ОПЕРАЦИОННАЯ СИСТЕМА АЛЬТ СЕРВЕР ВИРТУАЛИЗАЦИИ 10.4

Описание функциональных характеристик

Содержание

1 ОБЩИЕ СВЕДЕНИЯ ОБ ОС АЛЬТ СЕРВЕР ВИРТУАЛИЗАЦИИ 10.4

1.1 Краткое описание возможностей

Операционная система «Альт Сервер Виртуализации»/«Альт Виртуализация» (далее – ОС «Альт Сервер Виртуализации»), представляет собой совокупность интегрированных программ, созданных на основе ОС «Linux», и обеспечивает обработку, хранение и передачу информации в круглосуточном режиме эксплуатации. «Альт Виртуализация» – альтернативное название дистрибутива.

ОС «Альт Сервер Виртуализации» – серверный дистрибутив, нацеленный на предоставление функций виртуализации в корпоративной инфраструктуре. Дистрибутив включает в себя средства виртуализации:

- вычислений (ЦПУ и память);
- сети;
- хранения данных.

Управление системой виртуализации возможно через командный интерфейс, веб-интерфейс, с использованием API.

ОС «Альт Сервер Виртуализации» представляет собой решение уровня предприятия, позволяющее осуществить миграцию на импортозамещающее программное и аппаратное обеспечение.

ОС «Альт Сервер Виртуализации» обладает следующими функциональными характеристиками:

- обеспечивает возможность обработки, хранения и передачи информации;
- обеспечивает возможность функционирования в многозадачном режиме (одновременное выполнение множества процессов);
- обеспечивает возможность масштабирования системы: возможна эксплуатация ОС как на одной ПЭВМ, так и в информационных системах различной архитектуры;
- обеспечивает многопользовательский режим эксплуатации;
- обеспечивает поддержку мультипроцессорных систем;
- обеспечивает поддержку виртуальной памяти;
- обеспечивает поддержку запуска виртуальных машин;
- обеспечивает сетевую обработку данных, в том числе разграничение доступа к сетевым пакетам.

ОС «Альт Сервер Виртуализации» предоставляет 4 основных типа установки:

- «Базовый гипервизор». Включает в себя поддержку виртуализации KVM на уровне ядра Linux, утилиты запуска виртуальных машин qemu и унифицированный интерфейс создания и настройки виртуального окружения libvirt. Устанавливается на отдельно стоящий сервер или группу независимых серверов. Для управления используются интерфейс командной строки virsh или графическое приложение virt-manager на рабочей станции администратора.
- «Кластер серверов виртуализации на основе проекта PVE». Устанавливается на группу серверов (до 32 штук). Предназначено для управления виртуальным окружением KVM и контейнерами LXC, виртуальным сетевым окружением и хранилищем данных. Для управления используется интерфейс командной строки, а также веб-интерфейс. Возможна интеграция с корпоративными системами аутентификации (AD, LDAP и другие на основе PAM).
- «Облачная виртуализация уровня предприятия на основе проекта OpenNebula». Для использования необходимы 1 или 3 и более серверов управления (могут быть виртуальными), и группа серверов для запуска виртуальных окружений KVM или контейнеров LXC. Возможна интеграция с корпоративными системами аутентификации.
- «Контейнерная виртуализация». Для использования предлагаются Docker, Podman или LXC. Для построения кластера и управления контейнерами возможно использование Kubernetes.

ОС «Альт Сервер Виртуализации» поддерживает клиент-серверную архитектуру и может обслуживать процессы как в пределах одной компьютерной системы, так и процессы на других ПЭВМ через каналы передачи данных или сетевые соединения.

1.2 Структура программных средств

ОС «Альт Сервер Виртуализации» состоит из набора компонентов, предназначенных для реализации функциональных задач, необходимых пользователям (должностным лицам для выполнения определённых должностных инструкций, повседневных действий), и поставляется в виде дистрибутива и комплекта эксплуатационной документации.

В структуре ОС «Альт Сервер Виртуализации» можно выделить следующие функциональные элементы:

- ядро ОС;
- системные библиотеки;
- утилиты и драйверы;
- средства обеспечения информационной безопасности;
- системные приложения;

- средства обеспечения облачных и распределенных вычислений, средства виртуализации и системы хранения данных;
- системы мониторинга и управления;
- средства подготовки исполнимого кода;
- средства версионного контроля исходного кода;
- библиотеки подпрограмм (SDK);
- среды разработки, тестирования и отладки;
- интерактивные рабочие среды;
- программные серверы;
- веб-серверы;
- системы управления базами данных;
- командные интерпретаторы.

Ядро ОС «Альт Сервер Виртуализации» управляет доступом к оперативной памяти, сети, дисковым и прочим внешним устройствам. Оно запускает и регистрирует процессы, управляет разделением времени между ними, реализует разграничение прав и определяет политику безопасности, обойти которую, не обращаясь к нему, нельзя.

Ядро работает в режиме «супервизора», позволяющем ему иметь доступ сразу ко всей оперативной памяти и аппаратной таблице задач. Процессы запускаются в «режиме пользователя»: каждый жестко привязан ядром к одной записи таблицы задач, в которой, в числе прочих данных, указано, к какой именно части оперативной памяти этот процесс имеет доступ. Ядро постоянно находится в памяти, выполняя системные вызовы – запросы от процессов на выполнение этих подпрограмм.

Системные библиотеки – наборы программ (пакетов программ), выполняющие различные функциональные задачи и предназначенные для динамического подключения к работающим программам, которым необходимо выполнение этих задач.

ОС «Альт Сервер Виртуализации» предоставляет набор дополнительных служб, востребованных в инфраструктуре виртуализации любой сложности и архитектуры:

- сервер сетевой файловой системы NFS;
- распределённая сетевая файловая система CEPH;
- распределённая сетевая файловая система GlusterFS;
- поддержка iSCSI как в качестве клиента, так и создание сервера;
- сетевые службы DNS и DHCP;
- виртуальный сетевой коммутатор Open vSwitch;
- служба динамической маршрутизации bird с поддержкой протоколов BGP, OSPF и др.;
- сетевой балансировщик нагрузки HAProxy, keepallived;

- веб-серверы Apache и Nginx.

В ОС «Альт Сервер Виртуализации» входят агенты мониторинга (Zabbix, telegraf, Prometheus) и архивирования (Bacula, UrBackup), которые могут использоваться совместно с сервисами на ОС «Альт Сервер».

2 ЗАГРУЗКА ОПЕРАЦИОННОЙ СИСТЕМЫ

2.1 Настройка загрузки

Вызов ОС «Альт Сервер Виртуализации», установленной на жесткий диск, происходит автоматически и выполняется после запуска ПЭВМ и отработки набора программ BIOS. ОС «Альт Сервер Виртуализации» вызывает специальный загрузчик.

Загрузчик настраивается автоматически и включает в свое меню все системы, установку которых на ПЭВМ он определил. Поэтому загрузчик также может использоваться для вызова других ОС, если они установлены на компьютере.

Примечание. При наличии на компьютере нескольких ОС (или при наличии нескольких вариантов загрузки), оператор будет иметь возможность выбрать необходимую ОС (вариант загрузки). В случае если пользователем ни один вариант не был выбран, то по истечении заданного времени будет загружена ОС (вариант загрузки), заданные по умолчанию.

При стандартной установке ОС «Альт Сервер Виртуализации» в начальном меню загрузчика доступны несколько вариантов загрузки (Рис. 1): обычная загрузка, загрузка с дополнительными параметрами (например, «recovery mode» – загрузка с минимальным количеством драйверов), загрузка в программу проверки оперативной памяти (memtest).

Варианты загрузки

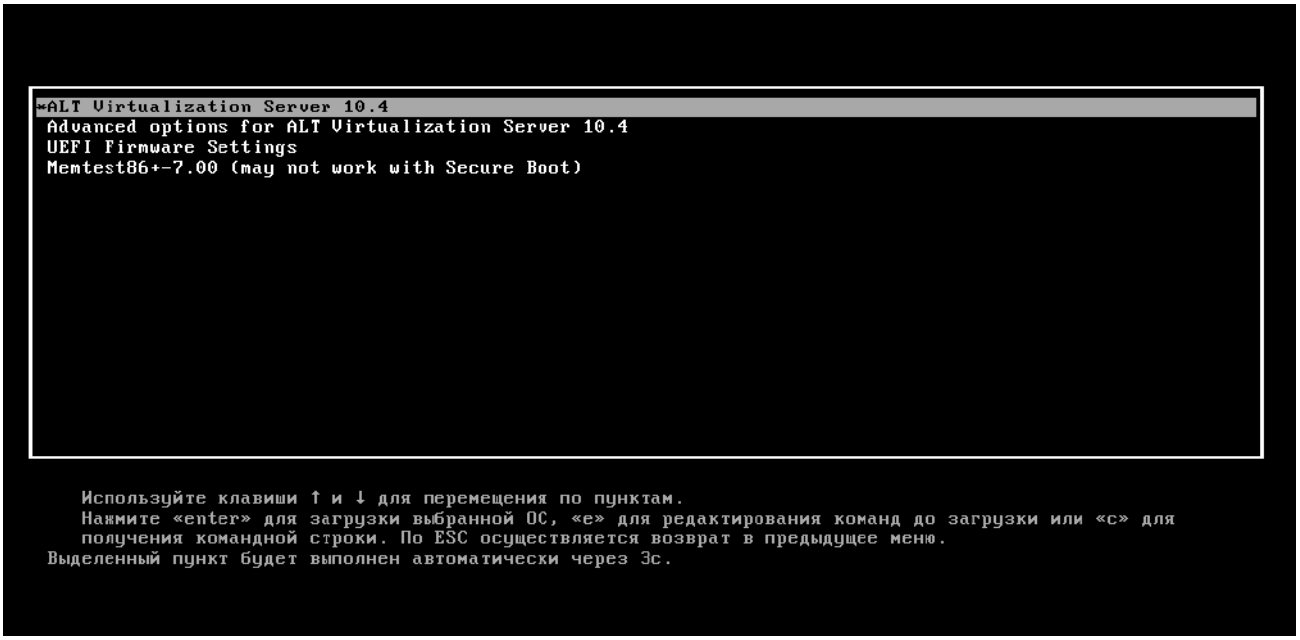


Рис. 1

По умолчанию, если не были нажаты управляющие клавиши на клавиатуре, загрузка ОС «Альт Сервер Виртуализации» продолжится автоматически после небольшого

времени ожидания (обычно несколько секунд). Нажав клавишу <Enter>, можно начать загрузку немедленно.

Для выбора дополнительных параметров загрузки нужно выбрать пункт «Дополнительные параметры для ALT Virtualization Server 10.4» («Advanced options for ALT Virtualization Server 10.4»).

Для выполнения тестирования оперативной памяти нужно выбрать пункт «Memtest86+-7.00».

Нажатием клавиши <E> можно вызвать редактор параметров загрузчика GRUB и указать параметры, которые будут переданы ядру ОС при загрузке.

П р и м е ч а н и е . Если при установке системы был установлен пароль на загрузчик, потребуется ввести имя пользователя «boot» и заданный на шаге «Установка загрузчика» пароль.

В процессе загрузки ОС «Альт Сервер Виртуализации» пользователь может следить за информацией процесса загрузки, которая отображает этапы запуска различных служб и программных серверов в виде отдельных строк (Рис. 2), на экране монитора.

Загрузка ОС

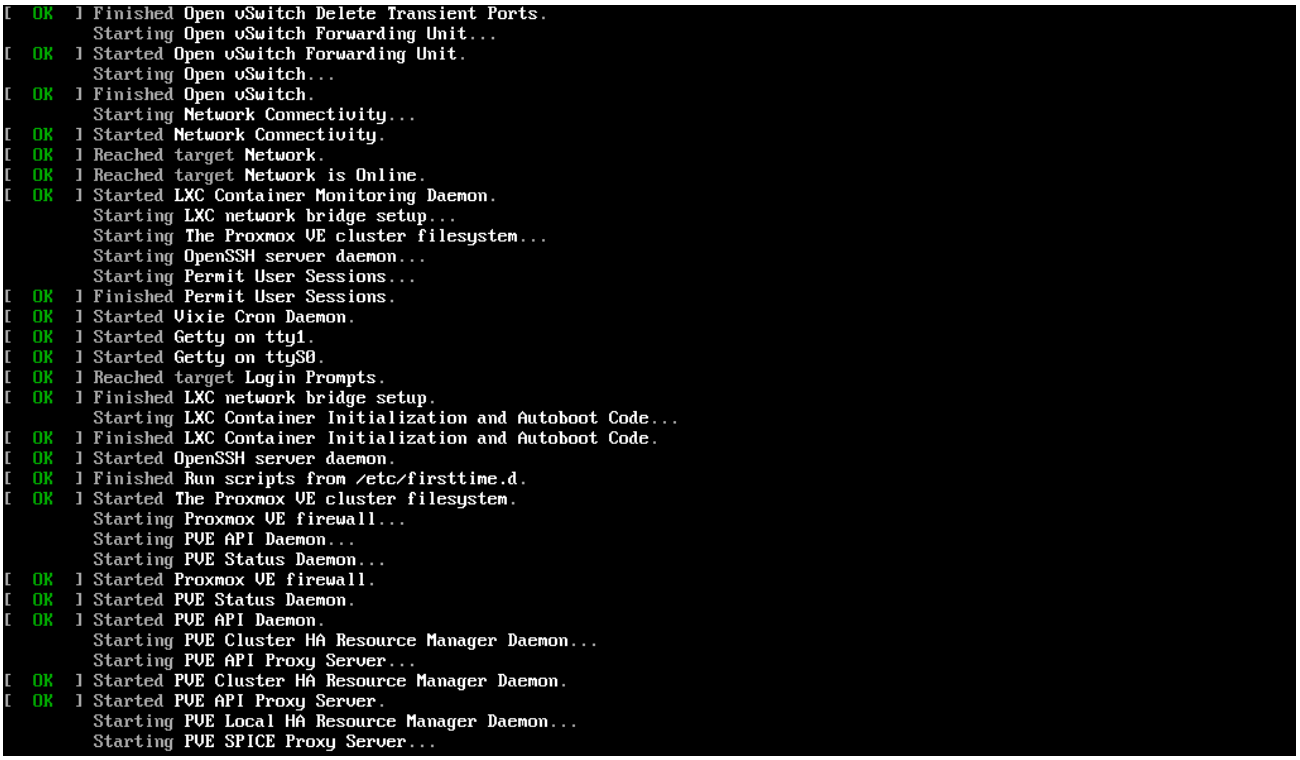


Рис. 2

При этом каждая строка начинается словом вида [XXXXXXX] (FAILED или OK), являющегося признаком нормального или ненормального завершения этапа загрузки. Слово XXXXXXX=FAILED (авария) свидетельствует о неуспешном завершении этапа загрузки, что требует вмешательства и специальных действий администратора системы.

Загрузка ОС может занять некоторое время, в зависимости от производительности компьютера. Основные этапы загрузки операционной системы – загрузка ядра, подключение (монтирование) файловых систем, запуск системных служб – периодически могут дополняться проверкой файловых систем на наличие ошибок. В этом случае время ожидания может занять больше времени, чем обычно.

2.2 Получение доступа к зашифрованным разделам

В случае если был создан шифрованный раздел, потребуется вводить пароль при обращении к этому разделу.

Например, если был зашифрован домашний раздел `/home`, то для того, чтобы войти в систему, потребуется ввести пароль этого раздела и затем нажать <Enter>.

Примечание. Если не ввести пароль за отведенный промежуток времени, то загрузка системы завершится ошибкой. В этом случае следует перезагрузить систему, нажав для этого два раза <Enter>, а затем клавиши <Ctrl>+<Alt>+<Delete>.

2.3 Вход и работа в системе в консольном режиме

Стандартная установка ОС «Альт Сервер Виртуализации» включает базовую систему, работающую в консольном режиме.

При загрузке в консольном режиме работа загрузчика ОС «Альт Сервер Виртуализации» завершается запросом на ввод логина и пароля учетной записи. Для дальнейшего входа в систему необходимо ввести логин и пароль учетной записи пользователя. (Рис. 3).

Приглашение для ввода команд

Рис. 3

В случае успешного прохождения процедуры аутентификации и идентификации будет выполнен вход в систему. ОС «Альт Сервер Виртуализации» перейдет к штатному режиму работы и предоставит дальнейший доступ к консоли

Примечание. После загрузки будут показаны имя и IP-адрес компьютера, а также, если были установлены OpenNebula или PVE, адрес доступа к панели управления (Рис. 4).

IP-адрес компьютера и адрес панели управления PVE

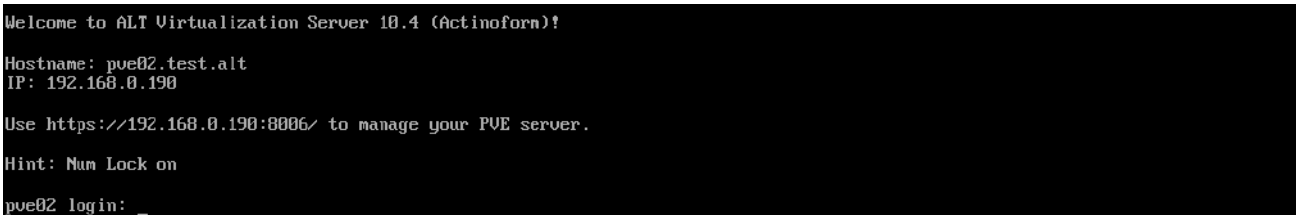


Рис. 4

2.4 Виртуальная консоль

В процессе работы ОС «Альт Сервер Виртуализации» активно несколько виртуальных консолей. Каждая виртуальная консоль доступна по одновременному нажатию клавиш <Ctrl>, <Alt> и функциональной клавиши с номером этой консоли от <F1> до <F6>.

На первых шести виртуальных консолях (от <Ctrl>+<Alt>+<F1> до <Ctrl>+<Alt>+<F6>) пользователь может зарегистрироваться и работать в текстовом режиме. Двенадцатая виртуальная консоль (<Ctrl>+<Alt>+<F12>) выполняет функцию системной консоли – на нее выводятся сообщения о происходящих в системе событиях.

3 OPENNEBULA

OpenNebula – это платформа облачных вычислений для управления разнородными инфраструктурами распределенных центров обработки данных. Платформа OpenNebula управляет виртуальной инфраструктурой центра обработки данных для создания частных, общедоступных и гибридных реализаций инфраструктуры как службы.

Облачная архитектура определяется тремя элементами: хранилищем данных, сетью и системой виртуализации.

OpenNebula состоит из следующих компонентов:

- Сервер управления (Front-end) – на нём выполняются сервисы OpenNebula;
- Серверы с виртуальными машинами;
- Хранилище данных – содержит образы ВМ;
- Физическая сеть – обеспечивает связь между хранилищем данных, серверами с ВМ, поддерживает VLAN-ы для ВМ, а также управление сервисами OpenNebula.

Примечание. Компоненты OpenNebula будут установлены в систему, если при установке дистрибутива выбрать профиль «Вычислительный узел Opennebula KVM», «Вычислительный узел Opennebula LXC» или «Сервер Opennebula».

3.1 Планирование ресурсов

3.1.1 Сервер управления

Минимальные требования к серверу управления показаны в таблице 1.

Таблица 1 – Минимальные требования к серверу управления

Ресурс	Минимальное значение
Оперативная память	2 ГБ
CPU	1 CPU (2 ядра)
Диск	100 ГБ
Сеть	2 интерфейса

Максимальное количество серверов (узлов виртуализации), управляемых одним сервером управления, зависит от инфраструктуры, особенно от производительности хранилища. Обычно рекомендуется не управлять более чем 500 серверами из одной точки, хотя существуют примеры с более чем 1000 серверами.

3.1.2 Серверы виртуализации

Серверы виртуализации – это физические машины, на которых выполняются виртуальные машины. Подсистема виртуализации – это компонент, который отвечает за связь с гипервизором,

установленным на узлах, и выполнение действий, необходимых для каждого этапа жизненного цикла виртуальной машины (ВМ).

Серверы (узлы) виртуализации имеют следующие характеристики и их рекомендованные значения:

- CPU – в обычных условиях каждое ядро, предоставляемое ВМ, должно быть реальным ядром физического процессора. Например, для обслуживания 40 ВМ с двумя процессорами в каждой, облако должно иметь 80 физических ядер. При этом они могут быть распределены по разным серверам: 10 серверов с восемью ядрами или 5 серверов с 16 ядрами на каждом. В случае перераспределения недостаточных ресурсов используются атрибуты CPU и VCPU: CPU определяет физические ядра, выделенные для ВМ, а VCPU – виртуальные ядра для гостевой ОС;
- Память – по умолчанию, OpenNebula не предоставляет памяти для гостевых систем больше, чем есть на самом деле. Желательно рассчитывать объём памяти с запасом в 10% на гипервизор. Например, для 45 ВМ с 2 ГБ памяти на каждой, необходимо 90 ГБ физической памяти. Важным параметром является количество физических серверов: каждый сервер должен иметь 10% запас для работы гипервизора, так, 10 серверов с 10 ГБ памяти на каждом могут предоставить по 9 ГБ для виртуальных машин и смогут обслужить 45 машин из этого примера (10% от 10 ГБ = 1 ГБ на гипервизор).

3.1.3 Хранилище данных

OpenNebula работает с двумя видами данных в хранилище: образцами виртуальных машин и образами (дисками) самих ВМ.

В хранилище образов (Images Datastore) OpenNebula хранит все зарегистрированные образы, которые можно использовать для создания ВМ.

Системное хранилище (System Datastore) – используется для хранения дисков виртуальных машин, работающих в текущий момент. Образы дисков перемещаются, или клонируются, в хранилище образов или из него при развертывании и отключении ВМ, при подсоединении или фиксировании мгновенного состояния дисков.

Одним из основных способов управления хранилищем данных является ограничение хранилища, доступного для пользователей, путем определения квот по максимальному количеству ВМ, а также максимального объема энергозависимой памяти, который может запросить пользователь, и обеспечения достаточного пространства хранения системных данных и образов, отвечающего предельным установленным квотам. OpenNebula позволяет администратору добавлять хранилища системных данных и образов.

Планирование хранилища – является критически важным аспектом, поскольку от него зависит производительность облака. Размер хранилищ сильно зависит от базовой технологии. Например, при использовании Ceph для среднего по размеру облака, необходимо взять как минимум 3 сервера в следующей конфигурации: 5 дисков по 1 ТБ, 16 ГБ памяти, 2 CPU по 4 ядра в каждом и как минимум 2 сетевые карты.

3.1.4 Сетевая инфраструктура

Сетевая инфраструктура должна быть спланирована так, чтобы обеспечить высокую надёжность и пропускную способность. Рекомендуется использовать два сетевых интерфейса на сервере управления и по четыре на каждом сервере виртуализации (публичный, внутренний, для управления и для связи с хранилищем).

3.2 Запуск сервера управления OpenNebula

3.2.1 Установка пароля для пользователя oneadmin

При установке OpenNebula система автоматически создает нового пользователя oneadmin, все дальнейшие действия по управлению OpenNebula необходимо выполнять от этого пользователя.

Примечание. Файл `/var/lib/one/.one/one_auth` будет создан со случайно сгенерированным паролем. Необходимо поменять этот пароль перед запуском OpenNebula.

Для установки пароля для пользователя oneadmin необходимо выполнить команду:

```
# passwd oneadmin
```

Теперь зайдя под пользователем oneadmin, следует заменить содержимое `/var/lib/one/.one/one_auth`. Он должен содержать следующее: `oneadmin: <пароль>`. Например:

```
$ echo "oneadmin:mypassword" > ~/.one/one_auth
```

3.2.2 Настройка MySQL (MariaDB) для хранения конфигурации

По умолчанию OpenNebula работает с SQLite. Если планируется использовать OpenNebula с MySQL, следует настроить данную конфигурацию перед первым запуском OpenNebula, чтобы избежать проблем с учетными данными oneadmin и serveradmin.

Примечание. Задать пароль root для mysql и настройки безопасности:

```
# mysql_secure_installation
```

Создать нового пользователя, предоставить ему привилегии в базе данных opennebula (эта база данных будет создана при первом запуске OpenNebula) и настроить уровень изоляции:

```
$ mysql -u root -p
Enter password:

MariaDB > GRANT ALL PRIVILEGES ON opennebula.* TO 'oneadmin' IDENTIFIED BY
'<thepassword>';
```

```
Query OK, 0 rows affected (0.003 sec)

MariaDB > SET GLOBAL TRANSACTION ISOLATION LEVEL READ COMMITTED;
Query OK, 0 rows affected (0.001 sec)

MariaDB > quit
```

Перед запуском сервера OpenNebula в первый раз необходимо настроить параметры доступа к базе данных в конфигурационном файле `/etc/one/oned.conf`:

```
#DB = [ BACKEND = "sqlite" ]
#       TIMEOUT = 2500 ]
# Sample configuration for MySQL
DB = [ BACKEND = "mysql",
       SERVER  = "localhost",
       PORT    = 0,
       USER    = "oneadmin",
       PASSWD  = "<thepassword>",
       DB_NAME = "opennebula",
       CONNECTIONS = 25,
       COMPARE_BINARY = "no" ]
```

3.2.3 Запуск OpenNebula

Для добавления в автозапуск и запуска OpenNebula необходимо выполнить следующие команды:

```
# systemctl enable –now opennebula
# systemctl enable –now opennebula-sunstone
```

3.2.4 Проверка установки

После запуска OpenNebula в первый раз, следует проверить, что команды могут подключаться к демону OpenNebula. Это можно сделать в командной строке или в графическом интерфейсе пользователя: Sunstone.

В командной строке:

```
$ oneuser show
USER 0 INFORMATION
ID              : 0
NAME            : oneadmin
GROUP           : oneadmin
PASSWORD        : 3bc15c8aae3e4124dd409035f32ea2fd6835efc9
AUTH_DRIVER     : core
ENABLED         : Yes
USER TEMPLATE
TOKEN_PASSWORD="ec21d27e2fe4f9ed08a396cbd47b08b8e0a4ca3c"
VMS USAGE & QUOTAS
```

```
VMS USAGE & QUOTAS - RUNNING
DATASTORE USAGE & QUOTAS
NETWORK USAGE & QUOTAS
IMAGE USAGE & QUOTAS
```

Затем можно попробовать войти в веб-интерфейс Sunstone. Для этого необходимо перейти по адресу http://<внешний адрес>:9869. Если все в порядке, будет предложена страница входа (Рис. 5).

Страница авторизации opennebula-sunstone

Рис. 5

Необходимо ввести в соответствующие поля имя пользователя (oneadmin) и пароль пользователя (тот, который находится в файле `/var/lib/one/.one/one_auth`).

После входа в систему будет доступна панель инструментов (Рис. 6).

Для смены языка интерфейса необходимо в левом меню выбрать пункт «Settings», и на открывшейся странице в выпадающем списке «Language» выбрать пункт «Russian (ru_RU)» (Рис. 7). Язык интерфейса будет изменён на русский (Рис. 8).

Панель инструментов opennebula-sunstone

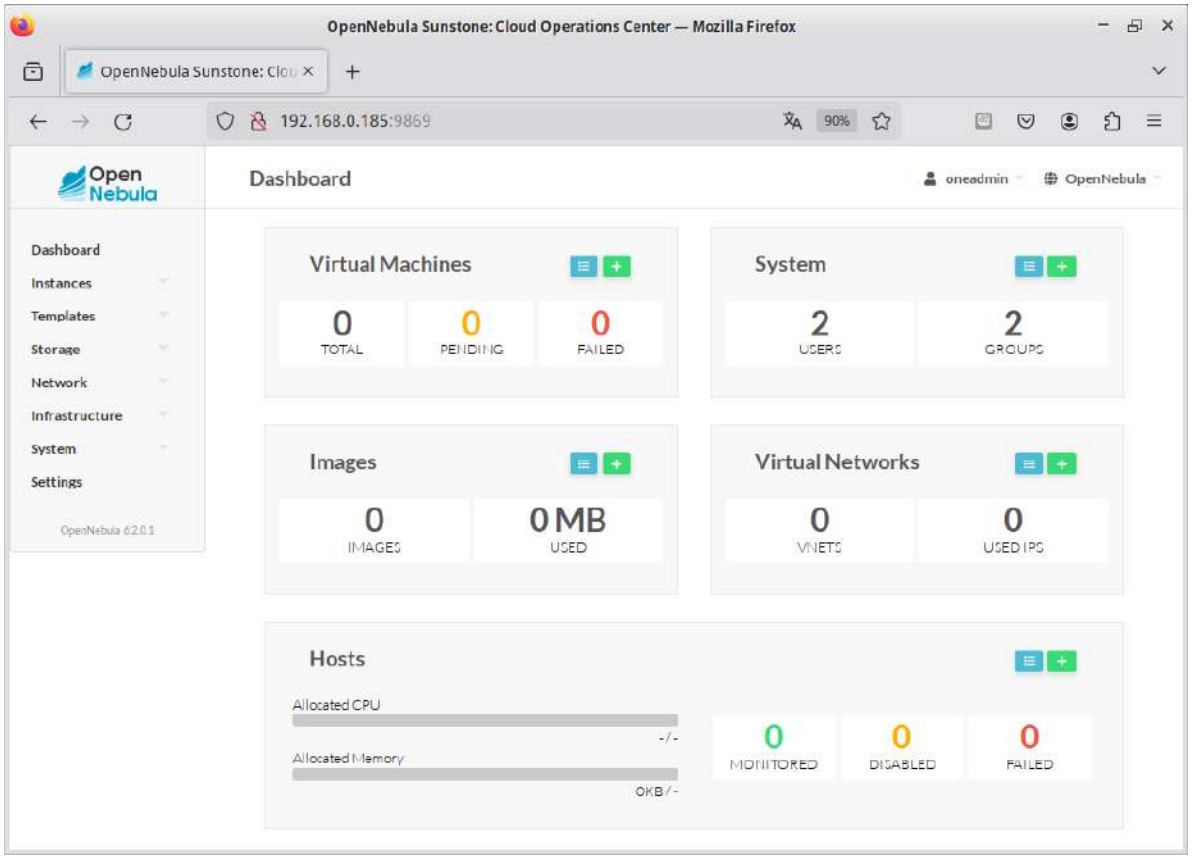


Рис. 6

Выбор языка интерфейса

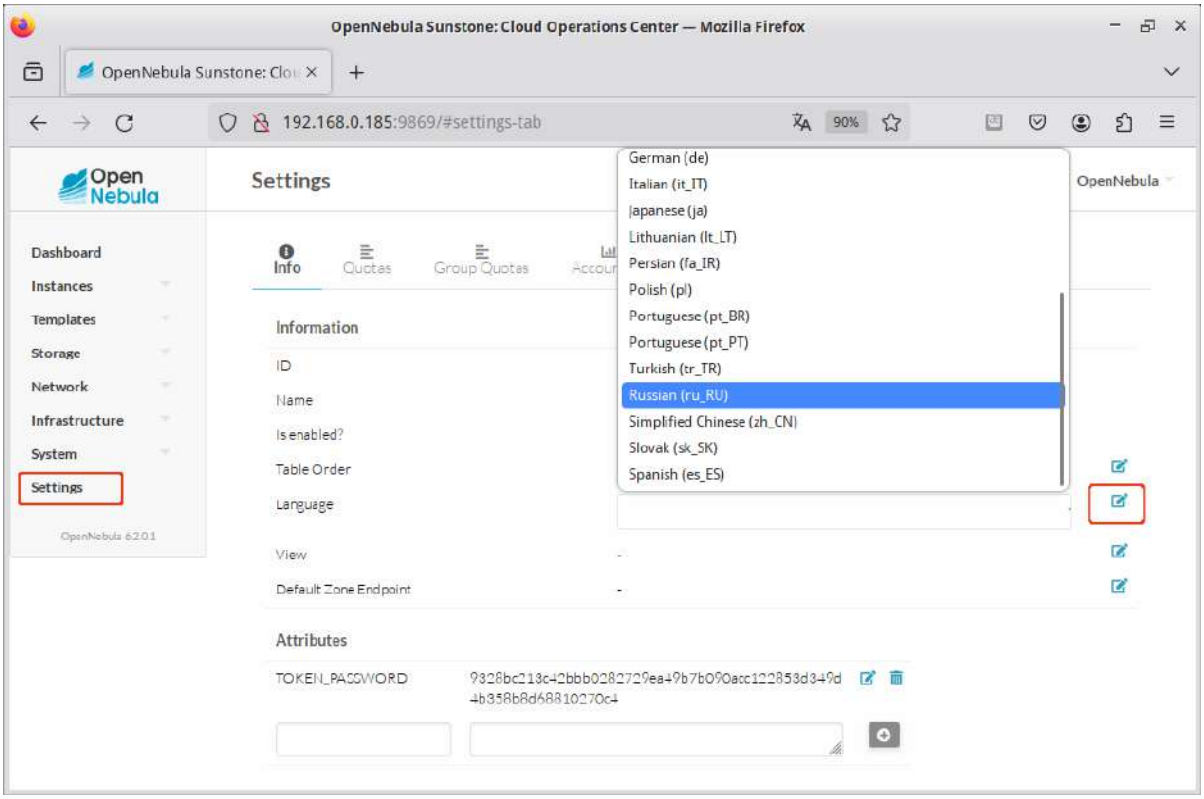


Рис. 7

Панель инструментов opennebula-sunstone с русским языком интерфейса

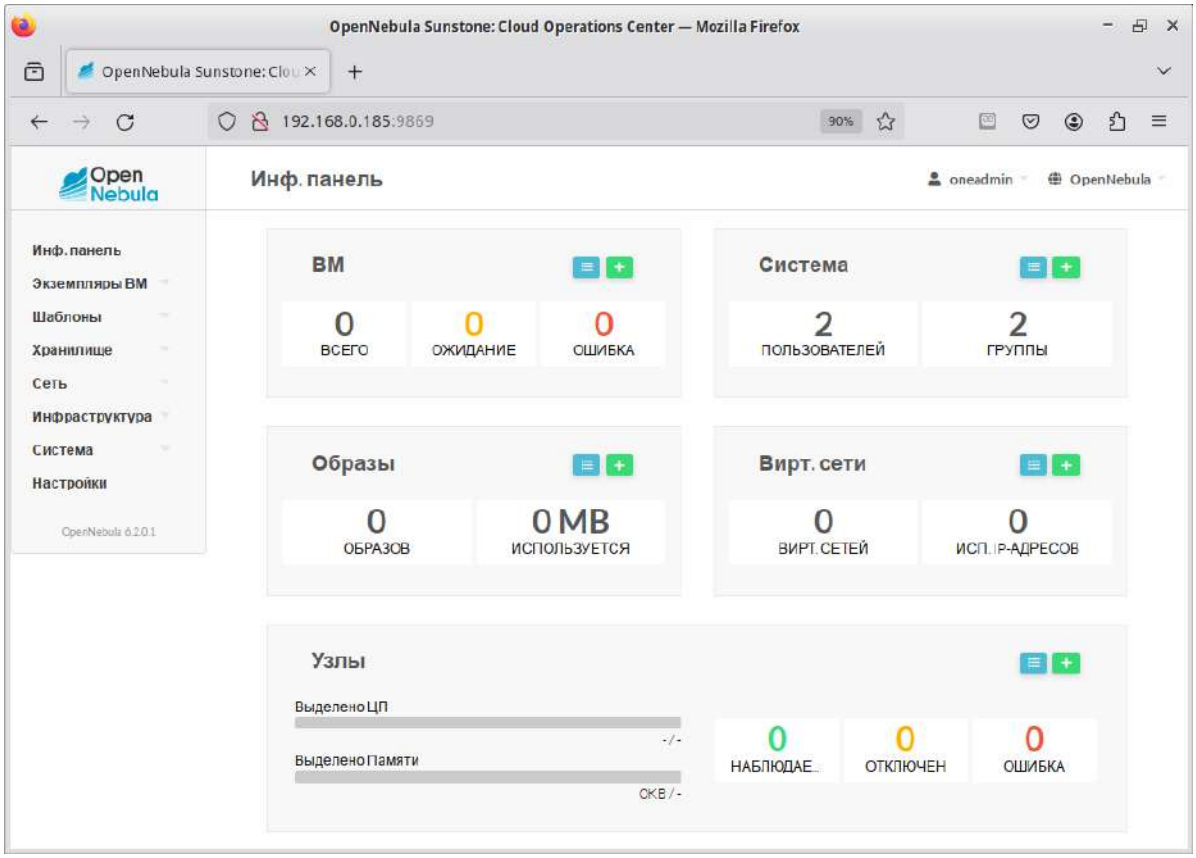


Рис. 8

3.2.5 Ключи для доступа по SSH

Сервер управления OpenNebula подключается к хостам гипервизора по SSH. Необходимо распространить открытый ключ пользователя oneadmin со всех машин в файл `/var/lib/one/.ssh/authorized_keys` на всех машинах.

При установке сервера управления OpenNebula ключ SSH был сгенерирован и добавлен в авторизованные ключи. Необходимо синхронизировать id_rsa, id_rsa.pub и authorized_keys сервера управления и узлов. Кроме того, следует создать файл known_hosts и также синхронизировать его с узлами. Чтобы создать файл known_hosts, необходимо выполнить следующую команду (от пользователя oneadmin на сервере управления) со всеми именами узлов и именем сервера управления в качестве параметров:

```
$ ssh-keyscan <сервер_управления> <узел1> <узел2> <узел3> ... >>
/var/lib/one/.ssh/known_hosts
```

Примечание. Команду ssh-keyscan необходимо выполнить, как для имён, так и для IP-адресов узлов:

```
$ ssh-keyscan 192.168.0.185 server 192.168.0.190 host-01 >>
/var/lib/one/.ssh/known_hosts
```

Далее необходимо скопировать каталог `/var/lib/one/.ssh` на все узлы. Самый простой способ – установить временный пароль для oneadmin на всех хостах и скопировать каталог с сервера управления:

```
$ scp -rp /var/lib/one/.ssh <узел1>:/var/lib/one/
$ scp -rp /var/lib/one/.ssh <узел2>:/var/lib/one/
$ scp -rp /var/lib/one/.ssh <узел3>:/var/lib/one/
...
```

После этого следует убедиться, что ни одно из этих подключений (под пользователем oneadmin) не заканчивается ошибкой, и пароль не запрашивается:

- от сервера управления к самому серверу управления;
- от сервера управления ко всем узлам;
- от всех узлов на все узлы;
- от всех узлов к серверу управления.

Эту проверку можно выполнить, например, на сервере управления:

```
# от сервера управления к самому серверу управления
ssh <сервер_управления>
exit

# от сервера управления к узлу, обратно на сервер управления и к другим узлам
ssh <узел1>
ssh <сервер_управления>
exit
ssh <узел2>
exit
ssh <узел3>
exit
exit
```

И так далее для всех узлов.

Если требуется дополнительный уровень безопасности, можно хранить закрытый ключ только на сервере управления, а не копировать его на весь гипервизор. Таким образом, пользователь oneadmin в гипервизоре не сможет получить доступ к другим гипервизорам. Это достигается путем изменения `/var/lib/one/.ssh/config` на сервере управления и добавления параметра `ForwardAgent` к хостам гипервизора для пересылки ключа:

```
$ cat /var/lib/one/.ssh/config
 Host host-01
    User oneadmin
    ForwardAgent yes
 Host host-02
    User oneadmin
```

```
ForwardAgent yes
```

3.2.6 Конфигурация сети

Сервисам, работающим на сервере управления, необходим доступ к узлам с целью управления гипервизорами и их мониторинга, а также для передачи файлов образов. Для этой цели рекомендуется использовать выделенную сеть.

Примечание. Настройка сети необходима только на серверах с виртуальными машинами. Точное имя ресурсов (br0, br1 и т.д.) значения не имеет, но важно, чтобы мосты и сетевые карты имели одно и то же имя на всех узлах.

3.3 Настройка узлов

3.3.1 Установка и настройка узла OpenNebula KVM

Перед добавлением узла типа KVM на сервер OpenNebula следует настроить узел KVM.

Для создания узла типа KVM при установке дистрибутива нужно выбрать профиль «Вычислительный узел Opennebula KVM» (Рис. 9).

Установка сервера виртуализации KVM

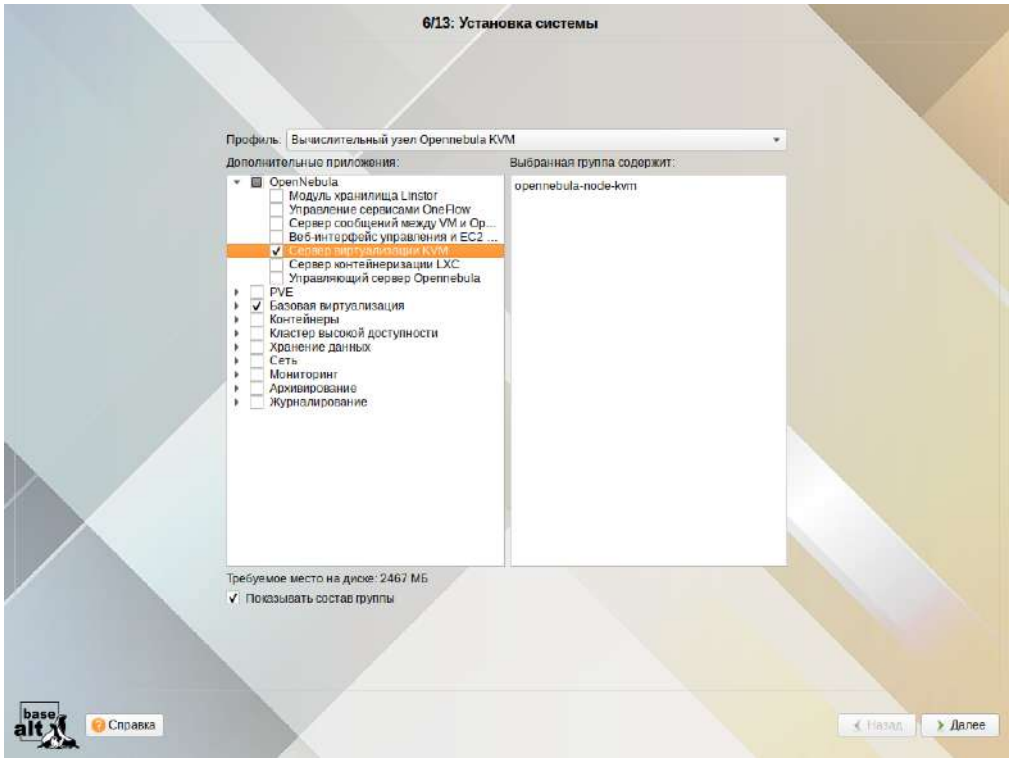


Рис. 9

Примечание. Для создания узла типа KVM в уже установленной системе необходимо выполнить следующие шаги:

- установить пакет opennebula-node-kvm:

```
# apt-get install opennebula-node-kvm
```

- добавить службу libvirtd в автозапуск и запустить её:

```
# systemctl enable --now libvirtd
```

После создания узла следует задать пароля для пользователя oneadmin:

```
# passwd oneadmin
```

и настроить доступ по SSH (см. раздел «Ключи для доступа по SSH»).

3.3.2 Настройка узла OpenNebula LXC

LXC – это гипервизор LXC контейнеров.

Примечание. Для работы с LXC в Opennebula должна быть настроена пара хранилищ (хранилище образов и системное) НЕ типа qcow2 (например, shared или ssh).

Перед добавлением хоста типа LXC на сервер OpenNebula следует настроить узел LXC.

Для создания узла типа LXC, при установке дистрибутива нужно выбрать профиль «Вычислительный узел Opennebula LXC» (Рис. 10).

Установка сервера контейнеризации LXC

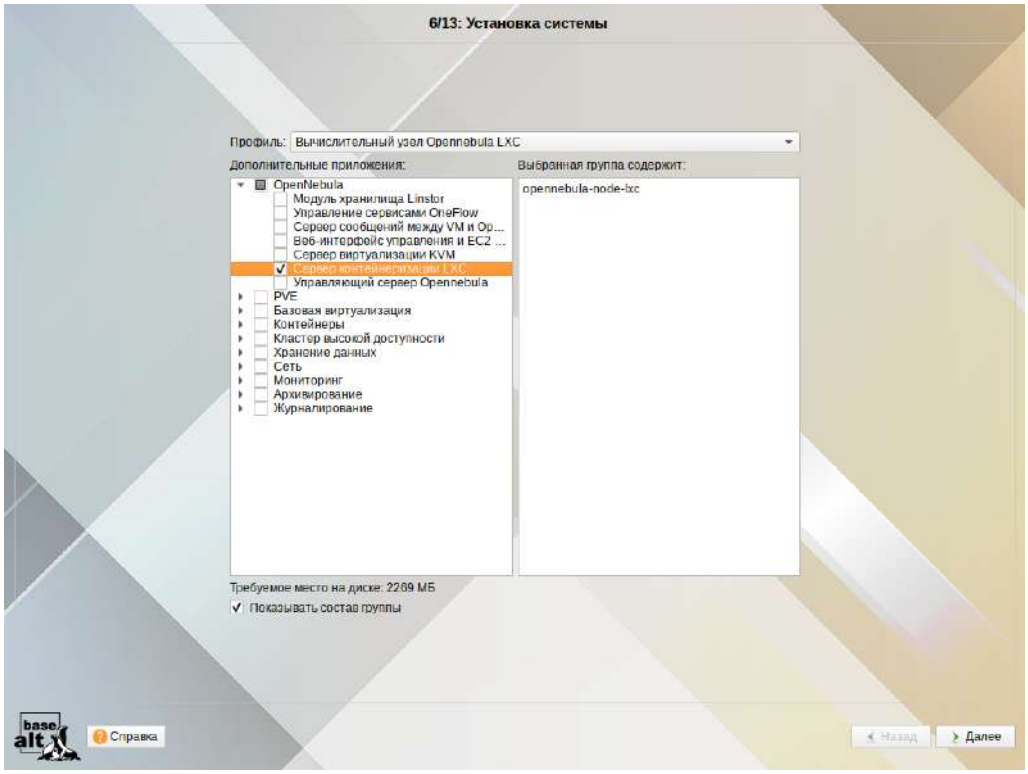


Рис. 10

Примечание. Для создания узла типа LXC в уже установленной системе необходимо выполнить следующие шаги:

- установить пакет opennebula-node-lxc:

```
# apt-get install opennebula-node-lxc
```

- запустить и добавить в автозапуск lxc.socket:

```
# systemctl enable --now lxc
```

После создания узла следует задать пароля для пользователя oneadmin:

```
# passwd oneadmin
```

и настроить доступ по SSH (см. раздел Ключи для доступа по SSH).

3.4 Добавление узлов в OpenNebula

Чтобы использовать существующие физические узлы, их необходимо зарегистрировать в OpenNebula. Регистрация узла в OpenNebula может быть выполнена в командной строке или в веб-интерфейсе Sunstone.

Примечание. Перед добавлением узла следует убедиться, что к узлу можно подключиться по SSH без запроса пароля.

3.4.1 Добавление узла типа KVM в OpenNebula-Sunstone

Для добавления узла, необходимо в левом меню выбрать «Инфраструктура» → «Узлы» и на загруженной странице нажать кнопку «+» (Рис. 11).

Добавление узла в OpenNebula-Sunstone

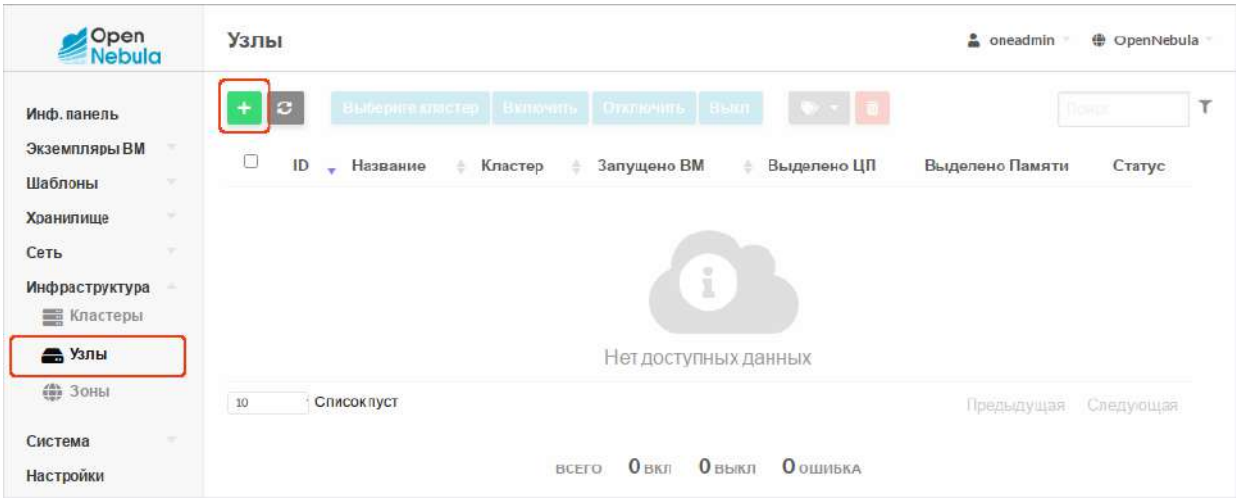


Рис. 11

Далее необходимо указать тип виртуализации, заполнить поле «Имя хоста» (можно ввести IP-адрес узла или его имя) и нажать кнопку «Создать» (Рис. 12).

