=test2`;
   - затем он исключит узлы, где есть ВМ с меткой `role=web-db2`;
   - среди оставшихся предпочтет те, где уже размещены ВМ с `authtor=dev`;
   - если ни один узел не соответствует всем жестким условиям, ВМ **не будет размещена**.

РАЗДЕЛЬНОЕ РАЗМЕЩЕНИЕ ВМ `antiAffinity`

`vmAntiAffinity` запрещает размещать ВМ вместе с другими ВМ с указанной меткой.

В примере ниже ВМ не будет размещена на узле, где есть ВМ с метками `authtor=dev` или `role=web-db2`.

```
...
affinity:

      vmAntiAffinity:
        requiredDuringSchedulingIgnoredDuringExecution:
          - labelSelector:
              matchExpressions:
                - key: authtor
                  operator: In
                  values:
                    - dev
                - key: role
                  operator: In
                  values:
                    - web-db2
...
```

ПРЕДПОЧТИТЕЛЬНОЕ РАЗМЕЩЕНИЕ РЯДОМ `colocation`

Пример предпочтительного размещения

```
apiVersion: v1
kind: api
metadata:
  plugin:
    scheduler:
```

```yaml
      request: add
spec:
  vcluster: "имя виртуального кластера"
  kind: vm
  name: "уникальное имя виртуальной машины"
  labels: "role=analytics,env=dev"
  descr: "описание виртуальной машины"
  data:

    affinity:

      vmAffinity:
        preferredDuringSchedulingIgnoredDuringExecution:
          - weight: 100
            vmAffinityTerm:
              labelSelector:
                matchExpressions:
                  - key: env
                    operator: In
                    values:
                      - dev
                  - key: role
                    operator: In
                    values:
                      - analytics

    nodeSelector:
      key: base
      disktype: ssd
      vcluster: test
    vms:
      - name: "уникальное имя виртуальной машины"
        descr: "Разместить рядом с другими analytics"
        vnc_passwd: "пароль для доступа по VNC"
        storage:
          sp:
            uuid: "идентификатор тома"
        networks: ["идентификатор сети, которая подключается к ВМ"]
        vram: "объем виртуальной памяти"
        vcpu: "количество выделяемых виртуальных процессоров"
```

Поведение планировщика:

- ВМ **желательно** разместить рядом с другими ВМ с `env=dev` и `role=analytics`.
- Если таких ВМ рядом нет, данная ВМ все равно будет размещена.

23.6. Оптимизация ресурсов ВМ

Глобальные параметры переподписки устанавливаются администратором кластера и по умолчанию применяются к каждой ВМ. Подробная информация описана в Руководстве администратора в командной строке.

При необходимости можно переопределить некоторые параметры переподписки индивидуально для ВМ.

23.6.1. Настройка переподписки ЦПУ

Переподписка ЦПУ позволяет выделить виртуальной машине больше ВЦПУ, чем физически доступно в кластере за счет разделения процессорного времени между несколькими ВЦПУ.

При установленных настройках Переподписки ЦПУ в кластере назначен глобальный коэффициент переподписки ЦПУ `default_cpu_overcommit_ratio`. Этот коэффициент применяется ко всем ВМ по умолчанию, но может быть переопределен для каждой виртуальной машины индивидуально.

При создании ВМ можно указать коэффициент переподписки `cpu_overcommit_ratio` в диапазоне `1-4`.

Пример создания ВМ с указанным коэффициентом переподписки ЦПУ

```
apiVersion: v1
kind: api
metadata:
  plugin:
    scheduler:
      request: add
spec:
  data:
    vms:
      - name: "vm"
        vcpu: 3
        cpu_overcommit_ratio: 2
        ...
```

Чем выше коэффициент переподписки ЦПУ, тем меньше доля ЦПУ на каждый ВЦПУ, то есть можно создать больше ВЦПУ.

