Sharx Base 6.2

Руководство администратора

SharxDC

Оглавление

1. Введение

Руководство администратора описывает последовательность действий пользователя с ролью **Администратор кластера** в Sharx Base 6.2 для управления виртуальной инфраструктурой:

- ВЦОД управления;
- ВЦОД;
- пользователями и ролями;
- IPAM, сетями, хранилищами и другими настройками.

Некоторые операции доступны пользователю с ролью **Администратор безопасности кластера** в пределах его прав доступа.

Функции Администратора кластера и Администратора безопасности кластера описаны в статье Права доступа и роли.

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.

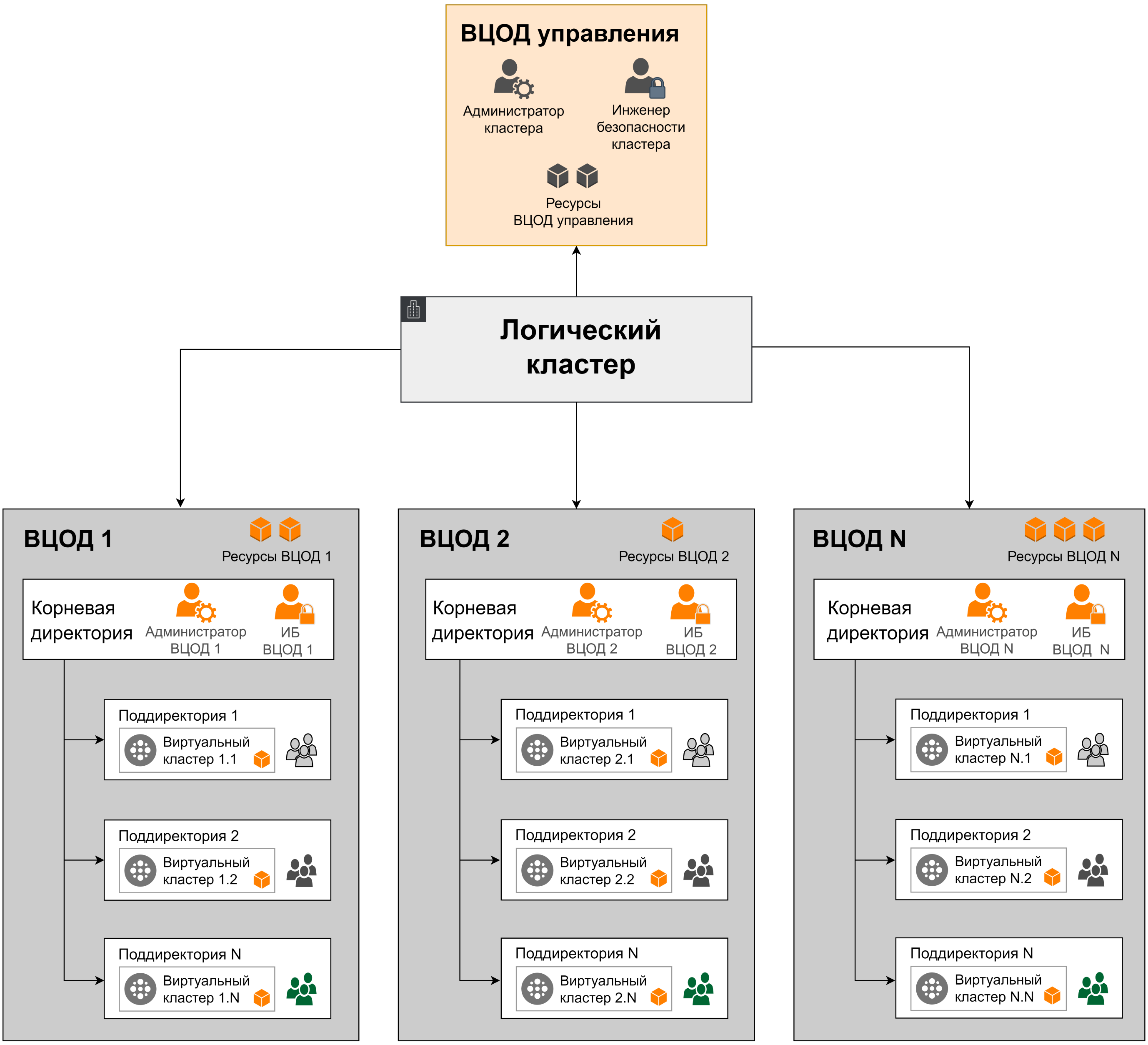
Sharx Base является гипервизором 1 типа, то есть включает все необходимые системные и прикладные компоненты для развертывания на аппаратных средствах без необходимости установки дополнительного ПО. Доступ к прикладным компонентам предоставляется пользователям и администраторам Sharx Base через графический 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.



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.

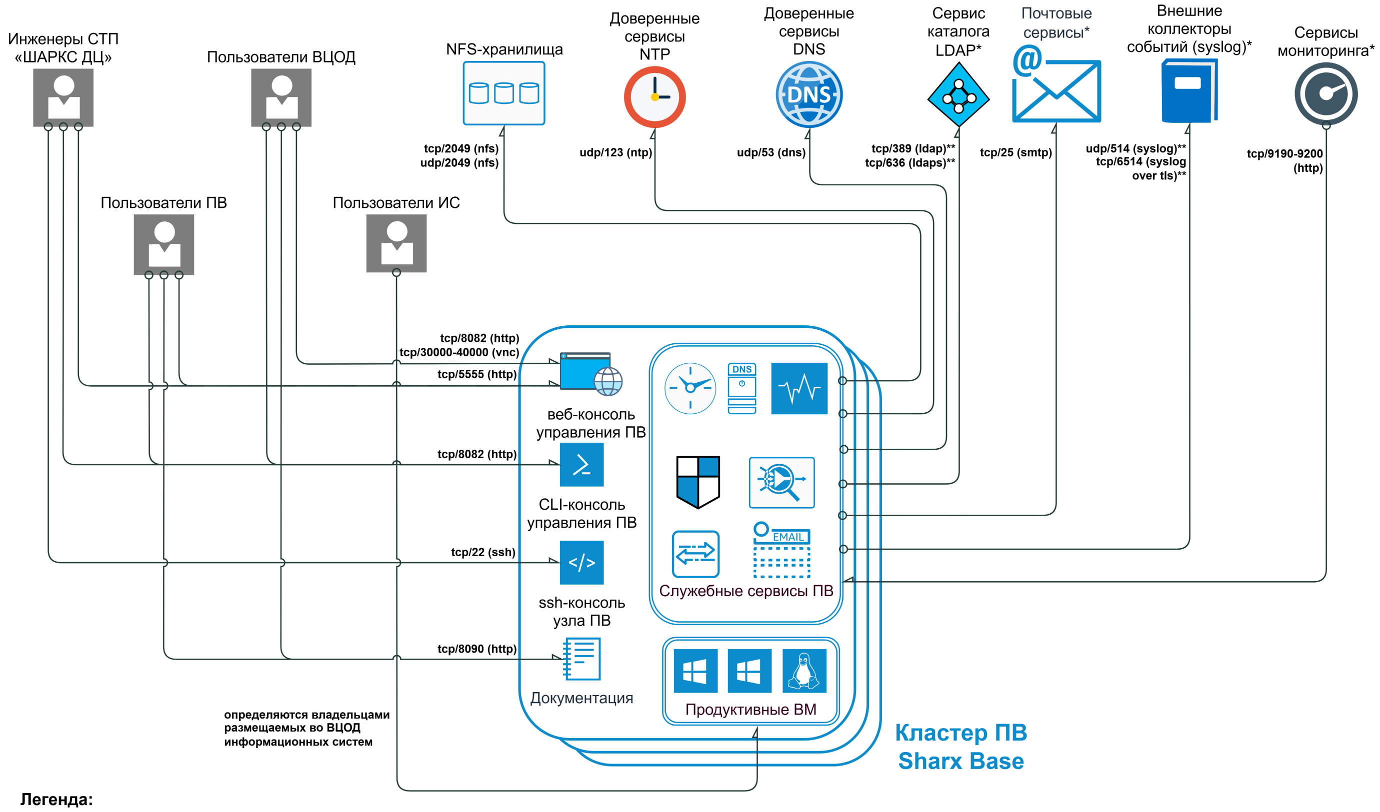
Подсеть РСХД используется для функционирования распределенного хранилища данных. Выделите VLAN и IP-адрес сети. Немаршрутизируемый VLAN. Трафик тегированный.
Для функционирования распределенной системы хранения данных (РСХД) используется RDMA и MTU 9000.
Не рекомендуется размещать в подсети РСХД узлы, 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.



Легенда:
* - настройка опциональна
** - протокол/порт по умолчанию, возможна смена значений на необходимые

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

Процесс настройки кластера разделен на обязательные шаги, без которых он не будет функционировать, и дополнительные, которые помогают оптимизировать и расширить функциональность кластера.

4.1. Обязательные шаги

1. Аутентифицируйтесь в кластере.
2. Просмотрите информацию об оборудовании кластера.
3. Настройте метки кластера.
   Данный шаг не является обязательным, но оптимально выполнить его на данном этапе для базовых меток.
4. Создайте параметры IPAM.
5. Настройте виртуальные сети кластера.
6. Настройте параметры переподписки.
7. Создайте ВЦОД.
8. Настройте необходимые хранилища:
   - РСХД: группы хранения и шаблоны хранения.
   - NFS: настроить подключение.

4.2. Дополнительные шаги

Эти шаги можно выполнить после прохождения обязательных шагов настройки кластера.

4.2.1. Оптимизация и организация

1. Настройте ресурсы кластера.
2. Настройте структуру директорий.
3. Настройте уведомления для отслеживания состояния кластера и ВЦОД управления.