Добавление узла типа KVM в OpenNebula-Sunstone

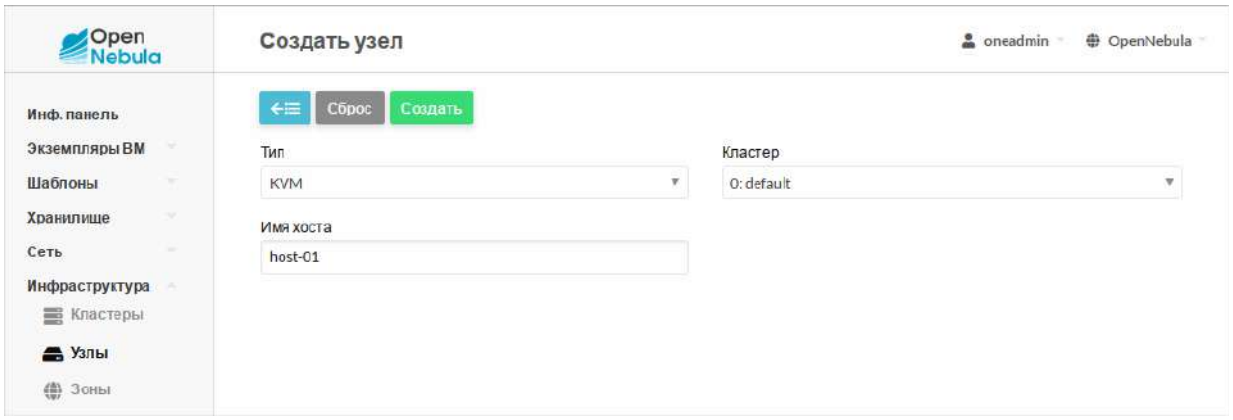


Рис. 12

Затем следует вернуться к списку узлов и убедиться, что узел перешел в состояние «Вкл» (это должно занять от 20 секунд до 1 минуты, можно нажать кнопку «Обновить» для обновления состояния) (Рис. 13).

Список узлов OpenNebula-Sunstone

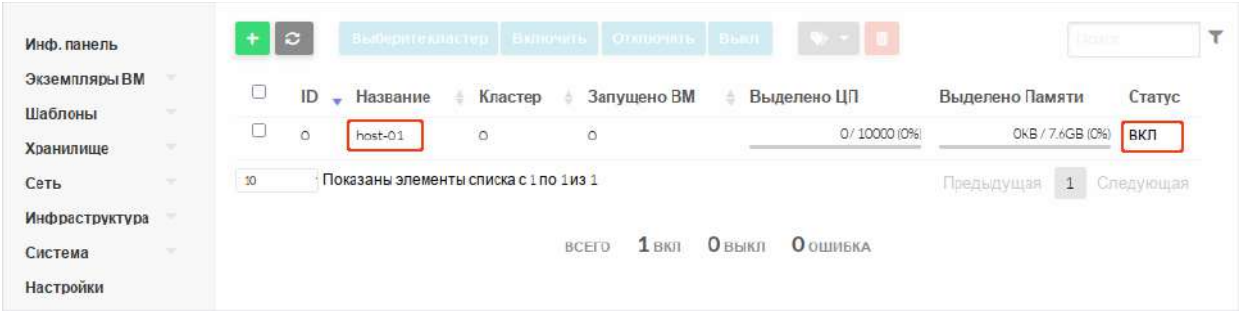


Рис. 13

3.4.2 Добавление узла типа LXC в OpenNebula-Sunstone

Для добавления узла типа LXC на сервере OpenNebula, необходимо в левом меню выбрать «Инфраструктура» → «Узлы» и на загруженной странице нажать кнопку «+».

Далее необходимо указать тип виртуализации – LXC, заполнить поле «Имя хоста» (можно ввести IP-адрес узла или его имя) и нажать кнопку «Создать» (Рис. 14).

Добавление узла типа LXC в OpenNebula-Sunstone

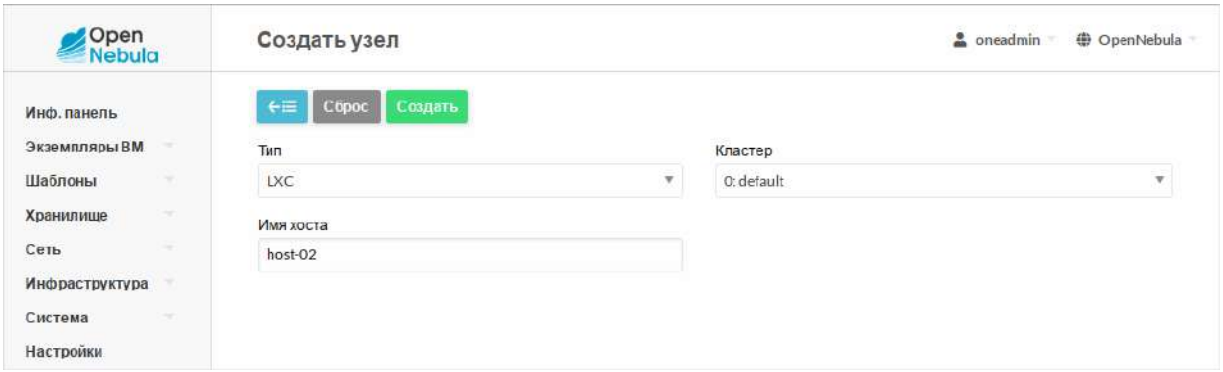


Рис. 14

Затем следует вернуться к списку узлов и убедиться, что узел перешел в состояние ВКЛ (это должно занять от 20 секунд до 1 минуты).

3.4.3 Работа с узлами в командной строке

onehost – это инструмент управления узлами в OpenNebula. Описание всех доступных опций утилиты onehost можно получить, выполнив команду:

```
$ man onehost
```

Для добавления узла KVM в облако, необходимо выполнить следующую команду от oneadmin на сервере управления:

```
$ onehost create host-01 --im kvm --vm kvm
ID: 0
```

Добавление узла типа LXC в командной строке:

```
$ onehost create host-02 --im lxc --vm lxc
ID: 1
```

Список узлов можно просмотреть, выполнив команду:

```
$ onehost list
  ID NAME          CLUSTER    TVM      ALLOCATED_CPU      ALLOCATED_MEM STAT
   1 host-02       default      0       0 / 100 (0%)     0K / 945M (0%) on
   0 host-01       default      0     0 / 10000 (0%)     0K / 7.6G (0%) on
```

Примечание. Если возникли проблемы с добавлением узла, то, скорее всего, неправильно настроен SSH. Ошибки можно просмотреть в `/var/log/one/oned.log`.

Для указания узла можно использовать его ID или имя. Например, удаление узла с указанием ID или имени:

```
$ onehost delete 1
$ onehost delete hos-t01
```

Изменение статуса узла:

```
$ onehost disable host-01 // деактивировать узел
$ onehost enable host-01 // активировать узел
$ onehost offline host-01 // полностью выключить узел
```

Просмотр информации об узле:

```
$ onehost show host-01
```

Вывод данной команды содержит:

- общую информацию об узле;
- информацию о процессоре и объёме оперативной памяти (Host Shares);
- информацию о локальном хранилище данных (Local System Datastore), если узел настроен на использование локального хранилища;
- данные мониторинга;
- информацию о ВМ, запущенных на узле.

3.5 Виртуальные сети

OpenNebula позволяет создавать виртуальные сети, отображая их поверх физических.

При запуске новой ВМ OpenNebula подключит свои виртуальные сетевые интерфейсы (определяемые атрибутами NIC) к сетевым устройствам гипервизора так, как это определено в соответствующей виртуальной сети. Это позволит ВМ иметь доступ к публичным и частным сетям.

onevnet – инструмент управления виртуальными сетями в OpenNebula. Описание всех доступных опций утилиты `onevnet` можно получить, выполнив команду:

```
$ man onevnet
```

Вывести список виртуальных сетей можно, выполнив команду:

```
$ onevnet list
  ID USER     GROUP    NAME            CLUSTERS   BRIDGE                LEASES
   2 oneadmin oneadmin VirtNetwork     0          onebr2                0
   0 oneadmin oneadmin LAN             0          vmbr0                 1
```

Вывести информацию о сети:

```
$ onevnet show 0
```

Создавать, редактировать, удалять и просматривать информацию о виртуальных сетях можно в веб-интерфейсе (Рис. 15).

Виртуальные сети

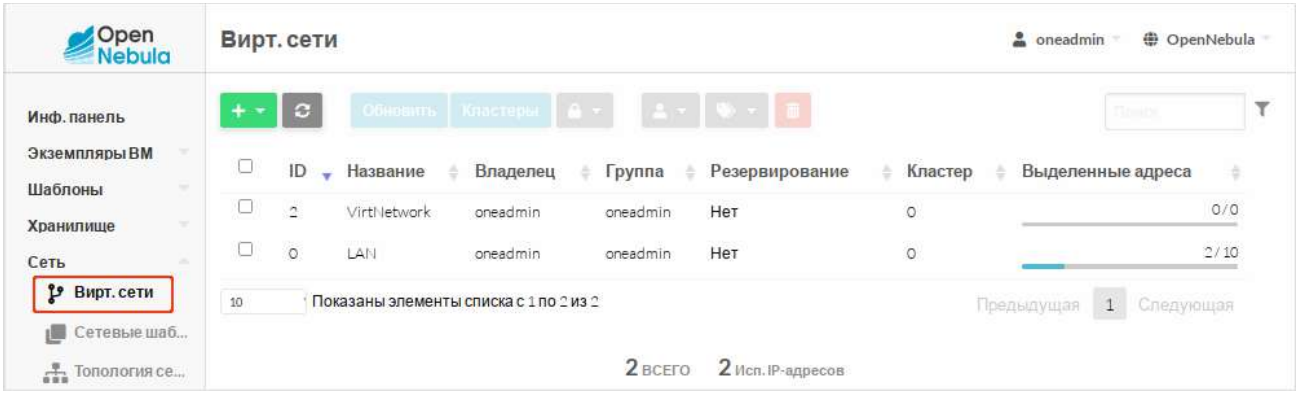


Рис. 15

OpenNebula поддерживает следующие сетевые режимы:

- Bridged (режим «Сетевой мост») – сетевой адаптер ВМ напрямую соединяется с существующим мостом в узле виртуализации;
- 802.1Q (режим «VLAN») – сетевой адаптер ВМ соединяется с существующим мостом в узле виртуализации, а виртуальная сеть настраивается для изоляции VLAN 802.1Q;
- VXLAN – сетевой адаптер ВМ соединяется с существующим мостом в узле виртуализации, а виртуальная сеть реализует изоляцию с помощью инкапсуляции VXLAN;
- Open vSwitch – сетевой адаптер ВМ соединяется с существующим мостом Open vSwitch в узле виртуализации, а виртуальная сеть дополнительно обеспечивает изоляцию VLAN 802.1Q;
- Open vSwitch–VXLAN – сетевой адаптер ВМ соединяется с существующим мостом Open vSwitch в узле виртуализации, а виртуальная сеть дополнительно обеспечивает изоляцию с помощью инкапсуляции VXLAN и, при необходимости, VLAN 802.1Q.

Атрибут VN_MAD виртуальной сети определяет, какой из вышеуказанных сетевых режимов используется.

3.5.1 Режим Bridged

В этом режиме трафик ВМ передается напрямую через мост Linux на узлах гипервизора. Мостовые сети могут работать в трёх различных режимах в зависимости от дополнительной фильтрации трафика, выполняемой OpenNebula:

- «Bridged» – сетевой мост без фильтрации, управляемый мост;
- «Bridged & Security Groups» (сетевой мост с группами безопасности) – для реализации правил групп безопасности устанавливаются правила iptables;
- «Bridged & ebtables VLAN»(сетевой мост с правилами ebtables) – аналогично «Bridged & Security Groups», плюс дополнительные правила ebtables для обеспечения изоляции L2 виртуальных сетей.

При фильтрации трафика необходимо учитывать следующее:

- в режимах «Bridged» и «Bridged & Security Groups» для обеспечения сетевой изоляции можно добавлять тегированные сетевые интерфейсы;
- режим «Bridged with ebtables VLAN» предназначен для небольших сред без соответствующей аппаратной поддержки для реализации VLAN. Данный режим ограничен сетями /24 и IP-адреса не могут перекрываться в виртуальных сетях. Этот режим рекомендуется использовать только в целях тестирования.

На узле виртуализации необходимо создать сетевой мост для каждой сети, в которой будут работать ВМ. При этом следует использовать одно имя сети на всех узлах.

Пример создания виртуальной сети с использованием конфигурационного файла:

1) создать файл `net-bridged.conf` со следующим содержимым:

```
NAME = "VirtNetwork"
VN_MAD = "bridge"
BRIDGE = "vmbr0"
PHYDEV = "enp3s0"

AR=[
    TYPE = "IP4",
    IP   = "192.168.0.140",
    SIZE = "5"
]
```

2) выполнить команду:

```
$ onevnet create net-bridged.conf
ID: 1
```

Параметры виртуальной сети в режиме Bridged приведены в табл. 2.

Т а б л и ц а 2 – Параметры виртуальной сети в режиме Bridged

Параметр	Значение	Обязательный
NAME	Имя виртуальной сети	Да
VN_MAD	Режим: bridge – режим без фильтрации; fw – фильтрация с группами безопасности; ebtables – фильтрация с изоляцией ebtables;	Да
BRIDGE	Имя сетевого моста в узлах виртуализации	Нет
PHYDEV	Имя физического сетевого устройства (на узле виртуализации), которое будет подключено к мосту	Нет
AR	Диапазон адресов, доступных в виртуальной сети	Нет

Пример создания виртуальной сети в веб-интерфейсе:

- в левом меню выбрать пункт «Сеть» → «Вирт. Сети», на загруженной странице нажать кнопку «+» и выбрать пункт «Создать»;
- на вкладке «Общие» указать название виртуальной сети (Рис. 16);
- на вкладке «Конфигурация» указать интерфейс сетевого моста, выбрать режим работы сети (Рис. 17);
- на вкладке «Адреса» указать диапазон IP-адресов, который будет использоваться при выделении IP-адресов для ВМ (Рис. 18);
- нажать кнопку «Создать».

Создание виртуальной сети. Вкладка «Общие»

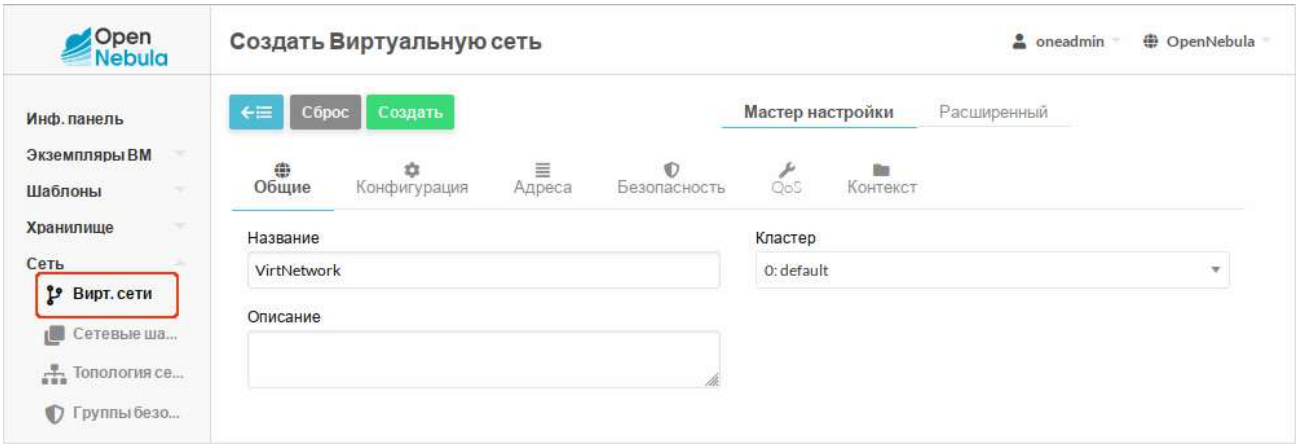


Рис. 16

Примечание. Если в качестве интерфейса моста указать интерфейс, через который производится управление и доступ к узлу, то при запуске ВМ этот интерфейс будет автоматически включен в сетевой мост и соединение с сервером будет потеряно. Поэтому в качестве интерфейса, на котором будут автоматически создаваться сетевые мосты (bridge) для виртуальных сетей, необходимо использовать отдельный сетевой интерфейс (в примере интерфейс enp3s0).

Создание виртуальной сети. Вкладка «Конфигурация»

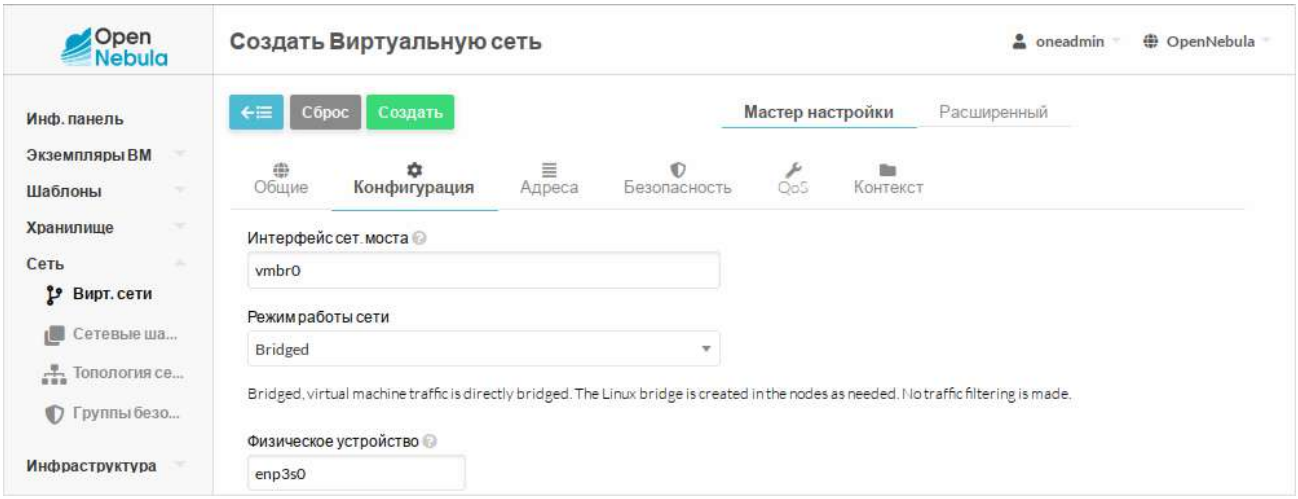


Рис. 17

Создание виртуальной сети. Вкладка «Адреса»

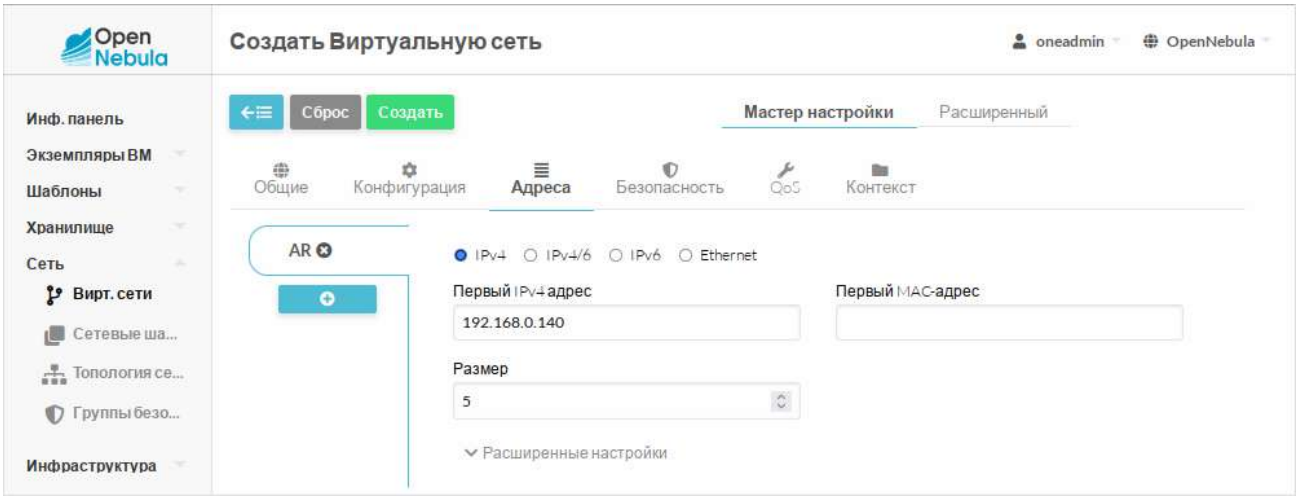


Рис. 18

3.5.2 Режим 802.1Q VLAN

В режиме 802.1Q для каждой виртуальной сети OpenNebula создаётся мост, сетевой интерфейс подключается к этому мосту с тегами VLAN. Этот механизм соответствует стандарту IEEE 802.1Q.

Идентификатор VLAN будет автоматически назначен OpenNebula и будет одинаков для каждого интерфейса в данной сети. Идентификатор VLAN также можно назначить принудительно, указав параметр VLAN_ID в шаблоне виртуальной сети.

Идентификатор VLAN рассчитывается в соответствии со следующим параметром конфигурации `/etc/one/oned.conf`:

```
VLAN_IDS = [
    START    = "2",
    RESERVED = "0, 1, 4095"
]
```

Драйвер сначала попытается выделить VLAN_IDS[START] + VNET_ID где:

- START – первый VLAN_ID, который будет использоваться;
- RESERVED – список VLAN_ID или диапазонов, которые не будут назначаться виртуальной сети (два числа, разделенные двоеточием, обозначают диапазон).

Параметры виртуальной сети в режиме 802.1Q приведены в табл. 3.

Т а б л и ц а 3 – Параметры виртуальной сети в режиме 802.1Q

Параметр	Значение	Обязательный
NAME	Имя виртуальной сети	Да
VN_MAD	802.1Q	Да
BRIDGE	Имя сетевого моста в узлах виртуализации (по умолчанию onebr<net_id> или onebr.<vlan_id>)	Нет
PHYDEV	Имя физического сетевого устройства (на узле виртуализации), которое будет подключено к мосту	Да
VLAN_ID	ID сети VLAN (если не указан, то идентификатор будет сгенерирован, а AUTOMATIC_VLAN_ID будет установлен в YES)	Да (если AUTOMATIC_VLAN_ID = "NO")
AUTOMATIC_VLAN_ID	Генерировать VLAN_ID автоматически	Да (если не указан VLAN_ID)
MTU	MTU для тегированного интерфейса и моста	Нет
AR	Диапазон адресов, доступных в виртуальной сети	Нет

Пример создания виртуальной сети с использованием конфигурационного файла:

1) создать файл `net-vlan.conf` со следующим содержимым:

```
NAME = "VLAN"
VN_MAD = "802.1Q"
BRIDGE = "vmbr1"
PHYDEV = "enp3s0"
AUTOMATIC_VLAN_ID = "Yes"
AR=[
    TYPE = "IP4",
    IP   = "192.168.0.150",
    SIZE = "5"
]
```

2) выполнить команду:

```
$ onevnet create net-vlan.conf
ID: 6
```

Пример создания виртуальной сети в режиме 802.1Q в веб-интерфейсе показан на Рис. 19.

В этом примере драйвер проверит наличие моста vmbr1. Если его не существует, он будет создан. Сетевой интерфейс enp3s0 будет помечен тегом (например, enp3s0.7) и подсоединён к vmbr1.

Создание виртуальной сети в режиме 802.1Q. Вкладка «Конфигурация»

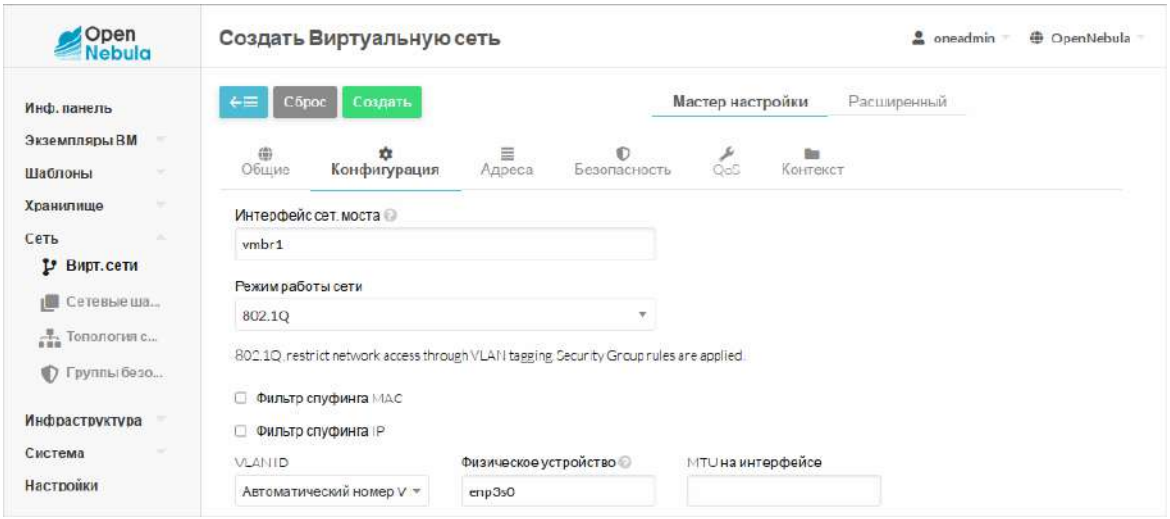


Рис. 19

3.5.3 Режим VXLAN

В режиме VXLAN для каждой виртуальной сети OpenNebula создаётся мост, сетевой интерфейс подключается к этому мосту с тегами VXLAN.

Идентификатор VLAN будет автоматически назначен OpenNebula и будет одинаков для каждого интерфейса в данной сети. Идентификатор VLAN также можно назначить принудительно, указав параметр VLAN_ID в шаблоне виртуальной сети.

С каждой сетью VLAN связывается адрес многоадресной рассылки для инкапсуляции широковещательного и многоадресного трафика L2. По умолчанию данный адрес будет принадлежать диапазону 239.0.0.0/8 в соответствии с RFC 2365 (многоадресная IP-адресация с административной областью). Адрес многоадресной рассылки получается путем добавления значения атрибута VLAN_ID к базовому адресу 239.0.0.0/8.

В данном сетевом режиме задействован стандартный UDP-порт сервера 8472.

Примечание. Сетевой интерфейс, который будет выступать в роли физического устройства, должен иметь IP-адрес.

Начальный идентификатор VXLAN можно указать в файле `/etc/one/oned.conf`:

```
VXLAN_IDS = [
    START = "2"
]
```

Параметры виртуальной сети в режиме VXLAN приведены в табл. 4.

Таблица 4 – Параметры виртуальной сети в режиме VXLAN

Параметр	Значение	Обязательный
NAME	Имя виртуальной сети	Да
VN_MAD	vxlan	Да

Параметр	Значение	Обязательный
BRIDGE	Имя сетевого моста в узлах виртуализации (по умолчанию onebr<net_id> или onebr.<vlan_id>)	Нет
PHYDEV	Имя физического сетевого устройства (на узле виртуализации), которое будет подключено к мосту	Да
VLAN_ID	ID сети VLAN (если не указан, то идентификатор будет сгенерирован, а AUTOMATIC_VLAN_ID будет установлен в YES)	Да (если AUTOMATIC_VLAN_ID = "NO")
AUTOMATIC_VLAN_ID	Генерировать VLAN_ID автоматически	Да (если не указан VLAN_ID)
MTU	MTU для тегированного интерфейса и моста	Нет
AR	Диапазон адресов, доступных в виртуальной сети	Нет

Пример создания виртуальной сети с использованием конфигурационного файла:

1) создать файл `net-vxlan.conf` со следующим содержимым:

```
NAME = "vxlan"
VN_MAD = "vxlan"
BRIDGE = "vxlan50"
PHYDEV = "enp3s0"
VLAN_ID = "50"
AR=[
    TYPE = "IP4",
    IP   = "192.168.0.150",
    SIZE = "5"
]
```

1) выполнить команду:

```
$ onevnet create net-vxlan.conf
ID: 7
```

Пример создания виртуальной сети в режиме VXLAN в веб-интерфейсе показан на Рис. 20.

Создание виртуальной сети в режиме VXLAN. Вкладка «Конфигурация»

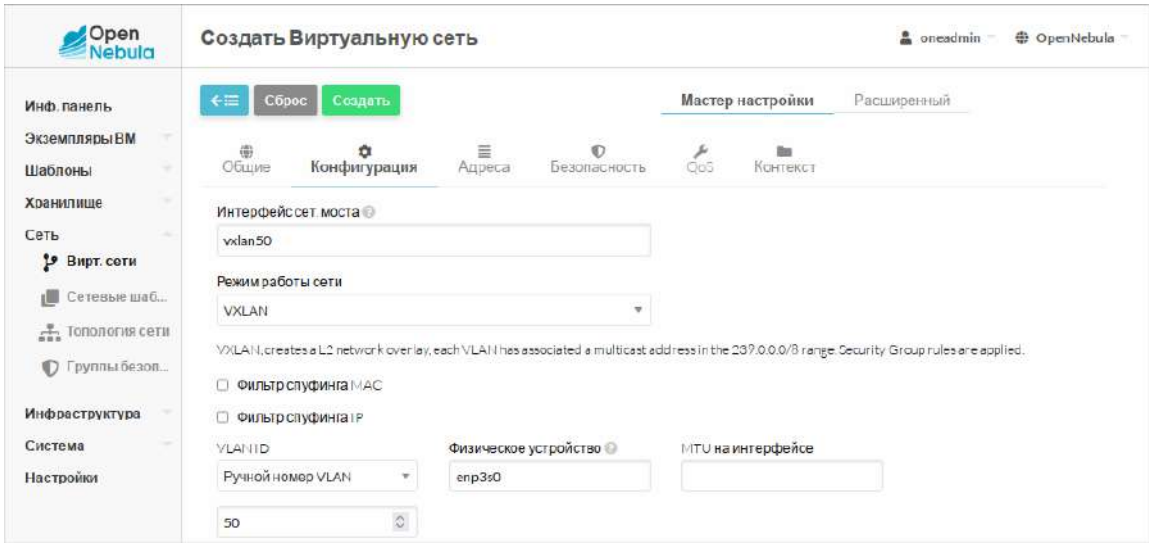


Рис. 20

В этом примере драйвер проверит наличие моста vxlan50. Если его не существует, он будет создан. Сетевой интерфейс enp3s0 будет помечен тегом (enp3s0.10) и подсоединён к vxlan50.

3.5.4 Режим Open vSwitch

Сети Open vSwitch создаются на базе программного коммутатора Open vSwitch.

Примечание. На узлах виртуализации необходимо установить пакет openvswitch:

```
# apt-get install openvswitch
```

А также запустить и добавить в автозагрузку службу Open vSwitch:

```
# systemctl enable --now openvswitch.service
```

Идентификатор VLAN будет автоматически назначен OpenNebula и будет одинаков для каждого интерфейса в данной сети. Идентификатор VLAN также можно назначить принудительно, указав параметр VLAN_ID в шаблоне виртуальной сети.

Идентификатор VLAN рассчитывается в соответствии со следующим параметром конфигурации `/etc/one/oned.conf`:

```
VLAN_IDS = [
    START    = "2",
    RESERVED = "0, 1, 4095"
]
```

Драйвер сначала попытается выделить VLAN_IDS[START] + VNET_ID где:

- START – первый VLAN_ID, который будет использоваться;
- RESERVED – список VLAN_ID или диапазонов, которые не будут назначаться виртуальной сети (два числа, разделенные двоеточием, обозначают диапазон).

Параметры виртуальной сети в режиме Open vSwitch приведены в табл. 5.

Т а б л и ц а 5 – Параметры виртуальной сети в режиме Open vSwitch

Параметр	Значение	Обязательный
NAME	Имя виртуальной сети	Да
VN_MAD	ovswitch	Да
BRIDGE	Имя сетевого моста Open vSwitch	Нет
PHYDEV	Имя физического сетевого устройства (на узле виртуализации), которое будет подключено к мосту	Нет (если не используются VLAN)
VLAN_ID	ID сети VLAN (если не указан, то идентификатор будет сгенерирован, а AUTOMATIC_VLAN_ID будет установлен в YES)	Нет
AUTOMATIC_ VLAN_ID	Генерировать VLAN_ID автоматически (игнорируется, если определен VLAN_ID)	Нет
MTU	MTU для моста Open vSwitch	Нет
AR	Диапазон адресов, доступных в виртуальной сети	Нет

Пример создания виртуальной сети с использованием конфигурационного файла:

1) создать файл `net-ovs.conf` со следующим содержимым:

```
NAME = "OVS"
VN_MAD = "ovswitch"
BRIDGE = "vmbr1"
PHYDEV = "enp3s0"
AR=[
    TYPE = "IP4",
    IP   = "192.168.0.150",
    SIZE = "5"
```

2) выполнить команду:

```
$ onevnet create net-ovs.conf
ID: 7
```

Пример создания виртуальной сети в режиме Open vSwitch в веб-интерфейсе показан на-Рис. 21.

Создание виртуальной сети в режиме Open vSwitch. Вкладка «Конфигурация»

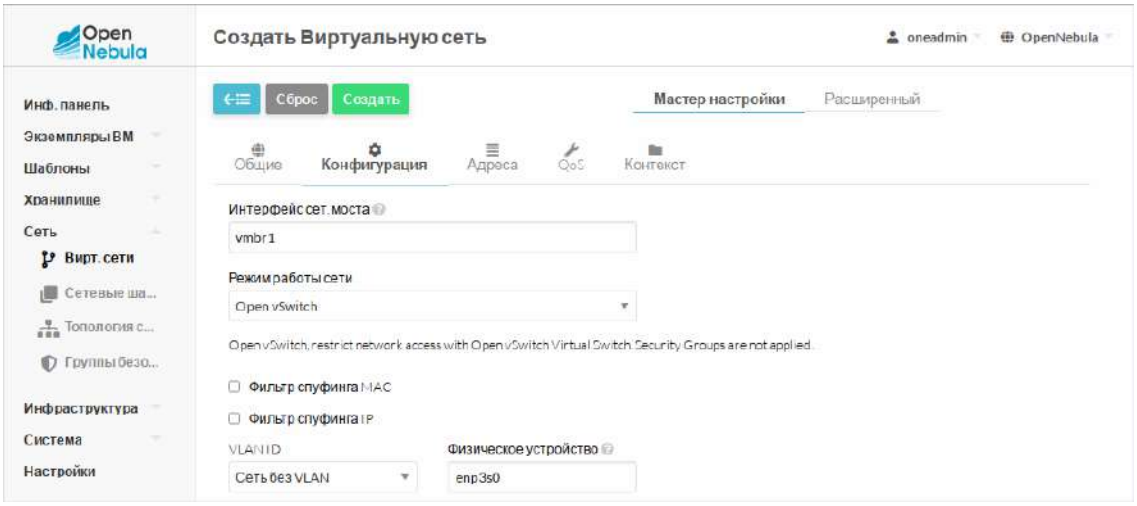


Рис. 21

3.5.5 Режим VXLAN в Open vSwitch

В качестве основы используется оверлейная сеть VXLAN с Open vSwitch (вместо обычного моста Linux). Трафик на самом низком уровне изолируется протоколом инкапсуляции VXLAN, а Open vSwitch обеспечивает изоляцию второго уровня с помощью тегов VLAN 802.1Q внутри инкапсулированного трафика. Основная изоляция всегда обеспечивается VXLAN, а не VLAN 802.1Q. Если для изоляции VXLAN требуется 802.1Q, драйвер необходимо настроить с использованием созданного пользователем физического интерфейса с тегами 802.1Q.

Параметры виртуальной сети в режиме Open vSwitch VXLAN приведены в табл. 6.

Пример создания виртуальной сети с использованием конфигурационного файла:

1) создать файл `net-ovsx.conf` со следующим содержимым:

```
NAME = "private"
VN_MAD = "ovswitch_vxlan"
PHYDEV = "eth0"
BRIDGE = "ovsvxbr0.10000"
OUTER_VLAN_ID = 10000
VLAN_ID = 50
AR=[
    TYPE = "IP4",
    IP   = "192.168.0.150",
    SIZE = "5"
]
```

2) выполнить команду:

```
$ onevnet create net-ovsx.conf
ID: 7
```

Т а б л и ц а 6 – Параметры виртуальной сети в режиме Open vSwitch VXLAN

Параметр	Значение	Обязательный
NAME	Имя виртуальной сети	Да
VN_MAD	ovswitch_vxlan	Да
BRIDGE	Имя сетевого моста Open vSwitch	Нет
PHYDEV	Имя физического сетевого устройства (на узле виртуализации), которое будет подключено к мосту	Нет (если не используются VLAN)
OUTER_VLAN_ID	ID внешней сети VXLAN (если не указан и AUTOMATIC_OUTER_VLAN_ID = "YES", то идентификатор будет сгенерирован)	Да (если AUTOMATIC_OUTER_VLAN_ID = "NO")
AUTOMATIC_OUTER_VLAN_ID	Генерировать ID автоматически (игнорируется, если определен OUTER_VLAN_ID)	Да (если не указан OUTER_VLAN_ID)
VLAN_ID	Внутренний идентификатор VLAN 802.1Q. (если не указан и AUTOMATIC_VLAN_ID = "YES", то идентификатор будет сгенерирован)	Нет
AUTOMATIC_VLAN_ID	Генерировать VLAN_ID автоматически	Нет
MTU	MTU для тегированного интерфейса и моста	Нет
AR	Диапазон адресов, доступных в виртуальной сети	Нет

В этом примере драйвер проверит наличие моста ovsvxbr0.10000. Если его нет, он будет создан. Также будет создан интерфейс VXLAN eth0.10000 и подключен к мосту Open vSwitch ovsvxbr0.10000. При создании экземпляра виртуальной машины ее порты моста будут помечены идентификатором 802.1Q VLAN ID 50.

Пример создания виртуальной сети в режиме Open vSwitch VXLAN в веб-интерфейсе показан на Рис. 22.

Создание виртуальной сети в режиме Open vSwitch VXLAN. Вкладка «Конфигурация»

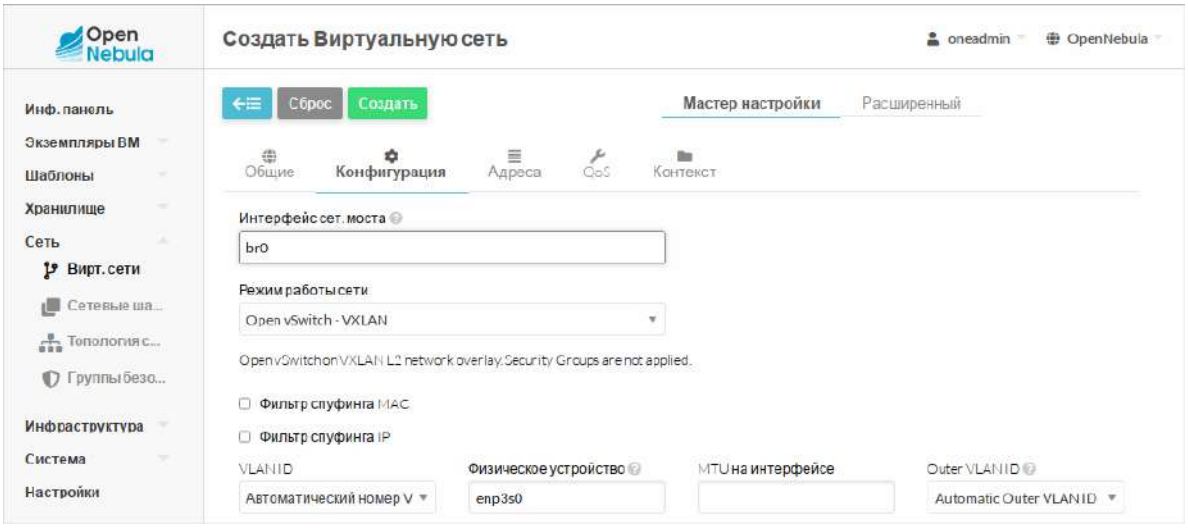


Рис. 22

3.6 Работа с хранилищами в OpenNebula

3.6.1 Типы хранилищ

OpenNebula использует три типа хранилищ данных:

- хранилище образов (Images Datastore) – используется для хранения образов ВМ, которые можно использовать для создания ВМ;
- системное хранилище (System Datastore) – используется для хранения дисков ВМ, работающих в текущий момент. Образы дисков перемещаются, или клонируются, в хранилище образов или из него при развертывании и отключении ВМ, при подсоединении или фиксировании мгновенного состояния дисков;
- хранилище файлов и ядер (Files & Kernels Datastore) – используется для хранения простых файлов, используемых в контекстуализации, или ядер ВМ, используемых некоторыми гипервизорами.

В зависимости от назначения выделяют два типа образов (Рис. 27):

- постоянные (persistent) – предназначены для хранения пользовательских данных (например, БД). Изменения, внесенные в такие образы, будут сохранены после завершения работы ВМ. В любой момент времени может быть только одна ВМ, использующая постоянный образ.
- непостоянные (non-persistent) – используются для хранения дисков ВМ, работающих в текущий момент. Образы дисков копируются, или клонируются, в хранилище образов или из него при развертывании и отключении ВМ, при подсоединении или фиксировании мгновенного состояния дисков. После удаления ВМ копия образа в системном хранилище также удаляется.

Схема взаимодействия хранилищ

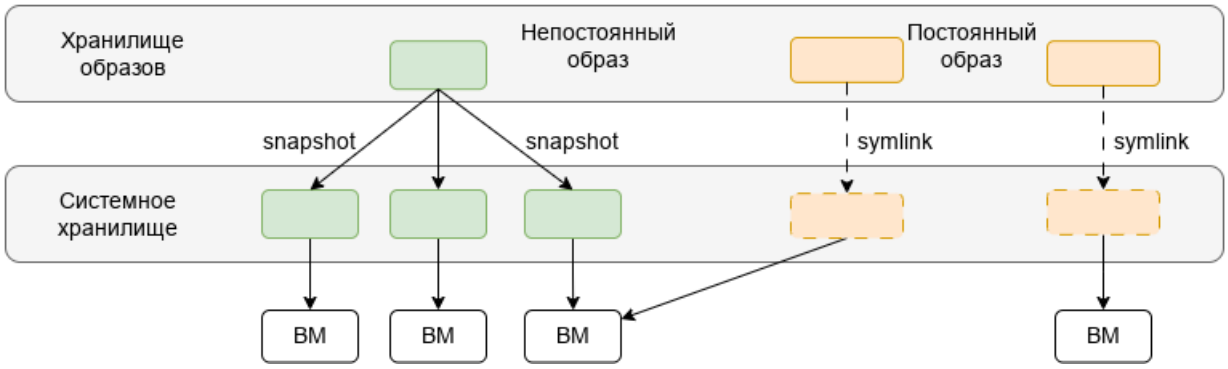


Рис. 23

Образы дисков передаются между хранилищем образов и системным хранилищем с помощью драйверов Transfer Manager (TM). Эти драйверы представляют собой специальные элементы ПО, которые выполняют низкоуровневые операции хранения.

Образы сохраняются в соответствующий каталог хранилища образов (/var/lib/one/datastores/<идентификатор_хранилища>). Кроме того, для каждой работающей ВМ существует каталог /var/lib/one/datastores/<идентификатор_хранилища>/<идентификатор_ВМ> в соответствующем системном хранилище. Эти каталоги содержат диски ВМ и дополнительные файлы, например, контрольные точки или снимки.

Например, система с хранилищем образов (1) с тремя образами и тремя ВМ (ВМ 0 и 2 работают, 7 – остановлена), развернутыми на системном хранилище (0), будет иметь следующую структуру:

```
/var/lib/one/datastores
|-- 0/
|   |-- 0/
|   |   |-- disk.0
|   |   `-- disk.1
|   |-- 2/
|   |   `-- disk.0
|   `-- 7/
|       |-- checkpoint
|       `-- disk.0
`-- 1
|-- 19217fdaaa715b04f1c740557826514b
|-- 99f93bd825f8387144356143dc69787d
`-- da8023daf074d0de3c1204e562b8d8d2
```

Драйвер передачи ssh (Рис. 24) использует локальную файловую систему узлов для размещения образов работающих ВМ. Все файловые операции выполняются локально, но образы всегда приходится копировать на узлы, что может оказаться очень ресурсоемкой операцией.

Драйвер передачи ssh

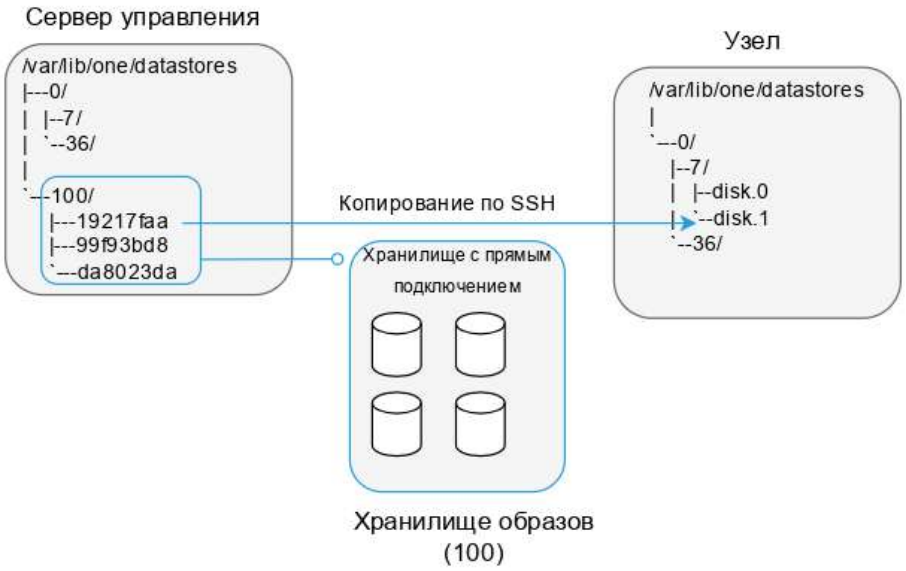


Рис. 24

Драйвер shared предполагает (Рис. 25), что на всех узлах установлена и настроена распределенная файловая система, например, NFS. Все файловые операции (ln, cp и т.д.) выполняются на узле виртуализации. Данный метод передачи сокращает время развертывания ВМ и обеспечивает возможность динамического перемещения.

Драйвер передачи shared

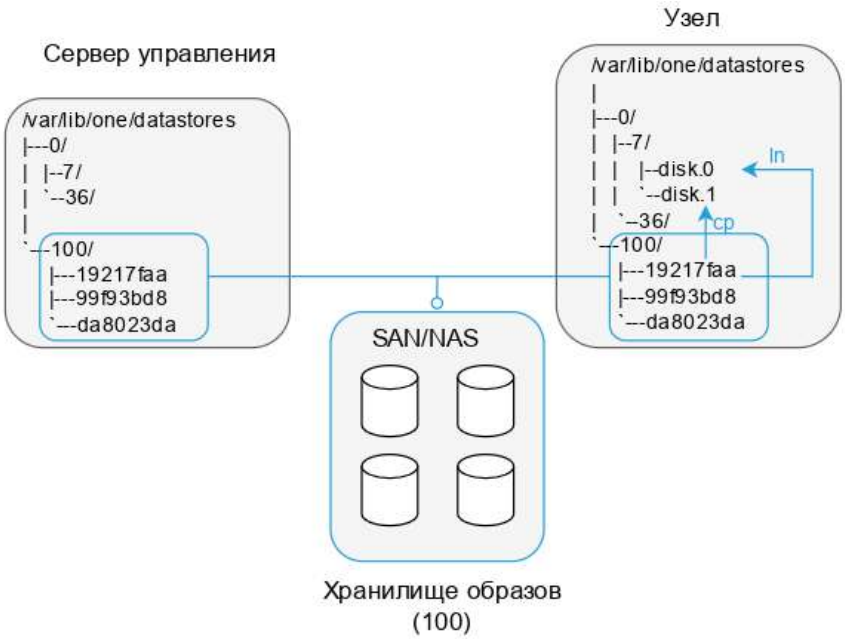


Рис. 25

Драйвер lvm (Рис. 26) рекомендуется использовать при наличии высокопроизводительной сети SAN. Один и тот же LUN можно экспортировать на все узлы, а ВМ будут работать непосредственно из SAN. При этом образы хранятся как обычные файлы (в /var/lib/one/datastores/<идентификатор_хранилища>) в хранилище образов, но при создании ВМ они будут сброшены в логические тома (LV). ВМ будут запускаться с логических томов узла.

Драйвер передачи lvm

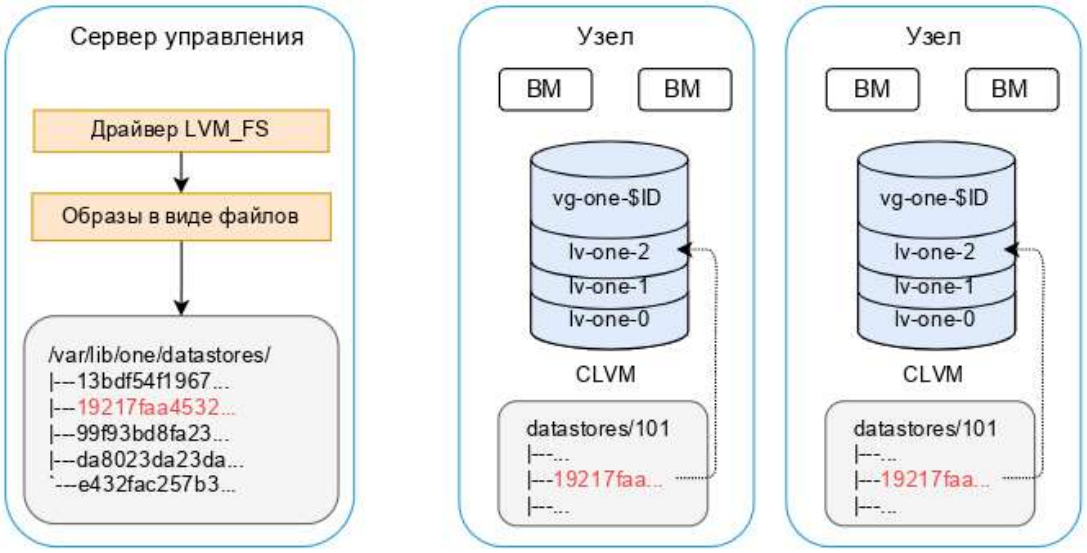


Рис. 26

3.6.2 Хранилища по умолчанию

По умолчанию в OpenNebula созданы три хранилища: хранилище образов (Images), системное (System) и файлов (Files).