Пример

ВМ с `vcpu: 3` и `cpu_overcommit_ratio: 2`. Каждый ВЦПУ использует `1 / 2 = 0.5` физического ЦПУ. Таким образом ВМ будет потреблять `3 x 0.5 = 1.5` ЦПУ

23.6.2. Настройка переподписки ОЗУ

Переподписка ОЗУ позволяет выделить ВМ больше виртуальной памяти, чем физически доступно в кластере, при условии, что в кластере настроена и включена переподписка ОЗУ.

При установленных настройках Переподписки ОЗУ в кластере назначено глобальное значение толерантности ВМ к переподписке `default_overcommit_tolerance`. Этот коэффициент применяется ко всем ВМ по умолчанию, но может быть переопределен для каждой виртуальной машины индивидуально.

При создании ВМ можно указать толерантность ВМ к переподписке `overcommit_tolerance` в диапазоне от `0-3`.

Правила работы коэффициента переподписки ОЗУ:

- ВМ с `overcommit_tolerance` больше `0` становится «донором памяти»;
- при достижении порога срабатывания балунинга ВМ начинает отдавать часть своей памяти другим ВМ;
- чем выше `overcommit_tolerance`, тем больше памяти может отдать ВМ;
- при `overcommit_tolerance: 0` ВМ не отдает свою память.

Пример создания ВМ с указанной толерантностью к переподписке ОЗУ

```yaml
apiVersion: v1
kind: api
metadata:
  plugin:
    scheduler:
      request: add
spec:
  data:
    vms:
      - name: "vm"
        vcpu: 3
        overcommit_tolerance: 2
        ...
```

23.7. Операции с виртуальными машинами

> **Примечание**
>
> Действия выполняются пользователем с ролью **Администратор ВЦОД** или **Разработчик ВМ**

Пользовательский ресурс — виртуальная машина, создаваемая на виртуальном кластере.

23.7.1. Создать ВМ

Создание ВМ возможно двумя способами: с помощью YAML-файлов или через командную строку. Первый вариант предпочтительнее, так как он минимизирует вероятность ошибок.

СОЗДАТЬ ВМ С ПОМОЩЬЮ YAML-ФАЙЛА

1. Создайте файл формата YAML.
2. Опишите в файле все необходимые параметры и правила для создания ВМ. Подробная информация описана в статье Правила размещения виртуальных машин.
3. При необходимости укажите IP-адрес из пула доступных адресов:

   a. Чтобы просмотреть доступные адреса, введите команду

   ```
   ipam addr free list --uuid <UUID>
                       [--full_output <FULL_OUTPUT>]
   ```

   где

   - `uuid` — идентификатор сети или подсети;
   - `--full_output` — отобразить весь список адресов. Без указания `--full_output` по умолчанию отобразятся только три доступных адреса.

   b. Чтобы просмотреть занятые адреса, введите

   ```
   ipam addr allocated list --uuid <UUID>
   ```

   c. Добавьте в YAML

   ```
   networks: ["идентификатор сети"]
   ip_addresses: ["{\"идентификатор сети\": \"IP-адрес из пула доступных адресов\"}"]
   ```

4. При необходимости ограничьте трафик для сетевых интерфейсов

   Пример для ВМ с одним сетевым интерфейсом

   ```
   networks: ["идентификатор сети"]
   inbounds:  ["лимит входящего трафика"]
   outbounds: ["лимит исходящего трафика"]
   ```

   Пример для ВМ с двумя сетевыми интерфейсами

   ```
   networks: ["идентификатор сети 1", "идентификатор сети 2"]
   inbounds:  ["-", "лимит входящего трафика сети 2"]
   outbounds: ["лимит исходящего трафика сети 1", "-"]
   ```

   Для ВМ с двумя сетевыми интерфейсами ограничения указываются в том же порядке, что и интерфейсы в `networks`, значение `"-"` означает отсутствие ограничения на сетевой интерфейс.

5. При необходимости определите индивидуальные параметры переподписки ЦПУ и ОЗУ.

6. Сохраните файл.
7. Загрузите файл на Платформу.