4.2.2. Детализация прав доступа

1. Настройте права доступа.
2. Настройте роли.
3. Настройте пользователей ВЦОД управления.

4.2.3. Настройки безопасности

1. Проверьте политику безопасности учетных записей.
2. Настройте валидацию объектов ВЦОД.
3. Выполните резервное копирование.
4. Выгрузите события журнала безопасности.

5. Настройте логирование.
6. Изучите мониторинг кластера.

4.2.4. Обслуживание кластера

1. Управляйте сервисным режимом узлов кластера.
2. Обновите компоненты до актуальных версий.
3. Расширяйте кластер или изменяйте состав узлов.

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. Далее введите

   ```
   login <admin_login> --cluster <cluster_name> --ns admins
   ```

   где

   - `login` — логин администратора, предоставленный поставщиком;
   - `cluster` — имя логического кластера;

- `ns` — имя ВЦОД управления.

3. В случае успешного входа отобразится перечень плагинов, установленных в кластере.

5.1. Использование TLS-соединения для взаимодействия с Sharx Base

TLS-соединение защищает передачу данных между узлами клиентской и серверной машины за счет криптографического шифрования информации.

Возможность использования TLS для взаимодействия с **Sharx Base** включает поставщик при сборке кластера.
После включения клиенту предоставляется файл `cert_name.crt` для дальнейшей настройки.

Чтобы настроить TLS для взаимодействия с Sharx Base, выполните следующие действия:

1. Поместите файл `cert_name.crt`, предоставленный поставщиком, в хранилище доверенных сертификатов

   ОС Debian

   Директория `/usr/local/share/ca-certificates/`

   ОС AlmaLinux

   Директория `/etc/pki/ca-trust/source/anchors/`

2. Обновите сертификаты

   ОС Debian

   Выполните команду

   ```
   update-ca-certificates
   ```

   ОС AlmaLinux

   Выполните команду

   ```
   update-ca-trust
   openssl x509 -in /etc/pki/ca-trust/extracted/openssl/ca-bundle.trust.crt -text | less
   ```

3. После выполненных действий вход в Sharx Base должен осуществляться с использованием флагов `--tls_verify` и `--tls`

   ```
   sdc-cli --tls_verify <path_to_cert_key> --tls yes
   ```

где `tls_verify` — путь к файлу сертификата.

---

6. ВЦОД управления

После аутентификации **Администратор кластера** находится во **ВЦОД управления**.

ВЦОД управления — пространство для управления доступными ресурсами (лимитами) логического кластера, используется только для настройки других пространств (ВЦОД) и не имеет собственной виртуализации (виртуальных кластеров и ВМ).

ВЦОД управления создается технической поддержкой при подготовке кластера.

Во ВЦОД управления по умолчанию созданы две роли:

- **Администратор кластера**;
- **Администратор безопасности кластера**.

Затем администратор кластера с помощью конфигурационного файла создает ВЦОД.

Созданный 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.

Действия по управлению ВЦОД и его виртуальными объектами выполняет пользователь с ролью Администратор ВЦОД. Описание действий приведены в отдельных разделах портала документации: в Руководстве пользователя в командной строке и в Руководстве пользователя в веб-интерфейсе.

Чтобы посмотреть информацию и статус ВЦОД управления, введите

```
aaa namespace show --cluster <cluster_name> --ns admins
```

> **Примечание**
>
> Действия для создания ВЦОД описаны в статье Создать ВЦОД

7. Информация об оборудовании кластера

Администратор кластера может получить актуальную информацию о физических ресурсах кластера. Обновление данных происходит регулярно. Полученная информация собирается и фильтруется для каждого отдельного логического кластера.

Чтобы просмотреть текущие данные о ресурсах конкретного узла, выполните команду

```
hardware event list --event_id <EVENT_ID>
                    --server_id <SERVER_ID>
                    [--limit <LIMIT>]
```

где

- `event_id` — тип сущности или атрибута, который необходимо отобразить. Возможные значения:
    - `baseboardlibvirt` — характеристики платы,
    - `cpulibvirt` — характеристики процессора,
    - `cpulibvirtinfo` — доступные ресурсы узла,
    - `cpudmi` — частота шины,
    - `ramlibvirt` — информация об объеме ОЗУ,
    - `ramutil` — использование ОЗУ,
    - `network` — подключенные сетевые интерфейсы,
    - `mnt` — точка монтирования;
- `server_id` — идентификатор узла, по которому выполняется запрос;
- `limit` — максимальное количество типов сущностей или атрибутов `event` для отображения.

8. Метки

Метки (labels) — пары ключ-значение формата `key=value`, которые описывают атрибуты объектов кластера. Они играют ключевую роль в механизме планирования размещения пользовательских ресурсов (ВМ) на узлах. Метки могут быть назначены в кластере, ВЦОД и виртуальном кластере.
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.
- **Делегирование метки** — предоставление возможности использовать метку объектам: ВЦОД или виртуальному кластеру.
- **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.
- **Ограничение** `taint` — свойство узла, не позволяющее планировать на него ресурсы, у которых нет допуска (toleration) к узлу с данным ограничением `taint`.
- **Метка-ограничение** — метка со свойством `taint`. На узлы с такой меткой ресурсы планируются по определенным правилам.
- **Допуски** `tolerations` — это параметры виртуальных машин, которые позволяют им размещаться на узлах с ограничениями. Допуск должен соответствовать введенному ограничению `taint` на узле `key=value:effect`.
- **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 Сервисное обслуживание
- **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. Подробная информация описана в Руководстве пользователя в командной строке.

8.1. Принципы выбора объекта для размещения ресурсов по меткам

> ❌ **Внимание**
>
> - 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

Планировщик размещает ресурсы только на узлах, содержащих **все** метки, указанные в запросе на создание ресурса в элементе `nodeSelector`. Игнорируемые метки **не участвуют в фильтрации узлов**.

Например, узлам A, B и C присвоены метки

node	labels
A	key=base, disk=ssd, graphics=fast, foo=bar
B	key=base
C	key=base, disk=ssd

Получен запрос на создание пользовательского ресурса

nodeSelector из запроса на создание ресурса

```
nodeSelector:
    key: base
    disk: ssd
    graphics: fast
```

Только узел A будет выбран планировщиком доступным для размещения ресурса, так как он единственный содержит **все** запрошенные метки.

8.2. Принцип размещения пользовательских ресурсов на узлы, содержащие метку-ограничение

При подборе узлов, где существует метка-ограничение, планировщик следует определенному алгоритму:

1. Если метка-ограничение не была указана пользователем в параметре `nodeSelector`, ресурс не будет запланирован на узлы, имеющие эту метку.
2. Если метка-ограничение была указана пользователем в параметре `nodeSelector`, ресурс автоматически получит допуск к размещению на узле, содержащем данную метку-ограничение.

Например, узлам A, B и C присвоены метки

node	labels
A	key=base, foo=bar:{taint}
B	key=base
C	key=base

где метка `foo=bar:{taint}` — метка-ограничение.

Получен запрос на создание пользовательского ресурса

nodeSelector из запроса на создание ресурса

```
nodeSelector:
  key: base
```

Планировщик выберет только узлы B и C, так как они не имеют метку-ограничение.

Если получен запрос с параметром `nodeSelector` вида

nodeSelector из запроса на создание ресурса

```
nodeSelector:
  foo: bar
```

Планировщик выберет только узел A, так как он единственный имеет метку-ограничение `foo=bar:{taint}`.

8.3. Управлять метками в кластере

8.3.1. Ограничения работы с метками в кластере

- При создании кластера каждому узлу по умолчанию присваивается метка `key=base`.
- Максимальное количество меток на узле — 150.
- Узлу нельзя назначить несколько меток с одинаковым ключом.
- С узла нельзя открепить все метки.
- Нельзя удалить метки, делегированные во ВЦОД и в виртуальные кластеры.

- Метки можно создавать и редактировать на узлах до создания первого ВЦОД (не считая ВЦОД управления). За исключением ситуаций, когда в кластер добавляется новый узел, и на нем еще нет меток.
- Из обычной метки можно сделать метку-ограничение. Также из метки-ограничения можно сделать обычную метку, удалив у нее свойство `taint`. Добавлять или удалять свойство `taint` меткам можно только до создания первого ВЦОД (не считая ВЦОД управления).
- Общие ограничения описаны выше.

8.3.2. Операции с метками кластера

1. Создать и назначить метки узлам кластера

> ❌ **Внимание**
>
> Игнорировать можно те метки, которые были предварительно делегированы во ВЦОД

```
scheduler labels add --labels <LABELS>
                     --nodes <NODES>
                     [--taints <TAINTS>]
                     [--descr <DESCR>]
```

где

- `labels` — список меток узла. Задается в формате `key=value`;
- `nodes` — список идентификаторов узлов, на которые устанавливается метка. При значении `*` метка устанавливается на все узлы;
- `taints` — свойство органичения для метки. Возможные значения:
  - `n` – метка останется обычной. Значение по умолчанию;
  - `y` — метка с ограничением `NoSchedule`;

  При указании `--taints y` метка будет помечена как метка-ограничение. ВМ будут размещаться только при явном указании этой метки в `nodeSelector`. Описание метки `descr` будет присвоено всем создаваемым меткам в данном запросе.
- `descr` — описание метки.

Пример

```
scheduler labels add --labels node=security
                     --nodes be492e22-d18d-46e6-9afc-b4c839ab584c bd2fcdfc-f4d3-4233-8a6b-f4ac4cbbf739
                     --taints y
                     --descr secure nodes for critical applications
```

2. Назначить существующие метки узлам кластера

```
scheduler labels assign --labels <LABELS>
                        --nodes <NODES>
```

где

- `labels` — список меток узла. Задается в формате `key=value`;
- `nodes` — список идентификаторов узлов, на которые устанавливается метка. При значении `*` метка устанавливается на все узлы.