По умолчанию хранилища настроены на использование локальной файловой системы (каталоги /var/lib/one/datastores/<идентификатор_хранилища>). При этом для передачи данных между хранилищем образов и системным хранилищем используется метод ssh.

Примечание. Стандартный путь для хранилищ /var/lib/one/datastores/ можно изменить, указав нужный путь в параметре DATASTORE_LOCATION в конфигурационном файле /etc/one/oned.conf.

onedatastore – инструмент управления хранилищами в OpenNebula. Описание всех доступных опций утилиты onedatastore можно получить, выполнив команду:

```
$ man onedatastore
```

Вывести список хранилищ данных можно, выполнив команду:

```
$ onedatastore list
  ID NAME                SIZE AVA CLUSTERS IMAGES TYPE DS      TM      STAT
   2 files               95.4G 91% 0             1 fil  fs      ssh     on
   1 default             95.4G 91% 0             8 img  fs      ssh     on
   0 system                  - -   0             0 sys  -       ssh     on
```

Информация о хранилище образов:

```
$ onedatastore show default
DATASTORE 1 INFORMATION
ID             : 1
NAME           : default
USER           : oneadmin
GROUP          : oneadmin
CLUSTERS       : 0
TYPE           : IMAGE
DS_MAD         : fs
TM_MAD         : ssh
BASE PATH      : /var/lib/one//datastores/1
DISK_TYPE      : FILE
STATE          : READY

DATASTORE CAPACITY
TOTAL:         : 95.4G
FREE:          : 55.9G
USED:          : 34.6G
LIMIT:         : -

PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : ---

DATASTORE TEMPLATE
ALLOW_ORPHANS="YES"
CLONE_TARGET="SYSTEM"
DISK_TYPE="FILE"
DS_MAD="fs"
LN_TARGET="SYSTEM"
RESTRICTED_DIRS="/"
SAFE_DIRS="/var/tmp"
TM_MAD="ssh"
TYPE="IMAGE_DS"

IMAGES
0
1
2
17
```

Информация о системном хранилище:

```
$ onedatastore show system
DATASTORE 0 INFORMATION
ID             : 0
NAME           : system
USER           : oneadmin
GROUP          : oneadmin
CLUSTERS       : 0
TYPE           : SYSTEM
DS_MAD         : -
TM_MAD         : ssh
BASE PATH      : /var/lib/one//datastores/0
DISK_TYPE      : FILE
STATE          : READY

DATASTORE CAPACITY
TOTAL:         : -
FREE:          : -
USED:          : -
LIMIT:         : -

PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : ---

DATASTORE TEMPLATE
ALLOW_ORPHANS="YES"
DISK_TYPE="FILE"
DS_MIGRATE="YES"
RESTRICTED_DIRS="/"
SAFE_DIRS="/var/tmp"
SHARED="NO"
TM_MAD="ssh"
TYPE="SYSTEM_DS"

IMAGES
```

Информация о хранилище содержит следующие разделы:

- INFORMATION – содержит базовую информацию о хранилище (название, путь к файлу хранилища, тип) и набор драйверов (DS_MAD и TM_MAD), используемых для хранения и передачи образов;

- CAPACITY – содержит основные показатели использования (общее, свободное и использованное пространство);
- TEMPLATE – содержит атрибуты хранилища;
- IMAGES – список образов, хранящихся в данный момент в этом хранилище.

В данном примере хранилище образов использует файловый драйвер (DS_MAD="fs") и протокол SSH для передачи (TM_MAD=ssh). Для системного хранилища определен только драйвер передачи (TM_MAD). Для системного хранилища также не указываются показатели использования (CAPACITY), так как драйвер ssh использует локальную область хранения каждого узла.

Примечание. Чтобы проверить доступное пространство на конкретном узле, можно воспользоваться командой `onehost show <hostid>`.

В зависимости используемого драйвера хранилища и инфраструктуры, для описания хранилища используются определенные атрибуты. Эти атрибуты описаны в следующих разделах. Кроме того, существует набор общих атрибутов, которые можно использовать в любом хранилище. Эти атрибуты описаны в табл. 7.

Таблица 7 – Общие атрибуты хранилищ

Атрибут	Описание
Description	Описание
RESTRICTED_DIRS	Каталоги, которые нельзя использовать для размещения образов. Список каталогов, разделенный пробелами
SAFE_DIRS	Разрешить использование каталога, указанного в разделе RESTRICTED_DIRS, для размещения образов. Список каталогов, разделенный пробелами
NO_DECOMPRESS	Не пытаться распаковать файл, который нужно зарегистрировать.
LIMIT_TRANSFER_BW	Максимальная скорость передачи при загрузке образов с URL-адреса http/https (в байтах/секунду). Могут использоваться суффиксы K, M или G
DATASTORE_CAPACITY_CHECK	Проверять доступную емкость хранилища данных перед созданием нового образа
LIMIT_MB	Максимально допустимая емкость хранилища данных в МБ
BRIDGE_LIST	Список мостов узла, разделенных пробелами, которые имеют доступ к хранилищу для добавления новых образов в хранилище
STAGING_DIR	Путь на узле моста хранения для копирования образа перед его перемещением в конечный пункт назначения. По умолчанию /var/tmp
DRIVER	Применение специального драйвера сопоставления изображений. Данный атрибут переопределяет DRIVER образа, установленный в атрибутах образа и шаблоне ВМ
COMPATIBLE_SYS_DS	Только для хранилищ образов. Установить системные хранилища данных, которые можно использовать с данным хранилищем образов (например, «0,100»)

Атрибут	Описание
CONTEXT_DISK_TYPE	Указывает тип диска, используемый для контекстных устройств: BLOCK или FILE (по умолчанию)

Примечание. Для использования BRIDGE_LIST, следует установить любой инструмент, необходимый для доступа к базовому хранилищу, а также универсальные инструменты, такие как qemu-img.

Системные хранилища можно отключить, чтобы планировщик не мог развернуть в них новую ВМ. При этом существующие ВМ продолжат работать. Отключение хранилища:

```
$ onedatastore disable system
$ onedatastore show system
DATASTORE 0 INFORMATION
ID             : 0
NAME           : system
...
STATE          : DISABLED
...
```

Создавать, включать, отключать, удалять и просматривать информацию о хранилищах можно в веб-интерфейсе (Рис. 27).

Работа с хранилищами в OpenNebula-Sunstone

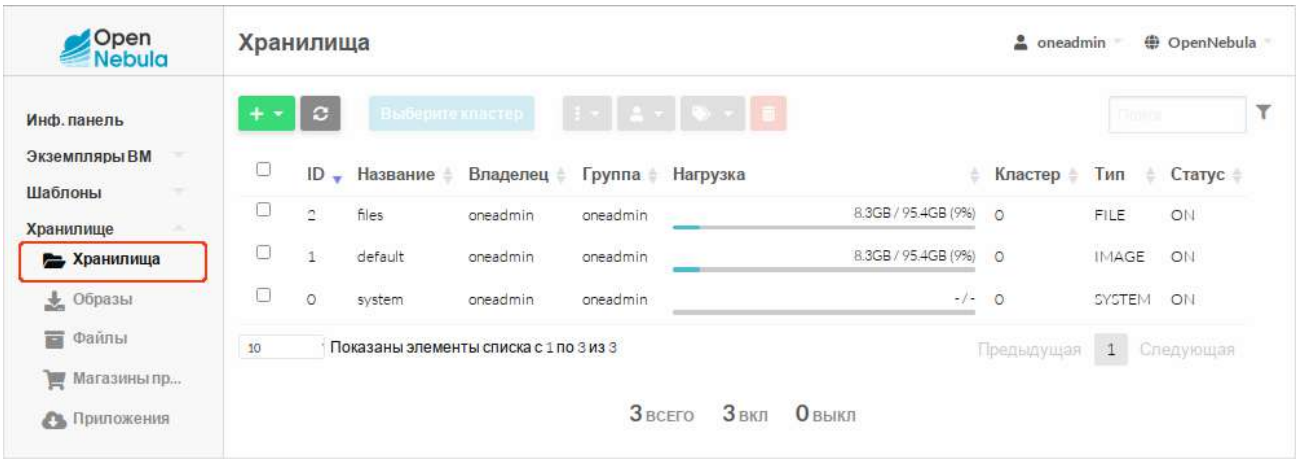


Рис. 27

3.6.3 Создание хранилищ

Для создания хранилища необходимо выполнить следующие действия:

- подготовить систему хранения данных в соответствии с выбранной технологией хранения;
- создать хранилище в OpenNebula, указав его имя, тип и метод передачи данных;
- смонтировать подготовленную систему хранения данных в каталог хранилища (на узле управления и узлах виртуализации).

3.6.3.1 Локальное хранилище

Данная конфигурация использует локальную область хранения каждого узла для запуска ВМ. Кроме того, потребуется место для хранения образа диска ВМ. Образы дисков передаются с сервера управления на узлы по протоколу SSH.

На сервере управления в `/var/lib/one/datastores/` должно быть достаточно места для:

- хранилища образов;
- системного хранилища (для временных дисков и файлов остановленных и неразвернутых ВМ).

На узле виртуализации в `/var/lib/one/datastores/` должно быть достаточно места для хранения дисков ВМ, работающих на этом узле.

Необходимо зарегистрировать два хранилища (системное и хранилище образов).

Чтобы создать новое системное хранилище, необходимо указать следующие параметры:

- NAME – название хранилища;
- TYPE – SYSTEM_DS;
- TM_MAD – shared (для режима общей передачи), qcow2 (для режима передачи qcow2), ssh (для режима передачи ssh).

Зарегистрировать системное хранилище можно как в веб-интерфейсе Sunstone (Рис. 28), так и в командной строке. Например, для создания системного хранилища можно создать файл `systemds.conf` со следующим содержимым:

```
NAME    = local_system
TM_MAD  = ssh
TYPE    = SYSTEM_DS
```

И выполнить команду:

```
$ onedatastore create systemds.conf
ID: 101
```

Чтобы создать новое хранилище образов, необходимо указать следующие параметры:

- NAME – название хранилища;
- DS_MAD – fs (драйвер хранилища данных);
- TYPE – IMAGE_DS;
- TM_MAD – shared (для режима общей передачи), qcow2 (для режима передачи qcow2), ssh (для режима передачи ssh).

Примечание. Необходимо использовать одинаковый метод передачи данных TM_MAD для системного хранилища и для хранилища образов.

Регистрация нового системного хранилища

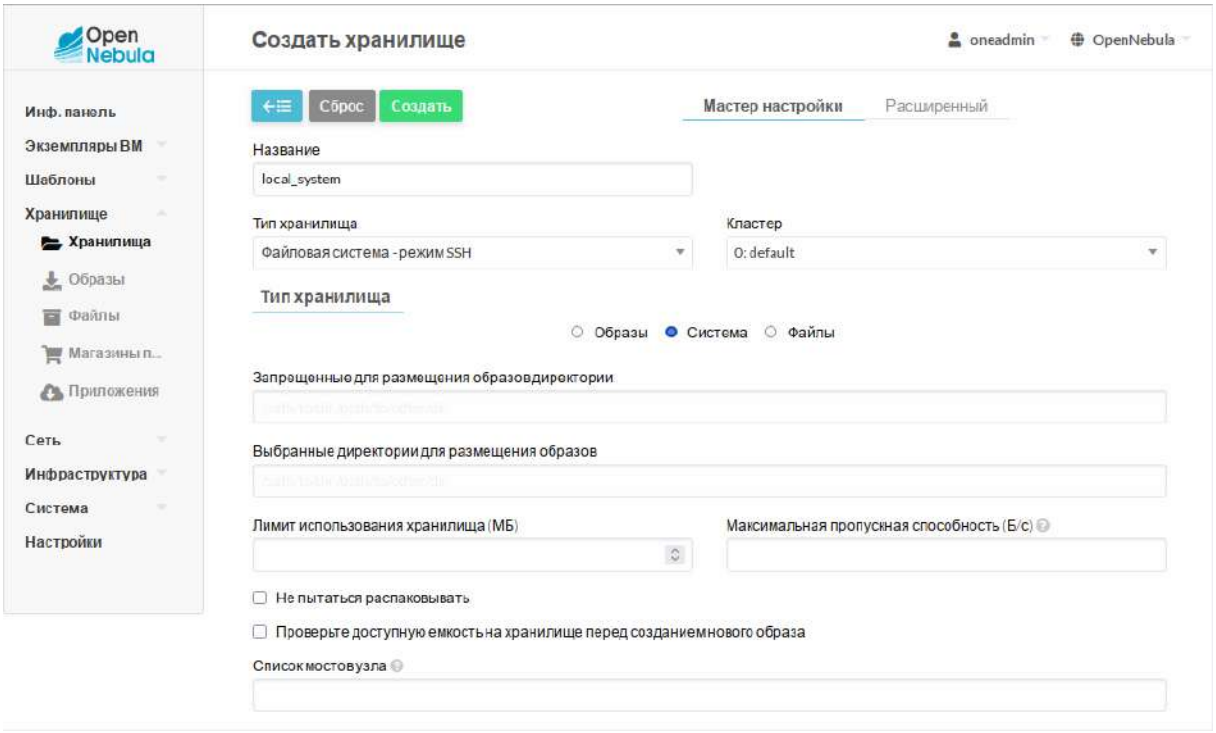


Рис. 28

Зарегистрировать хранилище образов можно как в веб-интерфейсе Sunstone, так и в командной строке. Например, для создания хранилища образов можно создать файл `imageds.conf` со следующим содержимым:

```
NAME    = local_image
TM_MAD  = ssh
TYPE    = IMAGE_DS
DS_MAD  = fs
```

И выполнить команду:

```
$ onedatastore create imageds.conf
ID: 102
```

3.6.3.2 Хранилище NFS/NAS

Эта конфигурация хранилища предполагает, что на узлах монтируются каталоги, расположенные на сервере NAS (сетевое хранилище). Эти каталоги используются для хранения файлов образов дисков ВМ. ВМ также будут загружаться с общего каталога.

Масштабируемость этого решения ограничена производительностью NAS-сервера.

Примечание. В `/var/lib/one/datastores/` можно смонтировать каталог с любого сервера NAS/SAN в сети.

Необходимо зарегистрировать два хранилища (системное и хранилище образов).

Чтобы создать новое системное хранилище, необходимо указать следующие параметры:

- NAME – название хранилища;

- TYPE – SYSTEM_DS;
- TM_MAD – shared (для режима общей передачи), qcow2 (для режима передачи qcow2), ssh (для режима передачи ssh).

Зарегистрировать системное хранилище можно как в веб-интерфейсе Sunstone, так и в командной строке. Например, для создания системного хранилища можно создать файл `systemds.conf` со следующим содержимым:

```
NAME    = nfs_system
TM_MAD  = shared
TYPE    = SYSTEM_DS
```

И выполнить команду:

```
$ onedatastore create systemds.conf
ID: 101
```

Чтобы создать новое хранилище образов, необходимо указать следующие параметры:

- NAME – название хранилища;
- DS_MAD – fs (драйвер хранилища данных);
- TYPE – IMAGE_DS;
- TM_MAD – shared (для режима общей передачи), qcow2 (для режима передачи qcow2), ssh (для режима передачи ssh).

Примечание. Необходимо использовать одинаковый метод передачи данных TM_MAD для системного хранилища и для хранилища образов.

Зарегистрировать хранилище образов можно как в веб-интерфейсе Sunstone, так и в командной строке. Например, для создания хранилища образов можно создать файл `imageds.conf` со следующим содержимым:

```
NAME    = nfs_image
TM_MAD  = shared
TYPE    = IMAGE_DS
DS_MAD  = fs
```

И выполнить команду:

```
$ onedatastore create imageds.conf
ID: 102
```

На узле управления (в `/var/lib/one/datastores/`) будут созданы два каталога: 101 и 102. На узлах виртуализации эти каталоги автоматически не создаются, поэтому на узлах виртализации требуется создать каталоги с соответствующими идентификаторами:

```
$ mkdir /var/lib/one/datastores/101
$ mkdir /var/lib/one/datastores/102
```

В каталог `/var/lib/one/datastores/<идентификатор_хранилища>` на узле управления и узлах виртуализации необходимо смонтировать удалённый каталог NFS. Например:

```
# mount -t nfs 192.168.0.157:/export/storage /var/lib/one/datastores/102
```

Для автоматического монтирования к NFS-серверу при загрузке необходимо добавить следующую строку в файл /etc/fstab:

```
192.168.0.157:/export/storage /var/lib/one/datastores/102   nfs
intr,soft,nolock,_netdev,x-systemd.automount    0 0
```

Примечание. Для возможности монтирования NFS-хранилища на всех узлах должен быть запущен nfs-client:

```
# systemctl enable --now nfs-client.target
```

Получить список совместных ресурсов с сервера NFS можно, выполнив команду:

```
# showmount -e 192.168.0.157
```

Примечание. При использовании файловой технологии хранения, после добавления записи об автоматическом монтировании в файле /etc/fstab и перезагрузки ОС, необходимо назначить на каталог этого хранилища владельца oneadmin. Например:

```
# chown oneadmin: /var/lib/one/datastores/102
```

3.6.3.3 NFS/NAS и локальное хранилище

При использовании хранилища NFS/NAS можно повысить производительность ВМ, разместив диски в локальной файловой системе узла. Таким образом, хранилище образов будет размещено на сервере NFS, но ВМ будут работать с локальных дисков.

Чтобы настроить этот сценарий, следует настроить хранилище образов и системное хранилища, как описано выше (TM_MAD=shared). Затем добавить системное хранилище (TM_MAD=ssh). Любой образ, зарегистрированный в хранилище образов, можно будет развернуть с использованием любого из этих системных хранилищ.

Чтобы выбрать (альтернативный) режим развертывания, следует добавить в шаблон ВМ атрибут:

```
TM_MAD_SYSTEM="ssh"
```

3.6.3.4 Хранилище SAN

Эта конфигурация хранилища предполагает, что узлы имеют доступ к устройствам хранения (LUN), экспортированным сервером сети хранения данных (SAN) с использованием подходящего протокола, такого как iSCSI или Fibre Channel.

Для организации хранилищ требуется выделение как минимум 2 LUN на системе хранения. Эти LUN должны быть презентованы каждому участнику кластера – узлам управления и узлам виртуализации.

Для хранения образов в виде файлов, используется хранилище файлового типа. Блочные устройства такого хранилища (созданные и презентованные выше LUN) должны быть отформатированы в кластерную файловую систему. Существует также возможность хранить образы исполняемых ВМ в виде томов LVM.

Хранилище SAN может получить доступ к файлам образов двумя способами:

- режим NFS – файлы образов доступны непосредственно на узлах виртуализации через распределенную файловую систему, например NFS или OCFS2 (fs_lvm);
- режим SSH – файлы образов передаются на узел по SSH (fs_lvm_ssh).

В любом режиме серверу управления необходимо иметь доступ к хранилищам образов путем монтирования соответствующего каталога в `/var/lib/one/datastores/<ID_хранилища>`. В случае режима NFS каталог необходимо смонтировать с сервера NAS. В режиме SSH можно смонтировать любой носитель данных в каталог хранилища.

Сервер управления также должен иметь доступ к общему LVM либо напрямую, либо через узел виртуализации, указав его в атрибуте BRIDGE_LIST в шаблоне хранилища.

3.6.3.4.1 Создание хранилищ

Чтобы создать новое системное хранилище, необходимо указать следующие параметры:

- NAME – название хранилища;
- TM_MAD – fs_lvm (для режима NFS), fs_lvm_ssh (для режима SSH);
- TYPE – SYSTEM_DS;
- BRIDGE_LIST – список узлов, имеющих доступ к логическим томам. НЕ требуется, если внешний интерфейс настроен на доступ к логическим томам.

Зарегистрировать системное хранилище можно как в веб-интерфейсе Sunstone, так и в командной строке. Например, для создания системного хранилища можно создать файл `systemds.conf` со следующим содержимым:

```
NAME    = lvm-system
TM_MAD  = fs_lvm_ssh
TYPE    = SYSTEM_DS
BRIDGE_LIST = "host-01 host-02"
```

И выполнить команду:

```
$ onedatastore create systemds.conf
ID: 101
```

Чтобы создать новое хранилище образов, необходимо указать следующие параметры:

- NAME – название хранилища;
- TM_MAD – fs_lvm (для режима NFS), fs_lvm_ssh (для режима SSH);
- DS_MAD – fs;
- TYPE – IMAGE_DS;
- BRIDGE_LIST – список узлов, имеющих доступ к логическим томам. НЕ требуется, если внешний интерфейс настроен на доступ к логическим томам;
- DISK_TYPE – BLOCK.

Зарегистрировать хранилище образов можно как в веб-интерфейсе Sunstone, так и в командной строке. Например, для создания хранилища образов можно создать файл `imageds.conf` со следующим содержимым:

```
NAME    = lvm-images
TM_MAD  = fs_lvm_ssh
TYPE    = IMAGE_DS
DISK_TYPE = "BLOCK"
DS_MAD  = fs
```

И выполнить команду:

```
$ onedatastore create imageds.conf
ID: 102
```

Примечание. Необходимо использовать одинаковый метод передачи данных TM_MAD для системного хранилища и для хранилища образов.

На узле управления (в `/var/lib/one/datastores/`) будут созданы два каталога: 101 и 102. На узлах виртуализации эти каталоги автоматически не создаются, поэтому требуется создать каталоги с соответствующими идентификаторами:

```
$ mkdir /var/lib/one/datastores/101
$ mkdir /var/lib/one/datastores/102
```

3.6.3.4.2 Разметка хранилища образов

Устройство, на котором размещается хранилище образов должно быть отформатировано кластерной ФС.

Ниже показано создание кластерной ФС ocfs2 на multipath-устройстве и подключение этого устройства в OpenNebula.

3.6.3.4.2.1 Кластерная ФС ocfs2

Ниже показано создание кластерной ФС ocfs2 на multipath-устройстве.

На всех узлах кластера необходимо установить пакет `ocfs2-tools`:

```
# apt-get install ocfs2-tools
```

Примечание. Основной конфигурационный файл для OCFS2 – `/etc/ocfs2/cluster.conf`. Этот файл должен быть одинаков на всех узлах кластера, при изменении в одном месте его нужно скопировать на остальные узлы. При добавлении нового узла в кластер, описание этого узла должно быть добавлено на всех остальных узлах до монтирования раздела ocfs2 с нового узла.

Создание кластерной конфигурации возможно с помощью команд или с помощью редактирования файла конфигурации `/etc/ocfs2/cluster.conf`.

Пример создания кластера из трёх узлов:

- в командной строке:

- создать кластер с именем mycluster:

```
# o2cb_ctl -C -n mycluster -t cluster -a name=mycluster
```

- добавить узел, выполнив команду для каждого:

```
# o2cb_ctl -C -n <имя_узла> -t node -a number=0 -a ip_address=<IP_узла> -a
ip_port=7777 -a cluster=mycluster
```

- редактирование конфигурационного файла /etc/ocfs2/cluster.conf:

```
cluster:
node_count = 3
heartbeat_mode = local
name = mycluster

node:
ip_port = 7777
ip_address = <IP_узла-01>
number = 0
name = <имя_узла-01>
cluster = mycluster

node:
ip_port = 7777
ip_address = <IP_узла-02>
number = 1
name = <имя_узла-02>
cluster = mycluster

node:
ip_port = 7777
ip_address = <IP_узла-03>
number = 2
name = <имя_узла-03>
cluster = mycluster
```

Примечание. Имя узла кластера должно быть таким, как оно указано в файле /etc/hostname.

Для включения автоматической загрузки сервиса OCFS2 можно использовать скрипт /etc/init.d/o2cb:

```
# /etc/init.d/o2cb configure
```

Для ручного запуска кластера нужно выполнить:

```
# /etc/init.d/o2cb load
checking debugfs...
Loading filesystem "ocfs2_dlmfs": OK
```

```
Creating directory '/dlm': OK
Mounting ocfs2_dlmfs filesystem at /dlm: OK
# /etc/init.d/o2cb online mycluster
checking debugfs...
Setting cluster stack "o2cb": OK
Registering O2CB cluster "mycluster": OK
Setting O2CB cluster timeouts : OK
```

Далее на одном из узлов необходимо создать раздел OCFS2, для этого следует выполнить следующие действия:

- создать физический раздел `/dev/mapper/mpatha-part1` на диске `/dev/mapper/mpatha`:

```
# fdisk /dev/mapper/mpatha
```

- отформатировать созданный раздел, выполнив команду:

```
# mkfs.ocfs2 -b 4096 -C 4k -L DBF1 -N 3 /dev/mapper/mpatha-part1
mkfs.ocfs2 1.8.7
Cluster stack: classic o2cb
Label: DBF1
…
mkfs.ocfs2 successful
```

Описание параметров команды mkfs.ocfs2 приведено в табл. 8.

Т а б л и ц а 8 – Параметры команды mkfs.ocfs2

Параметр	Описание
`-L метка_тома`	Метка тома, позволяющая его однозначно идентифицировать при подключении на разных узлах. Для изменения метки тома можно использовать утилиту tunefs.ocfs2
`-C размер_кластера`	Размер кластера — это наименьшая единица пространства, выделенная файлу для хранения данных. Возможные значения: 4, 8, 16, 32, 64, 128, 256, 512 и 1024 КБ. Размер кластера невозможно изменить после форматирования тома
`-N количество_узлов_кластера`	Максимальное количество узлов, которые могут одновременно монтировать том. Для изменения количества узлов можно использовать утилиту tunefs.ocfs2
`-b размер_блока`	Наименьшая единица пространства, адресуемая ФС. Возможные значения: 512 байт (не рекомендуется), 1 КБ, 2 КБ или 4 КБ (рекомендуется для большинства томов). Размер блока невозможно изменить после форматирования тома

П р и м е ч а н и е . Для создания нового раздела может потребоваться предварительно удалить существующие данные раздела на устройстве /dev/mpathX (следует использовать с осторожностью!):

```
# dd if=/dev/zero of=/dev/mapper/mpathX bs=512 count=1 conv=notrunc
```

3.6.3.4.2.2 OCFS2 в OpenNebula

На каждом узле OpenNebula необходимо добавить данную ФС OCFS2 к каталогам, которые будут автоматически монтироваться при загрузке узла:

- определить UUID раздела:

```
# blkid
/dev/mapper/mpatha-part1: LABEL="DBF1" UUID="df49216a-a835-47c6-b7c1-6962e9b7dcb6"
BLOCK_SIZE="4096" TYPE="ocfs2" PARTUUID="15f9cd13-01"
```

- добавить монтирование этого UUID в `/etc/fstab`:

```
UUID=<uuid> /var/lib/one/datastores/<идентификатор_хранилища> ocfs2 _netdev,defaults
0 0
```

Например:

```
UUID=df49216a-a835-47c6-b7c1-6962e9b7dcb6       /var/lib/one/datastores/102     ocfs2
_netdev,defaults 0 0
```

- убедиться, что монтирование прошло без ошибок, выполнив команду:

```
# mount -a
```

Результатом выполнения команды должен быть пустой вывод без ошибок.

- пример получившейся конфигурации:

```
# lsblk
NAME              MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINTS
sda                 8:0    0    59G  0 disk
`-sda1              8:1    0   255M  0 part  /boot/efi
sdb                 8:16   0 931.3G  0 disk
`-mpatha          253:0    0 931.3G  0 mpath
  `-mpatha-part1 253:1    0 931.3G  0 part  /var/lib/one/datastores/102
sdc                 8:32   0 931.3G  0 disk
|-sdc1              8:33   0 931.3G  0 part
`-mpatha          253:0    0 931.3G  0 mpath
  `-mpatha-part1 253:1    0 931.3G  0 part  /var/lib/one/datastores/102
sdd                 8:48   0 931.3G  0 disk
`-mpatha          253:0    0 931.3G  0 mpath
  `-mpatha-part1 253:1    0 931.3G  0 part  /var/lib/one/datastores/102
sde                 8:64   0 931.3G  0 disk
`-mpatha          253:0    0 931.3G  0 mpath
  `-mpatha-part1 253:1    0 931.3G  0 part  /var/lib/one/datastores/102
```

Примечание. Опция `_netdev` позволяет монтировать данный раздел только после успешного старта сетевой подсистемы.

Примечание. При использовании файловой технологии хранения, после добавления записи об автоматическом монтировании в файле `/etc/fstab` и перезагрузки ОС, необходимо назначить на каталог этого хранилища владельца oneadmin. Например:

```
# chown oneadmin: /var/lib/one/datastores/102
```

Примечание. Вывести все файловые системы OCFS2 на данном узле:

```
# mounted.ocfs2 -f
Device                    Stack  Cluster  F  Nodes
/dev/mapper/mpatha-part1  o2cb              server, host-02, host-01
# mounted.ocfs2 -d
Device                    Stack  Cluster  F UUID                          Label
/dev/mapper/mpatha-part1 o2cb             DF49216AA83547C6B7C16962B6 DBF
```

3.6.3.4.3 Разметка системного хранилища

LUN для системного хранилища будет обслуживаться менеджером томов LVM.

Предварительные условия:

- lvmetad должен быть отключен. Для этого следует установить параметр `use_lvmetad = 0` в `/etc/lvm/lvm.conf` (в разделе global) и отключить службу `lvm2-lvmetad.service`, если она запущена;
- пользователь oneadmin должен входить в группу disk:

```
# gpasswd -a oneadmin disk
```

- все узлы должны иметь доступ к одним и тем же LUN;
- для каждого хранилища необходимо создать LVM VG с именем vg-one-<идентификатор_хранилища> в общих LUN.

Примечание. LUN должен быть виден в системе по пути `/dev/mapper/`.

LUN должен быть не размечен. Его необходимо очистить от разметки и/или файловых систем на стороне СХД, или выполнить команду:

```
# wipefs -fa /dev/mapper/[LUN_WWID]
```

На узле управления:

- создать физический том (PV) на LUN, например:

```
# pvcreate /dev/mapper/mpathb
Physical volume "/dev/mapper/mpathb" successfully created.
```

- создать группу томов с именем `vg-one-<идентификатор_хранилища>`, например:

```
# vgcreate vg-one-101 /dev/mapper/mpathb
Volume group "vg-one-101" successfully created
```

- вывести информацию о физических томах:

```
# pvs
PV                    VG          Fmt  Attr PSize   PFree
/dev/mapper/mpathb    vg-one-101 lvm2 a--  931.32g 931.32g
```

Созданные хранилища будут отображаться в веб-интерфейсе OpenNebula. Индикация объёма хранилища должна соответствать выделенному объему на СХД для каждого LUN (Рис. 29).

SAN хранилища

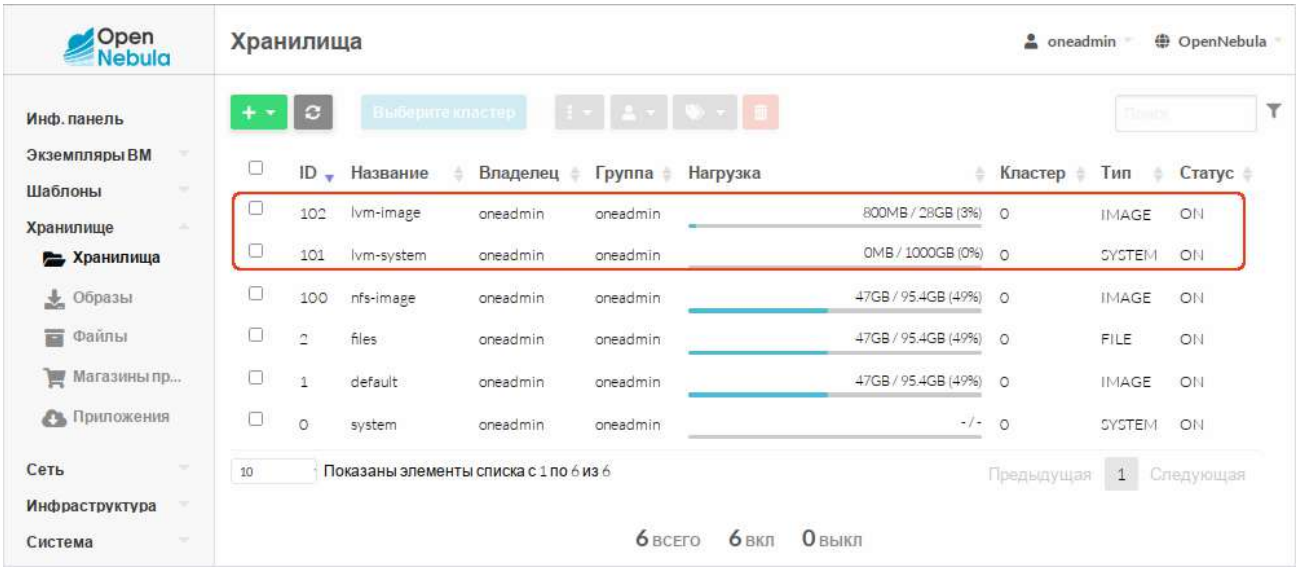


Рис. 29

После создания и запуска ВМ в данном хранилище будет создан логический раздел:

```
# lvscan
  ACTIVE            '/dev/vg-one-101/lv-one-52-0' [50,00 GiB] inherit
# lsblk
sde                                 8:64   0   931.3G  0 disk
└─mpathb                          253:1    0   931.3G  0 mpath
  └─vg--one--101-lv--one--52--0 253:3    0   51G     0 lvm
```

где 52 – идентификатор ВМ.

3.6.3.5 Хранилище файлов и ядер

Хранилище файлов и ядер позволяет хранить простые файлы, которые будут использоваться в качестве ядер ВМ, виртуальных дисков или любых других файлов, которые необходимо передать ВМ в процессе контекстуализации. Данное хранилище не предоставляет никакого специального механизма хранения, но представляет простой и безопасный способ использования файлов в шаблонах ВМ.

Чтобы создать новое хранилище файлов и ядер, необходимо указать следующие параметры:

- NAM – название хранилища;
- TYPE – FILE_DS;
- DS_MAD – fs;
- TM_MAD – ssh.

Зарегистрировать хранилище файлов и ядер можно как в веб-интерфейсе Sunstone, так и в командной строке. Например, для создания хранилища файлов и ядер можно создать файл `fileds.conf` со следующим содержимым:

```
NAME    = local_file
```

```
DS_MAD  = fs
TM_MAD  = ssh
TYPE    = FILE_DS
```

И выполнить команду:

```
$ onedatastore create fileds.conf
ID: 105
```

Примечание. Выше приведены рекомендованные значения DS_MAD и TM_MAD, но можно использовать любые другие.

Для того чтобы изменить параметры хранилища, можно создать файл с необходимыми настройками (пример см. выше) и выполнить команду:

```
$ onedatastore update <идентификатор_хранилища> <имя_файла>
```

3.7 Работа с образами в OpenNebula

Система хранилищ позволяет пользователям настраивать/устанавливать образы, которые могут быть образами ОС или данных, для использования в ВМ. Данные образы могут использоваться несколькими ВМ одновременно, а также предоставляться другим пользователями.

Типы образов для дисков ВМ (хранятся в хранилище образов):

- OS – образ загрузочного диска;
- CDROM – файл образа, содержащий CDROM. Эти образы предназначены только для чтения. В каждом шаблоне ВМ, можно использовать только один образ данного типа;
- DATABLOCK – файл образа, содержащий блок данных, создаваемый как пустой блок.

Типы файлов (хранятся в файловом хранилище):

- KERNEL – файл, который будет использоваться в качестве ядра ВМ (kernels);
- RAMDISK – файл для использования в качестве виртуального диска;
- CONTEXT – файл для включения в контекстный CD-ROM.

Образы могут работать в двух режимах:

- persistent (постоянные) – изменения, внесенные в такие образы, будут сохранены после завершения работы ВМ. В любой момент времени может быть только одна ВМ, использующая постоянный образ.
- non-persistent (непостоянный) – изменения не сохранятся после завершения работы ВМ. Непостоянные образы могут использоваться несколькими ВМ одновременно, поскольку каждая из них будет работать со своей собственной копией.

Управлять образами можно, используя команду `oneimage`. Управлять образами также можно в веб-интерфейсе на вкладке «Образы» (Рис. 30).

Управление образами в OpenNebula-Sunstone

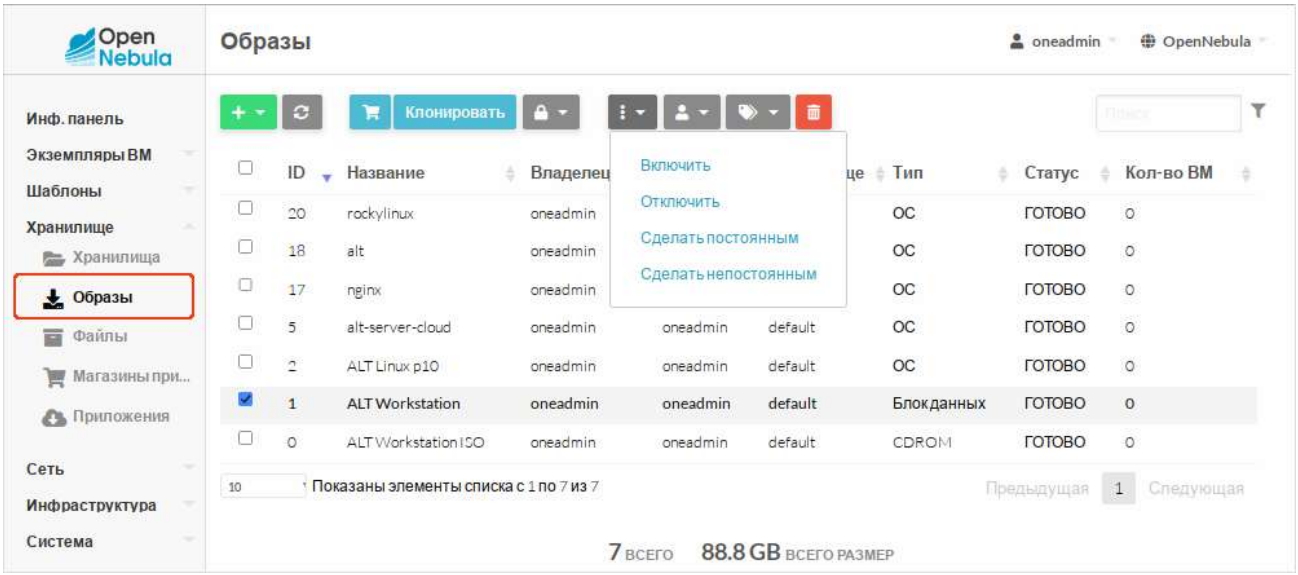


Рис. 30

3.7.1 Работа с образами в OpenNebula

Для создания образа ОС, необходимо подготовить ВМ и извлечь её диск.

3.7.1.1 Создание образов дисков

Создать образ типа CDROM с установочным ISO-образом.

Для этого перейти в раздел «Хранилище» → «Образы», на загруженной странице нажать «+» и выбрать пункт «Создать» (Рис. 31).

Создание образа в OpenNebula-Sunstone

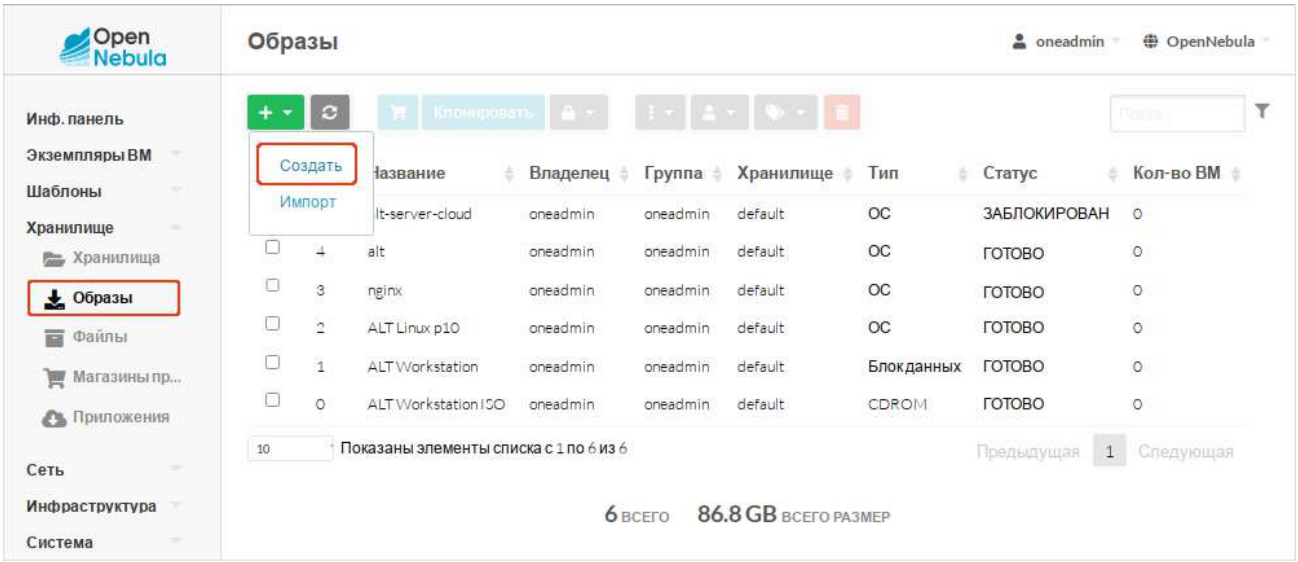


Рис. 31

В открывшемся окне заполнить поле «Название», выбрать в выпадающем списке «Тип» значение «CD-ROM только для чтения», выбрать хранилище, отметить в «Расположение образа» пункт «Путь/URL», указать путь к файлу (.iso) и нажать кнопку «Создать» (Рис. 32).

Создание образа типа CD-ROM

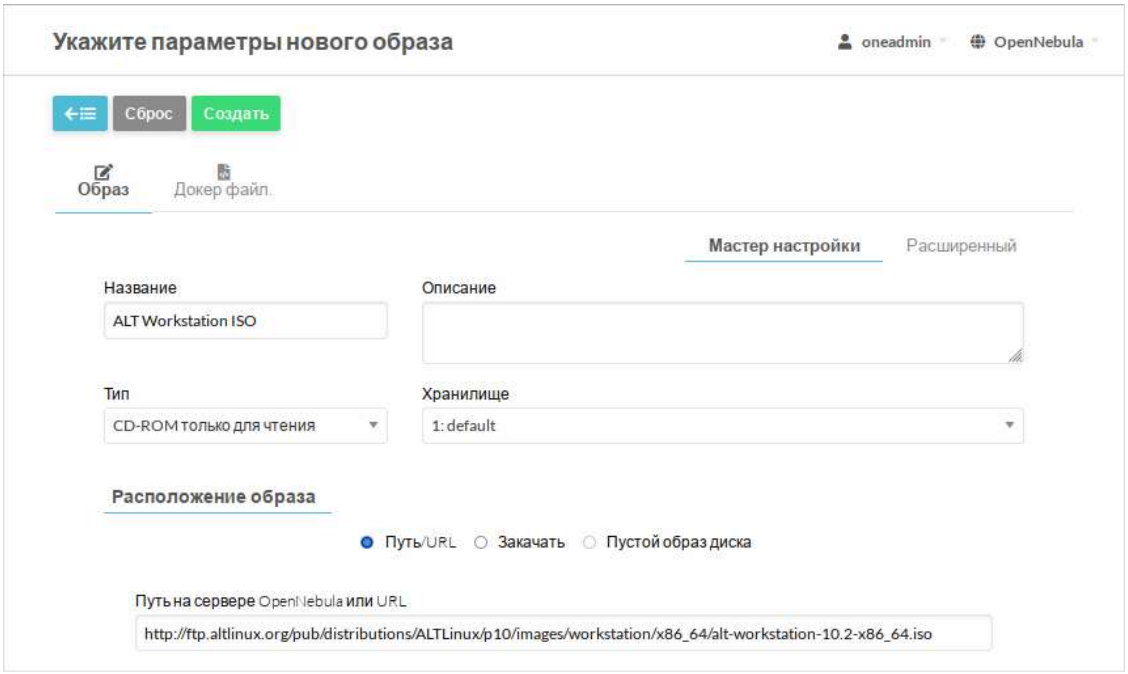


Рис. 32

Примечание. Если указывается путь на сервере OpnNebula, то ISO-образ должен быть загружен в каталог, к которому имеет доступ пользователь oneadmin.

Создать пустой образ диска, на который будет установлена операционная система.

Для этого создать новый образ. Заполнить поле «Название», в выпадающем списке «Тип» выбрать значение «Generic storage datablock», в выпадающем списке «Этот образ является постоянным» выбрать значение «Да», выбрать хранилище, в разделе «Расположение образа» выбрать пункт «Пустой образ диска», установить размер выбранного блока, например, 45ГБ, в разделе «Расширенные настройки» указать формат «qcow2» и нажать кнопку «Создать» (Рис. 33).

Эти же действия можно выполнить в командной строке.

Создать образ типа CDROM в хранилище данных по умолчанию (ID = 1):

```
$ oneimage create -d 1 --name "ALT Workstation ISO" --path /var/tmp/alt-workstation-10.0-x86_64.iso --type CDROM
ID: 31
```

Создать пустой образ диска (тип образа – DATABLOCK, размер 45 ГБ, драйвер qcow2):

```
$ oneimage create -d 1 --name "ALT Workstation" --type DATABLOCK --size 45G --persistent --driver qcow2
ID: 33
```

Создание диска

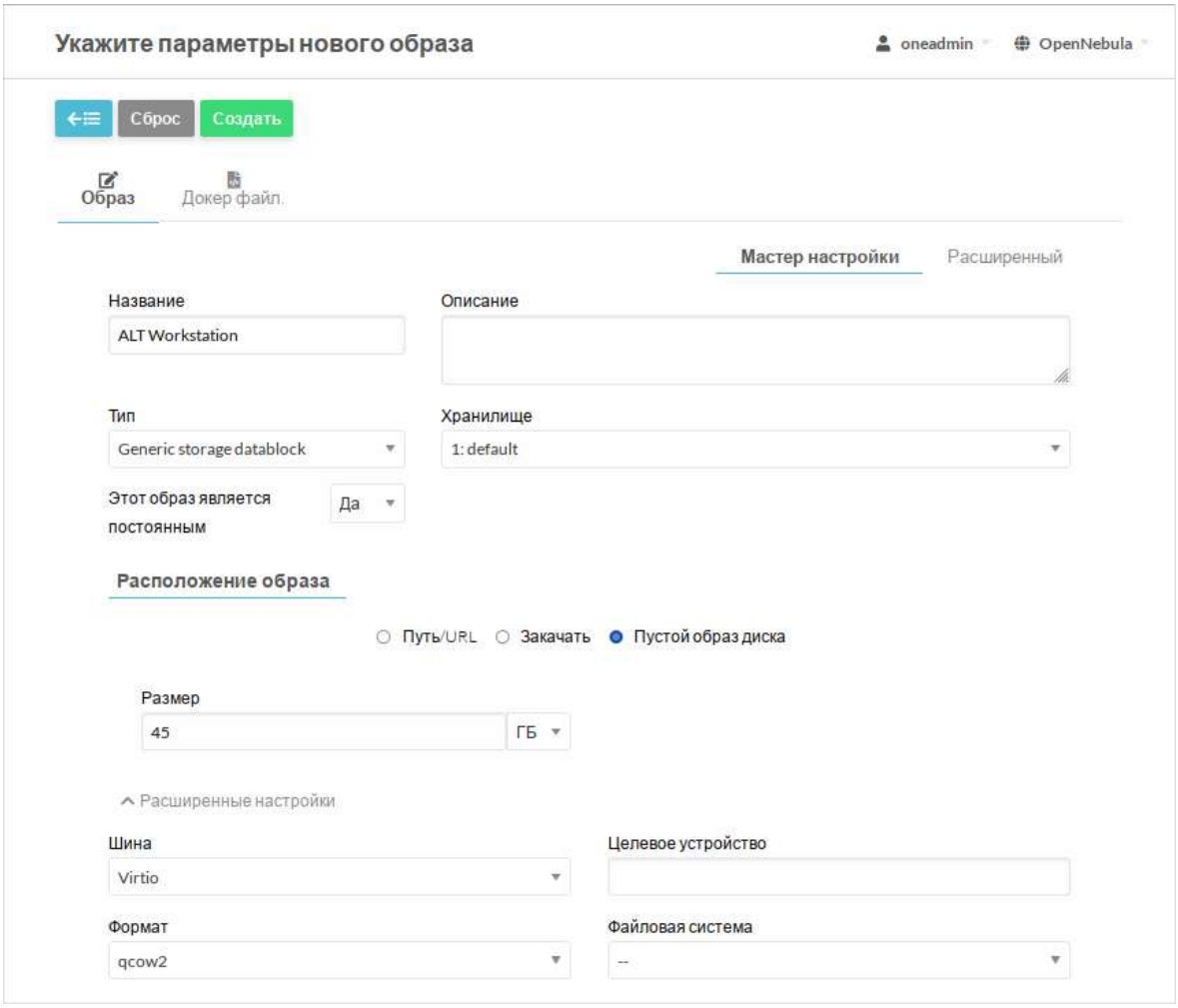


Рис. 33

3.7.1.2 Создание шаблона ВМ

Создание шаблона в командной строке:

1) Создать файл template со следующим содержимым:

```
NAME = "ALT Workstation"
CONTEXT = [
  NETWORK = "YES",
  SSH_PUBLIC_KEY = "$USER[SSH_PUBLIC_KEY]" ]
CPU = "0.25"
DISK = [
  IMAGE = "ALT Workstation ISO",
  IMAGE_UNAME = "oneadmin" ]
DISK = [
  DEV_PREFIX = "vd",
  IMAGE = "ALT Workstation",
  IMAGE_UNAME = "oneadmin" ]
GRAPHICS = [
  LISTEN = "0.0.0.0",
```

```
  TYPE = "SPICE" ]
HYPERVISOR = "kvm"
INPUTS_ORDER = ""
LOGO = "images/logos/alt.png"
MEMORY = "1024"
MEMORY_UNIT_COST = "MB"
NIC = [
  NETWORK = "VirtNetwork",
  NETWORK_UNAME = "oneadmin",
  SECURITY_GROUPS = "0" ]
NIC_DEFAULT = [
  MODEL = "virtio" ]
OS = [
  BOOT = "disk1,disk0" ]
SCHED_REQUIREMENTS = "ID=\"0\""
```

2) Создать шаблон:

```
$ onetemplate create template
ID: 22
```

Ниже рассмотрен пример создания шаблона в веб-интерфейсе.