   Чтобы загрузить файлы в Sharx Base, введите команду

   ```
   resource --spec <path_to_yaml_file>
   ```

   где `spec` — расположение YAML-файла.

   При локальном расположении на АРМ пользователя введите путь до файла.

   При расположении в стороннем репозитории укажите ссылку на данный файл.
8. После загрузки проверьте список всех созданных ВМ

   ```
   scheduler request list --vcluster <VCLUSTER>
   ```

Пример

Примеры файлов создания ВМ:

- для хранилища **Libvirt** с `nodeSelector`, указанием IP-адреса и ограничением на единственный сетевой интерфейс scheduler_request_libvirt.yaml.
- для хранилища **РСХД** с `nodeSelector`, указанием IP-адресов и ограничением только на второй сетевой интерфейс scheduler_request_sp.yaml.

СОЗДАТЬ ВМ ЧЕРЕЗ КОМАНДНУЮ СТРОКУ

Чтобы создать ВМ через командную строку, введите

```
scheduler request add --vcluster <VCLUSTER>
                      [--kind <KIND>]
                      --name <NAME>
                      [--descr <DESCR>]
                      [--labels <LABELS>]
                      --data <DATA>
```

где

- `vcluster` — имя виртуального кластера;
- `kind` — тип запроса. По умолчанию `vm`;
- `name` — имя ВМ;
- `descr` — описание виртуальной машины;

- `labels` — метки ВМ;
- `data` — определение спецификаций.

23.7.2. Просмотреть ВМ

1. Просмотреть список всех созданных ВМ

   ```
   scheduler request list --vcluster <VCLUSTER>
   ```

   где `vcluster` — имя виртуального кластера.

2. Просмотреть информацию о конкретной ВМ

   ```
   scheduler request show --vcluster <VCLUSTER>
                         [--kind <KIND>]
                          --name <NAME>
                         [--show_uuid <SHOW_UUID>]
                         [--metrics <METRICS>]
   ```

   где

   - `vcluster` — имя виртуального кластера;
   - `kind` — тип запроса. По умолчанию `vm`;
   - `name` — имя ВМ;
   - `show_uuid` — показать идентификатор объекта;
   - `metrics` — получить метрики ВМ. Возможные значения:
     - `yes` — включить получение метрик,
     - `no` — отключить получение метрик.

3. Просмотреть статус ВМ

   ```
   scheduler request status --vcluster <VCLUSTER>
                           [--kind <KIND>]
                           [--name <NAME>]
   ```

   где

   - `vcluster` — имя виртуального кластера;
   - `kind` — тип запроса. По умолчанию `vm`;
   - `name` — имя ВМ.

4. Просмотреть логины и роли пользователей, имеющих доступ к ВМ

```
scheduler request rights --vcluster <VCLUSTER>
                         [--kind <KIND>]
                         --name <NAME>
```

23.7.3. Обновить ВМ

```
scheduler request update --vcluster <VCLUSTER>
                         [--kind <KIND>]
                         --name <NAME>
                         [--descr <DESCR>]
                         [--labels <LABELS>]
                         --data <DATA>
```

где

- `vcluster` — имя виртуального кластера;
- `kind` — тип запроса. По умолчанию `vm`;
- `name` — имя ВМ;
- `descr` — описание виртуальной машины;
- `labels` — метки ВМ;
- `data` — спецификация.

23.7.4. Изменить статус ВМ

```
scheduler request state --vcluster <VCLUSTER>
                        [--kind <KIND>]
                        --name <NAME>
                        --state <STATE>
```

где

- `vcluster` — имя виртуального кластера;
- `kind` — тип запроса. По умолчанию `vm`;
- `name` — имя ВМ, у которой изменяется статус;
- `state` — статус ВМ. Возможные значения:
    - `START` — запуск ВМ,
    - `STOP` — остановка ВМ,
    - `RECREATE` — восстановление ВМ.