Пример

```
scheduler labels assign --labels graphics=fast disk=ssd
                        --nodes be492e22-d18d-46e6-9afc-b4c839ab584c bd2fcdfc-f4d3-4233-8a6b-f4ac4cbbf739
```

3. Просмотреть метки кластера со списками узлов

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

где

- `filter` — фильтрация меток. Возможные значения:
  - `a` — all, все метки;
  - `i` — ignored, игнорируемые метки;
  - `u` — unignored, все метки, кроме игнорируемых.

4. Просмотреть узлы кластера со списками их меток

```
scheduler labels list [--nodes <NODES>]
                      [--filter <FILTER>]
```

где

- `nodes` — список идентификаторов узлов. При значении `*` выводятся метки на всех узлах;
- `filter` — фильтрация меток. Возможные значения:
  - `a` — all, все метки;
  - `i` — ignored, игнорируемые метки;
  - `u` — unignored, все метки, кроме игнорируемых.

Пример

```
scheduler labels list --nodes be492e22-d18d-46e6-9afc-b4c839ab584c
```

5. Запросить список узлов кластера по определенным меткам

Внимание

Игнорируемые метки не будут учтены при поиске

```
scheduler labels nodes list --labels <LABELS>
                            [--strict <STRICT>]
                            [--silent <SILENT>]
```

где

- `labels` — список меток. Задается в формате `key=value`;
- `strict` — отображать узлы в соответствии с набором меток. Возможные значения:
  - при `strict y` отображать узлы, которые имеют **все** запрошенные метки;
  - при `strict n` отображать узлы, которые имеют **хотя бы одну** из запрошенных меток;
- `silent` — отображать ошибки, если какие-либо метки не существуют. Возможные значения:
  - `y` — отображать ошибки;
  - `n` — не отображать ошибки.

Пример

Узлам A и B присвоены метки

node	labels
A	key=base, disk=ssd, foo=bar
B	key=base

Запрос 1 вернет список [A, B]

Запрос 1

```
scheduler labels nodes list --labels key=base disk=ssd --strict n
```

Запрос 2 вернет только A

Запрос 2

```
scheduler labels nodes list --labels key=base disk=ssd --strict y
```

6. Игнорировать метки в пределах всего кластера.
   Если не указывать параметр `ns`, то метка заблокируется для всех новых и уже существующих ВЦОД: в новый ВЦОД метку добавить будет нельзя, а в уже существующих ВЦОД новые виртуальные кластеры будут создаваться без этой метки.

Внимание

Игнорировать можно те метки, которые были предварительно делегированы во ВЦОД

```
scheduler labels ignore --labels <LABELS>
                        [--ns <NS>]
```

где

- `labels` — список меток для игнорирования;
- `ns` — список ВЦОД. При значении `*` метки будут автоматически помечены как игнорируемые во всех ВЦОД кластера, куда они были делегированы.

Пример

```
scheduler labels ignore --labels disk=ssd key=base
```

7. Сделать обычную метку меткой-ограничением

```
scheduler labels taints add --labels <LABELS>
```

Меткам, указанным в `labels`, будет присвоено ограничение `NoSchedule`.

Пример

```
scheduler labels taints add --labels foo=bar
```

8. Сделать из метки-ограничения обычную метку

```
scheduler labels taints del --labels <LABELS>
```

Пример

```
scheduler labels taints del --labels foo=bar
```

9. Удалить метки кластера

Примечание

При удалении метки предварительно будет произведена попытка ее открепления со всех узлов кластера

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

где `labels` — список меток для удаления.

Пример

```
scheduler labels del --labels disk=ssd key=base
```

8.4. Управлять метками во ВЦОД

8.4.1. Ограничения работы с метками во ВЦОД

- При создании нового ВЦОД и при наличии в кластере базовой метки `key=base` и метки сервисного обслуживания `system_drain_node=yes` они будут автоматически делегированы в созданный ВЦОД.
- Во ВЦОД нельзя делегировать метки, игнорируемые в пределах кластера.
- При удалении ВЦОД все записи о делегированных ему метках удаляются.
- Общие ограничения описаны выше.

8.4.2. Операции с метками во ВЦОД

> **Примечание**
>
> Действия выполняются пользователем с ролью **Администратор кластера** во ВЦОД управления

1. Делегировать метки в заданные ВЦОД

```
scheduler labels delegate ns --ns <NS>
                             --labels <LABELS>
```

где

- `ns` — список имен ВЦОД. При значении `*` метка будет делегирована на все ВЦОД;
- `labels` — список делегируемых меток. Задается в формате `key=value`.

> **Пример**
>
> ```
> scheduler labels delegate ns --ns vdc --labels disk=ssd key=base
> ```

2. Просмотреть метки во ВЦОД

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

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

где

- `ns` — список имен ВЦОД для просмотра их меток.
  Администраторы кластера `admins` могут просмотреть метки всех ВЦОД кластера, добавив флаг `--ns *`.
  Чтобы ограничить вывод меток для определенных ВЦОД, администратор кластера должен указать их имена в `--ns`.
  Пользователям с другими ролями доступен просмотр меток только в своем ВЦОД без указания флага `--ns`.
- `filter` — фильтрация меток. Возможные значения:
  - `a` — all, все метки;
  - `i` — ignored, игнорируемые метки;
  - `u` — unignored, все метки, кроме игнорируемых.

3. Просмотр меток со списками ВЦОД, в которые они делегированы

```
scheduler labels ns list [--ns <NS>]
                         [--filter <FILTER>]
```

где

- `ns` — список имен ВЦОД для просмотра делегированных меток.
  Администраторы кластера `admins` могут просмотреть делегированные метки всех ВЦОД кластера, добавив флаг `--ns *`.
  Чтобы ограничить вывод меток для определенных ВЦОД, администратор кластера должен указать их имена в `--ns`.
  Пользователям с другими ролями доступен просмотр делегированных меток только в своем ВЦОД без указания флага `--ns`.
- `filter` — фильтрация меток. Возможные значения:
  - `a` — all, все метки;
  - `i` — ignored, игнорируемые метки;
  - `u` — unignored, все метки, кроме игнорируемых.

4. Игнорировать метки в пределах заданных ВЦОД

> ✖ **Внимание**
>
> Игнорировать можно те метки, которые были предварительно делегированы во ВЦОД

```
scheduler labels ignore --labels <LABELS>
                        [--ns <NS>]
```

где

- `labels` — список меток для игнорирования;
- `ns` — список ВЦОД. При значении `*` метки будут автоматически помечены как игнорируемые во всех ВЦОД кластера, куда они были делегированы.

Пример

```
scheduler labels ignore --ns vdc --labels disk=ssd key=base
```

9. Параметры IPAM и сетей

Управление IP-адресами (IPAM) — это интегрированный набор средств для комплексного планирования, развертывания, управления и мониторинга инфраструктуры IP-адресов с широкими возможностями взаимодействия с пользователем. IPAM автоматически определяет серверы инфраструктуры IP-адресов и DNS-серверы в сети и позволяет управлять ими из центрального интерфейса.

В статье описаны:

1. Настройка базовой сетевой конфигурации:
    - глобальные параметры IPAM для всего кластера;
    - диапазоны VID и зарезервированные сети;
    - настройки мостов и MTU для виртуальных коммутаторов.
2. Натсройка виртуальных сетей двух типов:
    - VLAN-сети. Классической сегментации трафика с поддержкой ACL.
    - VXLAN-сети. Для overlay-сетей. Без поддержки ACL.
3. Принцип работы со специализированной инфраструктурой:
    - OpenFlow-контроллер;
    - Автоматическое распределение IP-адресов.

Рекомендуемый порядок работы:

1. Начальная настройка. Задание глобальных параметров IPAM.

2. Создание сетей. Определение VLAN или VXLAN.
3. Настройка сервиса взаимодействия с OFC-контроллером.
4. Оперативное управление сетями. Мониторинг и изменение параметров.

9.1. IPAM

Первоначально задайте параметры IPAM, которые определяют базовые настройки.

1. В **Sharx Base** в командной строке введите

   ```
   ipam param add --allocated_vlans <ALLOCATED_VLANS ...>
                  --bridge <BRIDGE>
                  --mac_prefix <MAC_PREFIX ...>
                  --reserved_vlans <RESERVED_VLANS ...>
                  [--vtep_if <VTEP_IF>]
                  [--mtu <MTU>]
                  [--reserved_v4_nets <RESERVED_V4_NETS ...>]
                  [--vxlan_vid_range <VXLAN_VID_RANGE ...>]
   ```

   где

   - `allocated_vlans` — диапазон VID, сиспользуемых системой для создания виртуальных сетей;
   - `bridge` — имя бридж-моста для подключения виртуальных сетей;
   - `mac_prefix` — префикс MAC-адреса, который будет присваиваться виртуальным машинам;
   - `reserved_vlans` — зарезервированная VID, недоступные для создания виртуальных сетей;
   - `vtep_if` — имя VTEP-интерфейса;
   - `mtu` — максимальный размер пакета, который может быть отправлен единовременно без фрагментации. Рекомендуемое значение — `1500`;
   - `reserved_v4_nets` — зарезервированные IPv4-подсети для системных нужд;
   - `vxlan_vid_range` — список выделенных ID для VXLAN-сетей.