Для создания шаблона ВМ, необходимо в левом меню выбрать «Шаблоны» → «ВМ» и на загруженной странице нажать кнопку «+» и выбрать пункт «Создать» (Рис. 34).

Создание шаблона ВМ

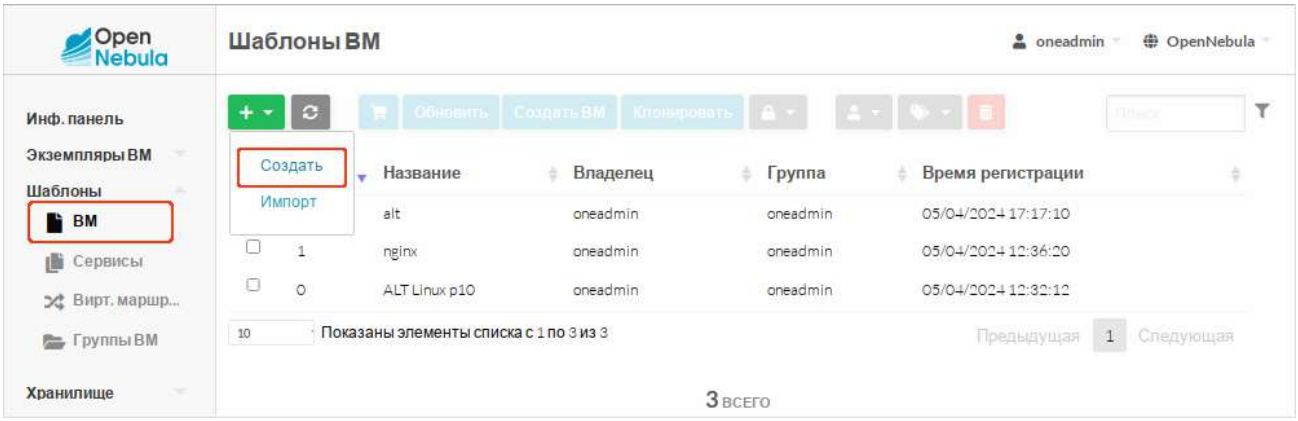


Рис. 34

На вкладке «Общие» необходимо указать параметры процессора, оперативной памяти, а также гипервизор (Рис. 35).

На вкладке «Хранилище» необходимо указать ранее созданный диск с установщиком ОС. Далее следует добавить новый диск и указать ранее созданный пустой диск (DATABLOCK), в разделе «Расширенные настройки» в выпадающем списке «Шина» выбрать «Virtio» (Рис. 36).

Создание шаблона ВМ. Вкладка «Общие»

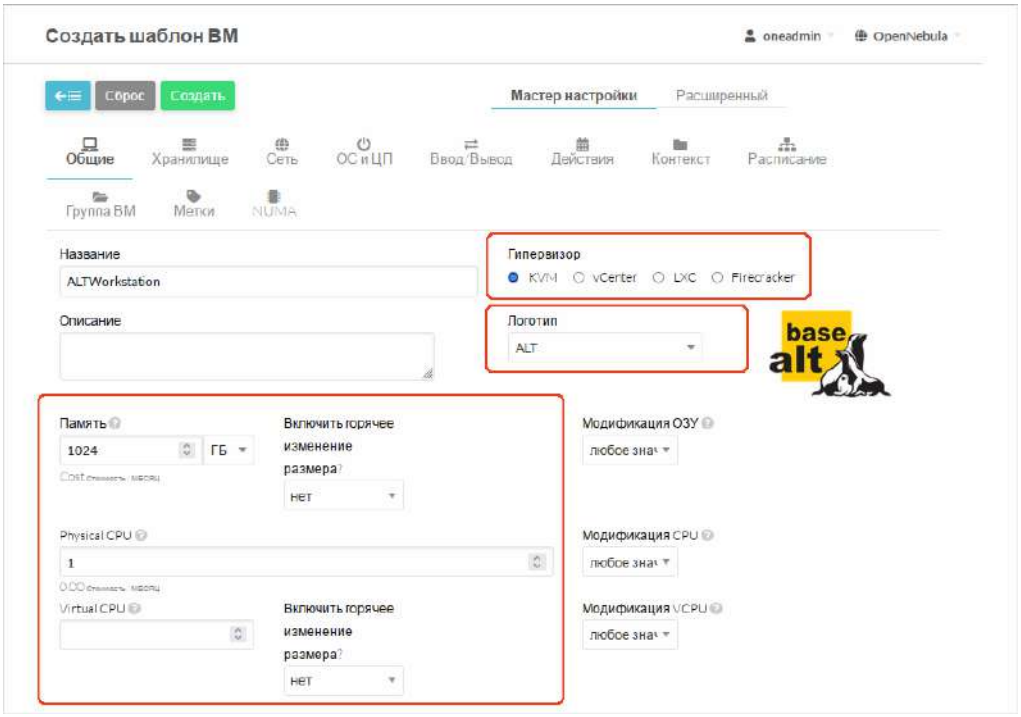


Рис. 35

Создание шаблона ВМ. Вкладка «Хранилище»

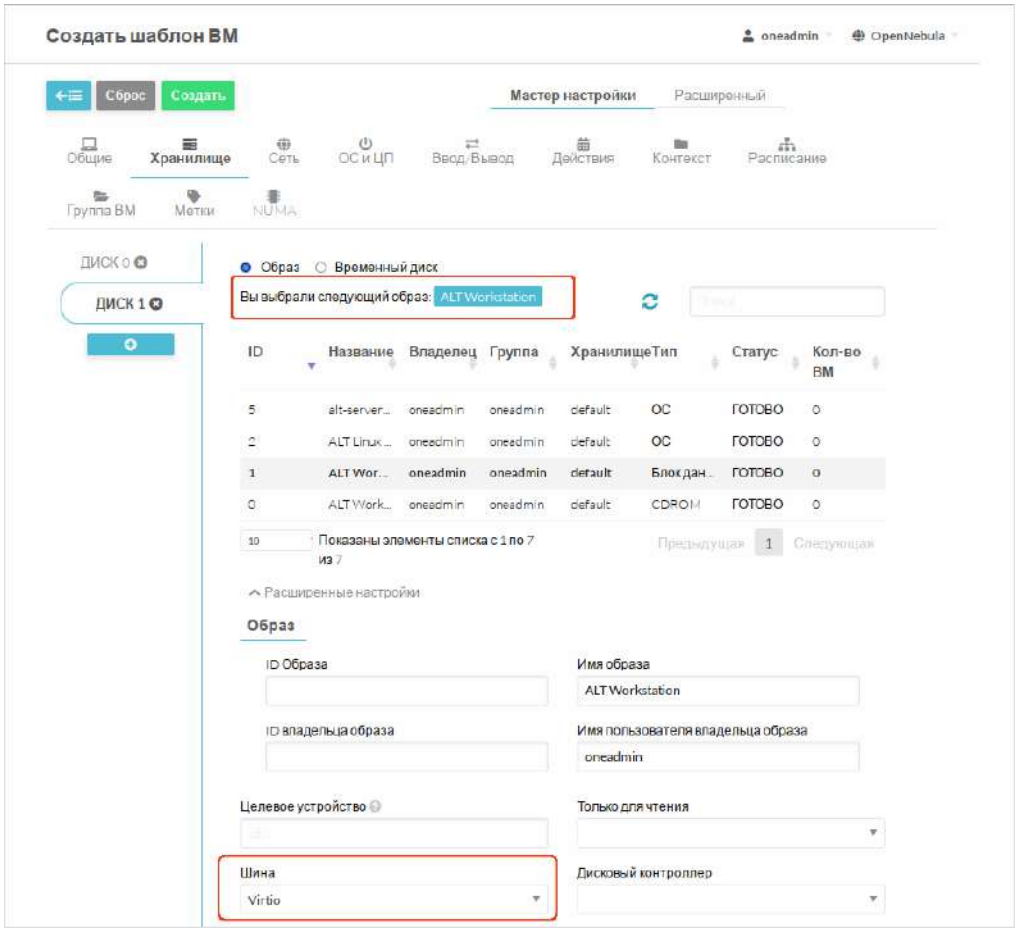


Рис. 36

На вкладке «Сеть» в поле «Default hardware model to emulate for all NICs» следует указать Virtio и если необходимо выбрать сеть (Рис. 37).

Создание шаблона ВМ. Вкладка «Сеть»

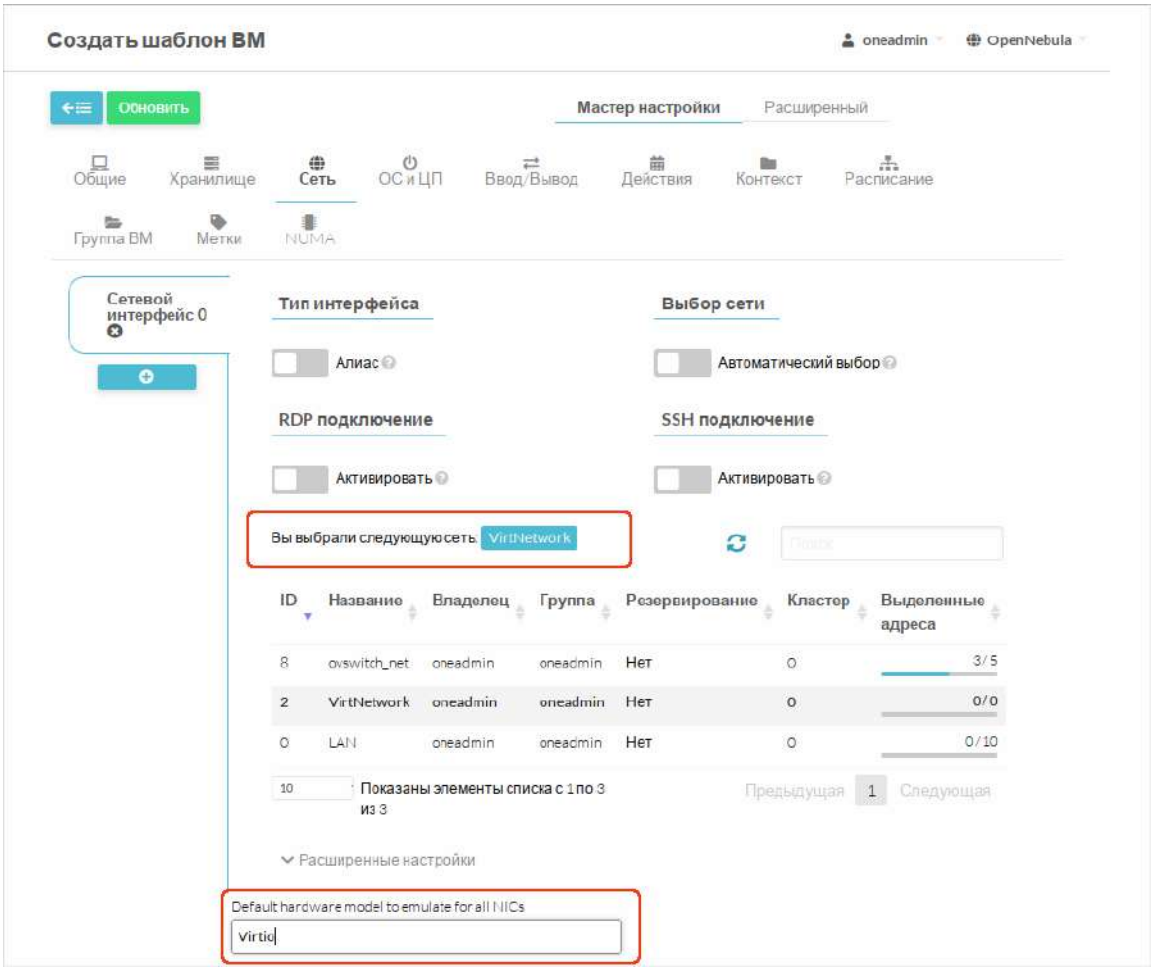


Рис. 37

На вкладке «ОС и ЦПУ» необходимо указать архитектуру устанавливаемой системы и выбрать порядок загрузки. Можно установить в качестве первого загрузочного устройства – пустой диск (DATABLOCK), а в качестве второго – CDROM (Рис. 38). При такой последовательности загрузочных устройств при пустом диске загрузка произойдёт с CDROM, а в дальнейшем, когда ОС будет уже установлена на диск, загрузка будет осуществляться с него.

На вкладке «Ввод/Вывод» следует включить «SPICE» (Рис. 39).

Создание шаблона ВМ. Вкладка «ОС и ЦПУ»

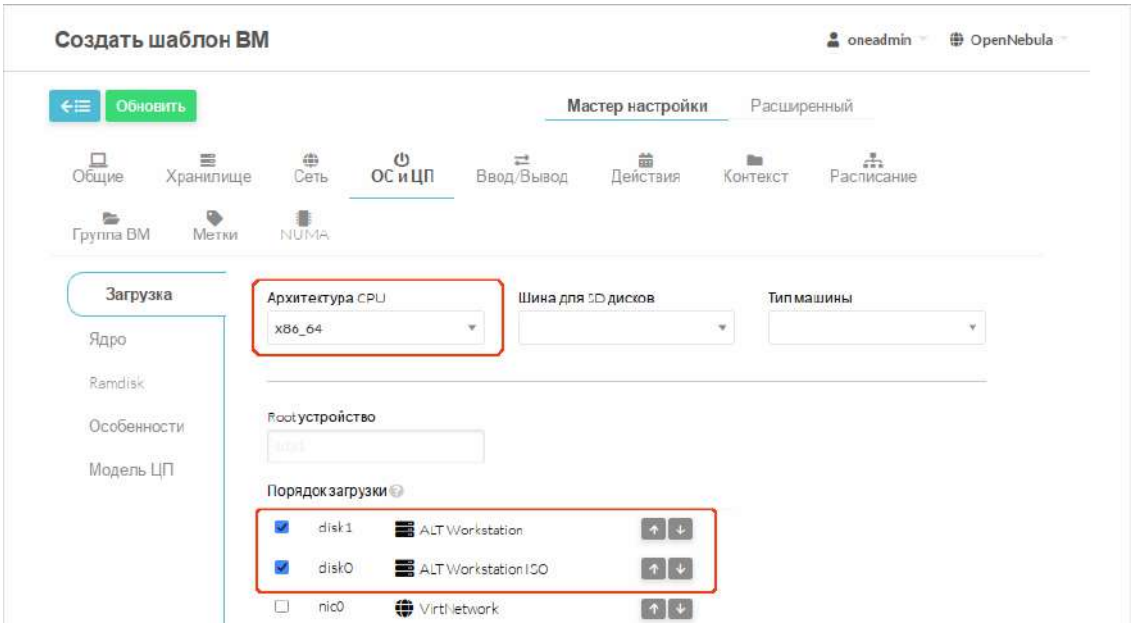


Рис. 38

Создание шаблона ВМ. Вкладка «Ввод/Вывод»

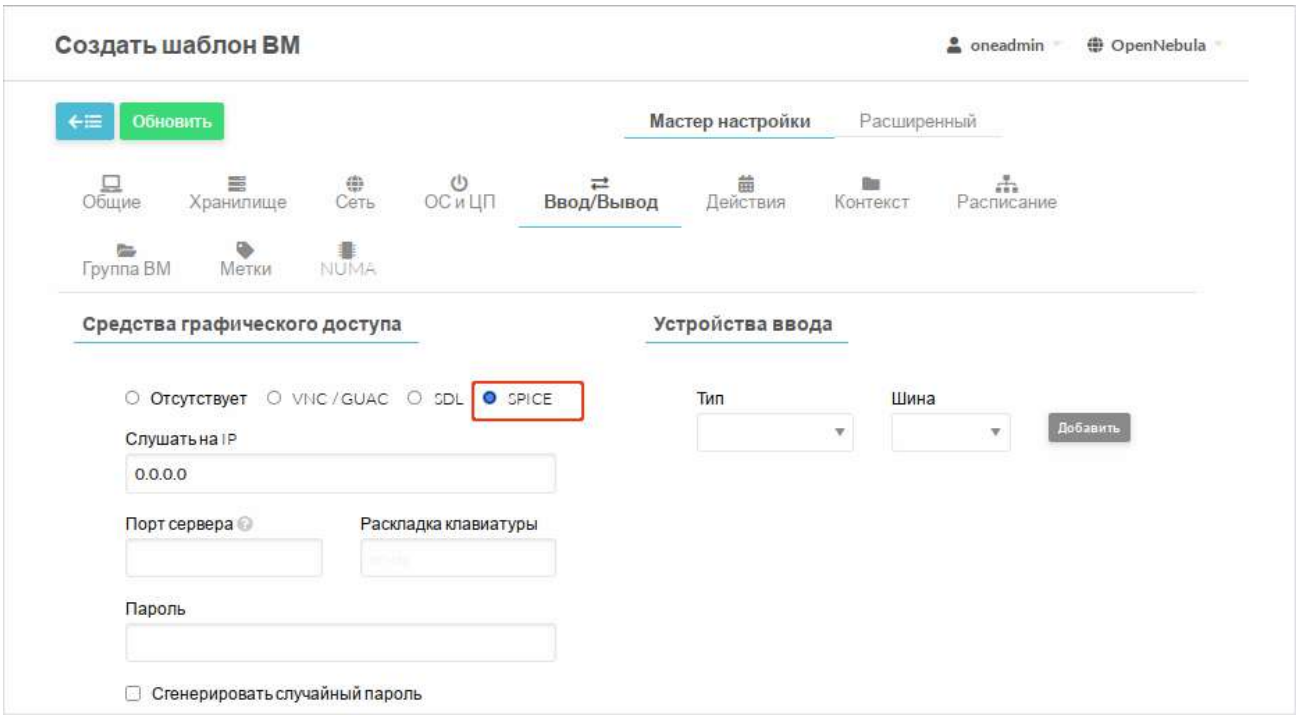


Рис. 39

На вкладке «Контекст» необходимо включить параметр «Использовать сетевое задание контекста», а также авторизацию по RSA-ключам (Рис. 40). Укажите открытый SSH (.pub) для доступа к ВМ по ключу. Если оставить это поле пустым, будет использована переменная $USER[SSH_PUBLIC_KEY].

Создание шаблона ВМ. Вкладка «Контекст»

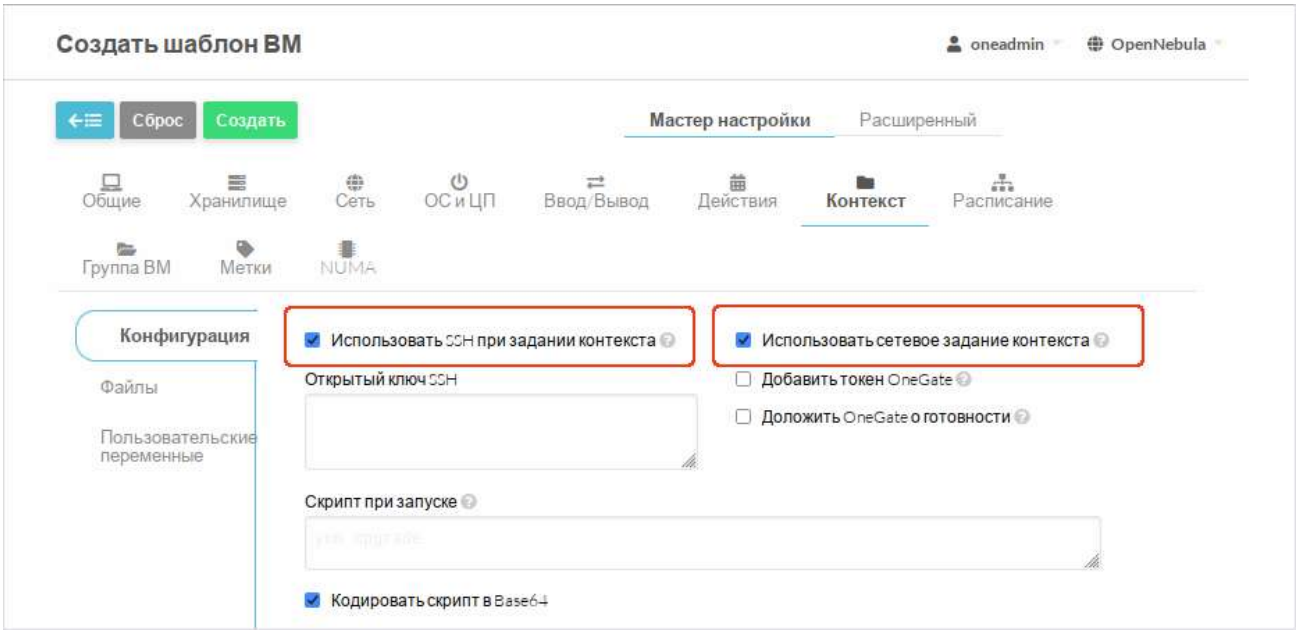


Рис. 40

На вкладке «Расписание» если необходимо можно выбрать кластер/узел, на котором будет размещаться виртуальное окружение (Рис. 41).

Создание шаблона ВМ. Вкладка «Расписание»

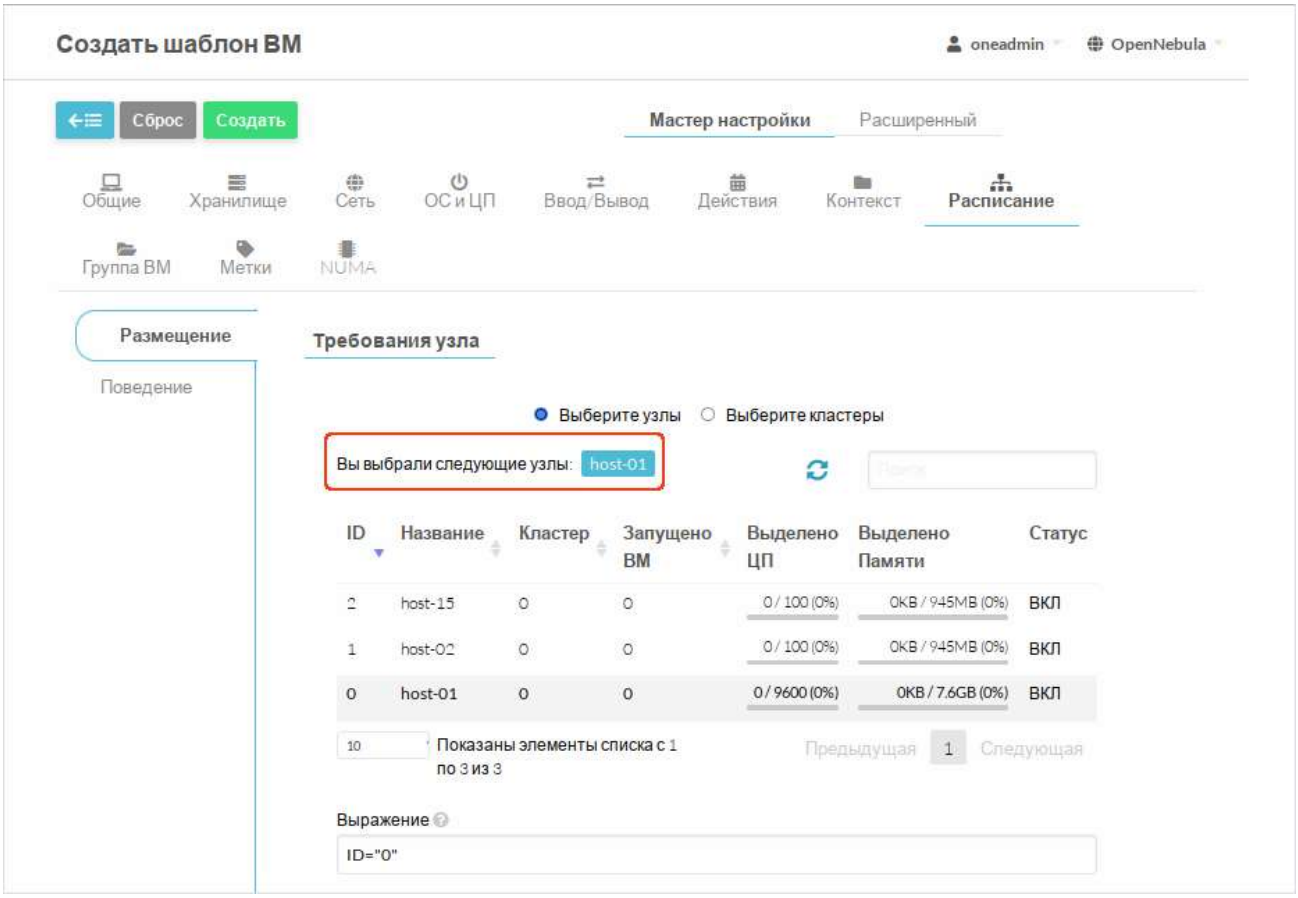


Рис. 41

Для создания шаблона ВМ нажать кнопку «Создать».

3.7.1.3 Создание ВМ

Для инициализации установки ОС из созданного шаблона в левом меню следует выбрать пункт «Шаблоны» → «ВМ», выбрать шаблон и нажать кнопку «Создать ВМ» (Рис. 42).

Создание экземпляра ВМ из шаблона

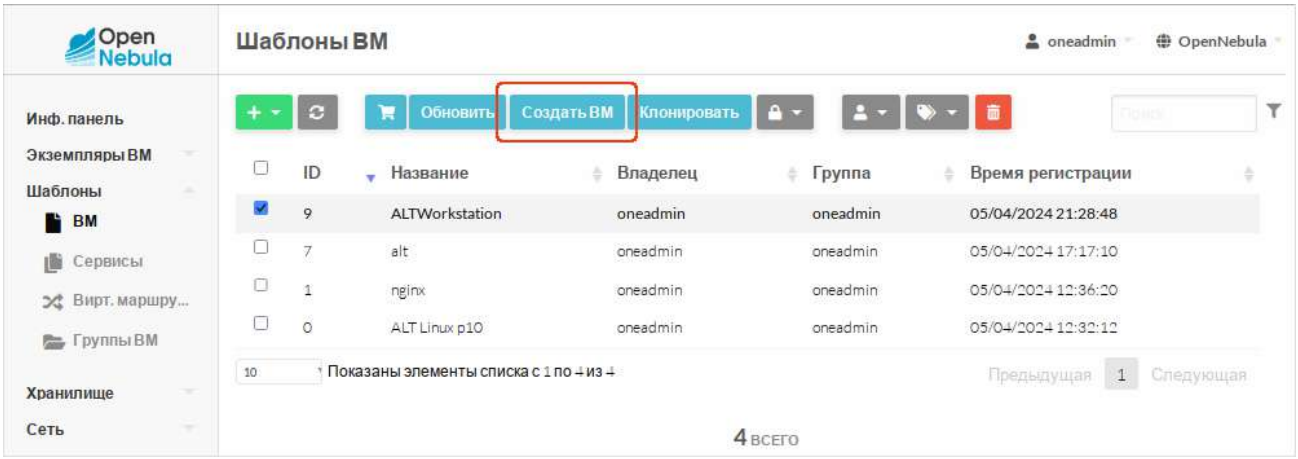


Рис. 42

В открывшемся окне необходимо указать имя ВМ и нажать кнопку «Создать ВМ» (Рис. 43).

Инициализация установки ОС из шаблона

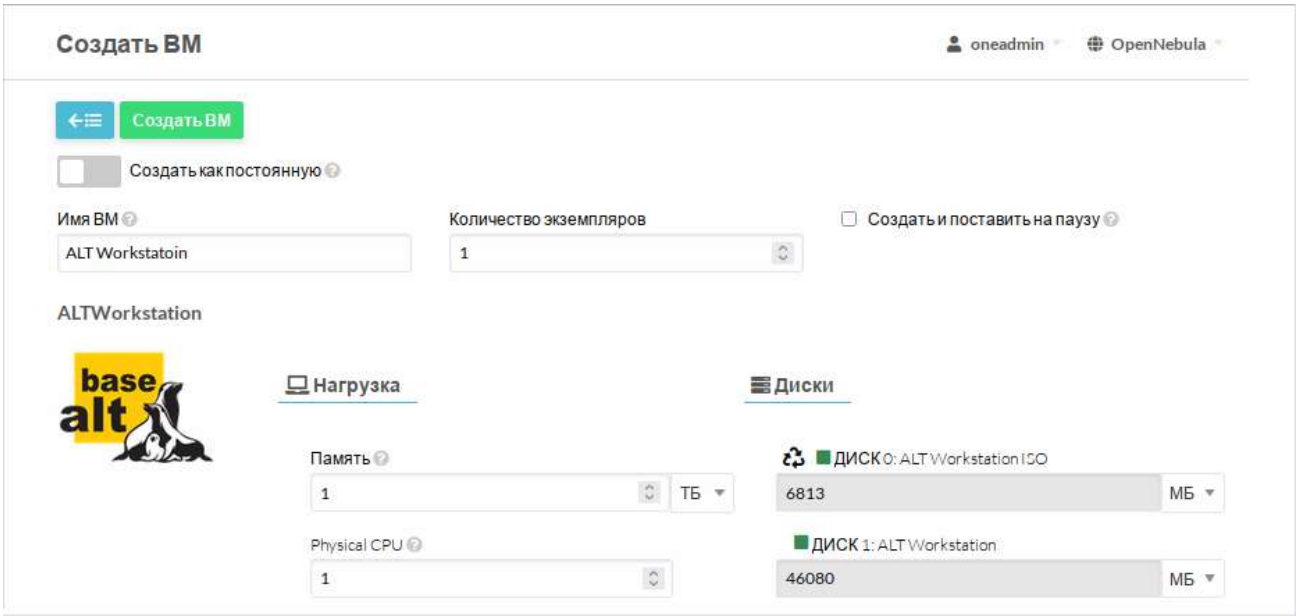


Рис. 43

Создание экземпляра ВМ из шаблона в командной строке:

```
$ onetemplate instantiate 9
VM ID: 5
```

3.7.1.4 Подключение к ВМ и установка ОС

Примечание. Процесс создания ВМ может занять несколько минут. Следует дождаться статуса – «ЗАПУЩЕНО» («RUNNING»).

Подключиться к ВМ можно как из веб-интерфейса Sunstone, раздел «Экземпляры ВМ» → «ВМ» выбрать ВМ и подключиться по SPICE (Рис. 44).

Подключение к ВМ

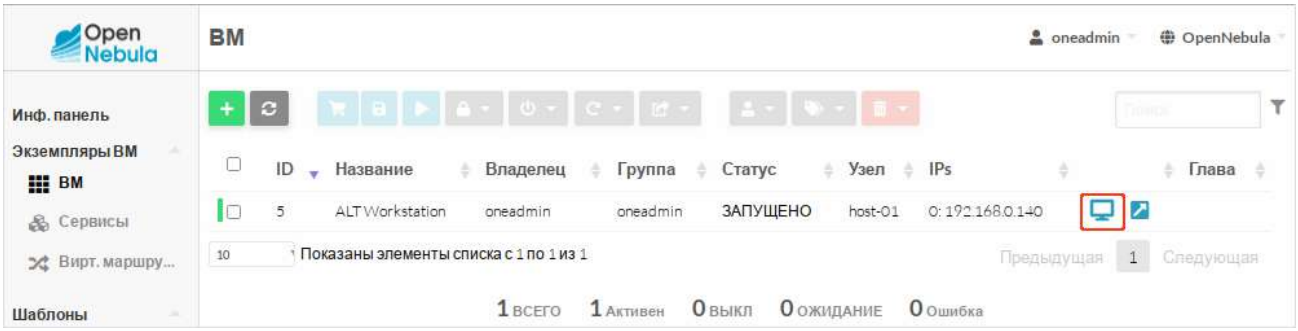


Рис. 44

Так и используя, любой клиент SPICE:

```
spice://192.168.0.180:5905
```

где 192.168.0.180 – IP-адрес узла с ВМ, а 5 – идентификатор ВМ (номер порта 5900 + 5).

Далее необходимо провести установку системы (Рис. 45).

Установка ВМ

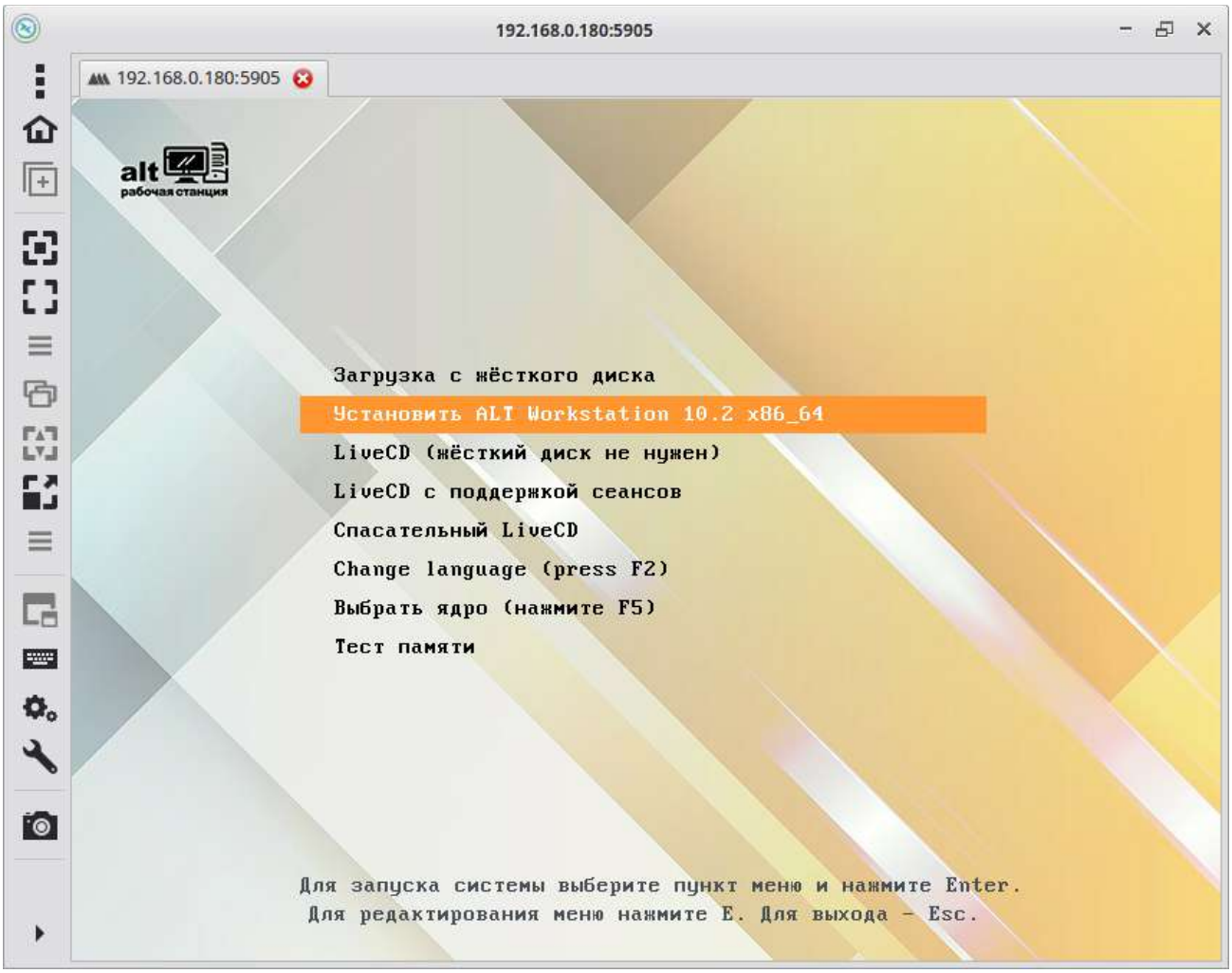


Рис. 45

3.7.1.5 Настройка контекстуализации

OpenNebula использует метод, называемый контекстуализацией, для отправки информации на ВМ во время загрузки. Контекстуализация позволяет установить или переопределить данные ВМ, имеющие неизвестные значения или значения по умолчанию (имя узла, IP-адрес, .ssh/ authorized_keys).

Пример настройки контекстуализации на ОС «Альт»:

1) подключиться к ВМ через SPICE или по SSH.

2) установить пакет opennebula-context:

```
# apt-get update && apt-get install opennebula-context
```

3) переключиться на systemd-networkd:

- установить пакет systemd-timesyncd:

```
# apt-get install systemd-timesyncd
```

- создать файл автонастройки всех сетевых интерфейсов по DHCP `/etc/systemd/network/lan.network` со следующим содержимым:

```
[Match]
Name = *

[Network]
DHCP = ipv4
```

- переключиться с etcnet/NetworkManager на systemd-networkd:

```
# systemctl disable network NetworkManager && systemctl enable systemd-networkd systemd-timesyncd
```

4) перезагрузить систему.

После перезагрузки доступ в систему будет возможен по ssh-ключу, ВМ будет назначен IP-адрес, который OpenNebula через механизм IPAM (подсистема IP Address Management) выделит из пула адресов.

3.7.1.6 Создание образа типа ОС

После завершения установки системы следует выключить и удалить ВМ. Диск находится в состоянии «Persistent», поэтому все внесенные изменения будут постоянными.

Для удаления ВМ в левом меню следует выбрать пункт «Экземпляры ВМ» → «ВМ», выбрать ВМ и нажать кнопку «Уничтожить» (Рис. 46).

Удаление ВМ

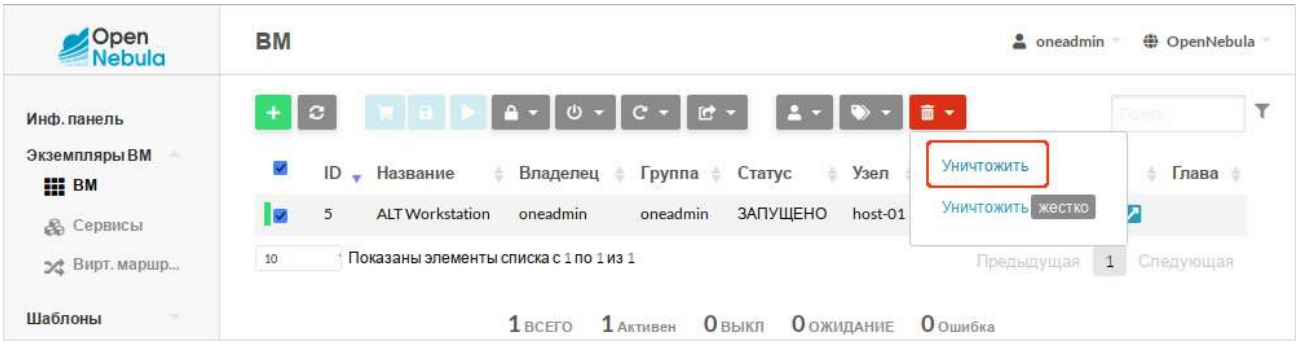


Рис. 46

Примечание. Удаление ВМ в командной строке:

```
$ onevm terminate 5
```

Затем перейти в «Хранилище» → «Образы ВМ», выбрать образ с установленной ОС (ALT Workstation) и изменить тип блочного устройства с «Блок данных» на «ОС» и состояние на «Не постоянный» (Рис. 47).

Примечание. Изменить тип блочного устройства на OS и состояние на Non Persistent в командной строке:

```
$ oneimage chtype 1 OS
$ oneimage nonpersistent 1
```

Образ готов. Далее можно использовать как имеющийся шаблон, так и создать новый на основе образа диска «ALT Workstation».

Изменение типа блочного устройства

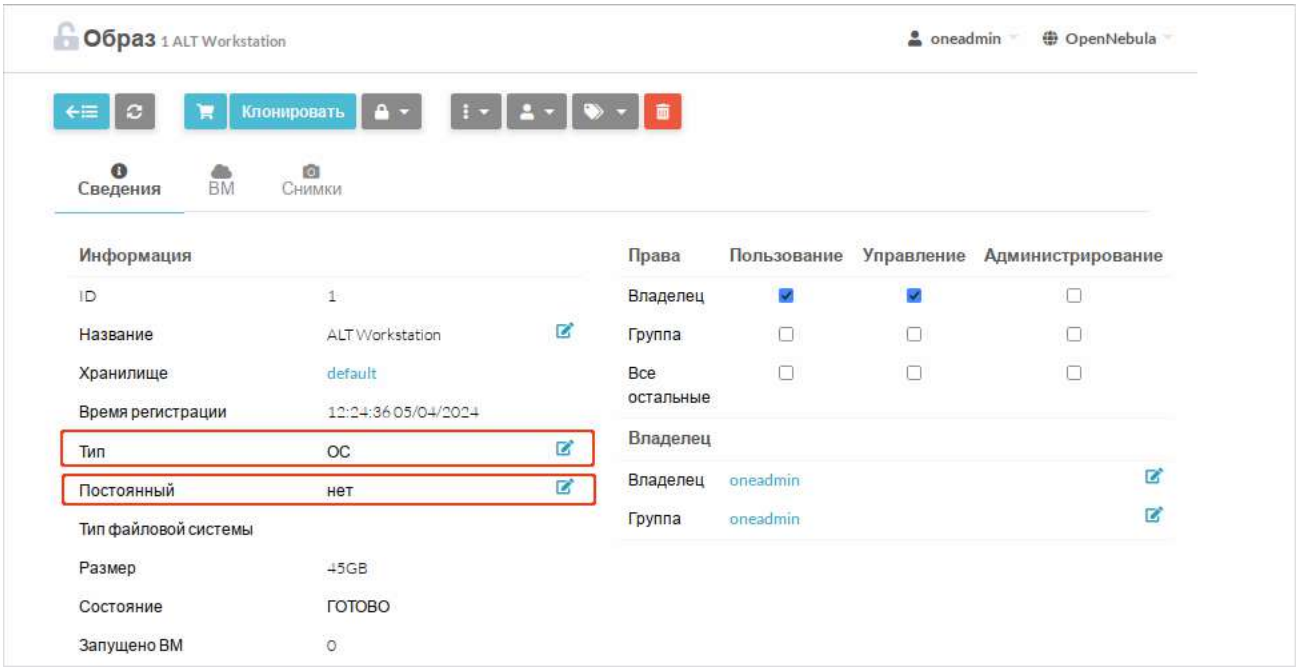


Рис. 47

3.7.2 Использование магазинов приложений OpenNebula

Магазины приложений OpenNebula предоставляют простой способ интеграции облака с популярными поставщиками приложений и изображений.

Список доступных магазинов можно увидеть, выбрав в меню «Хранилище» → «Магазины приложений» (Рис. 48).

Список магазинов приложений, настроенных в OpenNebula, можно вывести, выполнив команду:

```
$ onemarket list
  ID NAME                                SIZE AVAIL        APPS MAD      ZONE STAT
   3 DockerHub                             0M -             175 dockerh     0 on
   2 TurnKey Linux Containers              0M -               0 turnkey     0 on
   1 Linux Containers                      0M -               0 linuxco     0 on
   0 OpenNebula Public                     0M -              54 one         0 on
```

3.7.2.1 OpenNebula Public

OpenNebula Public – это каталог виртуальных устройств, готовых к работе в среде OpenNebula.

Для загрузки приложения из магазина OpenNebula Public необходимо выбрать «OpenNebula Public» → «Приложения». Появится список доступных приложений (Рис. 49).

Магазины приложений OpenNebula

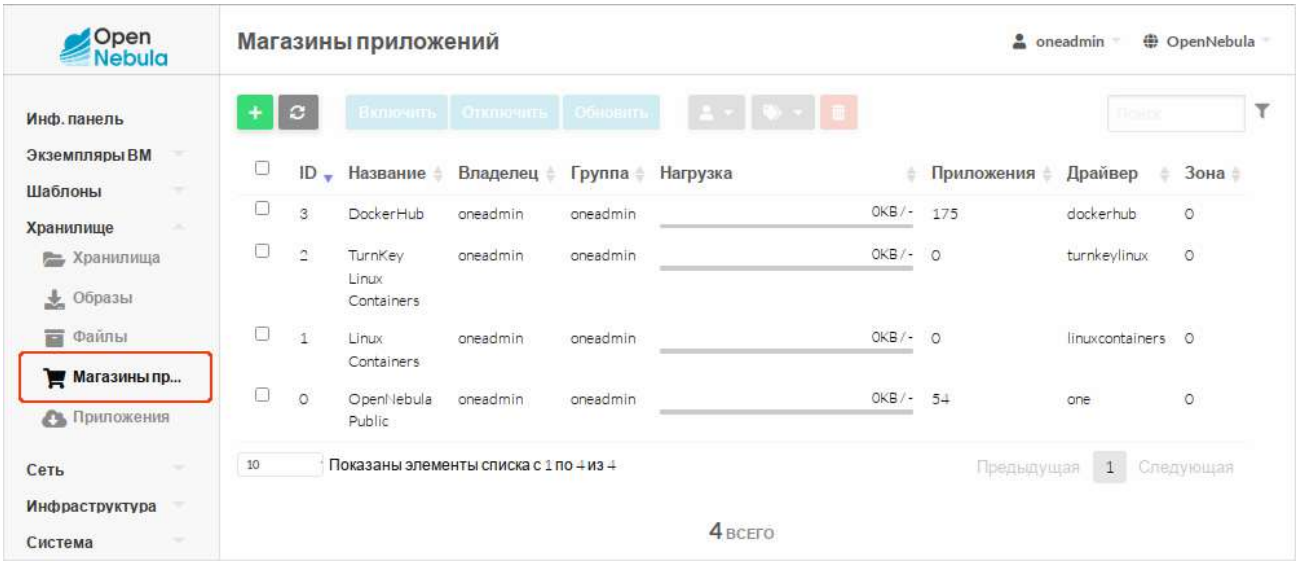


Рис. 48

Магазин приложений OpenNebula Public

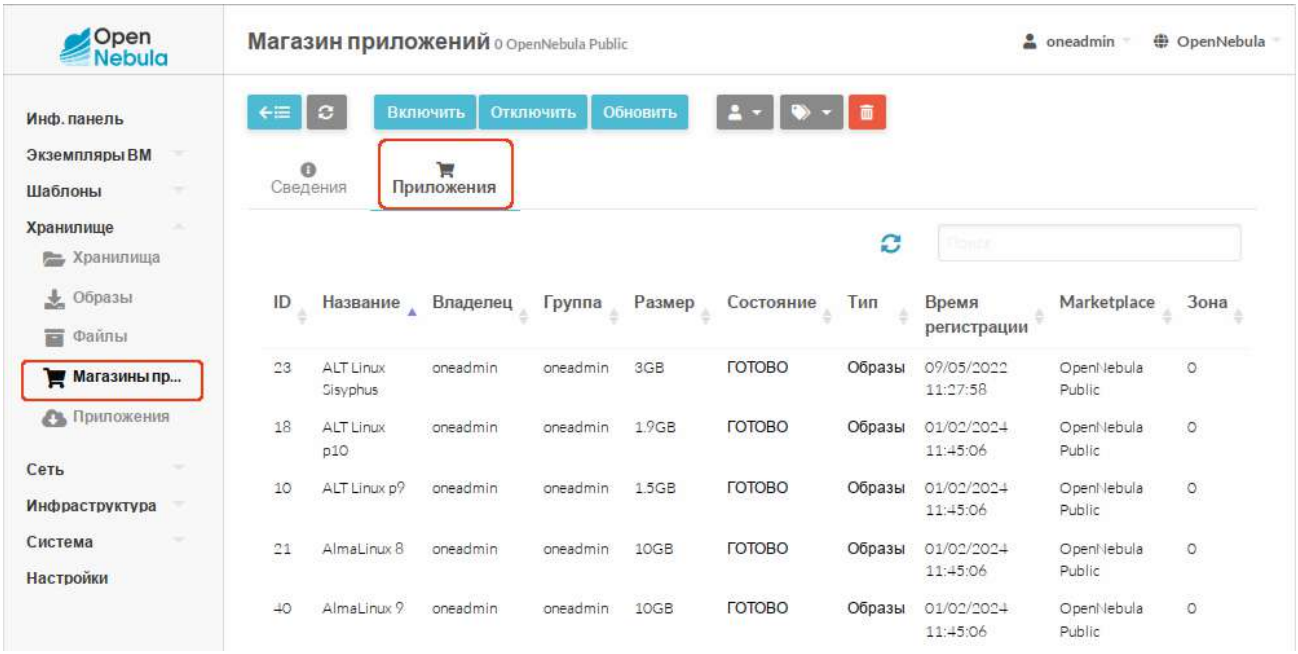


Рис. 49

Каждое приложение содержит образ и шаблон.

Чтобы импортировать приложение, необходимо его выбрать и нажать кнопку «Импорт в хранилище» (Рис. 50).