23.7.5. Назначить метку СКЗИ виртуальной машине

Метка СКЗИ или **криптографическая метка** применяется к ВМ, чтобы обозначить ее принадлежность к контуру криптографической защиты. Метка имеет ключ-значение `vm=crypto`. Для ВМ с меткой СКЗИ блокируются функции миграции и создания снимков, чтобы исключить риск утечки защищаемых данных.

> **Внимание**
>
> Криптографическую метку может назначить или снять только определенная роль — Администратор СКЗИ ВЦОД. Эта роль находится в конфигурации ВЦОД 'cb_virt' и будет оиисана в следующих версиях документации

1. Чтобы назначить метку СКЗИ виртуальной машине, введите команду

   ```
   scheduler request cips label add --vcluster <VCLUSTER>
                                    --name <NAME>
   ```

   где

   - `vcluster` — имя виртуального кластера;
   - `name` — имя ВМ.

2. Чтобы удалить метку СКЗИ с виртуальной машины, введите команду

   ```
   scheduler request cips label del --vcluster <VCLUSTER>
                                    --name <NAME>
   ```

23.7.6. Удалить ВМ

Чтобы удалить ВМ, введите

```
scheduler request del --vcluster <VCLUSTER>
                      [--kind <KIND>]
                      --name <NAME>
                      [--backup <BACKUP>]
```

где

- `vcluster` — имя виртуального кластера;
- `kind` — тип запроса. По умолчанию `vm`;
- `name` — имя ВМ;

- `backup` — создать снимок удаляемой ВМ. Снимок удалится через 1 день. Возможные значения `backup` — `yes`, `no`.

23.8. Запустить виртуальные машины

> **Примечание**
>
> Действия выполняются пользователем с ролью **Администратор ВЦОД** или **Разработчик ВМ**

Чтобы запустить ВМ, выполните команду

```
scheduler request state --vcluster <VCLUSTER>
                        [--kind <KIND>]
                        --name <NAME>
                        --state START
```

где

- `vcluster` — имя виртуального кластера;
- `kind` — тип запроса. По умолчанию `vm`;
- `name` — имя ВМ, у которой изменяется статус;
- `state` — статус ВМ. Возможные значения:
  - `START` — запуск ВМ,
  - `STOP` — остановка ВМ,
  - `RECREATE` — восстановление ВМ.

23.9. Снимки ВМ и томов

> **Примечание**
>
> Действия выполняются пользователем с ролью **Администратор ВЦОД** или **Разработчик ВМ**

Снимок — моментальный снимок тома или ВМ. Копия на уровне блоков физических или виртуальных систем, выполненная без остановки системных служб. Включает в себя структуру директорий, файлов и информацию о состоянии системы на фиксированный момент времени. Снимок не является резервной копией, применяется или используется как временный источник для создания согласованных резервных копий.

Важно

При удалении виртуальной машины снимки удаляются вместе с ней.

23.9.1. Снимки ВМ

Внимание

У ВМ, развернутой с использованием Libvirt-тома, может быть не более одного снимка. Создание снимка ВМ с меткой СКЗИ невозможно

1. Чтобы создать снимок ВМ, введите

```
scheduler request snapshot add  --vcluster <VCLUSTER>
                                [--kind <KIND>]
                                --name <NAME>
                                --snapshot_name <SNAPSHOT_NAME>
                                [--vms <VMS>]
                                [--descr <DESCR>]
```

   где

   - `vcluster` — имя виртуального кластера;
   - `kind` — тип запроса. По умолчанию `vm`;
   - `name` — имя пользовательского ресурса;
   - `snapshot_name` — имя снимка;
   - `vms` — список ВМ;
   - `descr` — описание снимка.

2. Просмотреть список снимков ВМ

```
scheduler request snapshot list --vcluster <VCLUSTER>
                                [--kind <KIND>]
                                --name <NAME>
```

3. Просмотреть информацию о конкретном снимке

```
scheduler request snapshot show --name <NAME>
                                [--kind <KIND>]
                                --vcluster <VCLUSTER>
                                --uuid <UUID>
```

4. Чтобы восстановить ВМ из снимка, введите

```
scheduler request snapshot revert --vcluster <VCLUSTER>
                                  [--kind <KIND>]
                                  --name <NAME>
                                  --uuid <UUID>
                                  [--vms <VMS>]
```

где

- `vcluster` — имя виртуального кластера;
- `kind` — тип запроса. По умолчанию `vm`;
- `name` — имя пользовательского ресурса;
- `uuid` — идентификатор снимка;
- `vms` — список ВМ.