2. Чтобы посмотреть созданные настройки, введите команду

   ```
   ipam param show
   ```

3. Обновить параметры

   ```
   ipam param update [--allocated_vlans <ALLOCATED_VLANS ...> ]
                     [--bridge <BRIDGE>]
                     [--mac_prefix <MAC_PREFIX ...>]
                     [--reserved_vlans <RESERVED_VLANS ...>]
   ```

```
                    [--vtep_if <VTEP_IF>]
                    [--mtu <MTU>]
                    [--reserved_v4_nets <RESERVED_V4_NETS ...>]
                    [--vxlan_vid_range <VXLAN_VID_RANGE ...>]
```

9.2. Определить сети

1. Чтобы определить сеть, в командной строке введите

```
ipam network add --name <NAME>
                 --ipv 4
                 --vlan <VLAN>
                 --v4_net <V4_NET>
                 [--descr <DESCR>]
                 [--mtu <MTU>]
                 [--ns <NS>]
                 [--v4_dns1 <V4_DNS1>]
                 [--v4_dns2 <V4_DNS2>]
                 [--v4_start <V4_START>]
                 [--v4_end <V4_END>]
                 [--v4_gw <V4_GW>]
                 [--v4_mask <V4_MASK>]
                 [--v4_pref <V4_PREF>]
```

   где

   - `name` — имя сети;
   - `ipv` — версия IP-сети;
   - `vlan` — идентификатор виртуальной локальной сети;
   - `v4_net` — адрес IPv4-сети. Например, `X.X.X.X`;
   - `descr` — описание сети;
   - `mtu` — максимальный размер пакета, который может быть отправлен единовременно без фрагментации. Рекомендуемое значение — *1500*;
   - `ns` — имя ВЦОД. Обязательно для `network`, необязательно для `vxlan`;
   - `v4_dns1` — первичный DNS-сервер сети;
   - `v4_dns2` — вторичный DNS-сервер сети;
   - `v4_start` — первый адрес для отдачи под ВМ;
   - `v4_end` — последний адрес для отдачи под ВМ;
   - `v4_gw` — адрес шлюза для данной подсети;
   - `v4_pref` — префикс подсети в формате CIDR.

2. Посмотреть список созданных сетей

```
ipam network list [--show_acl <SHOW_ACL>]
```

где `show_acl` — показать правила списков контроля доступа для каждой сети.

3. Посмотреть подробную информацию о конкретной сети

```
ipam network show --net_uuid <NET_UUID>
                  [--show_acl <SHOW_ACL>]
```

где `net_uuid` — идентификатор сети.

4. Обновить параметры сети

```
ipam network update --net_uuid <NET_UUID>
                    [--descr <DESCR>]
                    [--mtu <MTU>]
                    [--name <NAME>]
                    [--ns <NS>]
                    [--v4_dns1 <V4_DNS1>]
                    [--v4_dns2 <V4_DNS2>]
                    [--v4_end <V4_END>]
                    [--v4_gw <V4_GW>]
                    [--v4_start <V4_START>]
```

9.3. VXLAN

VXLAN — это технология виртуализации сетей, которая расширяет возможности обычных виртуальных локальных сетей. VXLAN позволяет создавать изолированные виртуальные сегменты поверх существующей сетевой инфраструктуры.

> ⚠ **Важно**
>
> Перед созданием инфраструктуры VXLAN необходимо задать базовые параметры плагина IPAM

9.3.1. Настроить базовые параметры плагина для работы с VXLAN

Чтобы задать или изменить базовые параметры плагина, введите

```
ipam param update --vtep_if <VTEP_IF>
                  --vxlan_vid_range <VXLAN_VID_RANGE>
```

где

- `vtep_if` — имя VTEP-интерфейса;
- `vxlan_vid_range` — диапазон ID (VID), из которого будут выделяться идентификаторы подсети VXLAN.

> **Важно**
>
> Значение `vxlan_vid_range` используется всеми VXLAN-сетями кластера. `vxlan_vid_range` не может пересекаться с диапазонами `allocated_vlans` и `reserved_vlans`

> **Пример**
>
> ```
> ipam param update --vtep_if <vtep_if_name> --vxlan_vid_range 1000-4094
> ```
>
> Каждой VXLAN-подсети при создании автоматически будет назначаться уникальный VID из указанного диапазона

9.3.2. VXLAN-сети

Особенности VXLAN-сетей:

- VXLAN-сеть выступает контейнером для изолированных подсетей с автоматической маршрутизацией между ними;
- для работы требуется предварительная настройка VTEP-интерфейса;
- каждому ВЦОД может быть назначена только одна VXLAN-сеть;
- нельзя изменить привязку к ВЦОД, если во VXLAN-сети уже созданы подсети;
- VXLAN-сеть нельзя удалить, если внутри нее созданы VXLAN-подсети.

Действия с VXLAN-сетями:

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

   ```
   ipam vxlan network add --name <NAME>
                          [--ns <NS>]
                          [--descr <DESCR>]
   ```

   где

   - `name` — имя VXLAN-сети;

- `ns` — имя ВЦОД. ВЦОД может быть привязан только к одной VXLAN-сети;
- `descr` — описание VXLAN-сети.

Пример

```
ipam vxlan network add --name vx-1 --ns vcod-1 --descr "vxlan network"
```

2. Просмотреть список VXLAN-сетей кластера

```
ipam vxlan network list
```

3. Изменить ВЦОД для VXLAN-сети

Внимание

Нельзя изменить привязку к ВЦОД, если во VXLAN-сети уже созданы подсети

```
ipam vxlan network update --vxlan_uuid <VXLAN_UUID>
                           [--ns <NS>]
                           [--name <NAME>]
                           [--descr <DESCR>]
```

где

- `vxlan_uuid` — идентификатор VXLAN-сети;
- `ns` — имя ВЦОД;
- `name` — имя VXLAN-сети;
- `descr` — описание VXLAN-сети.

Пример

```
ipam vxlan network update --vxlan_uuid 4ab2646c-1109-42ad-a3a3-506cb708c87c
                          --ns vcod-2
```

4. Удалить VXLAN-сеть

Внимание

Нельзя удалить VXLAN-сеть, если в ней есть подсети

```
ipam vxlan network del --vxlan_uuid <VXLAN_UUID>
```

> **Пример**
>
> ```
> ipam vxlan network del --vxlan_uuid 4ab2646c-1109-42ad-a3a3-506cb708c87c
> ```

5. Просмотреть сетевые интерфейсы конкретной подсети VXLAN

```
ipam iface list --vxlan_uuid <VXLAN_UUID>
```

где `vxlan_uuid` — идентификатор VXLAN-сети.

> **Пример**
>
> ```
> ipam iface list --vxlan_uuid 4ab2646c-1109-42ad-a3a3-506cb708c87c
> ```

9.3.3. Автоматическое управление сетевыми интерфейсами

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.

Дополнительные действия с сетевыми параметрами выполняет Администратор ВЦОД см. Руководство пользователя.

9.4. Взаимодействие с OFC-контроллером

Для взаимодействия с точками конфигурации OpenFlow в Sharx Base используется сервис для запуска, перезапуска или остановки системных сервисов на узлах виртуализации кластера.

Чтобы включить контроль за сервисом, используйте команду

```
services global systemd add --code VIP
                            --descr <service_description>
                            --name faucet
                            --state start
```

После выполнения команды на узлах кластера запускается сервис **faucet**, который обновляет описание OpenFlow-контроллера.

9.5. Удаление сетей и сетевых параметров

1. Чтобы удалить сеть, введите команду

   ```
   ipam network del --net_uuid <NET_UUID>
   ```

   где `net_uuid` — идентификатор сети.

2. Удалить параметры сети

   ```
   ipam param del
   ```

> **Примечание**
>
> Дополнительные действия с сетевыми параметрами выполняет Администратор ВЦОД см. Руководство пользователя

10. Управление ресурсами кластера

Раздел содержит настройку и управление ресурсами логического кластера.

Sharx Base использует систему управления ресурсами, которая обеспечивает доступность за счет резервирования ресурсов для системных компонентов и отказоустойчивости, эффективное использование с помощью переподписки для оптимизации загрузки оборудования и гибкое масштабирование для возможности расширения кластера без прерывания работы.

Для нового кластера выполните следующие шаги:

1. Настройте резервирование для высокой доступности.
2. Настройте резервирование для системных нужд.
3. Настройте переподписку.

4. Проверьте доступные ресурсы перед созданием ВЦОД.

При расширении кластера или изменении состава узлов в кластере:

1. Добавьте новые узлы в кластер.
2. Обновите резервирование системных ресурсов.

10.1. Резервирование ресурсов

> **Важно**
>
> Настройка параметров выполняется до создания ВЦОД

10.1.1. Резервировать ресурсы для высокой доступности

Высокая доступность кластера (отказоустойчивость, High Availability, HA) — это свойство Платформы обеспечивать максимально продолжительную доступность даже при отказе компонентов из-за аварии. При корректных настройках HA без дополнительных затрат минимизируется время простоя, вызванное сбоем серверов или операционных систем.

> **Примечание**
>
> Для кластера из 3 узлов рекомендуется устанавливать значение `30%`, так как HA предполагает возможность потери одного узла без нарушения работоспособности

1. Чтобы настроить отказоустойчивость, в командной строке введите

```
aaa cluster resource ha add --reserved_ha_cpu <RESERVED_HA_CPU>
                            --reserved_ha_ram <RESERVED_HA_RAM>
                            --reserved_ha_disk <RESERVED_HA_DISK>
                            --resources_share <RESOURCES_SHARE>
```

где

- `reserved_ha_disk` — процентное резервирование дискового пространства для отказоустойчивости на узле кластера. Рекомендуемое значение — *20*;
- `reserved_ha_ram` — процентное резервирование оперативной памяти для отказоустойчивости на узле кластера. Рекомендуемое значение — *20*;
- `reserved_ha_cpu` — процентное резервирование виртуального процессора для отказоустойчивости на узле кластера. Рекомендуемое значение — *20*;
- `resources_share` — совместное использование ресурсов кластера. По умолчанию — *no*.