Информация о приложении в магазине приложений OpenNebula Public

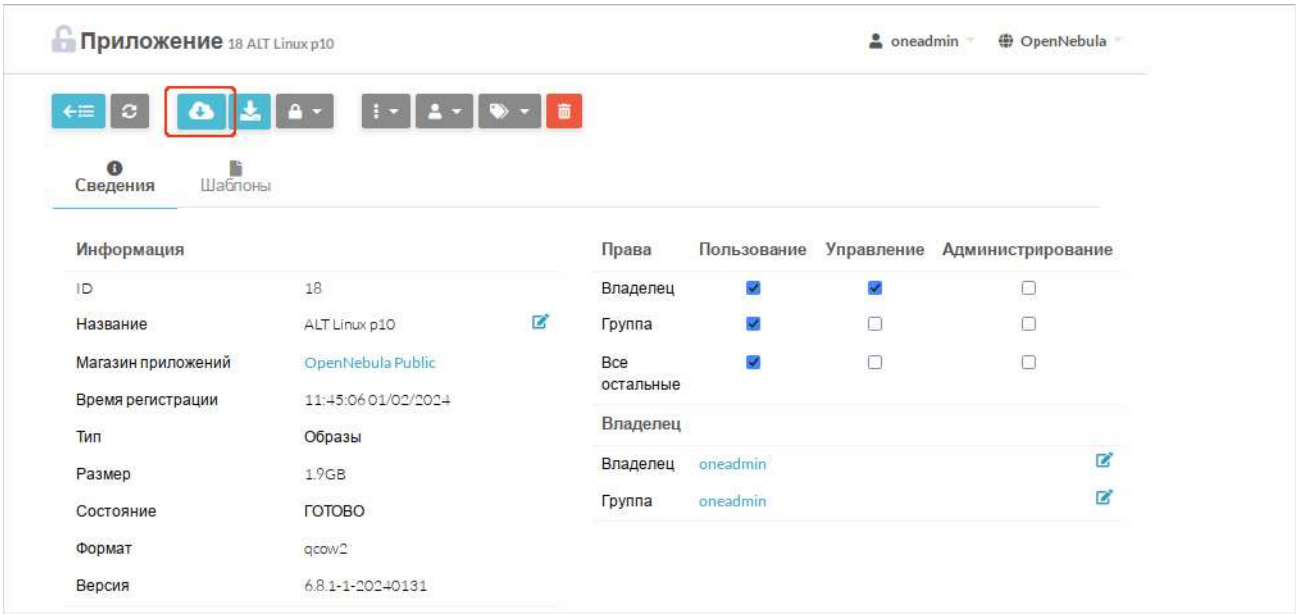


Рис. 50

В открывшемся окне указать имя для образа и шаблона, выбрать хранилище и нажать кнопку «Загрузить» (Рис. 51).

Импорт приложения из магазина приложений OpenNebula

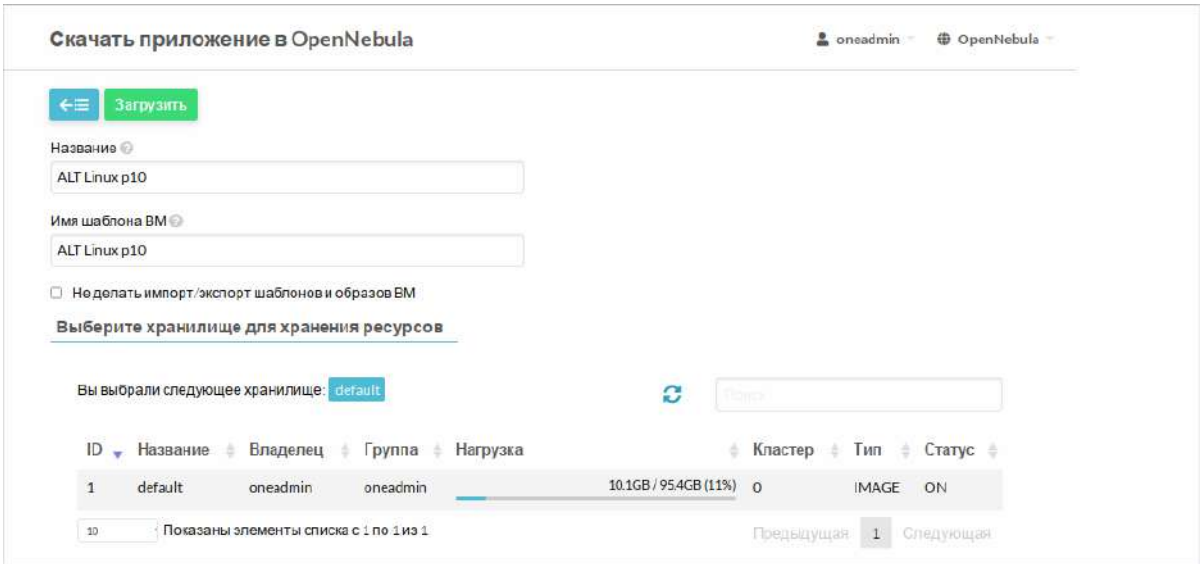


Рис. 51

Настройка образов, загруженных из магазина приложений:

1) изменить состояние образа на «Постоянный» (необходимо дождаться состояния «Готово»);

2) настроить шаблон;

3) создать на основе шаблона ВМ;

4) подключиться к ВМ. Установить/настроить необходимые компоненты;

5) удалить ВМ;

6) изменить состояние образа на «Не постоянный»;

7) далее можно создать новые шаблоны на основе этого образа или использовать существующий.

3.7.2.2 Скачивание шаблона контейнера из DockerHub

Магазин DockerHub предоставляет доступ к официальным образам DockerHub. Контекст OpenNebula устанавливается в процессе импорта образа, поэтому после импорта образ полностью готов к использованию.

Примечание. Для возможности загрузки контейнеров из магазина приложений DockerHub на сервере управления необходимо:

- установить Docker:

```
# apt-get install docker-engine
```

- добавить пользователя oneadmin в группу docker:

```
# gpasswd -a oneadmin docker
```

и выполнить повторный вход в систему

- запустить и добавить в автозагрузку службу docker:

```
# systemctl enable --now docker
```

- перезапустить opennebula:

```
# systemctl restart opennebula
```

Для загрузки контейнера из магазина DockerHub необходимо перейти в «Хранилище» → «Магазины приложений», выбрать «DockerHub» → «Приложения» (Рис. 52).

Магазин приложений DockerHub

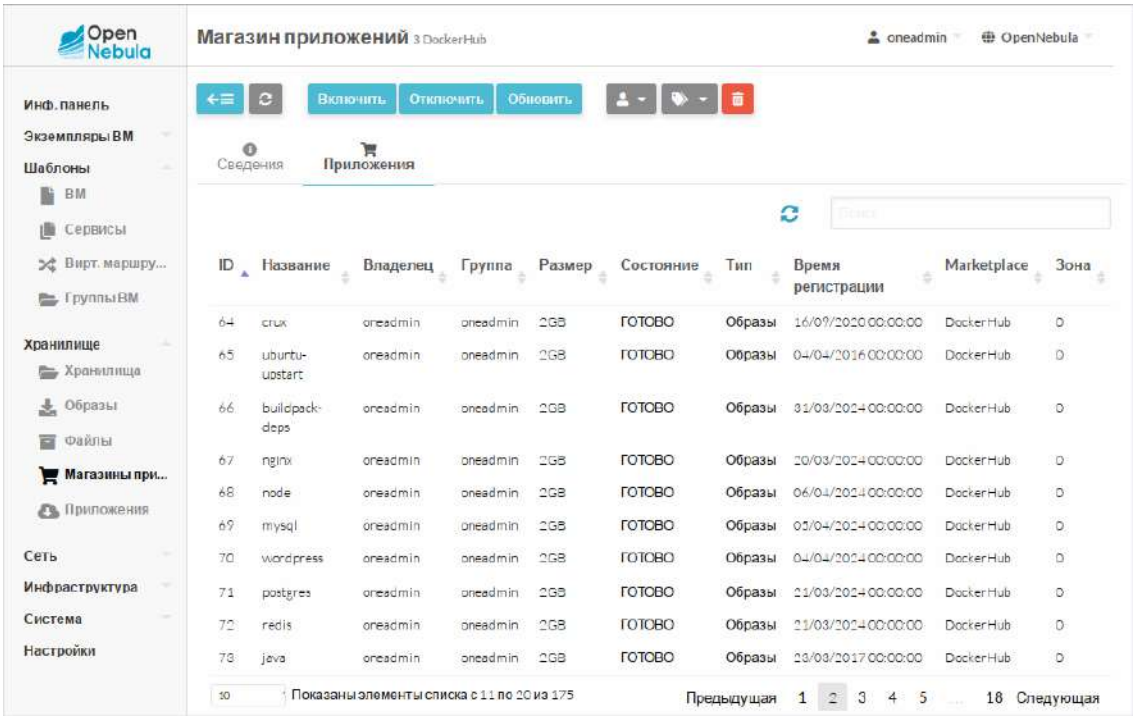


Рис. 52

Чтобы импортировать контейнер, необходимо его выбрать и нажать кнопку «Импорт в хранилище» (Рис. 53).

Информация о контейнере в магазине приложений DockerHub

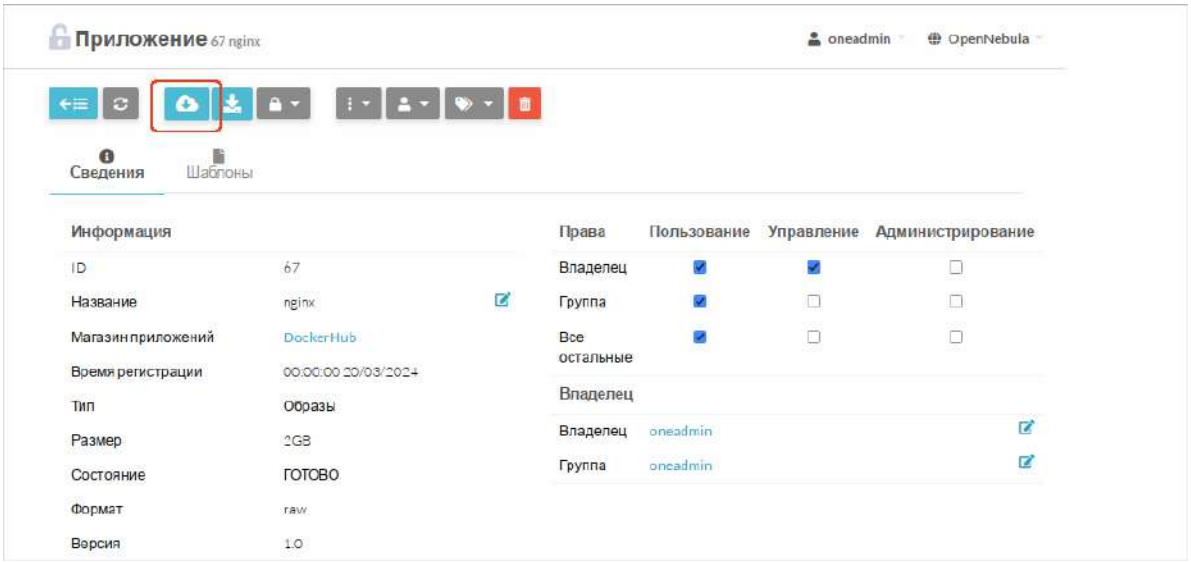


Рис. 53

Каждый контейнер содержит образ и шаблон.

В открывшемся окне указать название для образа и шаблона, выбрать хранилище и нажать кнопку «Загрузить» (Рис. 54).

Импорт контейнера из магазина DockerHub

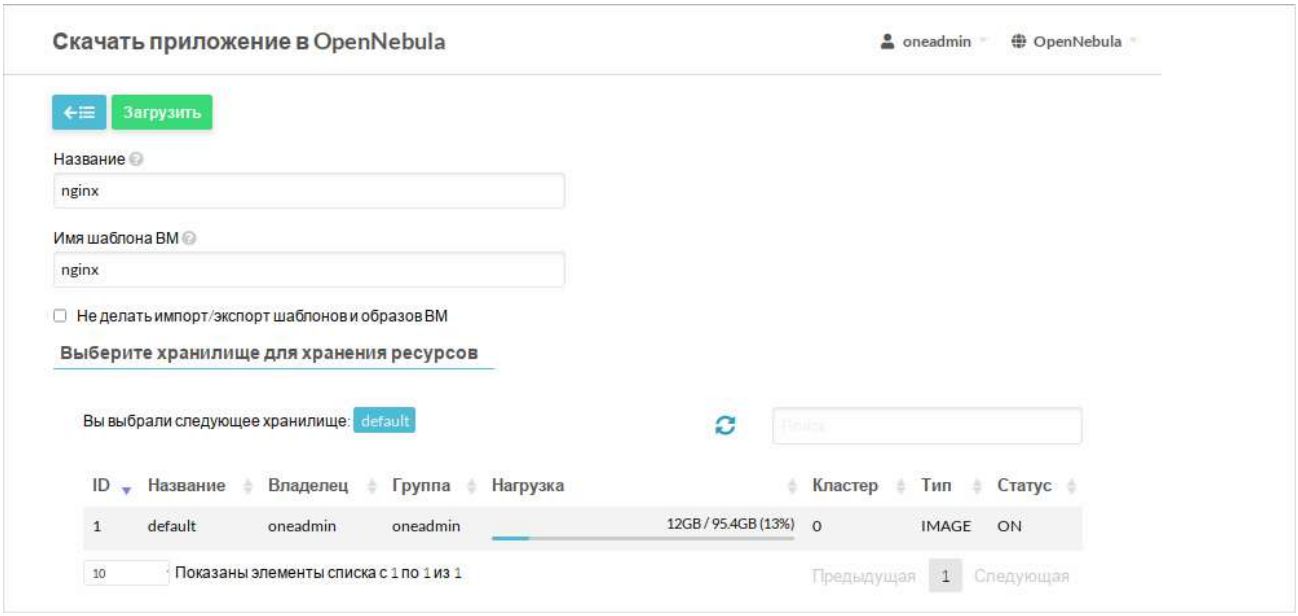


Рис. 54

Из полученного шаблона можно разворачивать контейнеры (ВМ в терминологии Opennebula). Процесс разворачивания контейнера из шаблона такой же, как и процесс разворачивания ВМ из шаблона (Рис. 55).

Разворачивание контейнера из шаблона

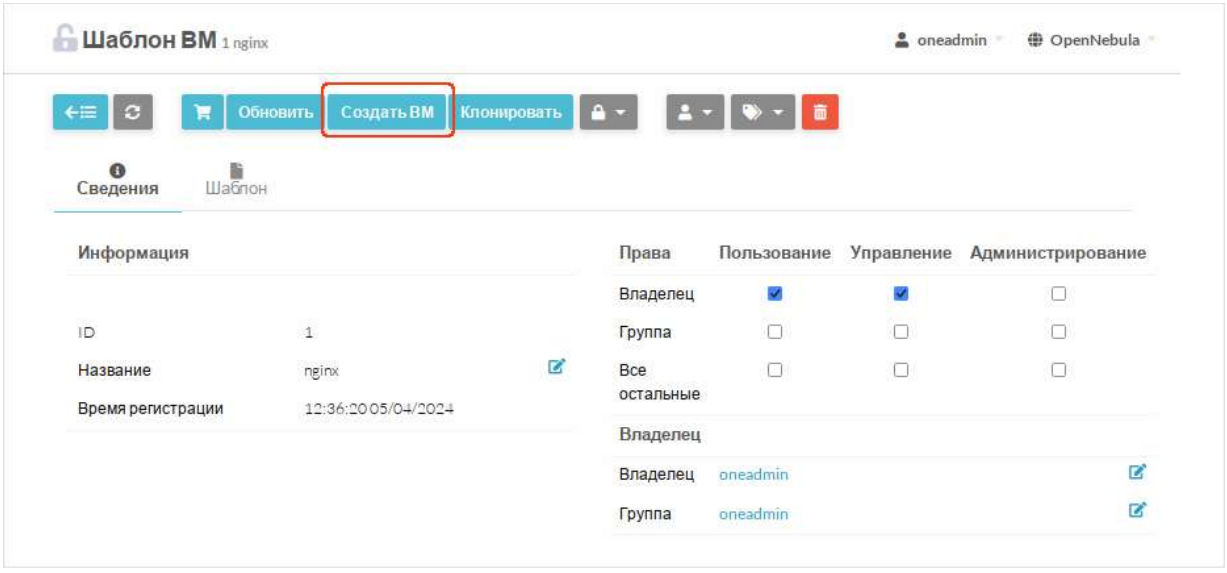


Рис. 55

В Магазине приложений DockerHub перечислены только официальные образы. Для того чтобы использовать неофициальный образ, следует создать образ (oneimage create), используя в качестве PATH (или опции --path) URL-адрес следующего формата:

```
docker://<image>?
size=<image_size>&filesystem=<fs_type>&format=raw&tag=<tag>&distro=<distro>
```

где:

- image – имя образа DockerHub;
- image_size – размер результирующего образа в МБ (этот размер должен быть больше фактического размера образа);
- fs_type – тип файловой системы (ext4, ext3, ext2 или xfs);
- tag – тег образа (по умолчанию latest).
- distro – дистрибутив образа (опционально).

Примечание. OpenNebula автоматически определяет дистрибутив образа, запуская контейнер и проверяя файл /etc/os-release. Если эта информация недоступна внутри контейнера, необходимо использовать аргумент distro.

Например, чтобы создать новый образ alt-p10 на основе образа alt из DockerHub размером 3 ГБ с использованием ext4 и тега p10, можно выполнить команду:

```
$ oneimage create --name alt-p10 --path 'docker://alt?
size=3072&filesystem=ext4&format=raw&tag=p10' --datastore 1
   ID: 22
```

Примечание. Этот формат URL-адреса также можно использовать в диалоговом окне создания образа в веб-интерфейсе (Рис. 56).

Новый образ из DockerHub

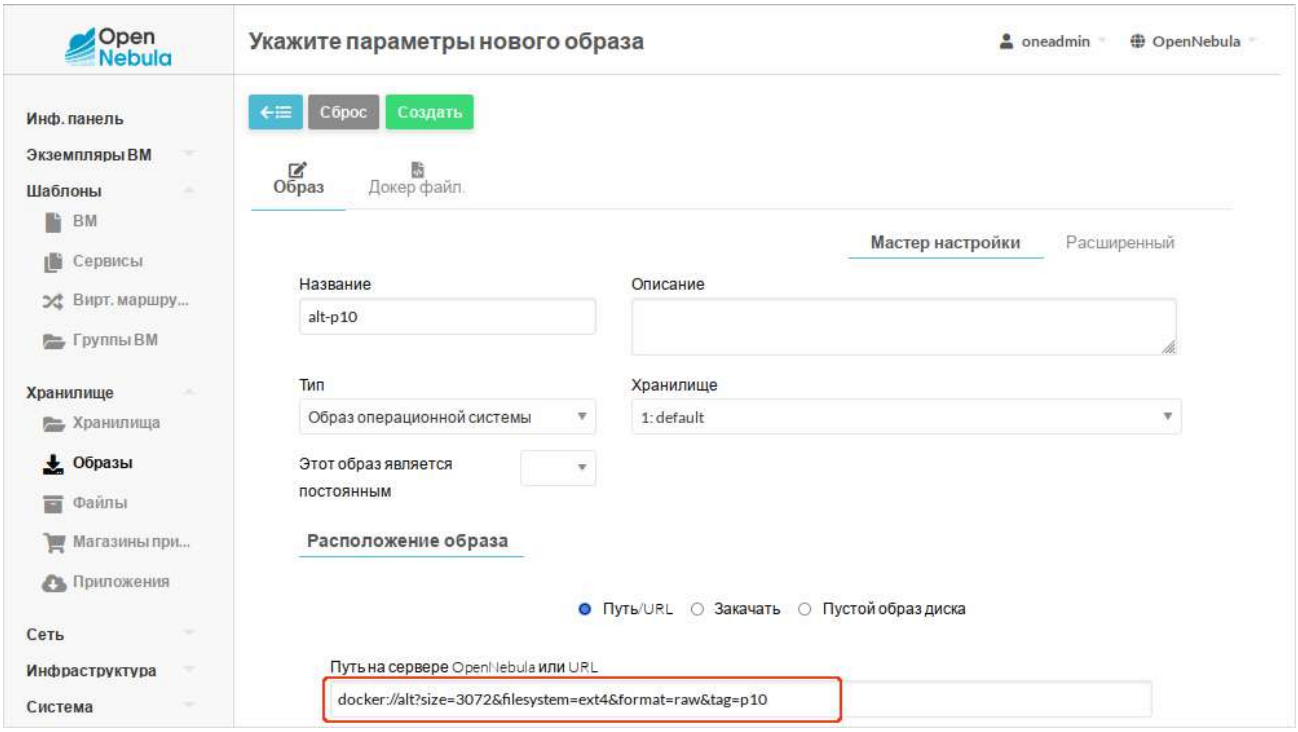


Рис. 56

3.8 Управление пользователями

OpenNebula поддерживает учётные записи пользователей и группы.

Ресурсы, к которым пользователь может получить доступ в OpenNebula, контролируются системой разрешений. По умолчанию только владелец ресурса может использовать и управлять им. Пользователи могут делиться ресурсами, предоставляя разрешения на использование или управление другим пользователям в своей группе или любому другому пользователю в системе.

3.8.1 Пользователи

Пользователь в OpenNebula определяется именем пользователя и паролем. Каждый пользователь имеет уникальный идентификатор и принадлежит как минимум к одной группе.

При установке OpenNebula создаются две административные учетные записи (oneadmin и serveradmin).

oneuser – инструмент командной строки для управления пользователями в OpenNebula.

Посмотр списка пользователей:

```
$ oneuser list
  ID NAME                ENAB GROUP    AUTH            VMS     MEMORY        CPU
   1 serveradmin         yes  oneadmin server_c    0 /   -      0M /   0.0 /   -
   0 oneadmin            yes  oneadmin core             -          -          -
```

Создание нового пользователя:

```
$ oneuser create <user_name> <password>
```

По умолчанию новый пользователь будет входить в группу users. Изменить группу пользователя можно, выполнив команду:

```
$ oneuser chgrp <user_name> oneadmin
```

Что бы удалить пользователя из группы, необходимо переместить его обратно в группу users.

Удалить пользователя:

```
$ oneuser delete <user_name>
```

Временно отключить пользователя:

```
$ oneuser disable <user_name>
```

Включить отключённого пользователя:

```
$ oneuser enable <user_name>
```

Все операции с пользователями можно производить в веб-интерфейсе (Рис. 57).

Управление пользователями в OpenNebula-Sunstone

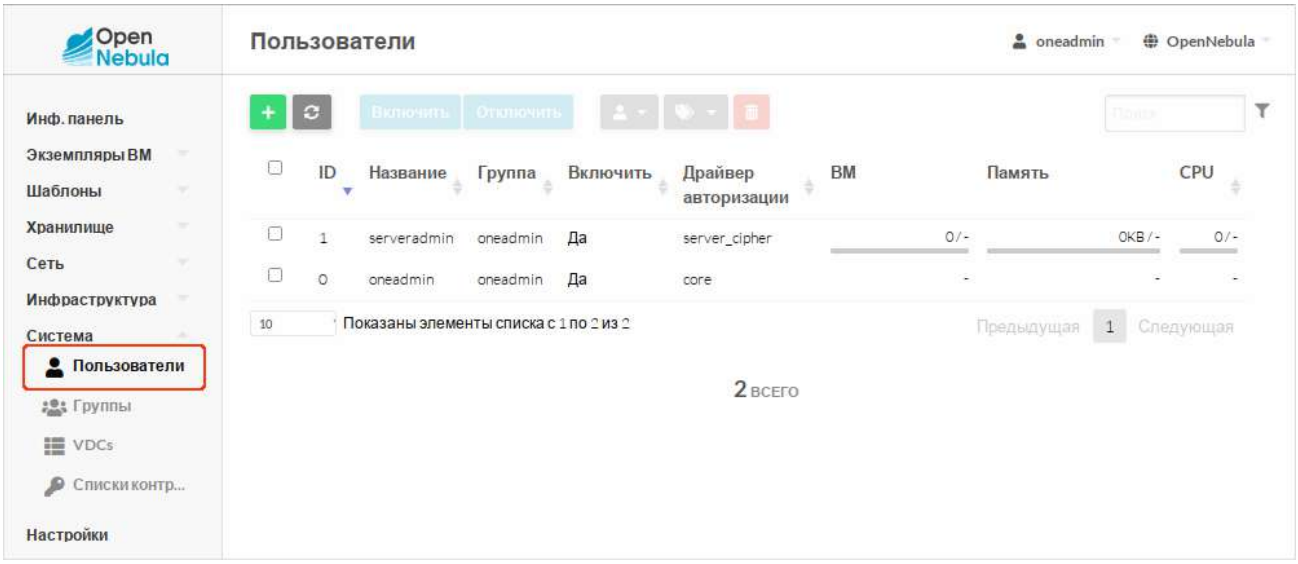


Рис. 57

Пользователь может аутентифицироваться в веб-интерфейсе OpenNebula и изменить настройки (изменить язык интерфейса, пароль, добавить ssh-ключ для доступа на ВМ и т.д.) (Рис. 58).

Примечание. Пользователи могут просматривать информацию о своей учётной записи и изменять свой пароль.

Панель пользователя в OpenNebula-Sunstone

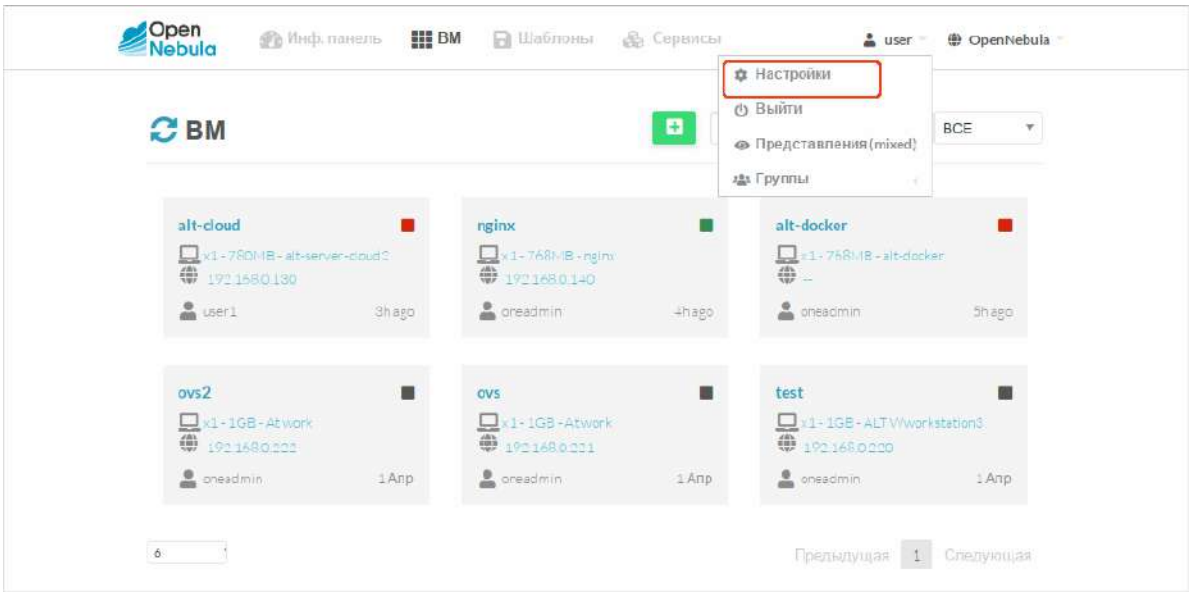


Рис. 58

3.8.2 Группы пользователей

При установке OpenNebula создаются две группы (oneadmin и users).

onegroup – инструмент командной строки для управления группами в OpenNebula.

Просмотр списка групп:

```
$ onegroup list
  ID NAME                 USERS       VMS             MEMORY          CPU
   1 users                    1   0 /   -       0M /       -  0.0 /    -
   0 oneadmin                 3         -                  -           -
```

Создание новой группы:

```
$ onegroup create group_name
ID: 100
```

Новая группа получила идентификатор 100, чтобы отличать специальные группы от групп, созданных пользователем.

После создания группы может быть создан связанный пользователь-администратор. По умолчанию этот пользователь сможет создавать пользователей в новой группе.

Пример создания новой группы с указанием, какие ресурсы могут быть созданы пользователями группы (по умолчанию VM+IMAGE+TEMPLATE):

```
$ onegroup create --name testgroup \
--admin_user testgroup-admin --admin_password somestr \
--resources TEMPLATE+VM
```

При выполнении данной команды также будет создан администратор группы.

Сделать существующего пользователя администратором группы:

```
$ onegroup addadmin <groupid_list> <userid>
```

Все операции с группами можно производить в веб-интерфейсе (Рис. 59).

Создание группы в OpenNebula-Sunstone

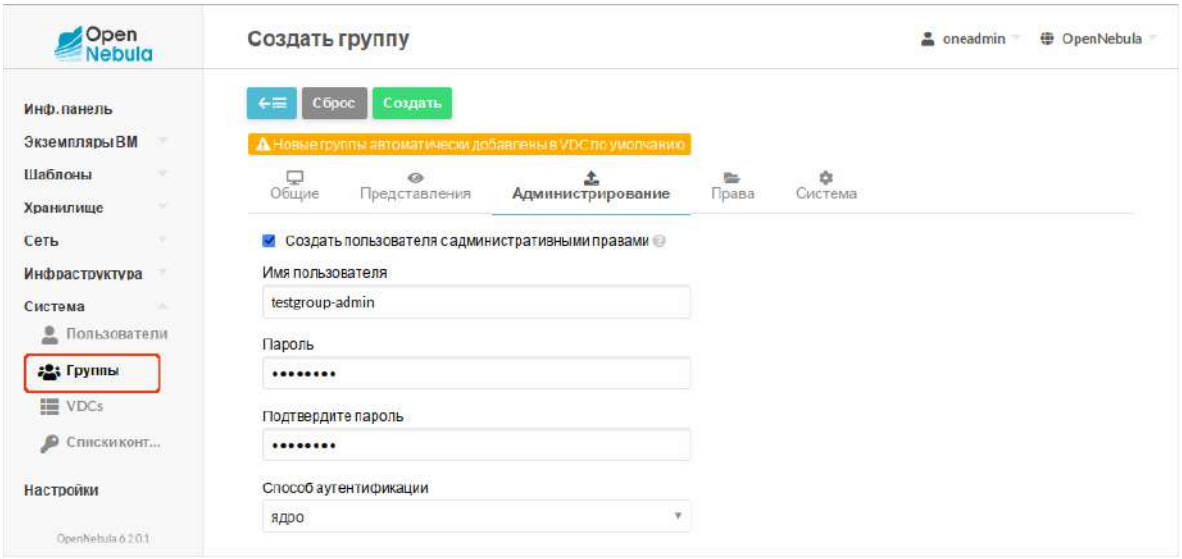


Рис. 59

3.8.3 Управление разрешениями

У ресурсов OpenNebula (шаблонов, ВМ, образов и виртуальных сетей) есть права назначенные владельцу, группе и всем остальным. Для каждой из этих групп можно установить три права: «Использование» (use), «Управление» (manage) и «Администрирование» (admin).

Просмотреть/изменить права доступа можно в веб-интерфейсе, выбрав соответсвующий ресурс (Рис. 60).

Управление разрешениями в OpenNebula-Sunstone

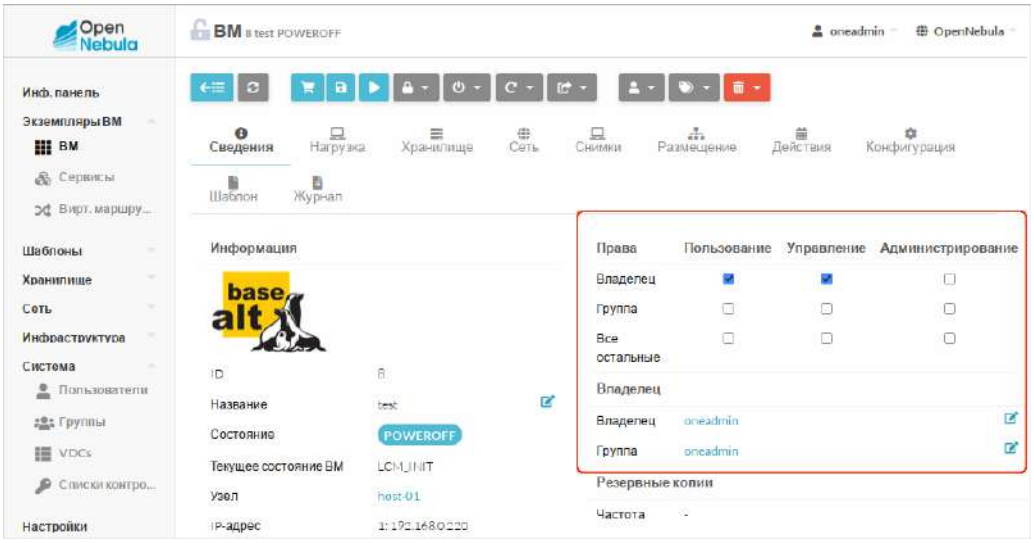


Рис. 60

Просмотреть права можно и в командной строке:

```
$ onevm show 8
VIRTUAL MACHINE 8 INFORMATION
ID                  : 8
NAME                : test
USER                : oneadmin
GROUP               : oneadmin
STATE               : POWEROFF
LCM_STATE           : LCM_INIT
LOCK                : None
RESCHED             : No
HOST                : host-01
CLUSTER ID          : 0
CLUSTER             : default
START TIME          : 04/08 16:12:53
END TIME            : -
DEPLOY ID           : 69ab21c2-22ad-4afb-bfc1-7b4e4ff2364f

VIRTUAL MACHINE MONITORING
ID                  : 8
```

```
TIMESTAMP           : 1712756284

PERMISSIONS
OWNER               : um-
GROUP               : ---
OTHER               : ---
…
```

В данном примере показаны права на DV с ID=8.

Для изменения прав в командной строке используется команда `chmod`. Права записываются в числовом формате. Пример изменения прав:

```
$ onevm chmod 8 607
$ onevm show 8
…
PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : uma
```

Примечание. Разрешения по умолчанию для создаваемых ресурсов могут быть установлены:

- глобально, в `oned.conf` (атрибут DEFAULT_UMASK);
- индивидуально для каждого пользователя с помощью команды `oneuser umask`.

Маска должна состоять из 3 восьмеричных цифр. Каждая цифра – это маска, которая, соответственно, отключает разрешение для владельца, группы и всех остальных. Например, если значение маски равно 137, то для нового объекта будут установлены права 640 (um- u-- ---).

3.8.4 Управление правилами ACL

Система правил ACL позволяет точно настроить разрешенные операции для любого пользователя или группы пользователей. Каждая операция генерирует запрос на авторизацию, который проверяется на соответствие зарегистрированному набору правил ACL. Затем ядро может дать разрешение или отклонить запрос.

Просмотреть список правил можно, выполнив команду:

```
$ oneacl list
   0       @1     V--I-T---O-S----P-      *      ---c      *
   1        *     ----------Z-------      *      u---      *
   2        *     --------------MA--      *      u---      *
   3       @1     -H----------------      *      -m--     #0
   4       @1     --N---------------      *      u---     #0
   5       @1     -------D----------      *      u---     #0
   6       #3     ---I--------------    #30      u---     #
```

Данные правила соответсвуют следующим:

```
@1      VM+IMAGE+TEMPLATE+DOCUMENT+SECGROUP/*   CREATE  *
*       ZONE/*                                  USE     *
*       MARKETPLACE+MARKETPLACEAPP/*            USE     *
@1      HOST/*                                  MANAGE  #0
@1      NET/*                                   USE     #0
@1      DATASTORE/*                             USE     #0
#3      IMAGE/#30                               USE     *
```

Первые шесть правил были созданы при начальной загрузке OpenNebula, а последнее – с помощью `oneacl`:

```
$ oneacl create "#3 IMAGE/#30 USE"
ID: 6
```

Столбцы в выводе `oneacl list`:

- ID – идентификатор правила;
- USER – пользователь. В этом поле может быть указан идентификатор пользователя (#), идентификатор группы (@) или все пользователи (*);
- Resources – тип ресурса, к которому применяется правило:
  - V – ВМ;
  - H – узел;
  - N – виртуальная сеть;
  - I – образ;
  - U – пользователь;
  - T – шаблон;
  - G – группа;
  - D – хранилище;
  - C – кластер;
  - O – документ;
  - Z – зона;
  - S – группа безопасности;
  - v – виртуальный дата центр (VDC);
  - R – виртуальный маршрутизатор;
  - M – магазин приложений;
  - A – приложение из магазина приложений;
  - P – группа ВМ;
  - t – шаблон виртуальной сети;

- RID – идентификатор ресурса. В этом поле может быть указан идентификатор отдельного объекта (#), группы (@) или кластера (%), или все объекты (*);
- Operations – разрешённые операции:
  - U – использовать;
  - M – управлять;
  - A – администрировать;
  - C – создавать;
- Zone – зоны, в которых применяется правило. В этом поле может быть указан идентификатор отдельной зоны (#), или всех зон (*).

Для удаления правила используется команда:

```
$ oneacl delete <ID>
```

Управлять правилами ACL удобно в веб-интерфейсе (Рис. 61).

Для создания нового правила ACL, следует нажать кнопку «Создать». В открывшемся диалоговом окне можно определить ресурсы, на которые распространяется правило, и разрешения которые им предоставляются (Рис. 62).

Управление правилами ACL в OpenNebula-Sunstone

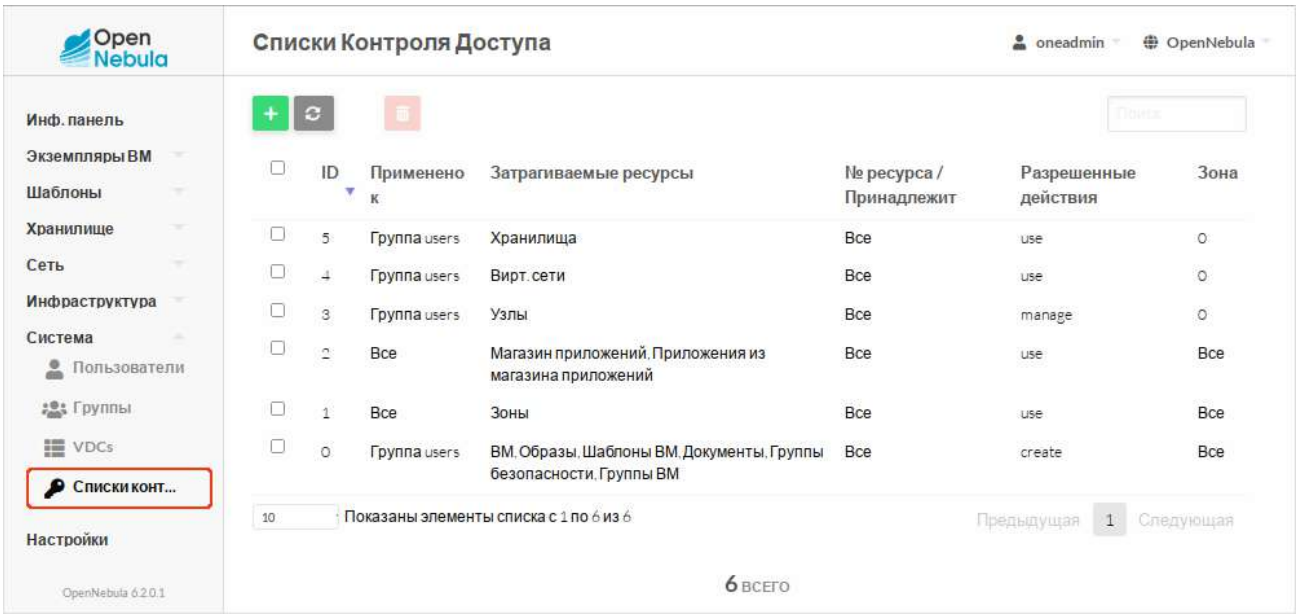


Рис. 61

Управление правилами ACL в OpenNebula-Sunstone

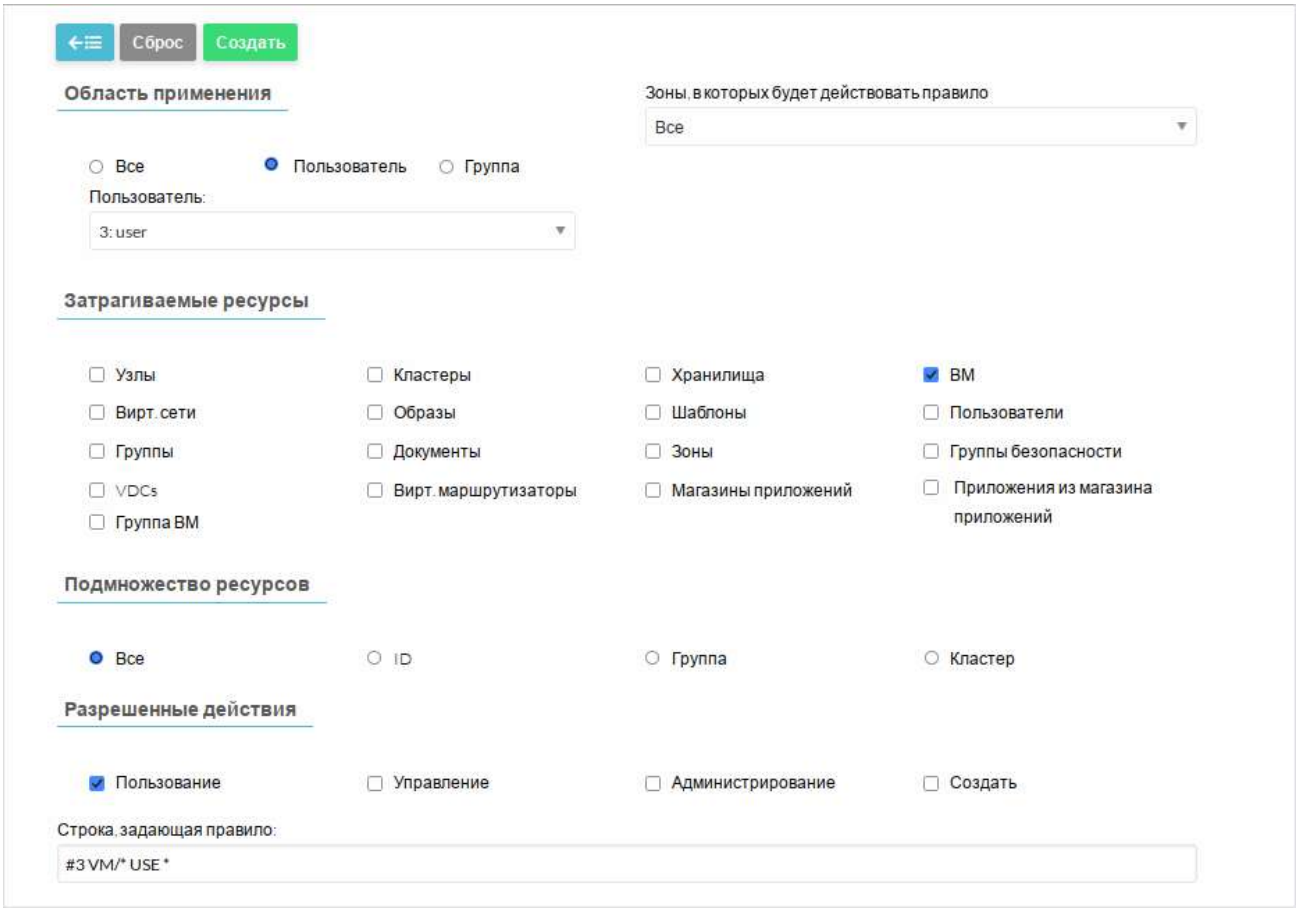


Рис. 62

Примечание. Каждое правило ACL добавляет новые разрешения и не может ограничивать существующие: если какое-либо правило даёт разрешение, операция разрешается.

3.8.5 Аутентификация пользователей

По умолчанию OpenNebula работает с внутренней системой аутентификации (пользователь/пароль). Но можно включить внешний драйвер аутентификации.

3.8.5.1 LDAP аутентификация

LDAP аутентификация позволяет пользователям и группам пользователей, принадлежащих практически любому аутентификатору на основе LDAP, получать доступ к виртуальным рабочим столам и приложениям.

Примечание. На сервере LDAP должна быть настроена отдельная учётная запись с правами чтения LDAP. От данной учетной записи будет выполняться подключение к серверу каталогов.

Для включения LDAP аутентификации необходимо в файл `/etc/one/oned.conf` добавить строку `DEFAULT_AUTH = "ldap"`:

…

```
AUTH_MAD = [
    EXECUTABLE = "one_auth_mad",
    AUTHN = "ssh,x509,ldap,server_cipher,server_x509"
]

DEFAULT_AUTH = "ldap"
…
```

Примечание. В файле /etc/one/sunstone-server.conf для параметра :auth должно быть указано значение opennebula:

```
:auth: opennebula
```

Ниже приведён пример настройки аутентификации в Active Directory (домен test.alt).

Для подключения к Active Directory в файле /etc/one/auth/ldap_auth.conf необходимо указать:

- :user – пользователь AD с правами на чтение (пользователь указывается в формате opennebula@test.alt);
- :password – пароль пользователя;
- :host – имя или IP-адрес сервера AD;
- :base – базовый DN для поиска пользователя;
- :user_field – для этого параметра следует установить значение sAMAccountName;
- :rfc2307bis – для этого параметра следует установить значение true.

Пример файла /etc/one/auth/ldap_auth.conf:

```
server 1:
    :user: 'opennebula@test.alt'
    :password: 'Pa$$word'
    :auth_method: :simple
    :host: dc1.test.alt
    :port: 389
    :base: 'dc=test,dc=alt'
    :user_field: 'sAMAccountName'
    :mapping_generate: false
    :mapping_timeout: 300
    :mapping_filename: server1.yaml
    :mapping_key: GROUP_DN
    :mapping_default: 100
    :rfc2307bis: true
:order:
    - server 1
```

Примечание. В параметре `:order` указывается порядк, в котором будут опрошены настроенные серверы. Элементы в параметре `:order` обрабатываются по порядку, пока пользователь не будет успешно аутентифицирован или не будет достигнут конец списка. Сервер, не указанный в параметре `:order`, не будет опрошен.

Примечание. Пример файла `/etc/one/auth/ldap_auth.conf` для настройки аутентификации в домене FreeIPA (домен example.test):

```
 server 1:
    :user: 'uid=admin,cn=users,cn=accounts,dc=example,dc=test'
    :password: '12345678'
    :auth_method: :simple
    :host: ipa.example.test
    :port: 389
    :base: 'dc=example,dc=test'
    :user_field: 'uid'
    :mapping_generate: false
    :mapping_timeout: 300
    :mapping_filename: server1.yaml
    :mapping_key: GROUP_DN
    :mapping_default: 100
    :rfc2307bis: true
:order:
    - server 1
```

После того как пользователь AD авторизуется в веб-интерфейсе OpenNebula, у администратора появится возможность изменять его настройки (Рис. 63).

Пользователи AD

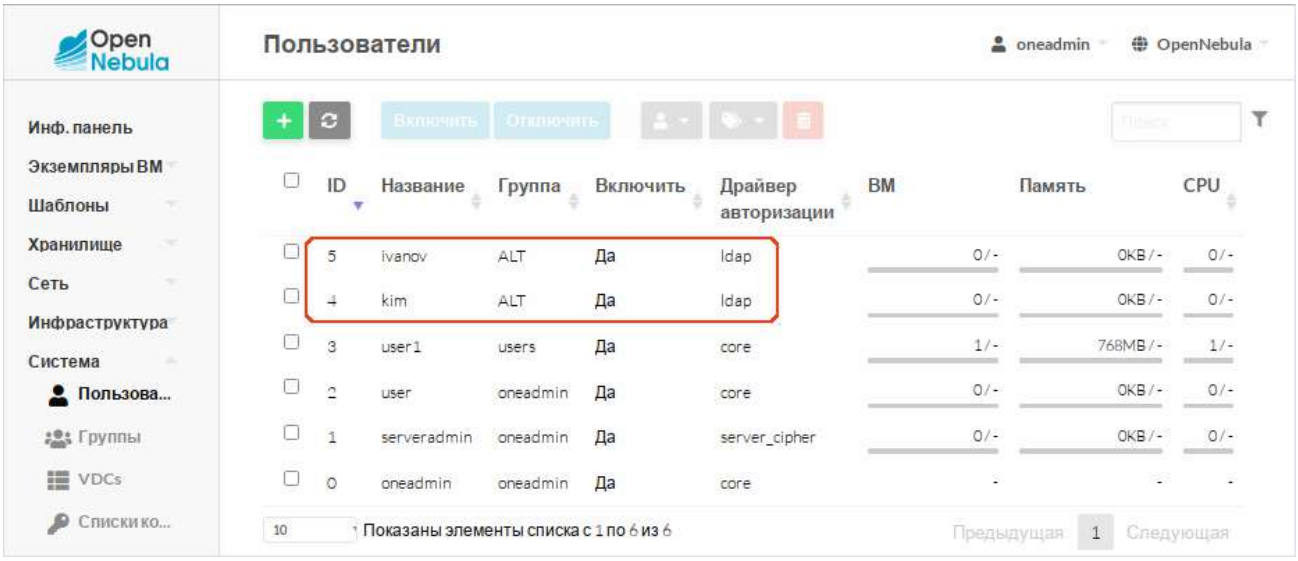


Рис. 63

Новых пользователей можно автоматически включать в определенную группу/группы. Для этого создается сопоставление группы AD с существующей группой OpenNebula. Эта система использует файл сопоставления, указанный в параметре `:mapping_file` (файл должен находиться в каталоге `/var/lib/one/`).