5. Удалить снимок ВМ

```
scheduler request snapshot del --name <NAME>
                               [--kind <KIND>]
                               --vcluster <VCLUSTER>
                               --uuid <UUID>
```

23.9.2. Снимки томов Libvirt

1. Чтобы создать снимок тома Libvirt, введите команду

```
storage libvirt snapshot add [--uuid <UUID>]
                             [--name <NAME>]
                             --vm_name <VM_NAME>
                             [--disks <DISKS>]
                             [--descr <DESCR>]
                             [--labels <LABELS>]
                             --vcluster_name <VCLUSTER_NAME>
```

где

- `uuid` — идентификатор снимка;
- `name` — имя снимка;
- `vm_name` — имя ВМ;
- `disks` — имя тома для снимка. При отсутствии параметра `disks` выполнится снимок всех томов, прикрепленных к данной ВМ;
- `descr` — описание снимка;

- `labels` — метка снимка;
- `vcluster_name` — имя виртуального кластера.

2. Просмотреть подробную информацию о конкретном снимке ВМ

```
storage libvirt snapshot show [--name <NAME>]
                              --vm_name <VM_NAME>
                              --vcluster_name <VCLUSTER_NAME>
```

где

- `name` — имя снимка;
- `vm_name` — имя ВМ;
- `vcluster_name` — имя виртуального кластера.

3. Чтобы удалить снимок, введите команду

```
storage libvirt snapshot del [--name <NAME>]
                              --vm_name <VM_NAME>
                              --vcluster_name <VCLUSTER_NAME>
```

где

- `name` — имя снимка;
- `vm_name` — имя ВМ;
- `vcluster_name` — имя виртуального кластера.

23.9.3. Снимки томов РСХД

1. Чтобы создать снимок тома, введите команду

```
storage sp snapshot add --name <NAME>
                        [--descr <DESCR>]
                        --volume_uuid <VOLUME_UUID>
```

где

- `name` — имя тома;
- `descr` — описание снимка;
- `volume_uuid` — идентификатор тома.

2. Просмотреть список всех снимков в пределах тома

```
storage sp snapshot list [--volume_uuid <VOLUME_UUID>]
```

где `volume_uuid` — идентификатор тома.

3. Просмотреть подробную информацию о конкретном снимке

```
storage sp snapshot show --uuid <UUID>
```

где `uuid` — идентификатор снимка.

4. Обновить параметры снимка

```
storage sp snapshot update --uuid <UUID>
                           [--name <NAME>]
                           [--descr <DESCR>]
                           [--delete_after <DELETE_AFTER>]
```

где

- `uuid` — идентификатор снимка;
- `name` — имя снимка;
- `descr` — описание снимка;
- `delete_after` — удалить снимок через N секунд. Установите период удаления относительно текущего времени. При отсутствии параметра снимок автоматически удаляться не будет.

5. Удалить снимок

```
storage sp snapshot del --uuid <UUID>
```

где `uuid` — идентификатор снимка.

23.9.4. Снимки томов NFS

1. Чтобы создать снимок тома, введите команду

```
storage nfs snapshot add --name <NAME>
                         --vm_name <VM_NAME>
                         --vcluster_name <VCLUSTER_NAME>
                         [--uuid <UUID>]
                         [--descr <DESCR>]
                         [--disks <DISKS>]
                         [--labels <LABELS>]
```

где

- `name` — имя создаваемого снимка;
- `vm_name` — имя ВМ;

- `vcluster_name` — имя виртуального кластера;
- `uuid` — идентификатор снимка;
- `descr` — описание снимка;
- `disks` — идентификатор или список идентификаторов томов у данной ВМ, для которых нужно создать снимок. По умолчанию снимки создаются для всех томов ВМ;
- `labels` — метки снимка.