2. Просмотр текущих настроек

```
aaa cluster resource ha show
```

3. Обновить текущие настройки высокой доступности

```
aaa cluster resource ha update --reserved_ha_cpu <RESERVED_HA_CPU>
                               --reserved_ha_ram <RESERVED_HA_RAM>
                               --reserved_ha_disk <RESERVED_HA_DISK>
                               --resources_share <RESOURCES_SHARE>
```

4. Удалить текущие настройки высокой доступности

```
aaa cluster resource ha del
```

10.1.2. Резервировать ресурсы под системные нужды

Sharx Base может автоматически рассчитать и зарезервировать ресурсы ЦПУ и ОЗУ для системных нужд на основе конфигурации контрольных групп на узлах кластера.
Чтобы функция резервирования выполнилась корректно, необходимо:

1. Убедиться в корректной настройке отслеживания контрольных групп.
2. Применить функцию автоматического резерва ресурсов под системные нужды.

При необходимости можно вручную изменить резервирование ресурсов согласно разделу Ручное управление резервированием.

КОНТРОЛЬНЫЕ ГРУППЫ

> **Важно**
>
> Перед настройкой резервирования ресурсов необходимо убедиться в корректной конфигурации контрольных групп на узлах кластера и настроить отслеживание этих групп

Контрольная группа — механизм контроля групп ресурсов (control groups или cgroups) для изоляции и ограничения потребления ресурсов разными компонентами системы.
Контрольные группы настраиваются в директории `/etc/cgconfig.d/` на каждом узле кластера. Каждая контрольная группа резервирует определенную долю физических ресурсов узла для конкретных задач.

По умолчанию в системе созданы следующие контрольные группы:

- `machine.slice` для расчета резерва ОЗУ;
- `storpool.slice` для РСХД;
- `user.slice` для пользовательских процессов;
- `system.slice` для системных служб операционной системы.

Система автоматического расчета резервирования отслеживает только те контрольные группы, которые указаны в переменной `SDC_CGROUPS_RESOURCES_MONITOR` в файле `/etc/sdc-env`.

Отслеживание контрольных групп по умолчанию

```
SDC_CGROUPS_RESOURCES_MONITOR="user,system,storpool"
```

Возможно ввести дополнительные группы, предварительно определив их конфигурацию в директории `/etc/cgconfig.d/`.

Отслеживание дополнительных контрольных групп

```
SDC_CGROUPS_RESOURCES_MONITOR="user,system,storpool,monitoring,security"
```

> **Внимание**
>
> После изменения конфигурации групп или переменной необходимо выполнить команду для перерасчета резервирования
>
> ```
> aaa cluster resource reserved apply
> ```

АВТОМАТИЧЕСКОЕ РЕЗЕРВИРОВАНИЕ РЕСУРСОВ КЛАСТЕРА

Система вычисляет объем ЦПУ и ОЗУ, который необходимо зарезервировать на уровне кластера. При расчете учитываются доступные лимиты ресурсов на виртуализацию, заданные на уровне операционной системы.

Чтобы зарезервировать ресурсы для системных нужд:

1. Опционально посмотрите предварительный расчет ресурсов кластера, которые будут зарезервированы на всех узлах

   ```
   aaa cluster resource reserved calc
   ```

Эта команда предназначена только для ознакомления с расчетами системы. Вы можете пропустить этот шаг и сразу перейти к применению настроек, так как система использует эти же внутренние расчеты для команды `apply`.

2. Чтобы применить рассчитанные системой значения резерва ресурсов, выполните команду

```
aaa cluster resource reserved apply
```

> **Важно**
>
> Данную команду необхоимо выполнять каждый раз при изменении конфигурации контрольных групп или переменной отслеживания

РУЧНОЕ УПРАВЛЕНИЕ РЕЗЕРВИРОВАНИЕМ

Рекомендуется устанавливать резерв ресурсов под системные нужды, рассчитанный с помощью команд в разделе Автоматическое резервирование ресурсов кластера. При необходимости Администратор может изменять данные значения, выполнив следующие действия:

1. Просмотрите текущие настройки резерва ресурсов

```
aaa cluster resource reserved show --uuids *
```

где `uuids` — список идентификаторов узлов. При `uuids`, равном `*`, выведется информация по всем узлам. Чтобы получить информацию по конкретному узлу, введите его идентификатор.

2. Чтобы изменить настройки резерва ресурсов, введите

```
aaa cluster resource reserved update --uuid <UUID>
                                     [--reserved_system_cpu <RESERVED_SYSTEM_CPU>]
                                     [--reserved_system_ram <RESERVED_SYSTEM_RAM]>
                                     [--reserved_system_disk <RESERVED_SYSTEM_DISK>]
```

где

- `uuid` — идентификатор узла;
- `reserved_system_cpu` — резервирование ЦПУ для системных нужд на узле кластера;
- `reserved_system_ram` — резервирование оперативной памяти для системных нужд на узле кластера;

- `reserved_system_disk` — резервирование дискового пространства для системных нужд на узле кластера.

Чтобы удалить текущие параметры резервирования

```
aaa cluster resource reserved del --uuids <UUIDS>
```

10.2. Переподписка ресурсов

Переподписка позволяет виртуально выделить виртуальным машинам больше ресурсов, чем физически доступно в кластере.

> **Внимание**
>
> Изменение параметров переподписки возможно только когда в кластере не созданы ВЦОД (за исключением ВЦОД управления)

10.2.1. Переподписка ЦПУ

Переподписка ЦПУ — это метод оптимизации использования ресурсов, который позволяет виртуальным машинам использовать больше виртуальных ядер (ВЦПУ), чем имеется физических ядер (ЦПУ) в кластере.

Основная формула перерасчета ВЦПУ в ЦПУ

```
CPU = VCPU/cpu_overcommit_ratio
```

Ключевой параметр `default_cpu_overcommit_ratio` — глобальный коэффициент переподписки в диапазоне от 1 до 4. Чем он выше, тем больше виртуальных ядер можно создать на одном физическом ядре. Этот коэффициент применяется ко всем ВМ по умолчанию, но может быть переопределен для каждой виртуальной машины индивидуально.

МЕХАНИЗМЫ ПЕРЕПОДПИСКИ ЦПУ

В Sharx Base доступны два механизма переподписки ЦПУ: тип доли `shares` и тип квоты `quotas`.

1. **Тип доли `shares`**.

   Механизм доли `shares` определяет относительные доли процессорного времени, которые получают ВМ. Он не гарантирует абсолютное время ЦПУ, но обеспечивает оптимальное распределение процессорных ресурсов относительно других ВМ.

Формула расчета доли

```
shares = 4096/cpu_overcommit_ratio
```

Команда настройки

```
aaa cluster resource overcommit add --cpu_overcommit_type shares --
default_cpu_overcommit_ratio {1-4}
```

где

- `cpu_overcommit_type` — тип переподписки ЦПУ. Установлено значение `share` (доли);
- `default_cpu_overcommit_ratio` — коэффициент переподписки ЦПУ в кластере по умолчанию. Возможные значения от `1` до `4`.

2. **Тип квоты `quotas`**.

Механизм квоты `quotas` задает фиксированную квоту процессорного времени в микросекундах, которое ВЦПУ может использовать в течение определенного периода. Квоты обеспечивают более строгий и предсказуемый контроль в распределении ресурсов.

Чем выше коэффициент переподписки, тем ниже квота процессорного времени, т.е. допустимое время выполнения ВЦПУ ВМ.

Формула расчета квоты

```
cpu_quota = cpu_quota_period/cpu_overcommit_ratio
```

Команда настройки

```
aaa cluster resource overcommit add --cpu_overcommit_type quotas --cpu_quota_period {10000-
500000} --default_cpu_overcommit_ratio {1-4}
```

где

- `cpu_overcommit_type` — тип переподписки ЦПУ. Установлено значение `quotas` (квоты);
- `cpu_quota_period` — период, в течение которого отслеживается квота. Задается в микросекундах. Возможные значения от `10000` до `500000`;
- `default_cpu_overcommit_ratio` — коэффициент переподписки ЦПУ в кластере по умолчанию. Возможные значения от `1` до `4`.

КОМАНДЫ УПРАВЛЕНИЯ ПЕРЕПОДПИСКОЙ ЦПУ

> **Внимание**
>
> Изменение или удаление параметров переподписки возможно только когда в кластере не созданы ВЦОД (за исключением ВЦОД управления)

1. Чтобы проверить статус переподписки ЦПУ, введите

```
aaa cluster resource overcommit show
```

2. Обновить параметры переподписки ЦПУ

```
aaa cluster resource overcommit update --cpu_overcommit_type <CPU_OVERCOMMIT_TYPE>
                                       --default_cpu_overcommit_ratio <DEFAULT_CPU_OVERCOMM
IT_RATIO>
```

3. Удалить параметры переподписки

> **Важно**
>
> Команда удаляет все параметры переподписки: и ЦПУ и ОЗУ

```
aaa cluster resource overcommit del
```

> **Примечание**
>
> Пример изменения коэффициента переподписки ЦПУ для определенной ВМ описан в **Руководстве пользователя в командной строке** в статье Оптимизация ресурсов

СРАВНЕНИЕ МЕХАНИЗМОВ ПЕРЕПОДПИСКИ ЦПУ

Таблица — Сравнение механизмов переподписки ЦПУ.

Критерий	Тип `shares` доли	Тип `quotas` квоты
Принцип работы		Абсолютная квота процессорного времени за период

Критерий	**Тип `shares` доли** Относительные доли процессорного времени	**Тип `quotas` квоты**
Гибкость	Высокая	Ограниченная
Предсказуемость	Относительная, зависит от общей загрузки	Высокая, гарантированная квота
Рекомендуемые сценарии	Смешанные нагрузки, веб-сервисы	Приложения с жесткими требованиями к производительности. Например, базы данных

Рекомендации по выбору

Тип `shares` рекомендуется для рабочих нагрузок с переменной активностью, где важно относительное распределение ресурсов.

Тип `quotas` рекомендуется для нагрузок, требующих гарантированного времени ЦПУ и предсказуемой производительности.