Файл сопоставления может быть сгенерирован автоматически с использованием данных в шаблоне группы, который определяет, какая группа AD сопоставляется с этой конкретной группой (для параметра `:mapping_generate` должно быть установлено значение true). Если в шаблон группы добавить строку (Рис. 64):

```
GROUP_DN="CN=office,CN=Users,DC=test,DC=alt"
```

И в файле `/etc/one/auth/ldap_auth.conf` для параметра `:mapping_key` установить значение GROUP_DN, то поиск DN сопоставляемой группы будет осуществляться в этом параметре шаблона. В этом случае файл `/var/lib/one/server1.yaml` будет сгенерирован со следующим содержимым:

```
---
CN=office,CN=Users,DC=test,DC=alt: '100'
```

и пользователи из группы AD office, будут сопоставлены с группой ALT (ID=100).

Шаблон группы

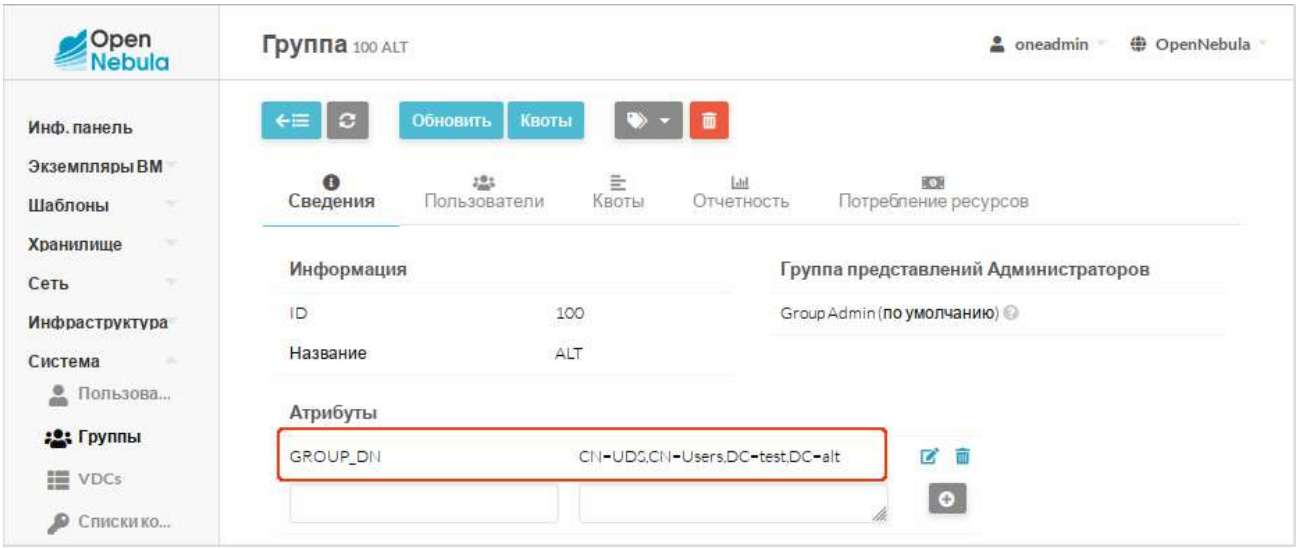


Рис. 64

Можно отключить автоматическую генерацию файла сопоставления, установив значение `:mapping_generate` равным false, и выполнить сопоставление вручную. Файл сопоставления имеет формат YAML и содержит хеш, где ключ – это DN группы AD, а значение – идентификатор группы OpenNebula. Например, если содержимое файла `/var/lib/one/server1.yaml`:

```
CN=office,CN=Users,DC=test,DC=alt: '100'
CN=Domain Admins,CN=Users,DC=test,DC=alt: '101'
```

то пользователи из группы AD office, будут сопоставлены с группой ALT (ID=100), а из группы AD Domain Admin – с группой Admin (ID=101) (Рис. 65).

Сопоставление пользователей AD

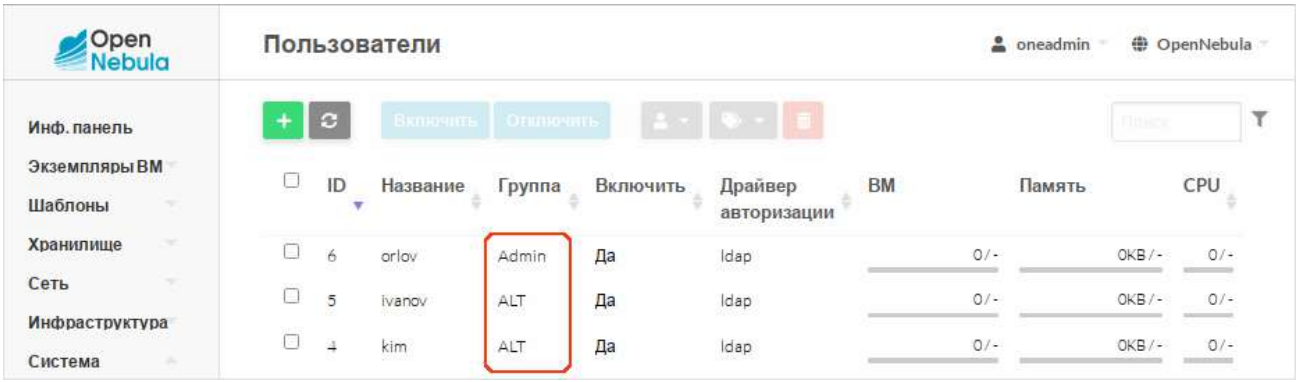


Рис. 65

3.9 Настройка отказоустойчивого кластера

В данном разделе рассмотрена настройка отказоустойчивого кластера (High Available, HA) для основных служб OpenNebula: core (oned), scheduler (mm_sched).

OpenNebula использует распределенный консенсусный протокол Raft, для обеспечения отказоустойчивости и согласованности состояний между службами. Алгоритм консенсуса построен на основе двух концепций:

- состояние системы – данные, хранящиеся в таблицах базы данных (пользователи, списки управления доступом или виртуальные машины в системе);
- журнал (log) – последовательность операторов SQL, которые последовательно применяются к базе данных OpenNebula на всех серверах для изменения состояния системы.

Чтобы сохранить согласованное представление о системе на всех серверах, изменения состояния системы выполняются через специальный узел, лидер или ведущий (Leader). Leader периодически посылает запросы (heartbeats) другим серверам, ведомым (Follower), чтобы сохранить свое лидерство. Если Leader не может послать запрос, Follower-серверы продвигаются к кандидатам и начинают новые выборы.

Каждый раз, когда система изменяется (например, в систему добавляется новая ВМ), Leader обновляет журнал и реплицирует запись у большинства Follower, прежде чем записывать её в базу данных. Таким образом, увеличивается задержка операций с БД, но состояние системы безопасно реплицируется, и кластер может продолжить свою работу в случае отказа узла.

Для настройки High Available требуется:

- нечетное количество серверов (рекомендуемый размер развертывания – 3 или 5 серверов, что обеспечивает отказоустойчивость при отказе 1 или 2 серверов соответственно);
- рекомендуется идентичная конфигурация серверов;

- идентичная программная конфигурация серверов (единственное отличие – это поле SERVER_ID в `/etc/one/oned.conf`);
- рекомендуется использовать подключение к базе данных одного типа (MySQL);
- серверы должны иметь беспарольный доступ для связи друг с другом;
- плавающий IP, который будет газначен лидеру;
- общая файловая система.

Добавлять дополнительные серверы или удалять старые можно после запуска кластера.

В данном примере показана настройка HA кластера из трех серверов:

- 192.168.0.186 opennebula
- 192.168.0.187 server02
- 192.168.0.188 server03
- 192.168.0.200 Floating IP

3.9.1 Первоначальная конфигурация Leader

Запустить сервис OpenNebula и добавить локальный сервер в существующую или новую зону (в примере зона с ID 0):

```
$ onezone list
C   ID NAME             ENDPOINT                                    FED_INDEX
*    0 OpenNebula       http://localhost:2633/RPC2                  -1

$ onezone server-add 0 --name opennebula --rpc http://192.168.0.186:2633/RPC2

$ onezone show 0
ZONE 0 INFORMATION
ID                : 0
NAME              : OpenNebula


ZONE SERVERS
ID NAME            ENDPOINT
 0 opennebula      http://192.168.0.186:2633/RPC2

HA & FEDERATION SYNC STATUS
ID NAME            STATE      TERM       INDEX      COMMIT     VOTE  FED_INDEX
 0 opennebula      solo       0          -1         0          -1    -1

ZONE TEMPLATE
ENDPOINT="http://localhost:2633/RPC2"
```

Остановить сервис opennebula и обновить конфигурацию SERVER_ID в файле /etc/one/oned.conf:

```
FEDERATION = [
    MODE          = "STANDALONE",
    ZONE_ID       = 0,
    SERVER_ID     = 0, # изменить с -1 на 0 (0—это ID сервера)
    MASTER_ONED   = ""
]
```

Включить Raft-обработчики, чтобы добавить плавающий IP-адрес в кластер (файл /etc/one/oned.conf):

```
RAFT_LEADER_HOOK = [
    COMMAND = "raft/vip.sh",
    ARGUMENTS = "leader enp0s3 192.168.0.200/24"
]

# Executed when a server transits from leader->follower
RAFT_FOLLOWER_HOOK = [
    COMMAND = "raft/vip.sh",
    ARGUMENTS = "follower enp0s3 192.168.0.200/24"
]
```

Запустить сервис OpenNebula и проверить зону:

```
$ onezone show 0
ZONE 0 INFORMATION
ID                : 0
NAME              : OpenNebula


ZONE SERVERS
ID NAME            ENDPOINT
 0 opennebula      http://192.168.0.186:2633/RPC2

HA & FEDERATION SYNC STATUS
ID NAME            STATE      TERM       INDEX      COMMIT     VOTE  FED_INDEX
 0 opennebula      leader     1          5          5          0     -1

ZONE TEMPLATE
ENDPOINT="http://localhost:2633/RPC
```

Сервер opennebula стал Leader-сервером, также ему присвоен плавающий адрес (Floating IP):

```
$ ip -o a sh enp0s3
```

```
2: enp0s3        inet 192.168.0.186/24 brd 192.168.0.255 scope global enp0s3\
valid_lft forever preferred_lft forever
2: enp0s3     inet 192.168.0.200/24 scope global secondary enp0s3\        valid_lft
forever preferred_lft forever
2: enp0s3    inet6 fe80::a00:27ff:fec7:38e6/64 scope link \       valid_lft forever
preferred_lft forever
```

3.9.2 Добавление дополнительных серверов

Примечание. Данная процедура удалит полностью базу на сервере и заменит её актуальной с Leader-сервера.

Примечание. Рекомендуется добавлять по одному узлу за раз, чтобы избежать конфликта в базе данных.

На Leader создать полную резервную копию актуальной базы и перенести её на другие серверы вместе с файлами из каталога `/var/lib/one/.one/`:

```
$ onedb backup -u oneadmin -d opennebula -p oneadmin
MySQL dump stored in /var/lib/one/mysql_localhost_opennebula_2021-6-23_13:43:21.sql
Use 'onedb restore' or restore the DB using the mysql command:
mysql -u user -h server -P port db_name < backup_file

$ scp /var/lib/one/mysql_localhost_opennebula_2021-6-23_13\:43\:21.sql <ip>:/tmp

$ ssh <ip> rm -rf /var/lib/one/.one
$ scp -r /var/lib/one/.one/ <ip>:/var/lib/one/
```

Остановить сервис OpenNebula на Follower-узлах и восстановить скопированную базу:

```
$ onedb restore -f -u oneadmin -p oneadmin -d opennebula
/tmp/mysql_localhost_opennebula_2021-6-23_13\:43\:21.sql
MySQL DB opennebula at localhost restored.
```

Перейти на Leader-сервер и добавить в зону новые узлы (рекомендуется добавлять серверы по-одному):

```
$ onezone server-add 0 --name server02 --rpc http://192.168.0.187:2633/RPC2
```

Проверить зону на Leader-сервере:

```
$ onezone show 0
ZONE 0 INFORMATION
ID                : 0
NAME              : OpenNebula


ZONE SERVERS
ID NAME            ENDPOINT
 0 opennebula      http://192.168.0.186:2633/RPC2
 1 server02        http://192.168.0.187:2633/RPC2
```

```
HA & FEDERATION SYNC STATUS
ID NAME            STATE      TERM       INDEX      COMMIT     VOTE  FED_INDEX
 0 opennebula      leader     4          22         22         0     -1
 1 server02        error      -          -          -          -     -

ZONE TEMPLATE
ENDPOINT="http://localhost:2633/RPC2"
```

Новый сервер находится в состоянии ошибки, так как OpenNebula на новом сервере не запущена. Следует запомнить идентификатор сервера, в данном случае он равен 1.

Переключиться на добавленный Follower-сервер и обновить конфигурацию SERVER_ID в файле `/etc/one/oned.conf` (указать в качестве SERVER_ID значение из предыдущего шага). Включить Raft-обработчики, как это было выполнено на Leader.

Запустить сервис OpenNebula на Follower-сервере и проверить на Leader-сервере состояние зоны:

```
$ onezone show 0
ZONE 0 INFORMATION
ID                : 0
NAME              : OpenNebula


ZONE SERVERS
ID NAME            ENDPOINT
 0 opennebula      http://192.168.0.186:2633/RPC2
 1 server02        http://192.168.0.187:2633/RPC2

HA & FEDERATION SYNC STATUS
ID NAME            STATE      TERM       INDEX      COMMIT     VOTE  FED_INDEX
 0 opennebula      leader     4          28         28         0     -1
 1 server02        follower   4          28         28         0     -1

ZONE TEMPLATE
ENDPOINT="http://localhost:2633/RPC2""
```

Повторить данные действия, чтобы добавить третий сервер в кластер.

Примечание. Добавлять серверы в кластер, следует только в случае нормальной работы кластера (работает Leader, а остальные находятся в состоянии Follower). Если в состоянии Error присутствует хотя бы один сервер, необходимо это исправить.

Проверка работоспособности кластера:

```
$ onezone show 0
ZONE 0 INFORMATION
```

```
ID                : 0
NAME              : OpenNebula


ZONE SERVERS
ID NAME            ENDPOINT
 0 opennebula      http://192.168.0.186:2633/RPC2
 1 server02        http://192.168.0.187:2633/RPC2
 2 server03        http://192.168.0.188:2633/RPC2

HA & FEDERATION SYNC STATUS
ID NAME            STATE      TERM       INDEX      COMMIT     VOTE  FED_INDEX
 0 opennebula      leader     4          35         35         0     -1
 1 server02        follower   4          35         35         0     -1
 2 server03        follower   4          35         35         0     -1

ZONE TEMPLATE
ENDPOINT="http://localhost:2633/RPC2"
```

Если какой-либо узел находится в состоянии ошибки, следует проверить журнал (`/var/log/one/oned.log`), как в текущем Leader (если он есть), так и в узле, который находится в состоянии Error. Все сообщения Raft будут регистрироваться в этом файле.

3.9.3 Добавление и удаление серверов

Команда добавления сервера:

```
$ onezone server-add <zoneid>
```

Параметры:

- -n, --name – имя сервера зоны;
- -r, --rpc – конечная точка RPC сервера зоны;
- -v, --verbose – подробный режим;
- --user name – имя пользователя, используемое для подключения к OpenNebula;
- --password password – пароль для аутентификации в OpenNebula;
- --endpoint endpoint – URL конечной точки интерфейса OpenNebula xmlrpc.

Команда удаления сервера:

```
$ onezone server-del <zoneid> <serverid>
```

Параметры:

- -v, --verbose – подробный режим;
- --user name – имя пользователя, используемое для подключения к OpenNebula;
- --password password – пароль для аутентификации в OpenNebula;
- --endpoint endpoint – URL конечной точки интерфейса OpenNebula xmlrpc.

3.9.4 Восстановление сервера

Если Follower -сервер в течение некоторого времени не работает, он может выпасть из окна восстановления. Чтобы восстановить этот сервер необходимо:

- Leader: создать резервную копию БД и скопировать её на отказавший сервер (повторно использовать предыдущую резервную копию нельзя).
- Follower: остановить OpenNebula.
- Follower: восстановить резервную копию БД.
- Follower: запустить OpenNebula.
- Leader: сбросить отказавший Follower, выполнив команду:

```
$ onezone server-reset <zone_id> <server_id_of_failed_follower>
```

3.9.5 Sunstone

Есть несколько способов развертывания Sunstone в среде HA. Базовым является Sunstone, работающий на каждом интерфейсном узле OpenNebula. Клиенты используют только один сервер – Leader с плавающим IP.

4 СРЕДСТВО УПРАВЛЕНИЯ ВИРТУАЛЬНЫМИ ОКРУЖЕНИЯМИ PVE

4.1 Краткое описание возможностей

Proxmox Virtual Environment (PVE) – средство управления виртуальными окружениями на базе гипервизора KVM и системы контейнерной изоляции LXC. Основными компонентами среды являются:

- физические серверы, на которых выполняются процессы гипервизора KVM, и процессы, работающие в контейнерах LXC;
- хранилища данных, в которых хранятся образы установочных дисков, образы дисков виртуальных машин KVM и файлы, доступные из контейнеров LXC;
- виртуальные сетевые коммутаторы, к которым подключаются сетевые интерфейсы виртуальных окружений.

PVE состоит из веб-интерфейса, распределенного хранилища данных конфигурации виртуальных окружений и утилит конфигурирования, работающих в командной строке. Все эти инструменты предназначены только для управления средой выполнения виртуальных окружений. За формирование среды отвечают компоненты системы, не входящие непосредственно в состав PVE. В первую очередь это сетевая и дисковая подсистемы, а также механизмы аутентификации.

4.1.1 Системные требования

Минимальные системные требования (для тестирования):

- CPU: 64bit (Intel EMT64 или AMD64), поддержка Intel VT/AMD-V CPU/Mainboard;
- минимум 1 Гб ОЗУ;
- жесткий диск;
- сетевая карта.

Рекомендуемые системные требования:

- CPU: мультипроцессорный 64bit (Intel EMT64 или AMD64), поддержка Intel VT/AMD-V CPU/Mainboard;
- минимум 2 Гб ОЗУ для ОС и сервисов PVE. Плюс выделенная память для гостевых систем. Для Ceph или ZFS требуется дополнительная память, примерно 1 ГБ ОЗУ на каждый ТБ используемого хранилища;
- хранилище для ОС: аппаратный RAID;
- хранилище для ВМ: аппаратный RAID для локального хранилища, или non-RAID для ZFS. Возможно также совместное и распределенное хранение;
- быстрые жёсткие диски 15krpm SAS, Raid10;

- сетевая карта.

Реальные системные требования определяются количеством и требованиями гостевых систем.

4.1.2 Веб-интерфейс

Веб-интерфейс PVE предназначен для решения следующих задач:

- создание, удаление, настройка виртуальных окружений;
- управление физическими серверами;
- мониторинг активности виртуальных окружений и использования ресурсов среды;
- фиксация состояний (snapshot-ы), создание резервных копий и шаблонов виртуальных окружений, восстановление виртуальных окружений из резервных копий.

Кроме решения пользовательских задач, веб-интерфейс PVE можно использовать еще и для встраивания в интегрированные системы управления – например, в панели управления хостингом. Для этого он имеет развитый RESTful API с JSON в качестве основного формата данных.

Для аутентификации пользователей веб-интерфейса можно использовать как собственные механизмы PVE, так и PAM. Использование PAM дает возможность, например, интегрировать PVE в домен аутентификации.

Так как используется кластерная файловая система (pmxcfs), можно подключиться к любому узлу для управления всем кластером. Каждый узел может управлять всем кластером.

Веб-интерфейс PVE доступен по адресу https://<имя-компьютера>:8006. Потребуется пройти аутентификацию (логин по умолчанию: root, пароль указывается в процессе установки ОС) (Рис. 66).

Пользовательский интерфейс PVE (Рис. 67) состоит из четырех областей:

- заголовок – верхняя часть. Показывает информацию о состоянии и содержит кнопки для наиболее важных действий;
- дерево ресурсов – левая сторона. Дерево навигации, где можно выбирать конкретные объекты;
- панель контента – центральная часть. Здесь отображаются конфигурация и статус выбранных объектов;
- панель журнала – нижняя часть. Отображает записи журнала для последних задач. Чтобы получить более подробную информацию или прервать выполнение задачи, следует дважды щелкнуть левой клавишей мыши по записи журнала.

Аутентификация в веб-интерфейсе PVE

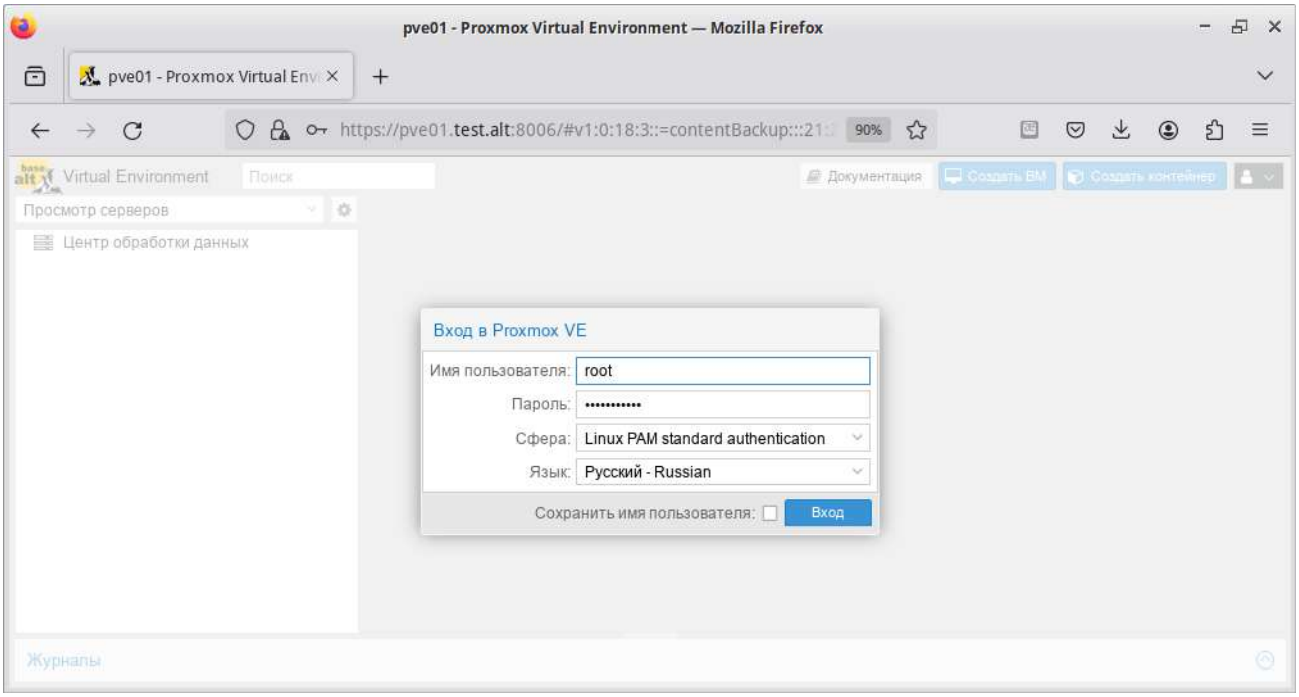


Рис. 66

Веб-интерфейс PVE

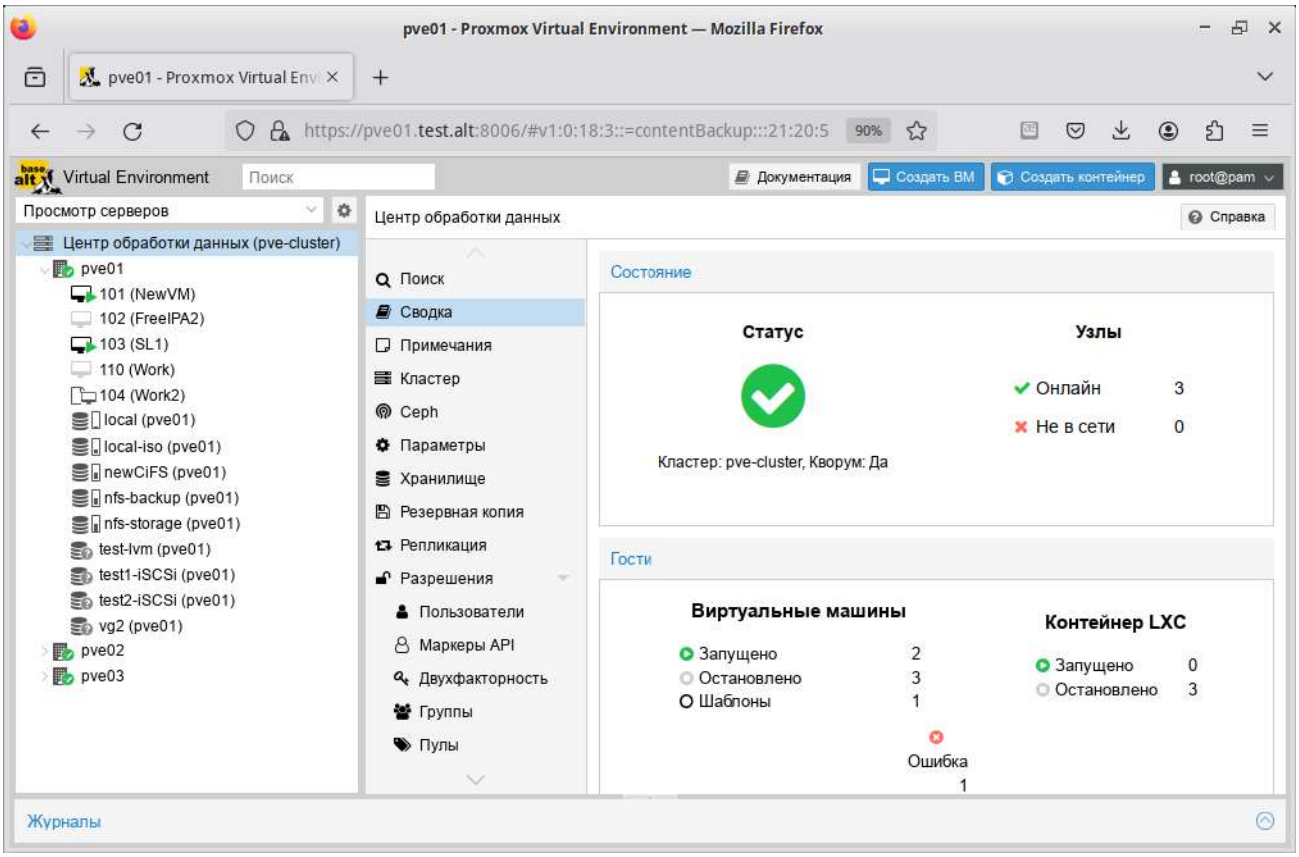


Рис. 67

Примечание. Есть возможность работы с PVE из мобильного приложения, например, Proxmox. В приложении можно получить доступ к узлам (Рис. 68), ВМ и контейнерам. Можно

зайти в консоль ВМ с помощью noVNC или SPICE, осуществлять необходимые манипуляции внутри ВМ (Рис. 69).

Работа с PVE из мобильного приложения

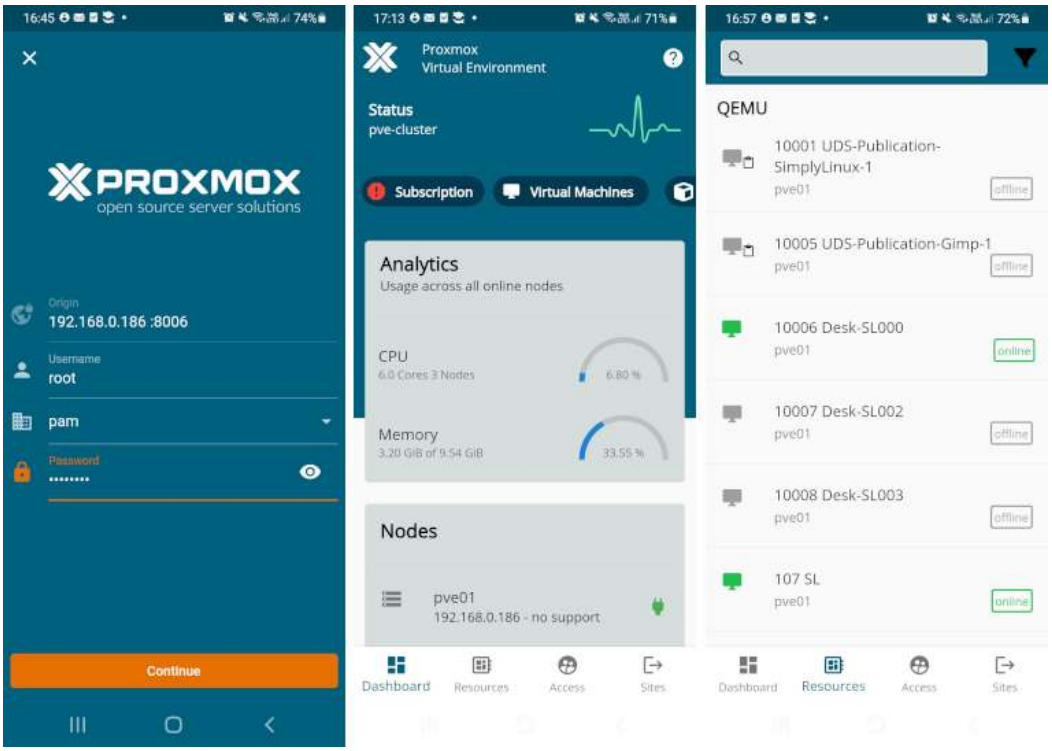


Рис. 68

Работа с ВМ из мобильного приложения

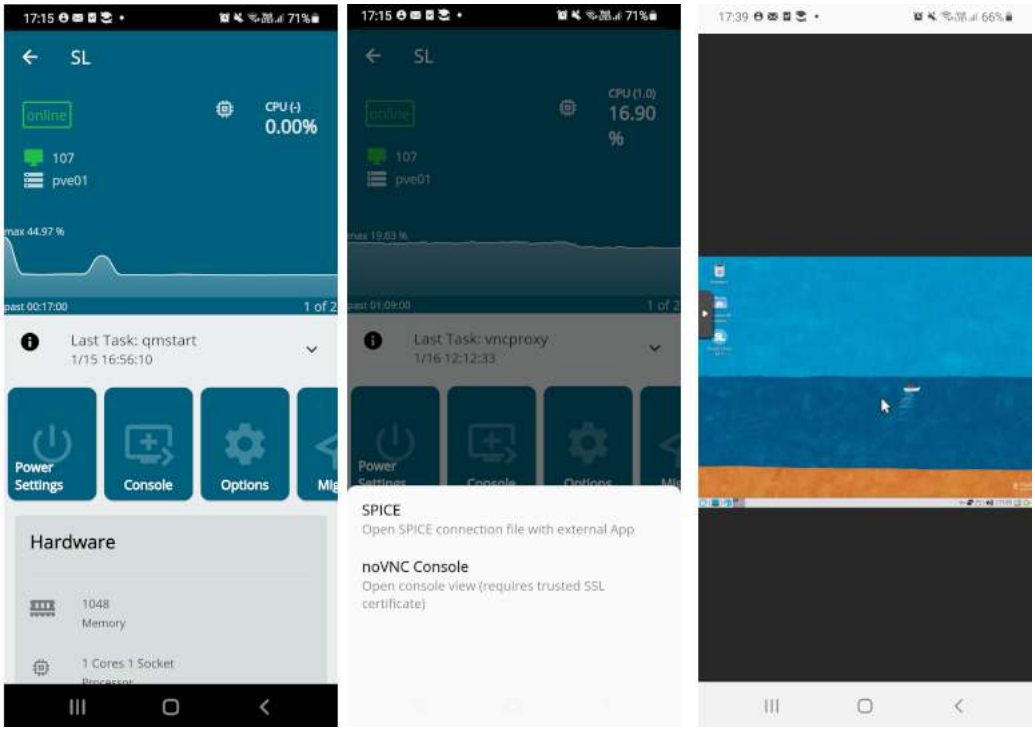


Рис. 69

4.1.3 Хранилище данных

В случае локальной установки PVE для размещения данных виртуальных окружений можно дополнительно ничего не настраивать и использовать локальную файловую систему сервера. Но в случае кластера из нескольких серверов имеет смысл настроить сетевую или распределенную сетевую файловую систему, обеспечивающую параллельный доступ к файлам со всех серверов, на которых выполняются процессы виртуальных окружений. В качестве таких файловых систем могут выступать, например, NFS или CEPH.

4.1.4 Сетевая подсистема

В отличие от хранилища данных, сетевая подсистема требует специальной настройки даже в случае локальной установки PVE. Это обусловлено тем, что сетевые интерфейсы виртуальных окружений должны подключаться к какому-то виртуальному устройству, обеспечивающему соединение с общей сетью передачи данных. Перед началом настройки сети следует принять решение о том, какой тип виртуализации (LXC/KVM) и какой тип подключения будет использоваться (маршрутизация/мост).

4.2 Установка и настройка PVE

Примечание. Компоненты PVE будут установлены в систему, если при установке дистрибутива выбрать профиль «Виртуальное окружение Proxmox». При установке дистрибутива также необходимо настроить Ethernet-мост vmbr0 и при заполнении поля с именем компьютера указать полное имя с доменом.

Все остальные настройки можно делать в веб-интерфейсе см. «Создание кластера PVE».

4.2.1 Настройка сетевой подсистемы

Для узла должно быть установлено полное имя с доменом (FQDN).

На всех узлах кластера необходимо настроить Ethernet-мост.

Примечание. Мосту должно быть назначено имя vmbr0 и оно должно быть одинаково на всех узлах.

Примечание. При использовании дистрибутива «Альт Сервер Виртуализации» интерфейс vmbr0 создаётся и настраивается в процессе установки системы.

4.2.1.1 Настройка Ethernet-моста в командной строке

Если интерфейсы, входящие в состав моста, являются единственными физически подключенными и настройка моста происходит с удаленного узла через эти интерфейсы, то требуется соблюдать осторожность, т.к. эти интерфейсы могут перестать быть доступны. В случае ошибки в конфигурации потребуется физический доступ к серверу. Для страховки, перед перезапуском сервиса network можно открыть еще одну консоль и запустить там, например, команду: `sleep 500 && reboot`.

Примечание. Далее предполагается, что для интерфейс, который будет входить в мост, сконфигурирован и имеет статический IP-адрес.

Для настройки Ethernet-моста с именем vmbr0, следует выполнить следующие команды:

```
# mkdir /etc/net/ifaces/vmbr0
# cp /etc/net/ifaces/enp0s3/* /etc/net/ifaces/vmbr0/
# rm -f /etc/net/ifaces/enp0s3/{i,r}*
# cat <<EOF > /etc/net/ifaces/vmbr0/options
BOOTPROTO=static
CONFIG_WIRELESS=no
CONFIG_IPV4=yes
HOST='enp0s3'
ONBOOT=yes
TYPE=bri
EOF
```

Имя интерфейса, обозначенного здесь как `enp0s3`, следует указать в соответствии с реальной конфигурацией сервера.

IP-адрес для интерфейса будет взят из `ipv4address`.

В опции HOST файла `options` нужно указать те интерфейсы, которые будут входить в мост. Если в него будут входить интерфейсы, которые до этого имели IP-адрес (например, `enp0s3`), то этот адрес должен быть удален (например, можно закомментировать содержимое файла `/etc/net/ifaces/enp0s3/ipv4address`).

Для того чтобы изменения вступили в силу, необходим перезапуск сервиса `network`:

```
# systemctl restart network
```

При старте сети сначала поднимаются интерфейсы, входящие в мост, затем сам мост (автоматически).

Установить имя узла, выполнив команду:

```
# hostnamectl set-hostname <имя узла>
```

Например:

```
# hostnamectl set-hostname pve01.test.alt
```

4.2.1.2 Настройка Ethernet-моста в веб-интерфейсе

При установленных пакетах alterator-net-eth и alterator-net-bridge, для настройки Ethernet-моста можно воспользоваться веб-интерфейсом центра управления системой (ЦУС).

Примечание. Должен также быть установлен пакет alterator-fbi и запущены сервисы ahttpd и alteratord:

```
# apt-get install alterator-fbi
# systemctl start ahttpd
# systemctl start alteratord
```

Веб-интерфейс ЦУС доступен по адресу https://ip-address:8080.

Для настройки Ethernet-моста необходимо выполнить следующие действия:

1) в группе «Сеть» выбрать пункт «Ethernet-интерфейсы»;

2) удалить IP-адрес и шлюз по умолчанию у интерфейса, который будет включен в сетевой мост, и нажать кнопку «Создать сетевой мост...» (Рис. 70);

Настройка сети в веб-интерфейсе

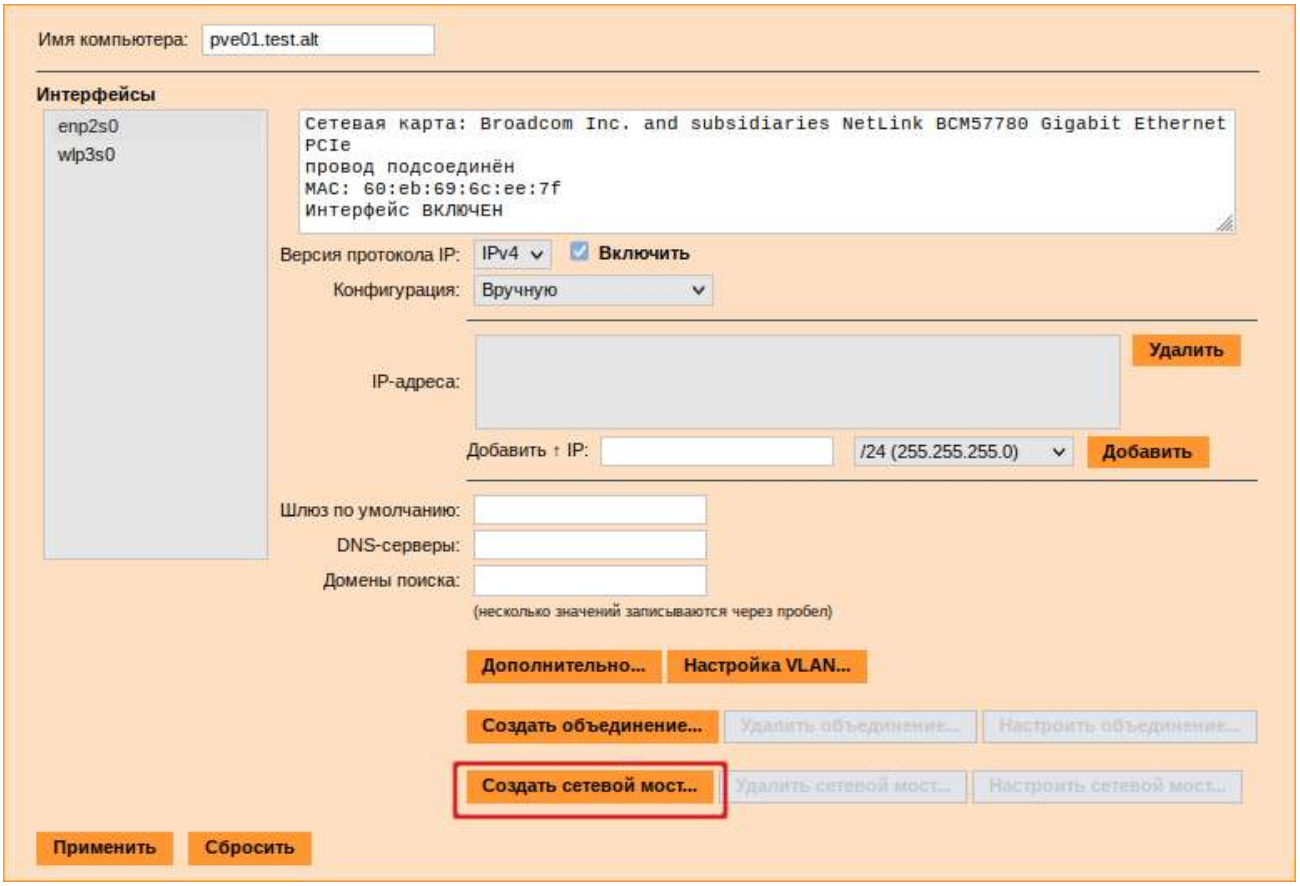


Рис. 70

3) в открывшемся окне (Рис. 71), задать имя моста vmbr0, выбрать сетевой интерфейс в списке «Доступные интерфейсы», переместить его в список «Члены» и нажать кнопку «Ок»;

4) настроить сетевой интерфейс vmbr0: задать IP-адрес и нажать кнопку «Добавить», ввести адрес шлюза по умолчанию и DNS-сервера (Рис. 72);

5) в поле «Имя компьютера» ввести имя компьютера (следует указать полное имя с доменом);

6) нажать кнопку «Применить».

Выбор сетевого интерфейса

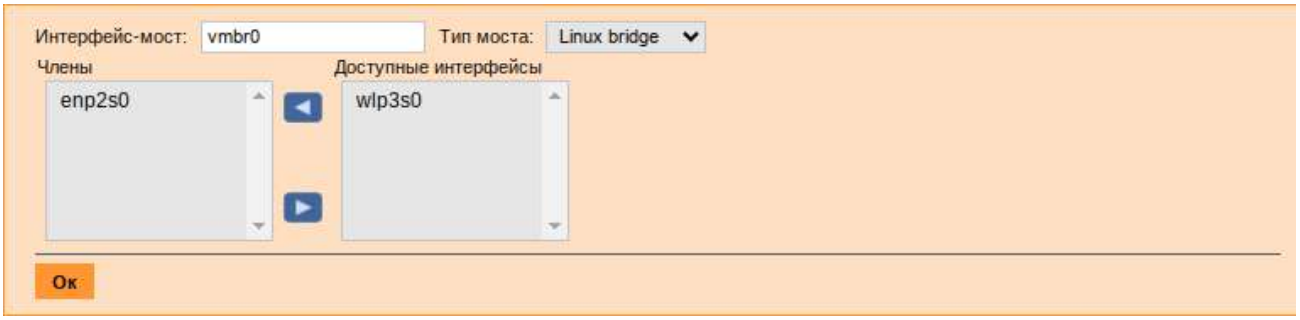


Рис. 71

Настройка параметров сетевого интерфейса vmbr0

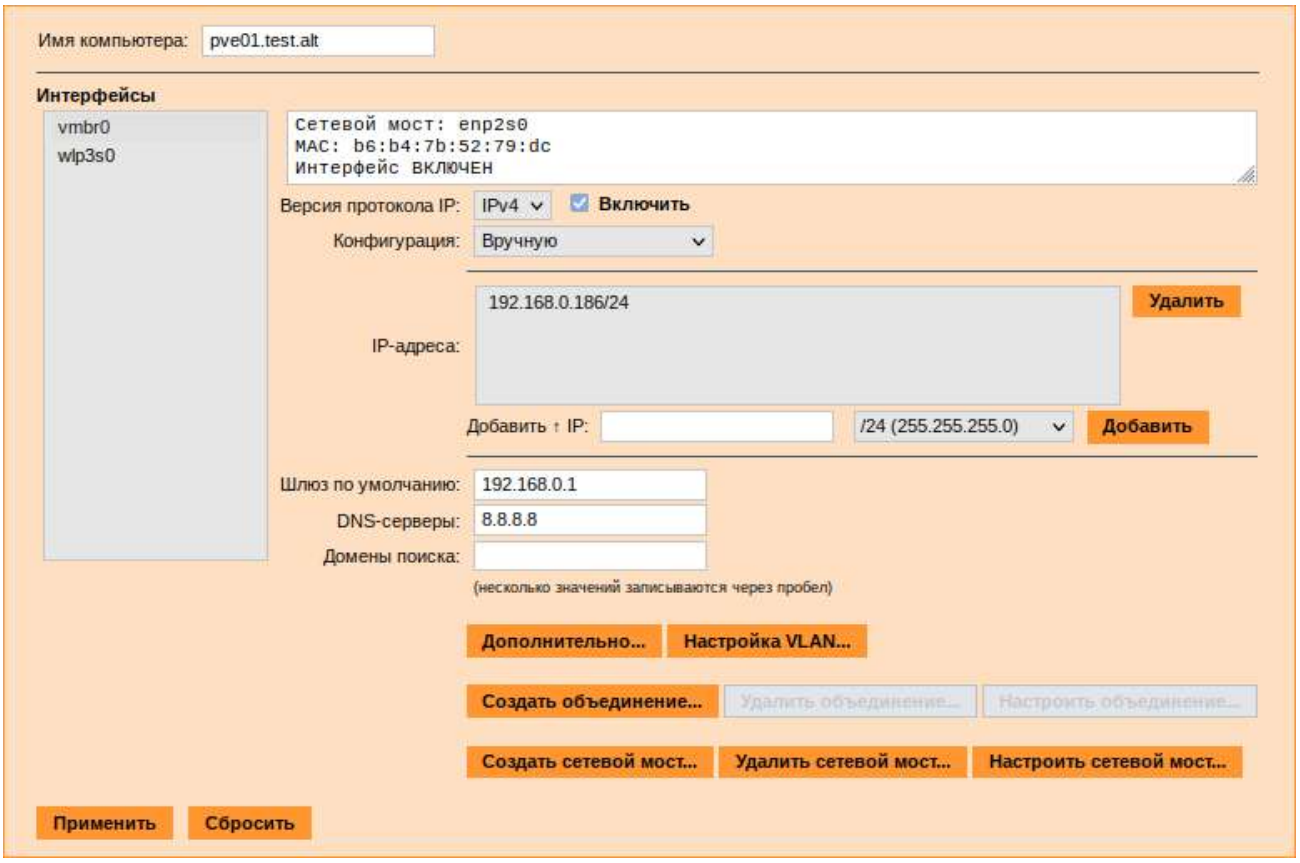


Рис. 72

4.2.2 Установка PVE

Установить пакет `pve-manager` (все необходимые компоненты при этом будут установлены автоматически):

```
# apt-get install pve-manager
```

Следует также убедиться в том, что пакет systemd обновлен до версии, находящейся в репозитории P10.

Добавить информацию об имени узла в файл /etc/hosts:

```
# echo "192.168.0.186 pve01.test.alt pve01" >> /etc/hosts
```

Запустить и добавить в автозагрузку службу pve-cluster:

```
# systemctl enable --now pve-cluster
```

Далее, запустить остальные службы и добавить их в список служб, запускаемых при старте узла:

```
# systemctl start lxc lxc-net lxc-monitord pve-lxc-syscalld pvedaemon pve-firewall pvestatd pve-ha-lrm pve-ha-crm spiceproxy pveproxy qmeventd
# systemctl enable corosync lxc lxc-net lxc-monitord pve-lxcsyscalld pve-cluster pvedaemon pve-firewall pvestatd pve-ha-lrm pveha-crm spiceproxy pveproxy pve-guests qmeventd
```

4.3 Создание кластера PVE

Рекомендации:

- для работы corosync все узлы должны иметь возможность обмениваться трафиком через UDP-порты 5404-5412;
- дата и время на всех узлах должны быть синхронизированы;
- между узлами используется SSH-туннель через TCP-порт 22;
- для работы механизма высокой доступности (HA) и формирования кворума необходимо как минимум три узла. На всех узлах должна быть установлена одна версия PVE;
- рекомендуется использовать выделенный сетевой адаптер для трафика кластера, особенно если используется общее хранилище.

Примечание. Для небольших кластеров из двух узлов можно использовать QDevice для обеспечения третьего голоса.

Кластер не создается автоматически на только что установленном узле PVE. В настоящее время создание кластера может быть выполнено либо в консоли (вход через ssh), либо в веб-интерфейсе.

Примечание. PVE при создании кластера включает парольную аутентификацию для root в sshd. В целях повышения безопасности, после включения всех серверов в кластер, рекомендуется отключить парольную аутентификацию для root в sshd:

```
# control sshd-permit-root-login without_password
# systemctl restart sshd
```

При добавлении в кластер нового сервера, можно временно включить парольную аутентификацию:

```
# control sshd-permit-root-login enabled
# systemctl restart sshd
```

А после того как сервер будет добавлен, снова отключить.

Кластеры используют ряд определенных портов (Табл. Таблица 9) для различных функций. Важно обеспечить доступность этих портов и отсутствие их блокировки межсетевыми экранами.

Т а б л и ц а 9 – Используемые порты

Порт	Функция
TCP 8006	Веб-интерфейс PVE
TCP 5900-5999	Доступ к консоли VNC
TCP 3128	Доступ к консоли SPICE
TCP 22	SSH доступ
UDP 5404, 5405	Широковещательный CMAN для применения настроек кластера

4.3.1 Настройка узлов кластера

PVE должен быть установлен на всех узлах. Следует убедиться, что каждый узел установлен с окончательным именем хоста и IP-конфигурацией. Изменение имени хоста и IP-адреса невозможно после создания кластера.

Необходимо обеспечить взаимно однозначное прямое и обратное преобразование имен для всех узлов кластера. Крайне желательно использовать DNS, в крайнем случае, можно обойтись соответствующими записями в локальных файлах `/etc/hosts`:

```
# echo "192.168.0.186 pve01.test.alt pve01" >> /etc/hosts
# echo "192.168.0.90 pve02.test.alt pve02" >> /etc/hosts
# echo "192.168.0.70 pve03.test.alt pve03" >> /etc/hosts
```

П р и м е ч а н и е . В PVE это можно сделать в панели управления: выбрать узел, перейти в «Система» → «Хосты», добавить все узлы, которые будут включены в состав кластера (Рис. 73).