2. Просмотр списка всех снимков в пределах виртуального кластера

```
storage nfs snapshot list --vcluster_name <VCLUSTER_NAME>
                          [--vm_name <VM_NAME>]
                          [--request_name <REQUEST_NAME>]
```

где `request_name` — имя ВМ.

3. Просмотр подробной информации о конкретном снимке

```
storage nfs snapshot show --uuid <UUID>
```

4. Удалить снимок

```
storage nfs snapshot del --uuid <UUID>
```

23.10. Резервные копии виртуальных машин

> **Примечание**
>
> Действия выполняются пользователем с ролью **Администратор ВЦОД** или **Разработчик ВМ**

Том с данными резервной копии ВМ находится в пуле, которому принадлежат исходные тома.
XML-данные резервной копии хранятся в БД и содержат полное описание конфигурации виртуальной машины.

Команды для резервного копирования:

1. Чтобы создать резервную копию диска, введите в командной строке

```
storage libvirt backup add --vm_name <VM_NAME>
                           [--sc_node <SC_NODE>]
                           [--disks <DISKS>]
                           [--timeout <TIMEOUT>]
```

где

- `vm_name` — имя ВМ;
- `sc_node` — идентификатор узла, на котором находится ВМ;
- `disks` — том ВМ для резервного копирования. При отсутствии параметра `disks` выполнится резервное копирование всех доступных томов ВМ;
- `timeout` — время ожидания для резервного копирования.

2. Просмотреть список всех **выполняемых** резервных копирований

> **Внимание**
>
> Данная команда доступна только Администратору ВЦОД

```
storage libvirt backup list [--sc_node <SC_NODE>]
```

где `sc_node` — идентификатор узла, на котором находится ВМ.

После успешного выполнения резервного копирования информация об операции автоматически удаляется из перечня и не отображается в результате данной команды.

3. Просмотреть подробную информации о статусе выполняемой операции

```
storage libvirt backup show [--name <VM_NAME>]
                            [--sc_node <SC_NODE>]
```

где

- `name` — имя ВМ;
- `sc_node` — идентификатор узла, на котором находится ВМ. Введите параметр `sc_node`, если запрос производится не с того узла, на котором находится запрашиваемая ВМ.

После успешного выполнения резервного копирования информация об операции автоматически удаляется из перечня и не отображается в результате данной команды.

4. В случае некорректной отработки операции резервного копирования можно удалить эту операцию из истории

```
storage libvirt backup del [--name <VM_NAME>]
                           [--sc_node <SC_NODE>]
```

где

- `name` — имя ВМ;
- `sc_node` — идентификатор узла, на котором находится ВМ.

Затем повторите резервное копирование, указав корректные параметры.

5. Получить XML-описание резервной копии ВМ

```
storage libvirt backup lookup  --name <VM_NAME>
```

где `name` — имя виртуальной машины.

> **Пример**
>
> Пример XML-описания резервной копии ВМ domain_backup.xml

23.11. Восстановить виртуальную машину

> **Примечание**
>
> Действия выполняются пользователем с ролью **Администратор ВЦОД** или **Разработчик ВМ**

Восстановление ВМ возможно при наличии резервной копии.
Команды резервного копирования подробно описаны в статье Резервные копии ВМ.

Механизм восстановления ВМ из резервной копии состоит из шагов, описанных ниже.

23.11.1. Подготовить резервную копию

1. Создайте резервную копию ВМ.
2. Если резервное копирование прошло успешно:
   a. Информация о совершенной операции автоматически удаляется из перечня и не отображается при выполнении команд `storage libvirt backup list` и `storage libvirt backup show`.
   b. В пуле хранилища появляются тома с данными резервной копии.
   c. Описание резервной копии в виде XML можно посмотреть командой

   ```
   storage libvirt backup lookup  --name <vm_name>
   ```

3. Если резервное копирование прошло неуспешно:
   a. При выполнении команд `storage libvirt backup list` и `storage libvirt backup show` висит информация об ошибке.

b. Удалите эту операцию из истории.

c. Затем повторите резервное копирование, указав корректные параметры.