Коэффициент переподписки `default_cpu_overcommit_ratio` следует выбирать в зависимости от ожидаемой нагрузки и требований к производительности.

10.2.2. Переподписка ОЗУ

Переподписка ОЗУ — технология увеличения доступной оперативной памяти для виртуальных машин через механизм `ballooning`, позволяющий ВМ делиться частью своей памяти с другими виртуальными машинами на том же узле.

Последовательность настройки переподписки ОЗУ:

1. Проверьте работу cgroup-экспортера.
2. Настройте параметры переподписки ОЗУ.
3. Включите механизм переподписки ОЗУ.
4. Контролируйте работу кластера.

ТРЕБОВАНИЯ К СИСТЕМЕ ДЛЯ ПЕРЕПОДПИСКИ ОЗУ

Для работы переподписки ОЗУ необходимо выполнить следующие условия на каждом узле кластера:

1. В конфигурации плагина `sdc-plgn-metrics` должен быть настроен cgroup-экспортер. Чтобы проверить работу cgroup-экспортера, введите

```
scheduler report mem
```

> **Пример**
>
> В результате выполнения команды должен вернуться список ресурсов по каждому узлу
>
> ```
> {
>    "2a808504-a385-4856-aa14-f7c1240d81cb": {
>       "CurrentMemory": "0Gb",
>       "MaxMemory": "0Gb",
>       "memballoon": "0Gb",
>       "memtotal": "1315Mb",
>       "memusage": "0Gb",
>       "memusage_ballooned": "0Gb",
>       "memusage_percent": 0
>    },
>    "336b18d5-9a2a-4ac5-90d2-6dc479ec709e": {
>       "CurrentMemory": "0Gb",
>       "MaxMemory": "0Gb",
>       "memballoon": "0Gb",
>       "memtotal": "1315Mb",
>       "memusage": "6Mb",
>       "memusage_ballooned": "6Mb",
>       "memusage_percent": 0
>    },
>    "d4de5a59-e11d-49aa-990c-8ca589611fe8": {
>       "CurrentMemory": "0Gb",
>       "MaxMemory": "0Gb",
>       "memballoon": "0Gb",
>       "memtotal": "1315Mb",
>       "memusage": "0Gb",
>       "memusage_ballooned": "0Gb",
>       "memusage_percent": 0
>    }
> }
> ```

Если cgroup-экспортер не настроен, то добавьте его в считывание командой

```
metrics exporter add --name cgroup_exporter --port 9198
```

После добавления экспортера проверьте его считывание

```
scheduler report mem
```

2. На каждом узле в файле `/etc/sdc-env` переменные окружения имеют фактические значения. В примерах ниже показаны их значения по умолчанию, которые система использует автоматически.

```
# Имя cgroup-экспортера. По умолчанию cgroup_exporter
SDC_CGROUP_EXPORTER_NAME="cgroup_exporter"

# Порт cgroup-экспортера.  По умолчанию 9198
SDC_CGROUP_EXPORTER_PORT="9198"
```

> **Примечание**
>
> Если вам необходимо изменить значения по умолчанию, вы можете явно задать их в файле `/etc/sdc-env`. Убедитесь, что заданные значения соответствуют фактической конфигурации экспортера

ПРИНЦИП РАБОТЫ И ОСНОВНЫЕ ПОНЯТИЯ

Основные понятия:

Балунинг `ballooning` — технология управления оперативной памятью в виртуализации, которая позволяет гипервизору временно «забирать» неиспользуемую память у одних виртуальных машин и перераспределять ее другим ВМ на том же физическом узле.

Порог срабатывания балунинга `ballooning_threshold` — уровень загруженности ОЗУ узла, с которого начинает работать механизм переподписки ОЗУ.

Лимит памяти узла `node_memory_limit` — максимальная загрузка ОЗУ узла, при достижении которой прекращается размещение новых ВМ.

Толерантность к переподписке `overcommit_tolerance` — уровень, определяющий объем памяти, которым может поделиться ВМ.

При включенной переподписке ОЗУ работает балунинг, при котором каждая виртуальная машина может отдавать часть своей памяти другим ВМ на том же узле.

Объем отдаваемой памяти зависит от показателей:

- Толерантности ВМ. Чем выше толерантность, тем больше памяти может отдать ВМ.
- Текущей загруженности узла. Чем выше загрузка, тем активнее работает балунинг.

> **Важно**
>
> При толерантности, равной `0`, ВМ не делится своей памятью

НАСТРОЙКА ПЕРЕПОДПИСКИ ОЗУ

Действия для настройки переподписки ОЗУ:

1. Чтобы настроить переподписку ОЗУ, введите

   ```
   aaa cluster resource overcommit update --ram_overcommit <RAM_OVERCOMMIT>
                                          --node_memory_limit <NODE_MEMORY_LIMIT>
                                          --default_overcommit_tolerance <DEFAULT_OVERCOMMIT_T
   OLERANCE>
   ```

   где

   - `ram_overcommit` — процент переподписки ОЗУ;
   - `node_memory_limit` — лимит использования ОЗУ узла в процентах;
   - `default_overcommit_tolerance` — толерантность ВМ к переподписке. Возможные значения от `0` до `3`.

   В результате команды Sharx Base рассчитывает порог срабатывания балунинга `ballooning_threshold` по формуле

   ```
   ballooning_threshold = 100 - ram_overcommit
   ```

   Критичные условия для назначения параметров переподписки ОЗУ:

   - Для запуска балунинга необходимо выполнить условие `ballooning_threshold` < `node_memory_limit`.
   - Разница между лимитами `node_memory_limit - 2% - ballooning_threshold ≥ 3%`.

Пример

```
aaa cluster resource overcommit update --ram_overcommit 40
                                       --node_memory_limit 98
                                       --default_overcommit_tolerance 0
```

В данном случае Sharx Base рассчитает

```
ballooning_threshold = 100 - 40 = 60%
```

Разница между лимитами

```
node_memory_limit - 2% - ballooning_threshold = 98% - 2% - 60% = 36%
```

Так как 36% ≥ 3%, параметры переподписки ОЗУ установлены корректно

2. Чтобы обновить параметры переподписки ОЗУ, введите

```
aaa cluster resource overcommit update --ram_overcommit <RAM_OVERCOMMIT>
                                       --node_memory_limit <NODE_MEMORY_LIMIT>
                                       --default_overcommit_tolerance <DEFAULT_OVERCOMMIT_TOLE
RANCE>
```

3. Удалить параметры переподписки

Важно

Команда удаляет все параметры переподписки: и ЦПУ и ОЗУ

```
aaa cluster resource overcommit del
```

Внимание

Изменение или удаление параметров переподписки возможно только когда в кластере не созданы ВЦОД (за исключением ВЦОД управления)

УПРАВЛЕНИЕ ПЕРЕПОДПИСКОЙ ОЗУ

> ❌ **Внимание**
>
> Переподписку ОЗУ можно включить или выключить только из ВЦОД управления

1. Чтобы включить переподписку ОЗУ, введите

   ```
   aaa cluster resource overcommit ballooning enable
   ```

2. Чтобы выключить переподписку ОЗУ, введите

   ```
   aaa cluster resource overcommit ballooning disable
   ```

   > ❌ **Внимание**
   >
   > Выключение переподписки возможно только при загруженности ОЗУ каждого узла ниже лимита использования ОЗУ `node_memory_limit`

3. Чтобы проверить статус переподписки ОЗУ, введите

   ```
   aaa cluster resource overcommit show
   ```

После выполнения всех шагов система начнет автоматически перераспределять память между ВМ при достижении порога срабатывания балунинга на узлах.

> **Примечание**
>
> Пример изменения толерантности определенной ВМ к переподписке описан в **Руководстве пользователя в командной строке** в статье Оптимизация ресурсов

10.3. Просмотр ресурсов и управление узлами кластера

10.3.1. Просмотр доступных ресурсов

Чтобы посмотреть доступные ресурсы для ВЦОД, введите команду

```
aaa cluster resource show
```

Чтобы посмотреть ресурсы на каждом узле, введите

```
aaa cluster nodes resource show
```

10.3.2. Управление узлами кластера

Sharx Base позволяет расширить или уменьшить общие ресурсы логического кластера. Чтобы управлять узлами кластера, выполните следующие шаги:

1. Просмотрите список доступных узлов

   ```
   cluster freenodes list
   ```

2. Выберите узел, который будет добавлен в кластер

   ```
   cluster nodes add --node <NODE>
   ```

   где `node` — идентификатор узла.

3. Чтобы удалить узел из кластера, выполните следующую команду

   ```
   cluster nodes del --node <NODE>
   ```

11. ВЦОД

ВЦОД — виртуальный центр обработки данных.

ВЦОД может создавать **Администратор кластера**, когда находится во ВЦОД управления.

11.1. Создать ВЦОД

> ❌ **Внимание**
>
> Перед созданием ВЦОД настройте резервирование и переподписку ресурсов

1. Чтобы создать или инициализировать ВЦОД, введите в командной строке

   ```
   aaa namespace add --cluster <CLUSTER>
                     --ns <NS>
                     --cpu <CPU>
                     --ram <RAM>
                     [--paths <PATHS>]
   ```

```
                    [--descr <DESCR>]
                    [--config_name <CONFIG_NAME>]
```

где

- `cluster` — имя кластера;
- `ns` — имя создаваемого ВЦОД;
- `cpu` — количество ЦПУ;
- `ram` — объем оперативной памяти. Указывается в мегабайтах `Mb` или гигабайтах `Gb`. Примеры: `2048Mb`, `2Gb`, `4Gb`;
- `descr` — описание ВЦОД;
- `paths` — структура директорий ВЦОД в формате `JSON`. По умолчанию имеет значение `"/"` (корневая директория ВЦОД);
- `config_name` — имя конфигурационного файла. Возможные значения:
    - `fstek_virt` — конфигурация по требованиям ФСТЭК,
    - `cb_virt` — конфигурация по требованиям ЦБ.