Редактирование записей в файле /etc/hosts

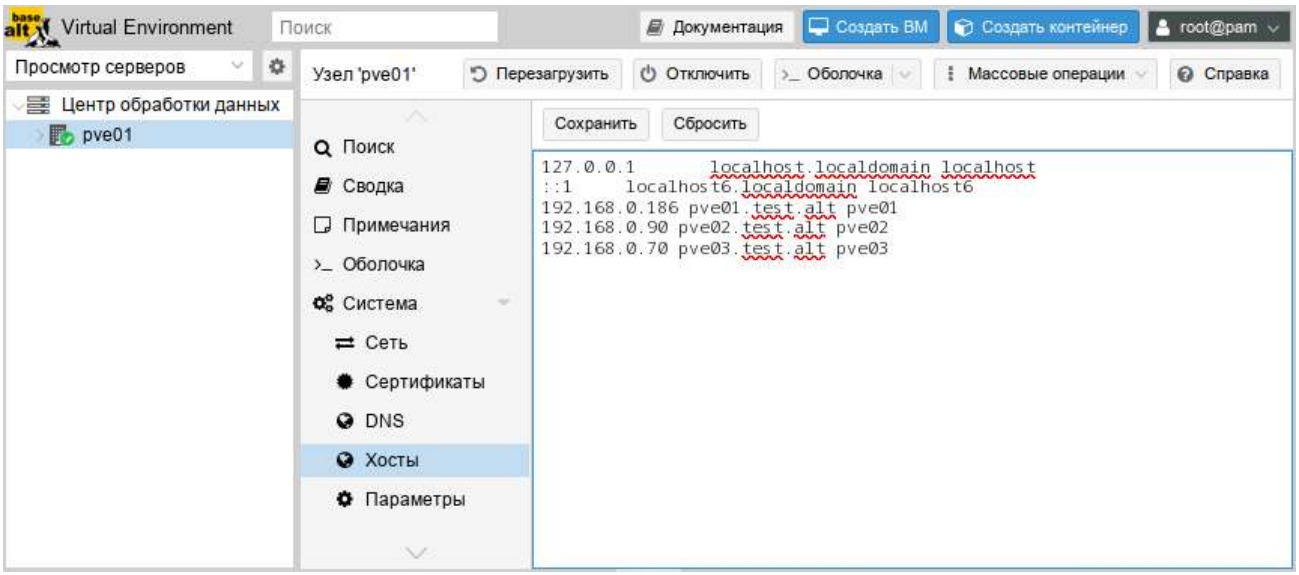


Рис. 73

П р и м е ч а н и е . Имя машины не должно присутствовать в файле `/etc/hosts` разрешающимся в 127.0.0.1.

4.3.2 Создание кластера в веб-интерфейсе

Для создания кластера необходимо выполнить следующие действия:

1) в панели управления любого узла кластера выбрать «Центр обработки данных» → «Кластер» и нажать кнопку «Создать кластер» (Рис. 74);

2) в открывшемся окне задать название кластера, выбрать IP-адрес интерфейса, на котором узел кластера будет работать, и нажать кнопку «Создать» (Рис. 75);

3) при успешном создании кластера будет выведена соответствующая информация (Рис. 76).

Создание кластера в веб-интерфейсе

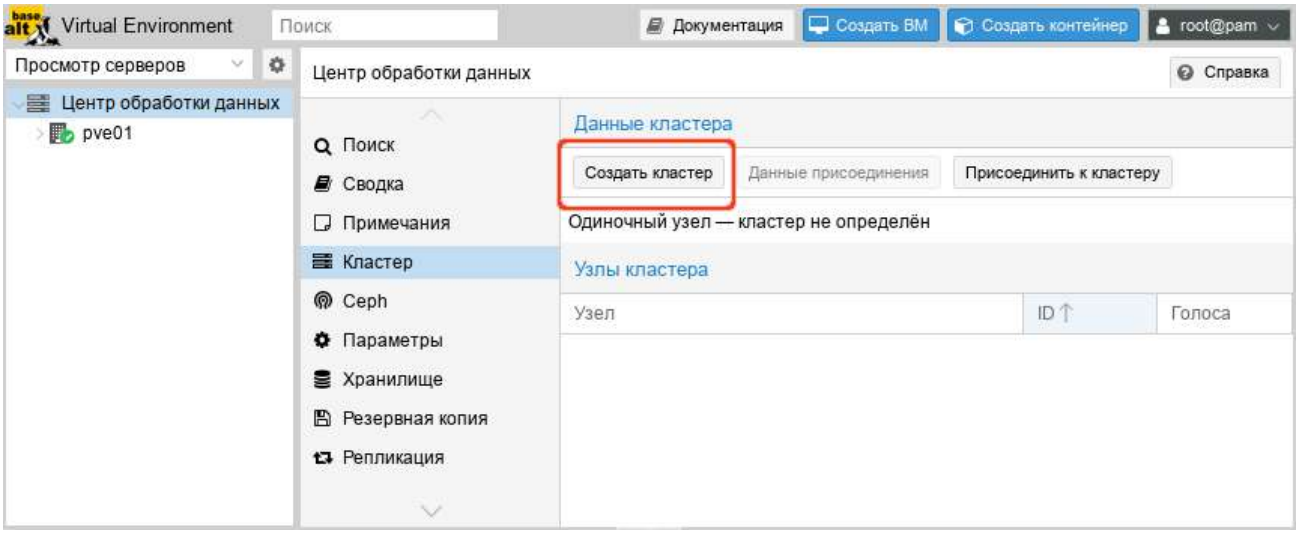


Рис. 74

Создание кластера в веб-интерфейсе. Название кластера

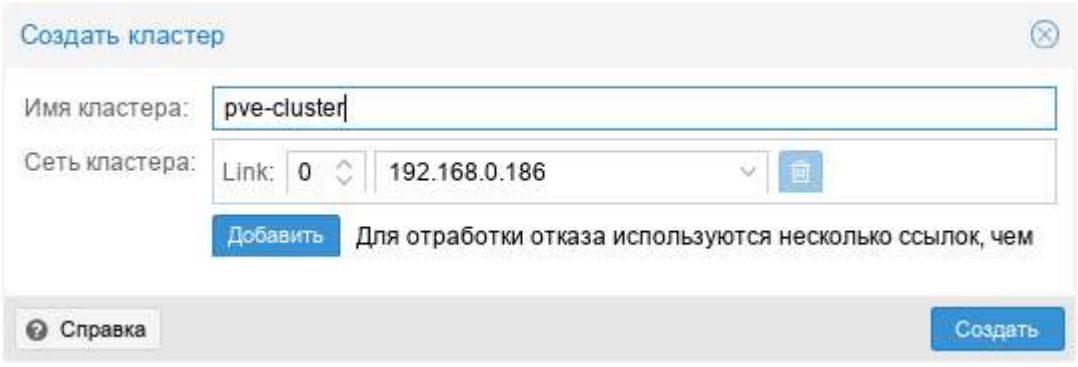


Рис. 75

Информация о создании кластера

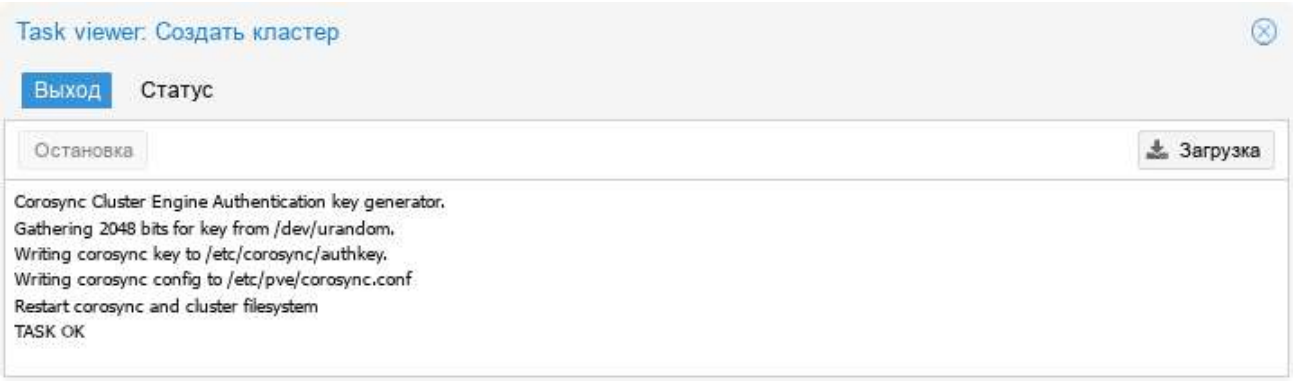


Рис. 76

Для добавления узла в кластер необходимо выполнить следующие действия:

1) в панели управления узла, на котором был создан кластер, выбрать «Центр обработки данных» → «Кластер» и нажать кнопку «Данные присоединения» (Рис. 77);

2) в открывшемся окне, нажав кнопку «Копировать данные» (Рис. 78), скопировать данные присоединения;

3) перейти в панель управления узла, который следует присоединить к кластеру. Выбрать пункт «Центр обработки данных» → «Кластер» и нажать кнопку «Присоединить к кластеру» (Рис. 79);

4) в поле «Данные» вставить данные присоединения, поля «Адрес сервера» и «Отпечаток» при этом будут заполнены автоматически. В поле «Пароль» ввести пароль пользователя root первого узла (Рис. 80) и нажать кнопку «Присоединить <имя кластера>»;

5) через несколько минут, после завершения репликации всех настроек, узел будет подключен к кластеру (Рис. 81).

Создание кластера в веб-интерфейсе. Получить данные присоединения

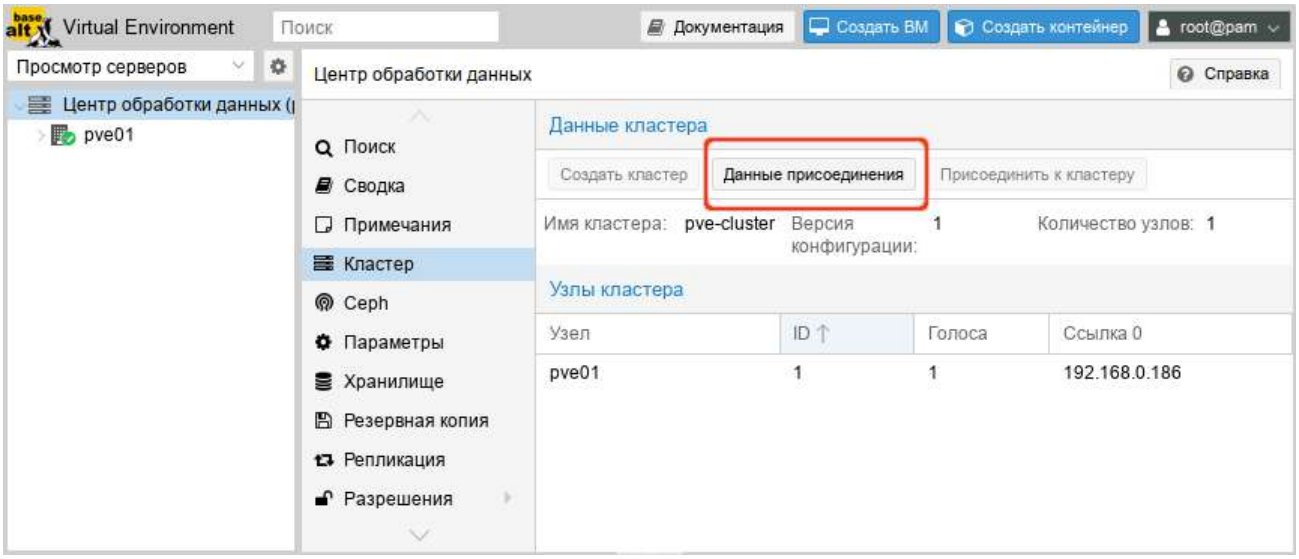


Рис. 77

Создание кластера в веб-интерфейсе. Данные присоединения

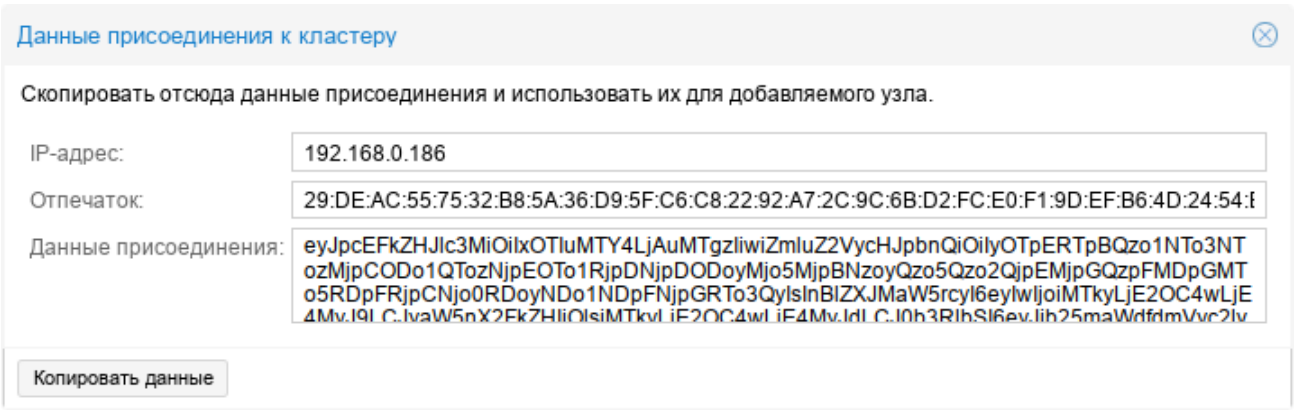


Рис. 78

Узел, который следует присоединить к кластеру

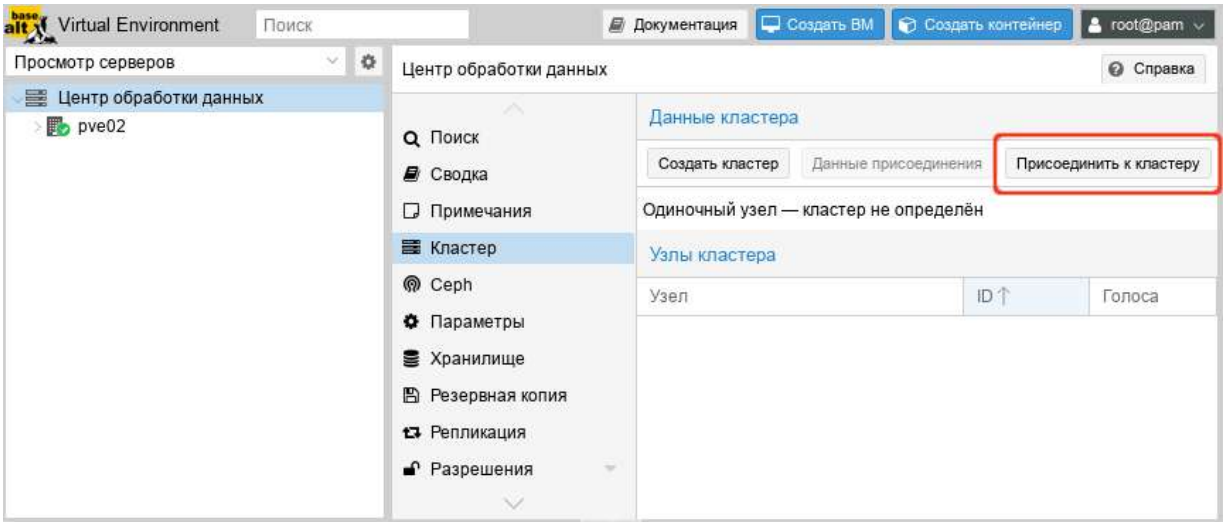


Рис. 79

Присоединение узла к кластеру

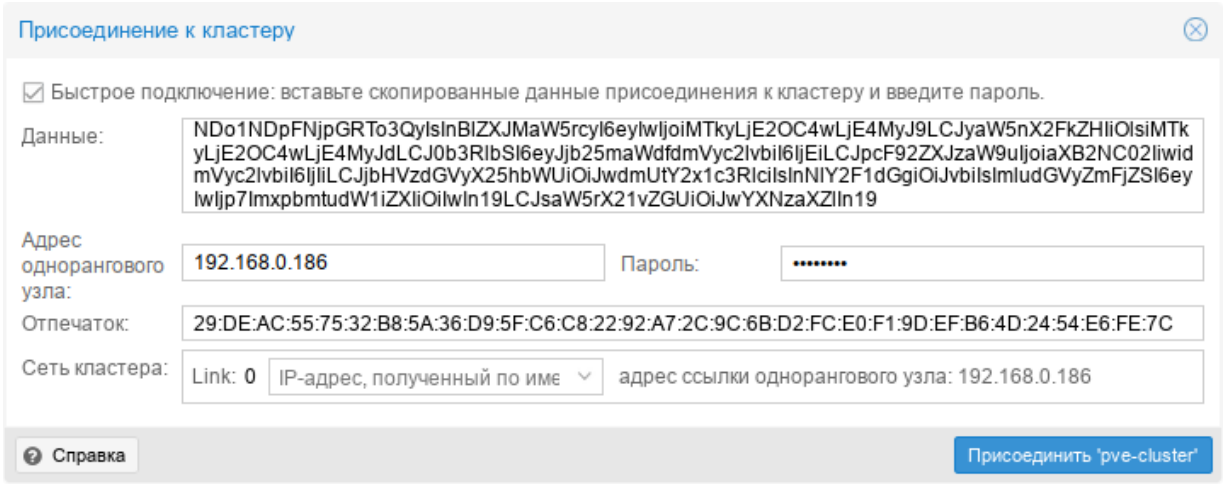


Рис. 80

Узлы кластера в веб-интерфейсе

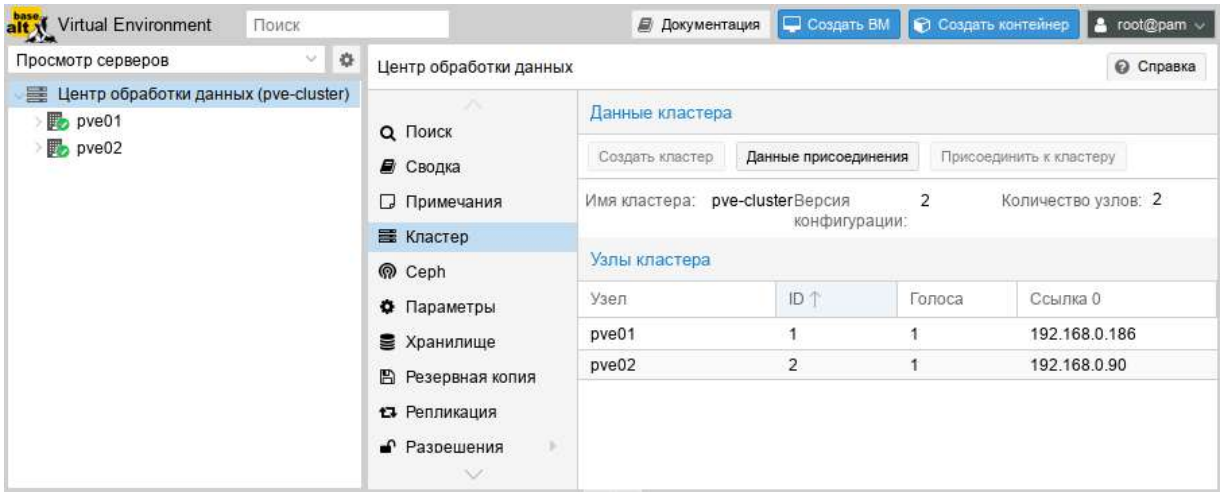


Рис. 81

Сразу после инициализации кластера в пределах каждого из узлов доступно одно локальное хранилище данных (Рис. 82).

Узлы кластера и локальные хранилища данных

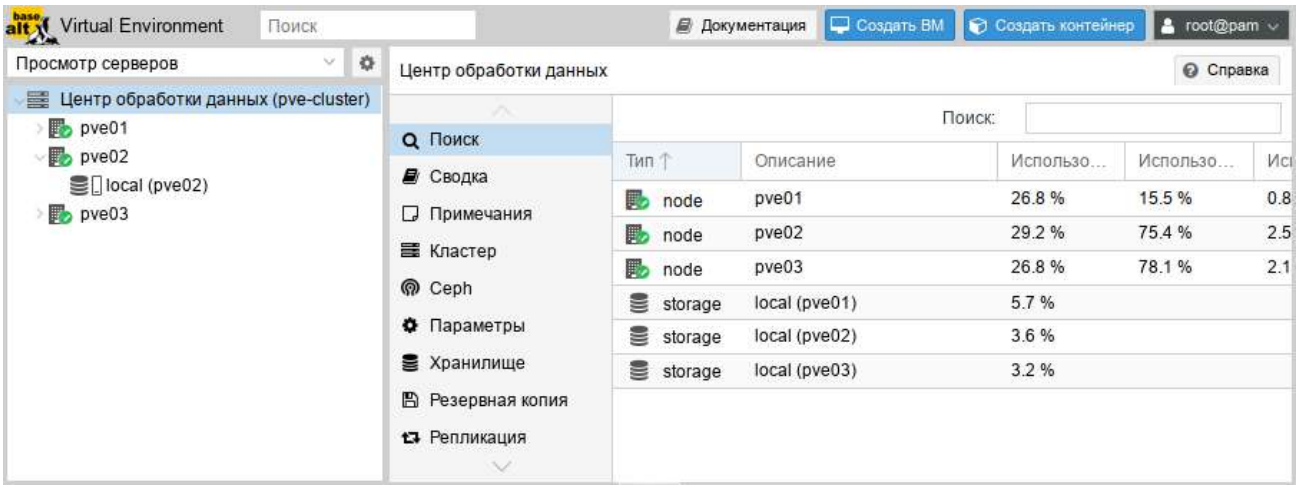


Рис. 82

4.3.3 Создание кластера в консоли

Команда создания кластера:

```
# pvecm create <cluster_name>
```

На «головном» узле кластера необходимо выполнить команду инициализации конкретного кластера PVE, в данном примере – «pve-cluster»:

```
# systemctl start pve-cluster
# pvecm create pve-cluster
```

Проверка:

```
# pvecm status
Cluster information
-------------------
```

```
Name:             pve-cluster
Config Version:   1
Transport:        knet
Secure auth:      on

Quorum information
------------------
Date:             Mon Apr  1 10:32:25 2024
Quorum provider:  corosync_votequorum
Nodes:            1
Node ID:          0x00000001
Ring ID:          1.5
Quorate:          Yes
Votequorum information
----------------------
Expected votes:   1
Highest expected: 1
Total votes:      1
Quorum:           1
Flags:            Quorate

Membership information
----------------------
    Nodeid      Votes Name
0x00000001          1 192.168.0.186 (local)
```

Команда создания кластера создает файл настройки `/etc/pve/corosync.conf`. По мере добавления узлов в кластер файл настройки будет автоматически пополняться информацией об узлах.

Команда для добавления узла в кластер:

```
# pvecm add <existing_node_in_cluster>
```

где `existing_node_in_cluster` – адрес уже добавленного узла (рекомендуется указывать самый первый).

Для добавления узлов в кластер, необходимо на «подчиненных» узлах выполнить команде:

```
# pvecm add pve01
```

где `pve01` – имя или IP-адрес «головного» узла.

При добавлении узла в кластер потребуется ввести пароль администратора главного узла кластера:

```
# pvecm add pve01
Please enter superuser (root) password for 'pve01': ***
Establishing API connection with host 'pve01'
```

```
Login succeeded.
Request addition of this node
Join request OK, finishing setup locally
stopping pve-cluster service
backup old database to '/var/lib/pve-cluster/backup/config-1625747072.sql.gz'
waiting for quorum...OK
(re)generate node files
generate new node certificate
merge authorized SSH keys and known hosts
generated new node certificate, restart pveproxy and pvedaemon services
successfully added node 'pve03' to cluster.
```

После добавления узлов в кластер, файл настройки кластера в `/etc/pve/corosync.-conf` должен содержать информацию об узлах кластера.

На всех узлах кластера должны быть запущены и добавлены в список служб, запускаемых при старте узла, службы:

```
# systemctl start pve-cluster pveproxy pvedaemon pvestatd pve-firewall pvefw-logger pve-ha-lrm pve-ha-crm spiceproxy lxc lxcfs lxc-net lxc-monitord qmeventd pvescheduler pve-lxc-syscalld
# systemctl enable pve-cluster pveproxy pvedaemon pvestatd pve-firewall pvefw-logger pve-guests pve-ha-crm pve-ha-lrm spiceproxy lxc lxcfs lxc-net lxc-monitord qmeventd pvescheduler corosync pve-lxc-syscalld
```

4.3.4 Удаление узла из кластера

Перед удалением узла из кластера необходимо переместить все ВМ с этого узла, а также убедиться, что нет никаких локальных данных или резервных копий, которые необходимо сохранить.

Для удаления узла из кластера необходимо выполнить следующие шаги:

1) войти в узел кластера, не подлежащий удалению (в примере pve01);
2) ввести команду `pvecm nodes`, чтобы определить идентификатор узла, который следует удалить:

```
# pvecm nodes
Membership information
----------------------
    Nodeid      Votes Name
         1          1 pve01 (local)
         2          1 pve02
         3          1 pve03
```

3) выключить подлежащий удалению узел (в данном случае pve02);
4) удалить узел из кластера, выполнив команду:

```
# pvecm delnode pve02
```

5) убедиться, что узел удален (в разделе «Membership information» команда отобразит список узлов кластера без удаленного узла):

```
# pvecm status
…
Membership information
----------------------
    Nodeid      Votes Name
0x00000001      1 192.168.0.186 (local)
0x00000003      1 192.168.0.70
```

Примечание. Крайне важно отключить узел перед удалением и убедиться, что он больше никогда не будет включен снова (в существующей сети кластера) с текущей конфигурацией. В противном случае кластер может выйти из строя, и восстановить его до рабочего состояния может быть сложно.

После удаления узла из кластера его каталог конфигурации /etc/pve/nodes/<имя_узла> не удаляется автоматически. Он может содержать конфигурации ВМ, хранилищ и другие данные, которые при необходимости можно сохранить или перенести.

После проверки и сохранения нужной информации данный каталог рекомендуется удалить вручную, чтобы избежать путаницы и возможных конфликтов при повторном добавлении узла с тем же именем.

Примечание. Если кворум в кластере нарушен, для удаления недоступного узла можно использовать следующую последовательность команд:

```
# pvecm expected 1
# pvecm delnode <недоступный_узел>
```

Команда `pvecm expected 1` применяется в ситуациях, когда:

- кворум утрачен;
- восстановление одного или нескольких узлов невозможно или не планируется;
- требуется удалить недоступный узел из конфигурации кластера.

Следует учитывать, что использование `pvecm expected 1` является аварийной мерой управления кластером, а не штатной операцией, и должно применяться только при осознанном принятии риска потери отказоустойчивости.

Если необходимо вернуть удаленный узел обратно в кластер, следует выполнить следующие действия:

- переустановить PVE на этом узле (это гарантирует, что все секретные ключи кластера/ssh и любые данные конфигурации будут уничтожены);
- присоединиться к кластеру.

4.3.5 Кворум

Для обеспечения согласованного состояния среди всех узлов кластера PVE использует метод на основе кворума. Кворум – это минимальное количество голосов, которые должны быть доступны для работы кластера.

Проверить наличие кворума можно, выполнив команду:

```
# pvecm status
…
Votequorum information
----------------------
Expected votes:   5
Highest expected: 5
Total votes:      5
Quorum:           3
Flags:            Quorate
…
```

В выводе команды видно, что в кластере пять узлов (`Expected votes`), из них для кворума необходимо не менее трех (`Quorum`), сейчас все пять узлов активны (`Total votes`), кворум соблюден (`Flags: Quorate`).

Если количество голосов окажется меньше, чем требуется для кворума, кластер переключится в режим только для чтения (read-only): ВМ продолжат работать, но любые операции с узлами или ВМ станут невозможными.

В PVE по умолчанию каждому узлу назначается один голос.

4.3.6 Поддержка внешнего арбитра corosync

Добавив в кластер PVE внешний арбитр, можно добиться того, что кластер сможет выдержать большее количество отказов узлов без нарушения работы кластера.

Для работы арбитра задействованы две службы:

- Corosync Quroum (QDevice) – служба, которая работает на каждом узле кластера PVE. Она предоставляет настроенное количество голосов подсистеме кворума на основе решения внешнего управляющего арбитра. Основное назначение этой службы – позволить кластеру выдержать большее количество отказов узлов, чем это позволяют стандартные правила кворума. Арбитр видит все узлы и может выбрать только один набор узлов, чтобы отдать свой голос (это будет сделано только в том случае, если указанный набор узлов при получении голоса арбитра сможет иметь кворум);
- внешний арбитр, который работает на независимом сервере. Единственное требование к арбитру – наличие сетевого доступа к кластеру.

В настоящее время Qdevices для кластеров с нечетным числом узлов не поддерживается. Это связано с разницей в количестве голосов, которые QDevice предоставляет для кластера.

Кластеры с чётным числом узлов получают один дополнительный голос, что только увеличивает доступность, поскольку сбой самого QDevice не влияет на работоспособность кластера.

Для кластера с нечётным числом узлов QDevice предоставляет (N-1) голосов, где N – количество узлов кластера. Этот алгоритм допускает сбой всех узлов, кроме одного и самого QDevice. При этом есть два недостатка:

- если произойдет сбой арбитра, ни один другой узел не может выйти из строя, или кластер немедленно потеряет кворум. Например, в кластере с 15 узлами 7 могут выйти из строя, прежде чем кластер станет неработоспособным. Но если в этом кластере настроен QDevice и он сам выйдет из строя, ни один узел из 15 не может выйти из строя. В этом случае QDevice действует почти как единая точка отказа;
- возможность выхода из строя всех узлов, кроме одного, плюс QDevice, может привести к массовому восстановлению служб высокой доступности (HA), что может привести к перегрузке единственного оставшегося узла. Кроме того, сервер Ceph прекратит предоставлять услуги, если в сети останется только ((N-1)/2) узлов или меньше.

Примечание. При настройке QDevice PVE копирует ключи кластера на внешний сервер. При добавлении QDevice можно временно включить парольную аутентификацию для root в sshd на внешнем сервере:

```
# control sshd-permit-root-login enabled
# systemctl restart sshd
```

А после того, как QDevice будет добавлен, отключить:

```
# control sshd-permit-root-login without_password
# systemctl restart sshd
```

Примечание. Пакеты corosync-qnetd, corosync-qdevice не входит в состав ISO-образа дистрибутива, их можно установить из репозитория p10. О добавлении репозиториев можно почитать в разделе «Добавление репозиториев».

Для настройки работы арбитра необходимо выполнить следующие действия:

1) на внешнем сервере:

- установить пакет corosync-qnetd:

```
# apt-get install corosync-qnetd
```

- запустить и добавить в автозагрузку службу corosync-qnetd:

```
# systemctl enable --now corosync-qnetd
```

2) на всех узлах PVE установить пакет `corosync-qdevice`:

```
# apt-get install corosync-qdevice
```

3) на одном из узлов PVE настроить QDevice, выполнив команду:

```
# pvecm qdevice setup 192.168.0.88
```

где 192.168.0.88 – IP-адрес арбитра (внешнего сервера).

SSH-ключи из кластера будут автоматически скопированы в QDevice.

4) на любом узле PVE проверить статус кластера:

```
# pvecm status
…
Votequorum information
----------------------
Expected votes:   5
Highest expected: 5
Total votes:      5
Quorum:           3
Flags:            Quorate Qdevice

Membership information
----------------------
    Nodeid      Votes    Qdevice Name
0x00000001          1    A,V,NMW 192.168.0.186 (local)
0x00000002          1    A,V,NMW 192.168.0.90
0x00000003          1    A,V,NMW 192.168.0.70
0x00000004          1    A,V,NMW 192.168.0.91
0x00000000          1            Qdevice
```

Для добавления нового узла или удаления существующего из кластера с настроенным QDevice, сначала необходимо удалить QDevice. После можно добавлять или удалять узлы в обычном режиме.

Команда удаления QDevice:

```
# pvecm qdevice remove
```

4.3.7 Кластерная файловая система PVE (pmxcfs)

Кластерная файловая система PVE (pmxcfs) – это файловая система на основе базы данных для хранения файлов конфигурации виртуальных окружений, реплицируемая в режиме реального времени на все узлы кластера с помощью `corosync`. Эта файловая система используется для хранения всех конфигурационных файлов связанных с PVE. Хотя файловая система хранит все данные в базе данных на диске, копия данных находится в оперативной памяти, что накладывает ограничение на максимальный размер данных, который в настоящее время составляет 30 МБ.

Преимущества файловой системы pmxcfs:

- мгновенная репликация и обновление конфигурации на все узлы в кластере;
- исключается вероятность дублирования идентификаторов виртуальных машин;

- в случае развала кворума в кластере, файловая система становится доступной только для чтения.

Все файлы и каталоги принадлежат пользователю root и имеют группу www-data. Только root имеет права на запись, но пользователи из группы www-data могут читать большинство файлов. Файлы в каталогах `/etc/pve/priv/` и `/etc/pve/nodes/${NAME}/priv/` доступны только root.

Файловая система смонтирована в `/etc/pve/`.

4.4 Системы хранения

Образы ВМ могут храниться в одном или нескольких локальных хранилищах или в общем (совместно используемом) хранилище, например, NFS или iSCSI (NAS, SAN). Ограничений нет, можно настроить столько хранилищ, сколько необходимо.

В кластерной среде PVE наличие общего хранилища не является обязательным, однако оно делает управление хранением более простой задачей. Преимущества общего хранилища:

- миграция ВМ в реальном масштабе времени;
- плавное расширение пространства хранения с множеством узлов;
- централизованное резервное копирование;
- многоуровневое кэширование данных;
- централизованное управление хранением.

4.4.1 Типы хранилищ в PVE

Существует два основных типа хранилищ:

- файловые хранилища – хранят данные в виде файлов. Технологии хранения на уровне файлов обеспечивают доступ к полнофункциональной файловой системе (POSIX). В целом они более гибкие, чем любое хранилище на уровне блоков, и позволяют хранить контент любого типа;
- блочное хранилище – позволяет хранить большие необработанные образы. Обычно в таких хранилищах невозможно хранить другие файлы (ISO-образы, резервные копии, и т.д.). Большинство современных реализаций хранилищ на уровне блоков поддерживают моментальные снимки и клонирование. RADOS и GlusterFS являются распределенными системами, реплицирующими данные хранилища на разные узлы.

Хранилищами данных удобно управлять через веб-интерфейс. В качестве бэкенда хранилищ PVE может использовать:

- сетевые файловые системы, в том числе распределенные: NFS, CEPH, GlusterFS;
- локальные системы управления дисковыми томами: LVM, ZFS;
- удаленные (iSCSI) и локальные дисковые тома;

- локальные дисковые каталоги.

Доступные типы хранилищ приведены в табл. 10.

Т а б л и ц а 10 – Доступные типы хранилищ

Хранилище	PVE тип	Уровень	Общее (shared)	Снимки (snapshots)
ZFS (локальный)	zfspool	файл	нет	да
Каталог	dir	файл	нет	нет (возможны в формате qcow2)
BTRFS	btrfs	файл	нет	да
NFS	nfs	файл	да	нет (возможны в формате qcow2)
CIFS	cifs	файл	да	нет (возможны в формате qcow2)
GlusterFS	glusterfs	файл	да	нет (возможны в формате qcow2)
CephFS	cephfs	файл	да	да
LVM	lvm	блок	нет[1]	нет
LVM-thin	lvmthin	блок	нет	да
iSCSI/kernel	iscsi	блок	да	нет
iSCSI/libiscsi	iscsidirect	блок	да	нет
Ceph/RBD	rbd	блок	да	да
ZFS over iSCSI	zfs	блок	да	да
Proxmox Backup	pbs	файл/блок	да	-

4.4.2 Конфигурация хранилища

Вся связанная с PVE информация о хранилищах хранится в файле `/etc/pve/storage.cfg`. Поскольку этот файл находится в `/etc/pve/`, он автоматически распространяется на все узлы кластера. Таким образом, все узлы имеют одинаковую конфигурацию хранилища.

Совместное использование конфигурации хранилища имеет смысл для общего хранилища, поскольку одно и то же «общее» хранилище доступно для всех узлов. Но также полезно для локальных типов хранения. В этом случае такое локальное хранилище доступно на всех узлах, но оно физически отличается и может иметь совершенно разное содержимое.

Каждое хранилище имеет <тип> и уникально идентифицируется своим <STORAGE_ID>. Конфигурация хранилища выглядит следующим образом:

```
<type>: <STORAGE_ID>
        <property> <value>
        <property> <value>
        ...
```

Строка `<type>: <STORAGE_ID>` определяет хранилище, затем следует список свойств.

Пример файла `/etc/pve/storage.cfg`:

```
# cat /etc/pve/storage.cfg
```

[1] Можно использовать LVM поверх хранилища на базе iSCSI или FC, получив таким образом общее хранилище LVM.

```
dir: local
      path /var/lib/vz
      content images,rootdir,iso,snippets,vztmpl
      maxfiles 0
nfs: nfs-storage
      export /export/storage
      path /mnt/nfs-vol
      server 192.168.0.105
      content images,iso,backup,vztmpl
      options vers=3,nolock,tcp
```

В данном случае файл содержит описание специального хранилища local, которое ссылается на каталог `/var/lib/vz`, и описание NFS-хранилища nfs-storage.

Некоторые параметры являются общими для разных типов хранилищ (табл. 11).

Т а б л и ц а 11 – Общие параметры хранилищ

Свойство	Описание
nodes	Список узлов кластера, где хранилище можно использовать/доступно. Можно использовать это свойство, чтобы ограничить доступ к хранилищу
content	Хранилище может поддерживать несколько типов содержимого. Это свойство указывает, для чего используется это хранилище. Доступные опции: - images – образы виртуальных дисков; - rootdir – данные контейнеров; - vztmpl – шаблоны контейнеров; - backup – резервные копии (vzdump); - iso – ISO-образы; - snippets – файлы фрагментов (сниппетов), например, скрипты-ловушки гостевой системы.
shared	Указать, что это единое хранилище с одинаковым содержимым на всех узлах (или на всех перечисленных в опции nodes). Данное свойство не делает содержимое локального хранилища автоматически доступным для других узлов, он просто помечает как таковое уже общее хранилище!
disable	Отключить хранилище
maxfiles	Устарело, следует использовать свойство prune-backups. Максимальное количество файлов резервных копий на ВМ
prune-backups	Варианты хранения резервных копий
format	Формат образа по умолчанию (raw|qcow2|vmdk)
preallocation	Режим предварительного выделения (off|metadata|falloc|full) для образов raw и qcow2 в файловых хранилищах. По умолчанию используется значение metadata (равносильно значению off для образов raw). При использовании сетевых хранилищ в сочетании с большими образами qcow2, использование значения off может помочь избежать таймаутов.

П р и м е ч а н и е . Не рекомендуется использовать один и тот же пул хранения в разных PVE-кластерах. Для некоторых операций требуется монопольный доступ к хранилищу, поэтому требуется правильная блокировка. Блокировка реализована внутри кластера, но не работает между разными кластерами.

4.4.3 Идентификатор тома

Для обращения к данным хранилищ используется специальная нотация. Том идентифицируется по <STORAGE_ID>, за которым через двоеточие следует имя тома, зависящее от типа хранилища. Примеры <VOLUME_ID>:

```
local:iso/slinux-10.2-x86_64.iso
local:101/vm-101-disk-0.qcow2
local:backup/vzdump-qemu-100-2023_08_22-21_12_33.vma.zst
nfs-storage:vztmpl/alt-p10-rootfs-systemd-x86_64.tar.xz
```

Чтобы получить путь к файловой системе для <VOLUME_ID> используется команда:

```
# pvesm path <VOLUME_ID>
```

Например:

```
# pvesm path local:iso/slinux-10.2-x86_64.iso
/var/lib/vz/template/iso/slinux-10.2-x86_64.iso
# pvesm path nfs-storage:vztmpl/alt-p10-rootfs-systemd-x86_64.tar.xz
/mnt/pve/nfs-storage/template/cache/alt-p10-rootfs-systemd-x86_64.tar.xz
```

Для томов типа image существует отношение владения. Каждый такой том принадлежит ВМ или контейнеру. Например, том local:101/vm-101-disk-0.qcow2 принадлежит ВМ 101. При удалении ВМ или контейнера система также удаляет все связанные тома, принадлежащие этой ВМ или контейнеру.

4.4.4 Работа с хранилищами в PVE

4.4.4.1 Использование командной строки

Утилита `pvesm` (PVE Storage Manager), позволяет выполнять общие задачи управления хранилищами.

Команды добавления (подключения) хранилища:

```
# pvesm add <TYPE> <STORAGE_ID> <OPTIONS>
# pvesm add dir <STORAGE_ID> --path <PATH>
# pvesm add nfs <STORAGE_ID> --path <PATH> --server <SERVER> --export <EXPORT>
# pvesm add lvm <STORAGE_ID> --vgname <VGNAME>
# pvesm add iscsi <STORAGE_ID> --portal <HOST[:PORT]> --target <TARGET>
```

Отключить хранилище:

```
# pvesm set <STORAGE_ID> --disable 1
```

Включить хранилище:

```
# pvesm set <STORAGE_ID> --disable 0
```

Для того чтобы изменить/установить опции хранилища, можно выполнить команды:

```
# pvesm set <STORAGE_ID> <OPTIONS>
# pvesm set <STORAGE_ID> --shared 1
# pvesm set local --format qcow2
# pvesm set <STORAGE_ID> --content iso
```

Удалить хранилище (при этом никакие данные не удаляются, удаляется только конфигурация хранилища):

```
# pvesm remove <STORAGE_ID>
```

Команда выделения тома:

```
# pvesm alloc <STORAGE_ID> <VMID> <name> <size> [--format <raw|qcow2>]
```

Выделить том 4 ГБ в локальном хранилище (имя будет сгенерировано):

```
# pvesm alloc local <VMID> '' 4G
```

Освободить место (уничтожает все данные тома):

```
# pvesm free <VOLUME_ID>
```

Вывести список хранилищ:

```
# pvesm status
```

Вывести список содержимого хранилища:

```
# pvesm list <STORAGE_ID> [--vmid <VMID>]
```

Вывести список ISO-образов:

```
# pvesm list <STORAGE_ID> --iso
```

4.4.4.2 Добавление хранилища в веб-интерфейсе PVE

Хранилища, которые могут быть добавлены в веб-интерфейсе PVE (Рис. 83):

- Локальные хранилища:
  - Каталог – хранение на существующей файловой системе;
  - LVM – локальные устройства, такие как, FC, DRBD и т.д.;
  - ZFS;
  - BTRFS;
- Сетевые хранилища:
  - LVM – сетевая поддержка с iSCSI target;
  - NFS;
  - CIFS;
  - GlusterFS;
  - iSCSI;
  - CephFS;
  - RBD;
  - ZFS over iSCSI.

Выбор типа добавляемого хранилища

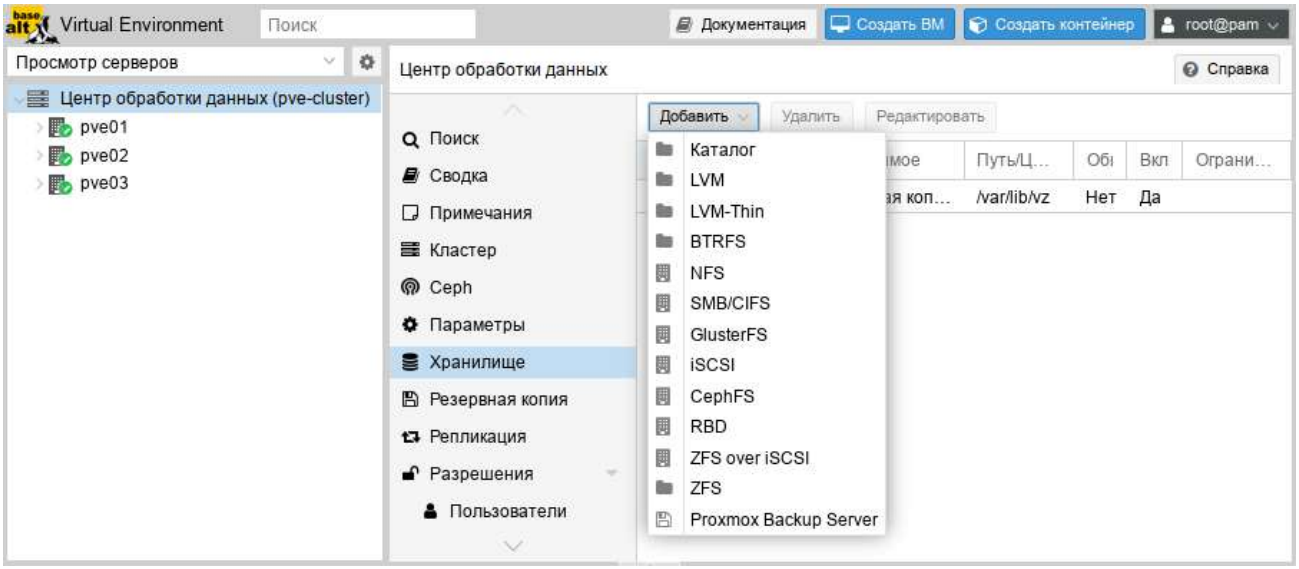


Рис. 83

При создании каждому хранилищу данных присваивается роль или набор ролей. Например, хранение контейнеров, образов виртуальных дисков, ISO-файлов и так далее. Список возможных ролей зависит от бэкенда хранилища. Например, для NFS или каталога локальной файловой системы доступны любые роли или наборы ролей (Рис. 84), а хранилища на базе RBD можно использовать только для хранения образов дисков и контейнеров.

Выбор ролей для хранилища

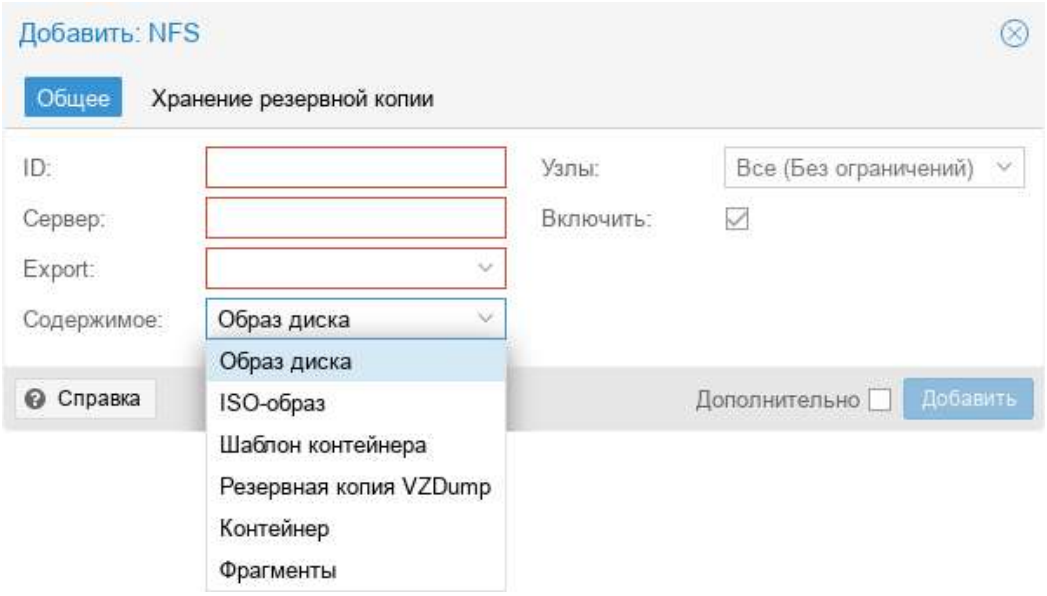


Рис. 84

Примечание. Можно добавить локальные хранилища (ZFS, LVM и LVM-Thin), расположенные на других узлах кластера. Для этого в мастере добавления хранилища следует выбрать узел для сканирования (Рис. 85).

LVM хранилище. Выбор узла для сканирования

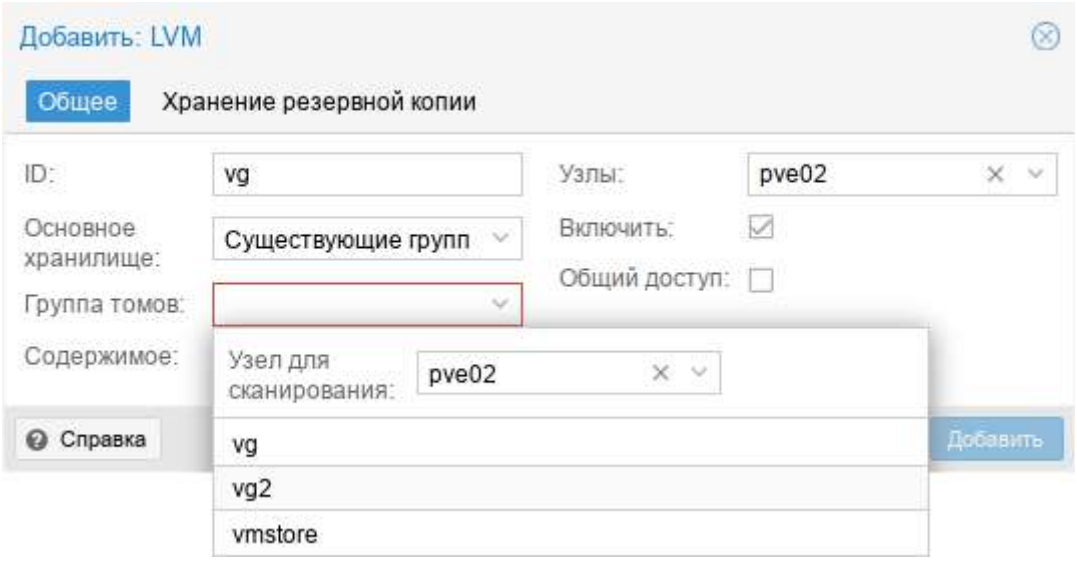


Рис. 85

4.4.4.3 Каталог

PVE может использовать локальные каталоги или локально смонтированные общие ресурсы для организации хранилища. Каталог – это файловое хранилище, поэтому в нем можно хранить данные любого типа, например, образы виртуальных дисков, контейнеры, шаблоны, ISO-образы или файлы резервных копий. Для хранения данных разного типа, используется предопределенная структура каталогов (табл. 12). Эта структура используется на всех файловых хранилищах.