23.11.2. Восстановить ВМ

> **Примечание**
>
> Том с данными резервной копии ВМ находится в пуле, которому принадлежат исходные тома. XML-данные резервной копии хранятся в БД и содержат полное описание конфигурации виртуальной машины

Чтобы восстановить ВМ из резервной копии, необходимо:

1. Получить XML-описание резервной копии ВМ

   ```
   storage libvirt backup lookup  --name <VM_NAME>
   ```

   где `name` — имя виртуальной машины.

   > **Пример**
   >
   > Пример XML-описания резервной копии ВМ domain_backup.xml

2. Создать YAML-файл для создания ВМ с параметрами из XML.

   > **Пример**
   >
   > YAML-файл создания ВМ request_libvirt_recovery.yaml

3. Загрузить файл на Платформу

   Чтобы загрузить файлы в Sharx Base, введите команду

   ```
   resource --spec <path_to_yaml_file>
   ```

   где `spec` — расположение YAML-файла.

   При локальном расположении на АРМ пользователя введите путь до файла.

   При расположении в стороннем репозитории укажите ссылку на данный файл.

4. После загрузки проверить список всех созданных ВМ

```
scheduler request list --vcluster <VCLUSTER>
```

5. Подробная информация о статусе конкретной ВМ

```
scheduler request show --name <NAME> --vcluster <VCLUSTER>
```

где

- `vcluster` — имя виртуального кластера;
- `name` — имя ВМ.

В результате действий будет создана виртуальная машина, аналогичная по данным и конфигурации восстанавливаемой ВМ.

24. Перевести узлы в сервисный режим

Сервисное обслуживание позволяет временно отключить отдельные узлы от процесса планирования ресурсов. Эта функция полезна при проведении технических работ с оборудованием.

При активации сервисного режима каждому узлу присваивается специальная метка `system_drain_node=yes`. После этого узел перестает быть доступным для размещения на нем новых ресурсов.

24.1. Перевести узлы в сервисный режим

Чтобы перевести узлы в сервисный режим, выполните команду

```
scheduler drain add --nodes <NODES>
```

где `nodes` — список идентификаторов узлов, которые требуется вывести из эксплуатации.

После выполнения команды:

- Узлам автоматически присваивается метка `system_drain_node=yes`.
- Узлы исключаются из размещения новых ресурсов.

24.2. Отключить сервисный режим

```
scheduler drain del --nodes <NODES>
```

После выполнения команда метка `system_drain_node` удаляется, и узлы возвращаются в общее планирование.

25. Мониторинг

В Sharx Base реализован мониторинг системы на основе взаимодействия с **экспортерами**, установленными на каждом узле кластера.

Экспортеры — сервисы, которые собирают метрики на узлах и предоставляют к ним доступ на определенном HTTP-порту.
В текущей версии Sharx Base для сбора показателей используются следующие экспортеры:

- **sdc-domain-exporter** — состояние виртуальных машин;
- **node_exporter_sharx** — показатели узлов;
- **ipmi_exporter_sharx** — состояние оборудования;
- **sp_exporter_sharx** — показатели хранилища РСХД;
- **disks_exporter_sharx** — метрики дисков.

Предустановки для мониторинга выполняются сотрудниками технической поддержки.

Настройка экспортеров выполняется **Администратором кластера**.

Администратор ВЦОД может работать с шаблонами, которые используются для извлечения информации о метриках из экспортера. Доступные действия **Администратора ВЦОД** описаны ниже:

1. Просмотреть список созданных шаблонов

```
metrics template list
```

2. Просмотреть подробную информацию о конкретном шаблоне

```
metrics template show --name <template_name>
```

3. Отобразить метрики по фильтрам, указанным в шаблоне

```
metrics template execute --name <template_name>
```