  Конфигурация ВЦОД может быть указана на уровне переменной окружения `CONFIG_ROLES_NAME`, тогда название конфигурационного файла указывать необязательно.

Пример `paths`

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

где

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

Важно

Для ВЦОД управления задайте значение `ram`, равное `0Gb`, значение `cpu`, равное `0`, так как ВЦОД управления предназначен только для настройки других ВЦОД, в нем будет отсутствовать виртуализация

2. Посмотреть информацию и статус ВЦОД

```
aaa namespace show --cluster <CLUSTER>
                   --ns <NS>
```

3. Проверить наличие ВЦОД в общем списке

```
aaa namespace list --cluster <CLUSTER>
```

Примечание

При необходимости настройте метки ВЦОД

11.2. Удалить ВЦОД

Перед удалением ВЦОД сначала проверьте, что в нем нет объектов, правил, пользователей и других связанных объектов. После проверки для удаления ВЦОД введите команду

```
aaa namespace del --cluster <CLUSTER>
                  --ns <NS>
```

Если во ВЦОД есть объекты, и вы уверены в удалении, используйте флаг `force`, указав в нем имя удаляемого ВЦОД

Внимание

Флаг `--force` приводит к полному удалению всех объектов ВЦОД

```
aaa namespace del --cluster <CLUSTER>
                  --ns <NS>
                  --force <NS>
```

Пример с force

```
aaa namespace del --cluster cluster-test-version
                  --ns ns-test
                  --force ns-test
```

12. Хранилище

12.1. Типы хранилищ

Sharx Base позволяет работать с тремя типами хранилища:

1. Libvirt.
2. РСХД.
3. NFS.

Каждый тип хранилища имеет свои особенности, преимущества и ограничения. Выбор хранилища зависит от требований к производительности, отказоустойчивости, масштабируемости и особенностей инфраструктуры.

12.1.1. Краткое описание типов хранилищ

Libvirt — локальное хранилище узла. Хранилище создается локально на дисках узла. Простое в настройке и использовании, не требует дополнительной сетевой инфраструктуры. Подходит для тестовых сред, небольших проектов, изолированных ВМ, не требующих высокой доступности.

Когда использовать:

- тестовые стенды;
- ВМ с невысокой критичностью;
- сценарии без требований к миграции и высокой доступности.

РСХД — распределенная система хранения данных: объединяет напрямую подключенные хранилища (жесткие диски или твердотельные накопители) со стандартных серверов для создания единого пула общего хранилища. Программное обеспечение объединяет производительность и емкость дисков, подключенных к серверам, в одно глобальное пространство имен. Поддерживает репликации и горизонтальное масштабирование. Обеспечивает миграцию виртуальных машин между узлами, балансировку нагрузки и высокую доступность.

Когда использовать:

- промышленные кластеры;
- системы с высокими требованиями к доступности;
- виртуальные среды, где важна производительность и масштабируемость.

NFS-хранилище — сетевое файловое хранилище. NFS обеспечивает доступ к файлам, расположенным на удаленных серверах, и позволяет работать с этими файлами точно так же, как и с локальными. NFS-хранилище простое в настройке, поддерживает миграцию ВМ. Зависит от качества сетевой инфраструктуры и конфигурации NFS-сервера.

Когда использовать:

- для хранения образов ВМ, ISO, резервных копий;
- в небольших и средних кластерах, где отказоустойчивость можно обеспечить средствами NFS-сервера;
- для ВМ со средними нагрузками.

12.1.2. Этапы работы с хранилищами

В Sharx Base правами на работу с различными объектами хранилищ обладают определенные роли: **Администратор кластера**, **Администратор ВЦОД** и **Разработчик ВМ**. В зависимости от логического этапа работы на Платформе в документации находятся статьи с информацией о настройке конкретного объекта хранилища. Если действия выполняет **Администратор ВЦОД** или **Разработчик ВМ**, подробное описание находится в

12.2. РСХД. Настроить лимиты использования ресурсов

В статье описана только настройка объектов РСХД: групп хранения и шаблонов, так как эти действия выполняет **Администратор кластера**. Остальные шаги настройки рассмотрены в

Лимиты использования ресурсов в хранилище РСХД — это ограничения, которые устанавливаются для управления и контроля потребления ресурсов системы хранения данных.
В рамках лимитов использования создаются **группы хранения** — наборы дисков, на которые будут реплицироваться объекты. Например, наборы по типу SSD.
Затем на основе групп хранения необходимо создать **шаблоны**, которые определяют параметры для создания томов в РСХД. Шаблоны позволяют упростить процесс создания новых томов, обеспечивая быстрое и единообразное развертывание томов с заданными характеристиками.

> ❌ **Внимание**
>
> Только Администратор кластера имеет права на настройку лимитов использования ресурсов в хранилище РСХД

12.2.1. Резервировать ресурсы

Резерв ресурсов происходит через создание и управление группами размещений.

1. Чтобы просмотреть созданные группы размещения, введите команду

```
storage sp placementGroup list
```

> **Пример**
>
> Пример вывода списка групп размещения
>
> ```
> json [ "ssd" ]
> ```

2. Создать группу размещения

```
storage sp placementGroup resources reserved add --place <ssd> --reserved <value>
```

где

- `place` — имя группы размещения;
- `reserved` — количество резервируемой физической памяти.

3. Просмотреть ресурсы, доступные для групп размещения

```
storage sp placementGroup resources show --place <name_place>
```

> **Пример**
>
> Пример вывода доступных ресурсов для групп размещения
>
> ```
> {
> "allocated": "100Gb",
> "available": "175Gb",
> "limit": "275Gb",
> "placementGroup": "ssd"
> }
> ```

4. Изменить количество резервируемой физической памяти у группы размещения

```
storage sp placementGroup resources reserved update --place <name_place> [--reserved <value
>]
```

5. Удалить группу размещения

```
storage sp placementGroup resources reserved del --place <name_place>
```

12.2.2. Создать шаблон

Внимание

Шаблоны создаются только во ВЦОД управления с учетом прав доступа.

Возможно создать шаблон только для существующего ВЦОД

Шаблон создается Администратором кластера для использования в рамках определенного ВЦОД.

1. Чтобы создать шаблон, введите команду

```
storage sp template add [--bw <BW>]
                        [--iops <IOPS>]
                        [--descr <DESCR>]
                        --name <name_template>
                        [--parent <PARENT>]
                        --place <name_place>
                        --replication <rf_quantity>
                        --capacity <volume_capacity>
                        --ns <namespace_name>
```

где

- `bw` — ограничение пропускной способности в килобайтах;
- `iops` — ограничение количества операций ввода-вывода в секунду;
- `descr` — описание шаблона;
- `name` — имя шаблона;
- `parent` — имя снимка, на основе которого будут созданы тома;
- `place` — имя группы размещения;
- `replication` — фактор репликации;
- `capacity` — ограничение размера шаблона;
- `ns` — имя ВЦОД.

> ⚠ **Важно**
>
> `capacity`, умноженная на `replication`, должен быть меньше или равен доступному ресурсу группы размещения

12.2.3. Команды для работы с шаблонами

1. Просмотреть список созданных шаблонов

```
storage sp template list
```

2. Просмотреть подробную информацию о шаблоне

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

где `name` — имя шаблона.

3. Обновить параметры шаблона

```
storage sp template update [--bw <BW>]
                           [--iops <IOPS>]
                           [--descr <DESCR>]
                           --name <name_template>
                           [--parent <PARENT>]
                           [--replication <rf_quantity>]
                           [--reuse_server <REUSE_SERVER>]
                           [--capacity <volume_capacity>]
                           [--ns <namespace_name>]
```

где

- `bw` — ограничение пропускной способности в килобайтах;
- `iops` — ограничение количества операций ввода-вывода в секунду;
- `descr` — описание шаблона;
- `name` — имя шаблона;
- `parent` — имя снимка, на основе которого будут созданы тома;
- `replication` — количество копий для репликации;
- `reuse_server` — разрешение размещения реплик на одном сервере;
- `capacity` — ограничение размера шаблона;
- `ns` — имя ВЦОД.

4. Просмотреть список шаблонов с лимитами и доступными ресурсами

```
storage sp template limit list
```

5. Шаблон может зависнуть в статусе *Pending (Ожидание)*. В таких случаях удалите запись о нем из базы данных командой

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

6. Удалить шаблон

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

После добавления шаблона во ВЦОД **Администратор ВЦОД** может создавать тома РСХД. При первой настройке системы рекомендуется подготавливать тома после создания виртуального кластера. При последующей работе в уже настроенной системе создавать и работать с томами можно в любой необходимый момент времени.

Правами на создание и работу с томами обладают **Администратор ВЦОД** и **Разработчик ВМ**.

12.3. NFS. Настроить хранилище

1. Добавьте информацию о новом подключении

> **Примечание**
>
> Команда производит только запись о подключении в БД. Монтирования при ее выполнении не происходит

```
storage nfs mount add --name <NAME>
                      --nfs_server <NFS_SERVER>
                      --shared_folder_path <SHARED_FOLDER_PATH>
                      [--mount_point <MOUNT_POINT>]
                      [--params <PARAMS>]
                      --nodes <NODES>
                      --ns <NS>
                      [--autocheck <AUTOCHECK>]
                      [--check_timeout <CHECK_TIMEOUT>]
                      [--descr <DESCR>]
```

где

- `name` — название подключения;
- `nfs_server` — IP-адрес сервера NFS;

- `shared_folder_path` — директория на сервере, которая будет смонтирована;
- `mount_point` — точка монтирования на одном или нескольких узлах. Если не указана, то формируется автоматически как `/nfs_shares/<cluster>/<ns>/<name>`, если в параметре `ns` указан определенный ВЦОД, и `/nfs_shares/<cluster>/<name>`, если в параметре `ns` указаны все ВЦОД;
- `params` — параметры, которые должны быть записаны в файл `/etc/fstab`. Если не указаны, то автоматически используется строка `defaults`;
- `ns` — список ВЦОД, которые должны иметь доступ к этому подключению. Можно указать конкретные ВЦОД или все сразу;
- `nodes` — список узлов, на которые нужно будет монтировать директорию;
- `autocheck` — нужно ли будет проводить автоматическую проверку доступности для этого подключения после монтирования;
- `check_timeout` — с какой периодичностью нужно проводить автоматическую проверку. Можно указать любое значение между 60 и 300 секундами. Значение по умолчанию — 300;
- `descr` — описание подключения.

2. Просмотрите список всех подключений

```
storage nfs mount list
```

3. Чтобы просмотреть подробную информацию об определенном подключении, введите

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

где `uuid` — идентификатор подключения из БД.

4. Выполните монтирование директории на узлы

```
storage nfs mount connect --uuid <UUID>
```

где `uuid` — идентификатор подключения из БД.

Монтирование производится сразу на все узлы, указанные при создании подключения. На каждом узле в файл `/etc/fstab` записывается информация о подключении и выполняется команда `mount <mount_point>`.

5. Чтобы обновить настройки подключения, введите

> ❌ **Внимание**
>
> Команду обновления можно выполнять только для размонтированного подключения

```
storage nfs mount update --uuid <UUID>
                         [--name <NAME>]
                         [--nfs_server <NFS_SERVER>]
                         [--shared_folder_path <SHARED_FOLDER_PATH>]
                         [--mount_point <MOUNT_POINT>]
                         [--params <PARAMS>]
                         [--autocheck <AUTOCHECK>]
                         [--check_timeout <CHECK_TIMEOUT>]
                         [--descr <DESCR>]
```

где

- `uuid` — идентификатор подключения из БД;
- `name` — название подключения;
- `nfs_server` — IP-адрес сервера NFS;
- `shared_folder_path` — директория на сервере, которая будет смонтирована;
- `mount_point` — точка монтирования на одном или нескольких узлах;
- `params` — параметры, которые должны быть записаны в файл `/etc/fstab`;
- `autocheck` — нужно ли будет проводить автоматическую проверку доступности для этого подключения после монтирования;
- `check_timeout` — с какой периодичностью нужно проводить автоматическую проверку. Можно указать любое значение между 60 и 300 секундами. Значение по умолчанию — 300;
- `descr` — описание подключения.

6. Добавить новые узлы к уже смонтированному подключению

```
storage nfs mount nodes add --nodes <NODES>
                            --uuid <UUID>
```

где

- `nodes` — список узлов, которые нужно добавить;
- `uuid` — идентификатор подключения из БД.

При выполнении команды монтирование директории на новых узлах выполнится автоматически.

7. Удаление узлов из подключения

> ✖ **Внимание**
>
> Команду можно выполнить только для несмонтированного подключения

```
storage nfs mount nodes del --nodes <NODES>
                            --uuid <UUID>
```

где

- `nodes` — список узлов, которые нужно удалить;
- `uuid` — идентификатор подключения из БД.

8. Тестировать доступность подключения

```
storage nfs mount test --uuid <UUID>
                       [--node <NODES>]
```

где

- `uuid` — идентификатор подключения из БД;
- `node` — узел, на котором нужно провести проверку. Если узел не указан, тест выполнится на всех подключенных узлах.

Тест подключения на каждом узле состоит из двух этапов: выполнения команды `findmnt <mount_point>` и теста на запись в точке монтирования `timeout 10 dd if=/dev/zero bs=4096 count=20 > <mount_point>/test-<node_uuid>`.
Статус подключения и ошибки выполнения теста сохраняются в БД.
Если при добавлении подключения был указан параметр `autocheck`, то тест подключения будет выполняться автоматически с указанной периодичностью в `check_timeout`.

9. Размонтировать подключение

> ❌ **Внимание**
>
> Подключение можно размонтировать только после удаления всех ВЦОД из него

```
storage nfs mount disconnect --uuid <UUID>
```

где `uuid` — идентификатор подключения из БД.

Размонтирование происходит на всех подключенных узлах сразу. На каждом узле из файла `/etc/fstab` удаляется запись о подключении и выполняется команда `umount <mount_point>`.

10. Удалить подключение из БД

> ❌ **Внимание**
>
> Подключение можно удалить только после размонтирования

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

где `uuid` — идентификатор подключения из БД.

11. Добавить новые ВЦОД к подключению

```
storage nfs mount ns add --ns <NS>
                         --uuid <UUID>
```

где

- `uuid` — идентификатор подключения из БД;
- `ns` — список ВЦОД, которые нужно добавить.

12. Удалить ВЦОД из подключения

> **Внимание**
>
> Команду можно выполнить только для несмонтированного подключения

```
storage nfs mount ns del --ns <NS>
                         --uuid <UUID>
```

где

- `uuid` — идентификатор подключения из БД;
- `ns` — список ВЦОД, которые нужно удалить.

> **Примечание**
>
> Работа с NFS-томами описана в Руководстве пользователя в командной строке

13. Структура ВЦОД

Ресурсы логического кластера разделяются на виртуальные единицы: множество ВЦОД — виртуальных центров обработки данных (см. статью Общие сведения).

По умолчанию у ВЦОД управления имеется корневая директория, в которой размещены **Администратор кластера** и **Администратор безопасности кластера**.

Далее в пределах ВЦОД управления создаются поддиректории, изолированные друг от друга. При этом Администратор кластера, находясь в корневой директории ВЦОД управления, имеет доступ ко всем ресурсам пользователей, находящихся в поддиректориях ВЦОД управления.

Директории ВЦОД управления — это средство управления доступом пользователей к объектам, находящимся внутри этих директорий.

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

---

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.

14. Уведомления

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**.

14.1. 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>
   ```

14.2. Сформировать список адресов

1. Чтобы добавить адреса, введите в командной строке

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

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>
```

14.3. 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_n
ame>
```

2. Запрос информации о всех шаблонах сообщений

```
notif template list
```

3. Запрос информации об определенном шаблоне

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

14.4. 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>
```

14.5. 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.

14.6. 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.

15. Права доступа и роли

15.1. 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.

15.2. 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.

Во ВЦОД управления по умолчанию создаются две роли: администратор кластера и администратор безопасности кластера.

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.

15.2.1. Роли по умолчанию

АДМИНИСТРАТОР КЛАСТЕРА

Функции **Администратора кластера**:

- настройка параметров и операции с программными модулями (плагинами) Sharx Base;
- настройка параметров безопасности и интеграции Sharx Base с каталогом LDAP;
- настройка ролей во ВЦОД управления;
- делегирование и отзыв прав доступа во ВЦОД;
- управление сессиями доступа во ВЦОД управления;
- создание, удаление, настройка параметров учетных записей пользователей ВЦОД управления;
- настройка параметров использования ресурсов кластера;
- просмотр журналов и конфигурационных параметров журналирования событий Sharx Base;
- настройка параметров и выполнение операций с сетевым стеком и виртуальными сетями платформы виртуализации;
- настройка параметров и выполнение операций с журналами событий узлов кластера;
- настройка параметров механизма уведомлений и рассылок Sharx Base;
- перевод, удаление узлов кластера из сервисного режима;
- выполнение операций с метками и их привязка к ВЦОД;
- настройка параметров и выполнение операций с ключами и сертификатами;
- настройка параметров работы Sharx Base с СХД (РСХД, Libvirt, NFS);
- настройка параметров и выполнение операций по обновлению ПО Sharx Base;
- настройка параметров и выполнение операции с лицензиями;
- просмотр параметров и результатов операций контроля целостности;
- настройка параметров и выполнение операций мониторинга.

АДМИНИСТРАТОР БЕЗОПАСНОСТИ КЛАСТЕРА

Функции **Администратора безопасности кластера**:

- просмотр всех конфигурационных параметров Sharx Base;
- настройка параметров и выполнение операций контроля целостности;
- настройка параметров журналирования событий и просмотр событий;
- выполнение операций мониторинга.

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.

15.2.2. Действия с ролями

Администратор кластера может управлять ролями в рамках ВЦОД управления.

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

16. Пользователи ВЦОД управления

> **Важно**
>
> При передаче пользователю контроля над ВЦОД управления формируется специальный пользователь Технической поддержки ООО «ШарксДЦ»: **sharx_tech**. У него точно такие же права, как и у Администратора кластера

Во ВЦОД управления по умолчанию созданы два пользователя на основе ролей **Администратор кластера** и **Администратор безопасности кластера**. При необходимости можно создать дополнительных пользователей.

16.1. Создать пользователя

Примечание

В статье описано ручное создание пользователя.

Также Sharx Base поддерживает создание пользователей с использованием LDAP. Подробная инструкция описана в Руководстве пользователя в командной строке.

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>
```

16.2. 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

17. 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. В данном случае — это ВЦОД управления

18. 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>
```

19. 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)

Параметр ВЦОД	**Описание**
`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. 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`	

Параметр ВЦОД	**Описание**
	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: • *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`	

Параметр ВЦОД	Описание
	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
`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:

Параметр ВЦОД	Описание
	• *no* — выключена (значение по умолчанию), • *yes* — включена
`whitelist_networks`	Список белых адресов, с которых можно подключаться к Sharx Base

20. 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.

Параметр ВЦОД	Описание
	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
`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*

Параметр ВЦОД	Описание
`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
`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

21. Валидация объектов 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.

21.1. Управление ключами ВЦОД управления

1. Чтобы посмотреть список ключей во ВЦОД управления, 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.

21.2. 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 кластера** 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