Т а б л и ц а 12 – Структура каталогов файлового хранилища

Тип данных	Подкаталог
Образы дисков ВМ	images/<VMID>/
ISO-образы	template/iso/
Шаблоны контейнеров	template/cache/
Резервные копии VZDump	dump/
Фрагменты (сниппеты)	snippets/

П р и м е ч а н и е . Дополнительные хранилища можно подключить в `/etc/fstab`, а затем определить хранилище каталогов для этой точки монтирования. Таким образом, можно использовать любую файловую систему (ФС), поддерживаемую Linux.

П р и м е ч а н и е . Большинство ФС «из коробки» не поддерживают моментальные снимки. Чтобы обойти эту проблему, этот бэкэнд может использовать возможности внутренних снимков qcow2.

Для создания нового хранилища типа «Каталог» необходимо выбрать «Центр обработки данных» → «Хранилище», нажать кнопку «Добавить» и в выпадающем меню выбрать пункт «Каталог» (Рис. 83). На Рис. 86 показан диалог создания хранилища local-iso типа «Каталог» для хранения ISO-образов и шаблонов контейнеров, которое будет смонтировано в каталог `/mnt/iso`.

Добавление хранилища «Каталог»

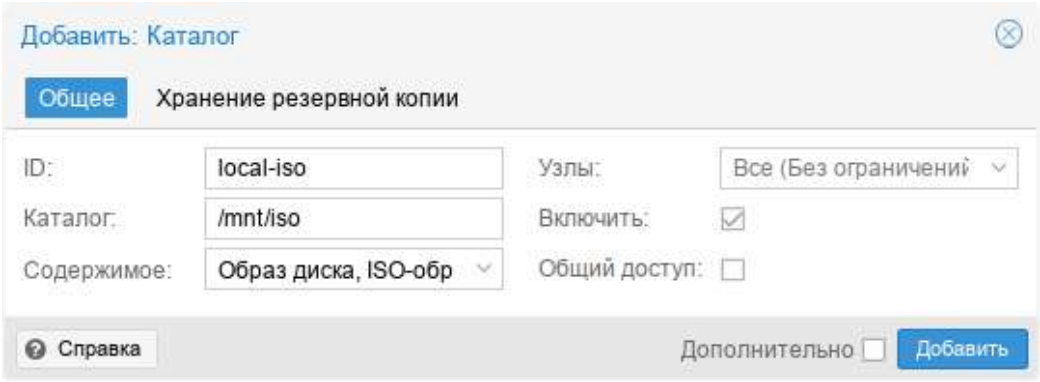


Рис. 86

Данное хранилище поддерживает все общие свойства хранилищ и дополнительно свойства:

- path – указание каталога (это должен быть абсолютный путь к файловой системе);
- content-dirs (опционально) – позволяет изменить макет по умолчанию. Состоит из списка идентификаторов, разделенных запятыми, в формате:

`vtype=path`

где vtype – один из разрешенных типов контента для хранилища, а path – путь относительно точки монтирования хранилища.

Пример файла конфигурации (`/etc/pve/storage.cfg`):

```
dir: backup
      path /mnt/backup
      content backup
      prune-backups keep-last=7
      shared 0
      content-dirs backup=custom/backup
```

Данная конфигурация определяет пул хранения резервных копий. Этот пул может использоваться для хранения последних 7 резервных копий на виртуальную машину. Реальный путь к файлам резервных копий – `/mnt/backup/custom/backup`.

Примечание. Флаг `shared` («Общий доступ») можно установить только для кластерных ФС (например, ocfs2).

Хранилище «Каталог» использует следующую схему именования образов ВМ:

`VM-<VMID>-<NAME>.<FORMAT>`

где:

<VMID> – идентификатор ВМ;

<NAME> – произвольное имя (ascii) без пробелов. По умолчанию используется disk-[N], где [N] заменяется целым числом;

<FORMAT> – определяет формат образа (raw|qcow2|vmdk).

Пример:

```
# ls /var/lib/vz/images/101
vm-101-disk-0.qcow2 vm-101-disk-1.qcow2
```

При создании шаблона ВМ все образы дисков ВМ переименовываются, чтобы указать, что они теперь доступны только для чтения и могут использоваться в качестве базового образа для клонов:

```
base-<VMID>-<NAME>.<FORMAT>
```

4.4.4.4 NFS

Хранилище NFS аналогично хранению каталогов и файлов на диске, с дополнительным преимуществом совместного хранения и миграции в реальном времени. Свойства хранилища NFS во многом совпадают с хранилищем типа «Каталог». Структура каталогов и соглашение об именах файлов также одинаковы. Основным преимуществом является то, что можно напрямую настроить свойства сервера NFS, и общий ресурс будет монтироваться автоматически (без необходимости изменения `/etc/fstab`).

Данное хранилище поддерживает все общие свойства хранилищ кроме флага shared, который всегда установлен. Кроме того, для настройки NFS используются следующие свойства:

- server – IP-адрес сервера или DNS-имя. Предпочтительнее использовать IP-адрес вместо DNS-имени (чтобы избежать задержек при поиске DNS);
- export – совместный ресурс с сервера NFS (список можно просмотреть, выполнив команду `pvesm scan nfs <server>`);
- path – локальная точка монтирования (по умолчанию `/mnt/pve/<STORAGE_ID>/`);
- content-dirs (опционально) – позволяет изменить макет по умолчанию. Состоит из списка идентификаторов, разделенных запятыми, в формате:

```
vtype=path
```

где vtype – один из разрешенных типов контента для хранилища, а path – путь относительно точки монтирования хранилища;

- options – параметры монтирования NFS (см. man nfs).

Пример файла конфигурации (`/etc/pve/storage.cfg`):

```
nfs: nfs-storage
      export /export/storage
      path /mnt/pve/nfs-storage
      server 192.168.0.105
      content images,iso,backup,vztmpl
```

```
options vers=3,nolock,tcp
```

Примечание. По истечении времени ожидания запрос NFS по умолчанию повторяется бесконечно. Это может привести к неожиданным зависаниям на стороне клиента. Для содержимого, доступного только для чтения, следует рассмотреть возможность использования NFS-опции `soft`, в этом случае будет выполняться только три запроса.

Примечание. Для возможности монтирования NFS хранилища должен быть запущен nfs-client:

```
# systemctl enable --now nfs-client.target
```

На Рис. 87 показано присоединение хранилища NFS с именем nfs-storage с удаленного сервера 192.168.0.105.

Создание хранилища NFS

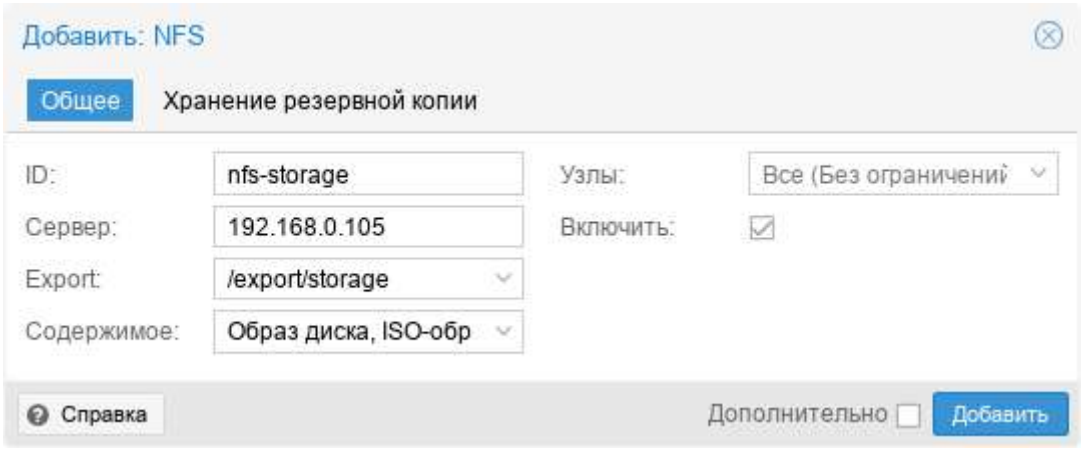


Рис. 87

После ввода IP-адреса удаленного сервера доступные ресурсы будут автоматически просканированы и будут отображены в выпадающем списке «Export». В данном примере обнаруженная в блоке диалога точка монтирования – `/export/storage`.

Пример добавления хранилища NFS в командной строке с помощью утилиты `pvesm`:

```
# pvesm add nfs nfs-storage --path /mnt/pve/nfs-storage --server 192.168.0.105 --
options vers=3,nolock,tcp --export /export/storage --content images,iso,vztmpl,backup
```

Получить список совместных ресурсов с сервера NFS:

```
# pvesm nfsscan <server>
```

4.4.4.5 BTRFS

Свойства хранилища BTRFS во многом совпадают с хранилищем типа «Каталог». Основное отличие состоит в том, что при использовании этого типа хранилища диски в формате raw будут помещены в subvolume (подтом), для возможности создания снимков (снапшотов) и поддержки автономной миграции хранилища с сохранением снимков.

Примечание. BTRFS учитывает флаг O_DIRECT при открытии файлов, что означает, что ВМ не должны использовать режим кеширования none, иначе возникнут ошибки контрольной суммы.

Пример файла конфигурации (`/etc/pve/storage.cfg`):

```
btrfs: btrfs-storage
      path /mnt/data/btrfs-storage
      content rootdir,images
      is_mountpoint /mnt/data
      nodes pve02
      prune-backups keep-all=1
```

На Рис. 88 показан диалог создания хранилища brtfs-storage типа BTRFS для хранения образов дисков и контейнеров.

Создание хранилища BTRFS

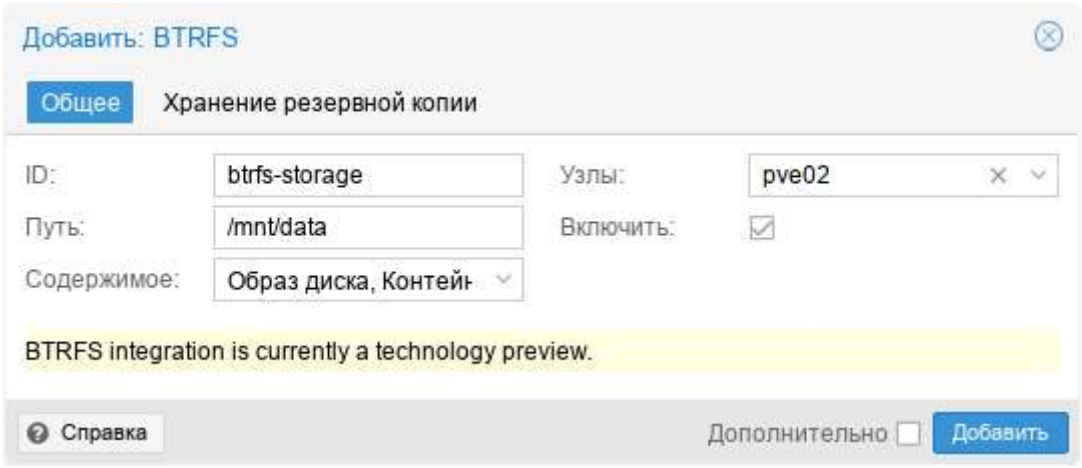


Рис. 88

Пример добавления хранилища BTRFS в командной строке с помощью утилиты `pvesm`:

```
# pvesm add btrfs btrfs-storage --path /mnt/data/btrfs-storage --is_mountpoint /mnt/data --content images,rootdir
```

4.4.4.5.1 Администрирование BTRFS

В этом разделе приведены некоторые примеры работы с ФС BTRFS.

Пример создания ФС BTRFS на диске /dev/sdd:

```
# mkfs.btrfs -m single -d single -L My-Storage /dev/sdd
```

Параметры `-m` и `-d` используются для установки профиля для метаданных и данных соответственно. С помощью необязательного параметра `-L` можно установить метку.

Можно создать RAID1 на двух разделах /dev/sdb1 и/dev/sdc1:

```
 # mkfs.btrfs -m raid1 -d raid1 -L My-Storage /dev/sdb1 /dev/sdc1
```

Новую ФС можно смонтировать вручную, например:

```
# mkdir /mnt/data
# mount /dev/sdd /mnt/data
```

Для автоматического монтирования BTRFS раздела, следует добавить этот раздел в /etc/fstab. Рекомендуется использовать значение UUID (выведенное при выполнении команды mkfs.btrfs), например:

```
UUID=5a556184-43b2-4212-bc21-eee3798c8322 /mnt/data btrfs defaults 0 0
```

Выполнить проверку монтирования:

```
# mount -a
```

Результатом выполнения команды должен быть пустой вывод без ошибок.

Примечание. UUID можно также получить, выполнив команду:

```
# blkid
/dev/sdd:        LABEL="My-Storage"        UUID="5a556184-43b2-4212-bc21-eee3798c8322"
BLOCK_SIZE="4096" TYPE="btrfs"
```

Создание подтома (файловая система BTRFS должна быть примонтирована):

```
# btrfs subvolume create /mnt/data/btrfs-storage
```

Создание снимка (снимок – это подтом, который имеет общие данные и метаданные с другим подтомом):

```
# btrfs subvolume snapshot -r /mnt/data/btrfs-storage /mnt/data/new
```

Будет создан доступный только для чтения «клон» подтома /mnt/data/btrfs-storage. Чтобы из снимка, доступного только для чтения, создать его версию, доступную для записи, следует просто создать его снимок без опции `-r`.

Просмотреть список текущих подтомов:

```
# btrfs subvolume list /mnt/data
ID 256 gen 17 top level 5 path btrfs-storage
ID 257 gen 14 top level 5 path new
```

Удаление подтома:

```
# btrfs subvolume delete /mnt/data/btrfs-storage
```

Отображение занятого/свободного места:

```
# btrfs filesystem usage /mnt/data
```

или:

```
$ btrfs filesystem df /mnt/data
```

4.4.4.6 SMB/CIFS

Хранилище SMB/CIFS расширяет хранилище типа «Каталог», поэтому ручная настройка монтирования CIFS не требуется.

Примечание. Для возможности просмотра общих ресурсов на каждом узле кластера необходимо установить пакет `samba-client`.

Данное хранилище поддерживает все общие свойства хранилищ кроме флага `shared`, который всегда установлен. Кроме того, для настройки CIFS используются следующие свойства:

- server – IP-адрес сервера или DNS-имя. Предпочтительнее использовать IP-адрес вместо DNS-имени (чтобы избежать задержек при поиске DNS);
- share – совместный ресурс с сервера CIFS (список можно просмотреть, выполнив команду `pvesm scan cifs <server>`);
- username – имя пользователя для хранилища CIFS (необязательно, по умолчанию «guest»);
- password – пароль пользователя (опционально). Пароль будет сохранен в файле, доступном только для чтения root-пользователю (`/etc/pve/priv/storage/<STORAGE_ID>.pw`);
- domain – устанавливает домен пользователя (рабочую группу) для этого хранилища (опционально);
- smbversion – версия протокола SMB (опционально, по умолчанию 3);
- path – локальная точка монтирования (по умолчанию `/mnt/pve/<STORAGE_ID>/`);
- content-dirs (опционально) – позволяет изменить макет по умолчанию. Состоит из списка идентификаторов, разделенных запятыми, в формате:

`vtype=path`

где vtype – один из разрешенных типов контента для хранилища, а path – путь относительно точки монтирования хранилища;

- options – дополнительные параметры монтирования CIFS (см. man mount.cifs). Некоторые параметры устанавливаются автоматически, и их не следует задавать в этом параметре. PVE всегда устанавливает опцию soft;
- subdir – подкаталог общего ресурса, который необходимо смонтировать. Необязательно, по умолчанию используется корневой каталог общего ресурса.

Пример файла конфигурации (`/etc/pve/storage.cfg`):

```
cifs: newCIFS
      path /mnt/pve/newCIFS
      server 192.168.0.105
      share smb_data
```

Получить список совместных ресурсов с сервера CIFS можно, выполнив команду:

```
# pvesm cifsscan <server> [--username <username>] [--password]
```

Команда добавления общего ресурса в качестве хранилища:

```
# pvesm add cifs <storagename> --server <server> --share <share> [--username <username>] [--password]
```

На Рис. 89 показано присоединение хранилища SMB/CIFS с именем newCIFS с удаленного сервера 192.168.0.105.

После ввода IP-адреса удаленного сервера доступные ресурсы будут автоматически просканированы и будут отображены в выпадающем списке «Share».

Примечание. При создании CIFS хранилища в веб-интерфейсе, PVE предполагает, что удаленный сервер поддерживает CIFS v3. В веб-интерфейсе нет возможности указать версию CIFS, поэтому в случае, если удалённый сервер поддерживает версии CIFS отличные от v3, создать хранилище можно в командной строке, например:

```
# pvesm add cifs newCIFS --server 192.168.0.105 --share smb_data --smbversion 2.1
```

Добавление CIFS хранилища

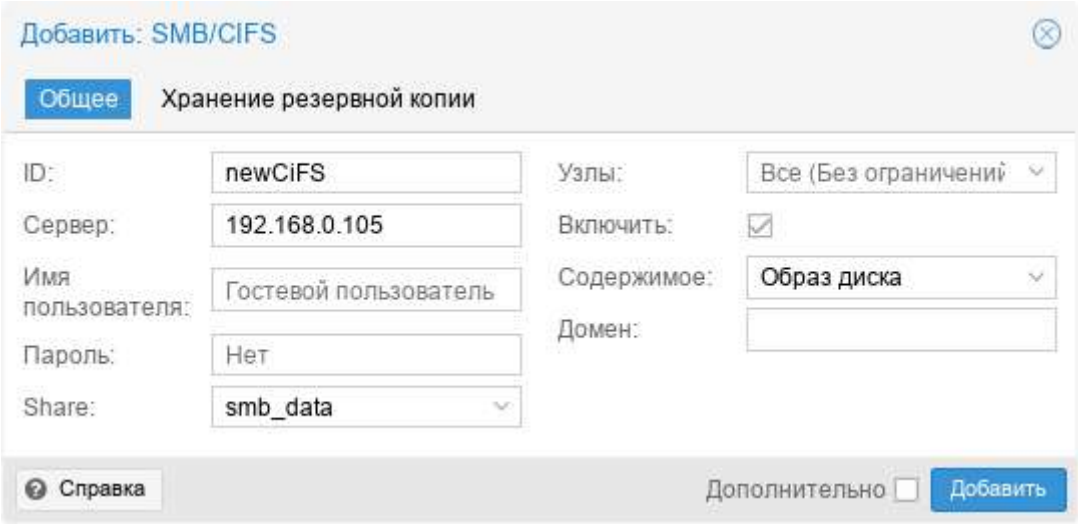


Рис. 89

4.4.4.7 GlusterFS

GlusterFS – это масштабируемая сетевая файловая система. Система использует модульную конструкцию, работает на аппаратном оборудовании и может обеспечить высокодоступное корпоративное хранилище при низких затратах. Такая система способна масштабироваться до нескольких петабайт и может обрабатывать тысячи клиентов.

Примечание. После сбоя узла GlusterFS выполняет полную синхронизацию, чтобы убедиться в согласованности данных. Для больших файлов это может занять очень много времени, поэтому это хранилище не подходит для хранения больших образов ВМ.

Данное хранилище поддерживает общие свойства (content, nodes, disable) хранилищ, и дополнительно используются следующие свойства:

- server – IP-адрес или DNS-имя сервера GlusterFS;
- server2 – IP-адрес или DNS-имя резервного сервера GlusterFS;
- volume – том GlusterFS;
- transport – транспорт GlusterFS: tcp, unix или rdma.

Пример файла конфигурации (`/etc/pve/storage.cfg`):

```
glusterfs: gluster-01
             server 192.168.0.105
             server2 192.168.0.110
             volume glustervol
```

```
content images,iso
```

На Рис. 90 показано присоединение хранилища GlusterFS с именем gluster-01 с удаленного сервера 192.168.0.105.

Создание хранилища GlusterFS

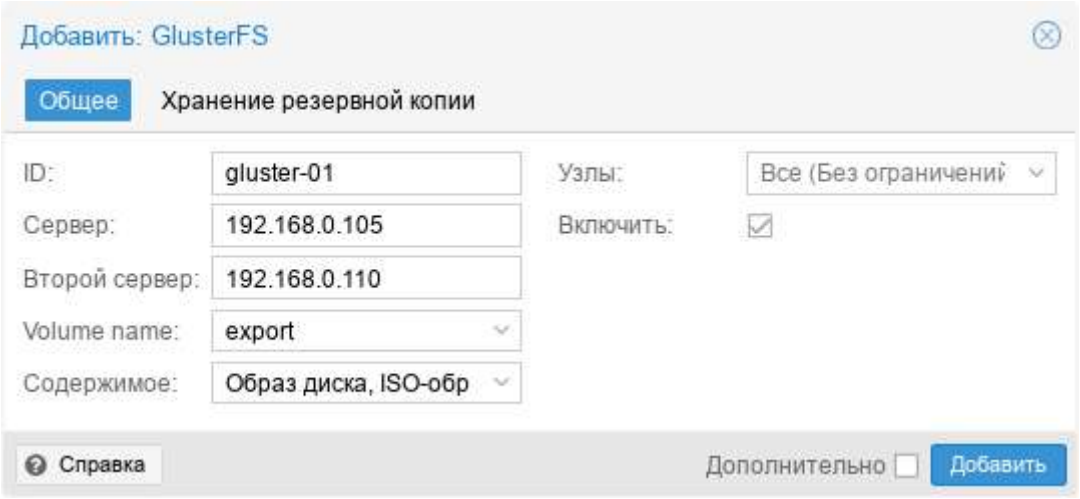


Рис. 90

4.4.4.8 Локальный ZFS

Примечание. Для работы с локальным ZFS хранилищем должен быть установлен модуль ядра `kernel-modules-zfs-std-def`. Включить модуль:

```
# modprobe zfs
```

Чтобы не вводить эту команду после перезагрузки, следует раскомментировать строку: `#zfs` в файле `/etc/modules-load.d/zfs.conf`.

Локальный ZFS позволяет получить доступ к локальным пулам ZFS (или файловым системам ZFS внутри таких пулов). Данное хранилище поддерживает общие свойства (content, nodes, disable) хранилищ, кроме того, для настройки ZFS используются следующие свойства:

- pool – пул/файловая система ZFS;
- blocksize – размер блока;
- sparse – использовать тонкое выделение ресурсов;
- mountpoint – точка монтирования пула/файловой системы ZFS. Изменение этого параметра не влияет на свойство точки монтирования набора данных, видимого zfs. По умолчанию /<pool>.

Пул ZFS поддерживает следующие типы RAID:

- RAID-0 (Single Disk) – размер такого пула – сумма емкостей всех дисков. RAID0 не добавляет избыточности, поэтому отказ одного диска делает том не пригодным для использования (минимально требуется один диск);

- пул RAID-1 (Mirror) – данные зеркалируются на все диски (минимально требуется два диска);
- пул RAID-10 – сочетание RAID0 и RAID1 (минимально требуется четыре диска);
- пул RAIDZ-1 – вариация RAID-5, одинарная четность (минимально требуется три диска);
- пул RAIDZ-2 – вариация на RAID-5, двойной паритет (минимально требуется четыре диска);
- пул RAIDZ-3 – разновидность RAID-5, тройная четность (минимально требуется пять дисков).

Пример файла конфигурации (`/etc/pve/storage.cfg`):

```
zfspool: vmdata
            pool vmdata
            content images,rootdir
            mountpoint /vmdata
            nodes pve03
```

Возможные типы содержимого: `rootdir` (данные контейнера), `images` (образ виртуального диска в формате raw или subvol).

Используется следующая схема именования образов дисков ВМ:

- vm-<VMID>-<NAME> – образ ВМ;
- base-<VMID>-<NAME> – шаблон образа ВМ (только для чтения);
- subvol-<VMID>-<NAME> – файловая система ZFS для контейнеров.

Примечание. Если в ВМ созданной в ZFS хранилище будет создан диск с LVM разделами, то гипервизор не позволит удалить этот диск. Пример ошибки:

```
cannot destroy 'data/vm-101-disk-0': dataset is busy
```

Чтобы избежать этой ситуации следует исключить ZFS-диски из области сканирования LVM, добавив в конфигурацию LVM (файл `/etc/lvm/lvm.conf`) в секцию `devices{}` строки:

```
# Do not scan ZFS zvols (to avoid problems on ZFS zvols snapshots)
filter = [ "r|^/dev/zd*|" ]
global_filter = [ "r|^/dev/zd*|" ]
```

4.4.4.8.1 Создание локального хранилища ZFS в веб-интерфейсе

Для создания локального хранилища ZFS в веб-интерфейсе следует выбрать узел, на котором будет создано хранилище, в разделе «Диски» выбрать пункт «ZFS» и нажать кнопку «Создать: ZFS» (Рис. 91).

В открывшемся окне (Рис. 92) следует задать параметры ZFS хранилища: имя хранилища, выбрать диски, уровень RAID и нажать кнопку «Создать».

Статус пула можно просмотреть выбрав его в списке и нажав кнопку «Подробно» (Рис. 93).

Добавление ZFS хранилища

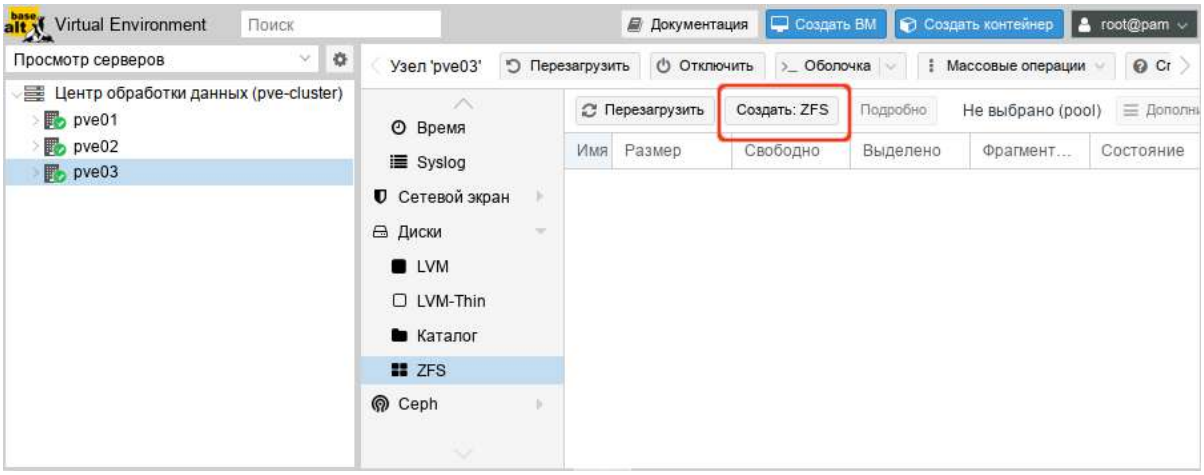


Рис. 91

Параметры ZFS хранилища

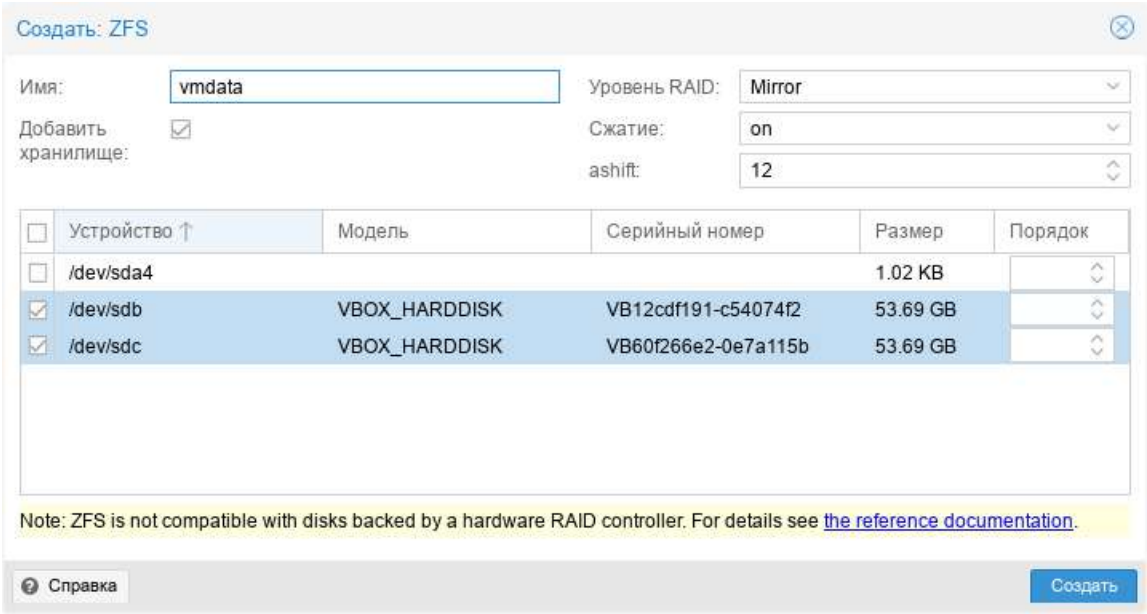


Рис. 92

Локальное ZFS хранилище

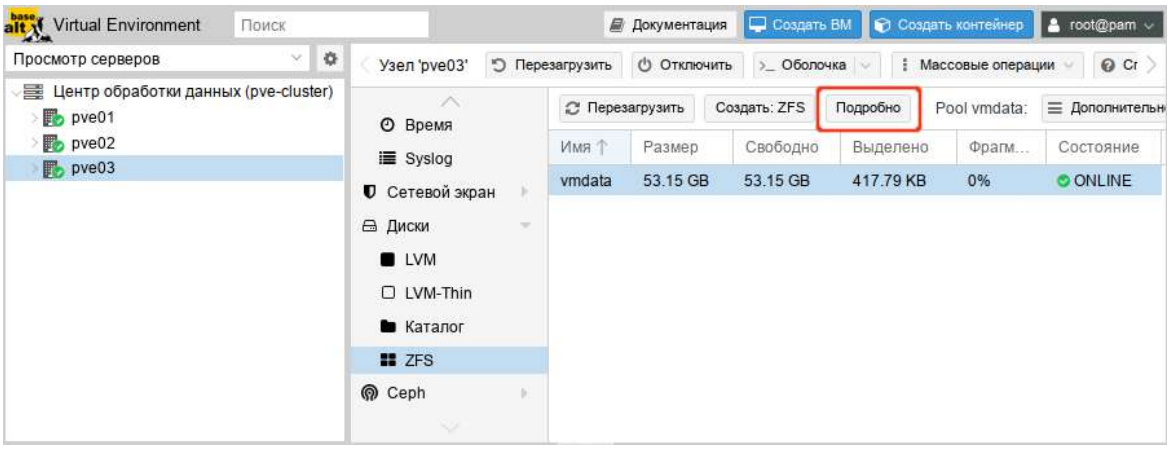


Рис. 93

Для того чтобы внести изменения в настройки ZFS хранилища, следует выбрать «Центр обработки данных» → «Хранилище», затем нужное хранилище и нажать кнопку «Редактировать» (Рис. 94). В открывшемся окне (Рис. 95) можно изменить тип содержимого контейнера, включить/отключить хранилище, включить дисковое резервирование.

Выбор хранилища для редактирования

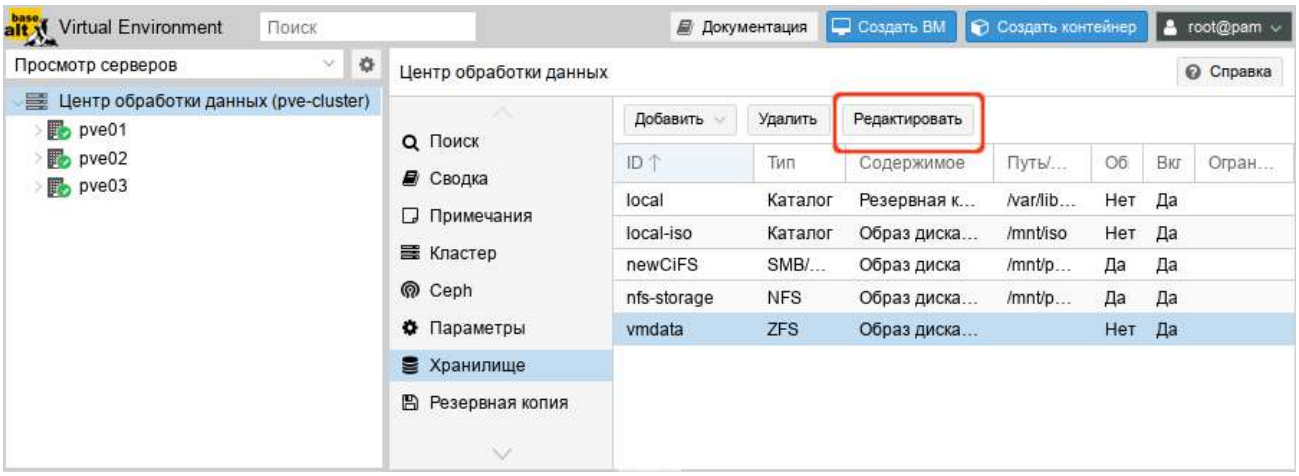


Рис. 94

Редактирование ZFS хранилища

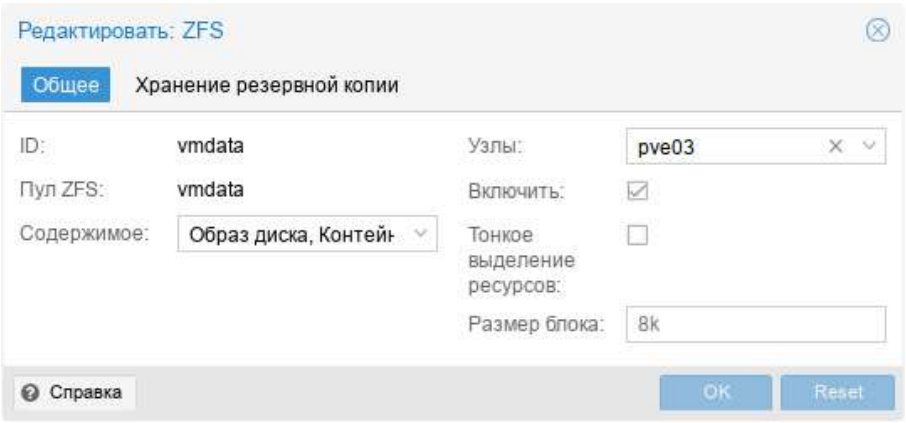


Рис. 95

4.4.4.8.2 Администрирование ZFS

Основными командами для управления ZFS являются zfs и zpool.

Для создания нового пула необходим как минимум один пустой диск.

Создание нового пула RAID-0 (минимум 1 диск):

```
# zpool create -f -o ashift=12 <pool> <device1> <device2>
```

Создание нового пула RAID-1 (минимум 2 диска):

```
# zpool create -f -o ashift=12 <pool> mirror <device1> <device2>
```

Создание нового пула RAID-10 (минимум 4 диска):

```
# zpool create -f -o ashift=12 <pool> mirror <device1> <device2> mirror <device3> <device4>
```

Создание нового пула RAIDZ-1 (минимум 3 диска):

```
# zpool create -f -o ashift=12 <pool> raidz1 <device1> <device2> <device3>
```

Создание нового пула RAIDZ-2 (минимум 4 диска):

```
# zpool create -f -o ashift=12 <pool> raidz2 <device1> <device2> <device3> <device4>
```

Смена неисправного устройства:

```
# zpool replace -f <pool> <old device> <new device>
```

Включить сжатие:

```
# zfs set compression=on <pool>
```

Получить список доступных ZFS файловых систем:

```
# pvesm zfsscan
```

Пример создания RAID1(mirror) с помощью zfs:

```
# zpool create -f vmdata mirror sdb sdc
```

Просмотреть созданные в системе пулы:

```
# zpool list
NAME     SIZE  ALLOC   FREE  CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP  HEALTH  ALTROOT
vmdata  17,5G   492K  17,5G        -         -     0%     0%  1.00x  ONLINE  -
```

Просмотреть статус пула:

```
# zpool status
  pool: vmdata
 state: ONLINE
  scan: none requested
config:

      NAME        STATE     READ WRITE CKSUM
      vmdata      ONLINE       0     0     0
        mirror-0  ONLINE       0     0     0
          sdb     ONLINE       0     0     0
          sdc     ONLINE       0     0     0
errors: No known data errors
```

4.4.4.9 LVM

LVM (Logical Volume Management) это система управления дисковым пространством. Позволяет логически объединить несколько дисковых пространств (физические тома) в одно, и уже из этого пространства (дисковой группы или группы томов – VG), можно выделять разделы (логические тома – LV), доступные для работы.

Использование LVM групп обеспечивает лучшую управляемость. Логические тома можно легко создавать/удалять/перемещать между физическими устройствами хранения. Если база хранения для группы LVM доступна на всех PVE узлах (например, ISCSI LUN) или репликах (например, DRBD), то все узлы имеют доступ к образам ВМ, и возможна live-миграция.

Данное хранилище поддерживает общие свойства (content, nodes, disable) хранилищ, кроме того, для настройки LVM используются следующие свойства:

- vgname – имя группы томов LVM (должно указывать на существующую группу томов);
- base – базовый том. Этот том автоматически активируется перед доступом к хранилищу. Это особенно полезно, когда группа томов LVM находится на удаленном сервере iSCSI;
- saferemove – обнуление данных при удалении LV. При удалении тома это гарантирует, что все данные будут удалены;
- saferemove_throughput – очистка пропускной способности (значение параметра `cstream -t`).

Пример файла конфигурации (`/etc/pve/storage.cfg`):

```
lvm: vg
      vgname vg
      content rootdir,images
      nodes pve03
      shared 0
```

Возможные типы содержимого: `rootdir` (данные контейнера), `images` (образ виртуального диска в формате raw).

4.4.4.9.1 Создание локального хранилища LVM в веб-интерфейсе

Примечание. Для создания локального LVM хранилища в веб-интерфейсе необходимо, чтобы в системе имелся хотя бы один пустой жесткий диск.

Для создания локального LVM хранилища в веб-интерфейсе следует выбрать узел, на котором будет создано хранилище, в разделе «Диски» выбрать пункт «LVM» и нажать кнопку «Создать: Volume Group» (Рис. 96). В открывшемся окне (Рис. 97) следует выбрать диск, задать имя группы томов, отметить пункт «Добавить хранилище» (если этот пункт не отмечен будет создана только группа томов) и нажать кнопку «Создать».

Пункт «LVM» в разделе «Диски»

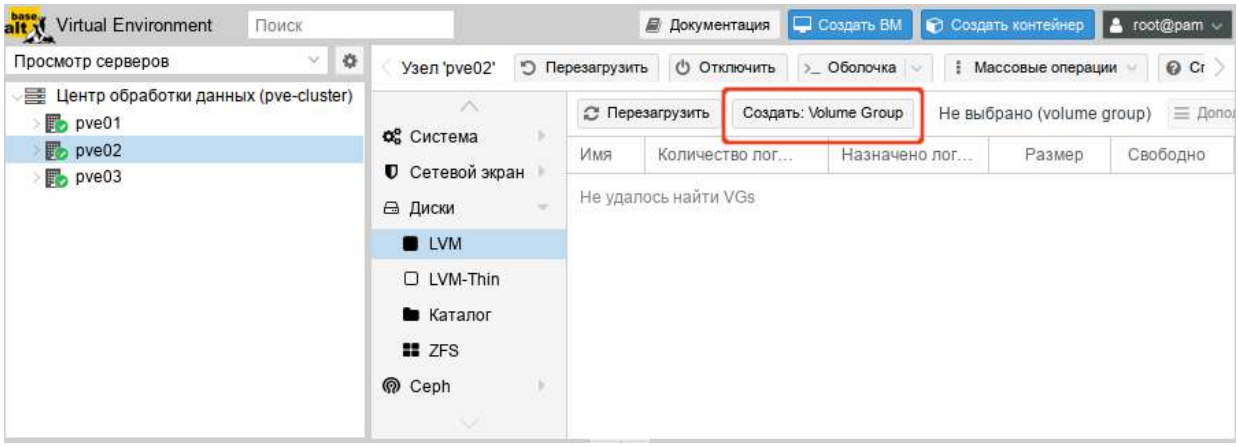


Рис. 96

Создание группы томов

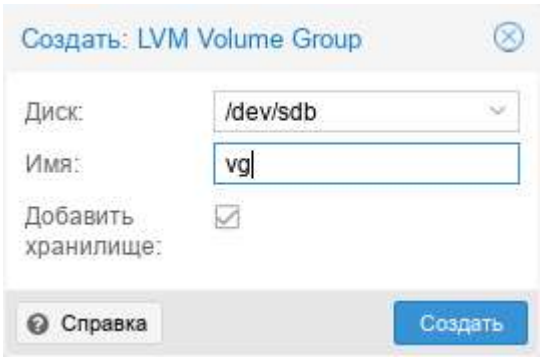


Рис. 97

Для того чтобы внести изменения в настройки LVM хранилища, следует выбрать «Центр обработки данных» → «Хранилище», затем нужное хранилище и нажать кнопку «Редактировать». В открывшемся окне (Рис. 98) можно изменить тип содержимого контейнера, включить/отключить хранилище.

Редактирование LVM хранилища

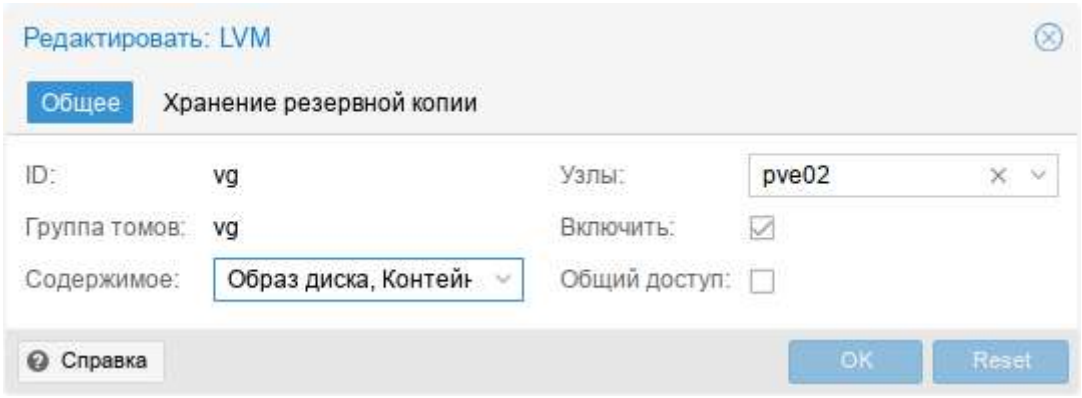


Рис. 98

Одним из преимуществ хранилища LVM является то, что его можно использовать поверх общего хранилища, например, iSCSI LUN (Рис. 99). Сам бэкэнд реализует правильную блокировку на уровне кластера.

Добавление хранилища типа LVM (over iSCSI)

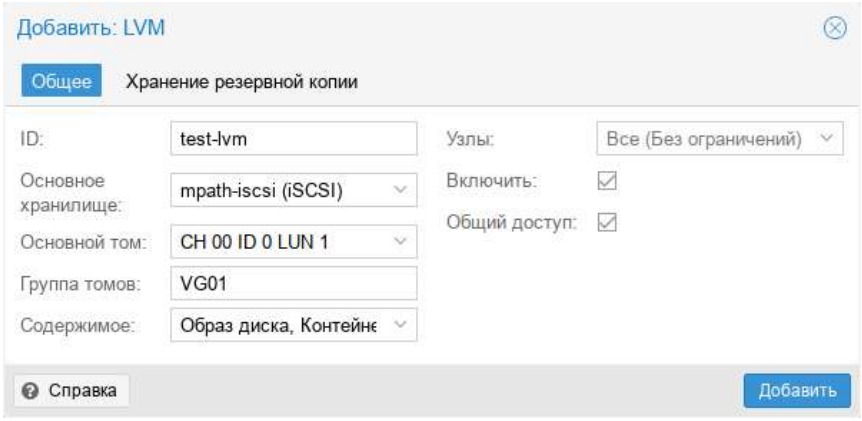


Рис. 99

4.4.4.9.2 Создание хранилища LVM в командной строке

Пример создания LVM хранилища на пустом диске /dev/sdd:

1) создать физический том (PV):

```
# pvcreate /dev/sdd
  Physical volume "/dev/sdd" successfully created.
```

2) создать группу томов (VG) с именем vg:

```
# vgcreate vg /dev/sdd
  Volume group "vg" successfully created
```

3) показать информацию о физических томах:

```
# pvs
PV         VG        Fmt  Attr PSize   PFree
  /dev/sdd   vg        lvm2 a--  <18,00g <3,00g
```

4) показать информацию о группах томов:

```
# vgs
  VG        #PV #LV #SN Attr   VSize   VFree
  vg          1   2   0 wz--n- <18,00g <3,00g
```

5) получить список доступных PVE групп томов:

```
# pvesm lvmscan
vg
```

6) создать LVM хранилище с именем myspace:

```
# pvesm add lvm myspace --vgname vg --nodes pve03
```

4.4.4.10 LVM-thin

LVM-thin (thin provision) – это возможность использовать какое-либо внешнее блочное устройство в режиме только для чтения как основу для создания новых логических томов LVM. Такие разделы при создании уже будут выглядеть так, будто они заполнены данными исходного блочного устройства. Операции с томами изменяются налету таким образом, что чтение данных выполняется с исходного блочного устройства (или с тома, если данные уже отличаются), а запись – на том.

Такая возможность может быть полезна, например, при создании множества однотипных ВМ или для решения других аналогичных задач, т.е. задач, где нужно получить несколько изменяемых копий одних и тех же исходных данных.

Данное хранилище поддерживает общие свойства хранилищ, кроме того, для настройки LVM-thin используются следующие свойства:

- vgname – имя группы томов LVM (должно указывать на существующую группу томов);
- thinpool – название тонкого пула LVM.

Пример файла конфигурации (`/etc/pve/storage.cfg`):

```
lvmthin: vmstore
```

```
thinpool vmstore
vgname vmstore
content rootdir,images
nodes pve03
```

Возможные типы содержимого: `rootdir` (данные контейнера), `images` (образ виртуального диска в формате raw).

LVM thin является блочным хранилищем, но полностью поддерживает моментальные снимки и клоны. Новые тома автоматически инициализируются с нуля.

Тонкие пулы LVM не могут совместно использоваться несколькими узлами, поэтому их можно использовать только в качестве локального хранилища.

4.4.4.10.1 Создание локального хранилища LVM-Thin в веб-интерфейсе

Примечание. Для создания локального LVM-Thin хранилища в веб-интерфейсе необходимо, чтобы в системе имелся хотя бы один пустой жесткий диск.

Для создания локального LVM-Thin хранилища в веб-интерфейсе следует выбрать узел, на котором будет создано хранилище, в разделе «Диски» выбрать пункт «LVM-Thin» и нажать кнопку «Создать: Thinpool» (Рис. 100). В открывшемся окне (Рис. 101) следует выбрать диск, задать имя группы томов, отметить пункт «Добавить хранилище» (если этот пункт не отмечен будет создана только группа томов) и нажать кнопку «Создать».

Пункт «LVM-Thin» в разделе «Диски»

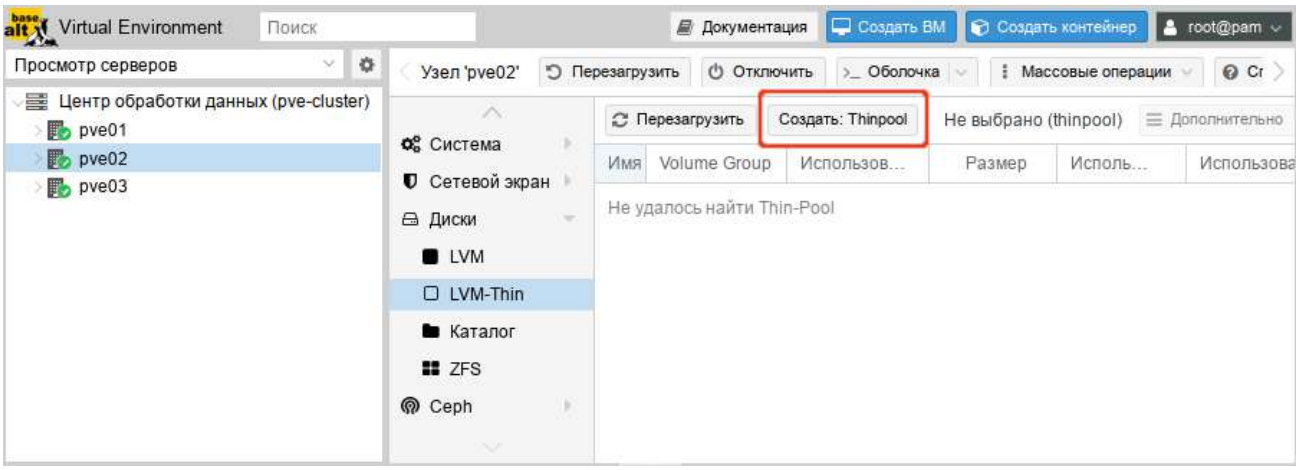


Рис. 100

Создание LVM-Thin хранилища

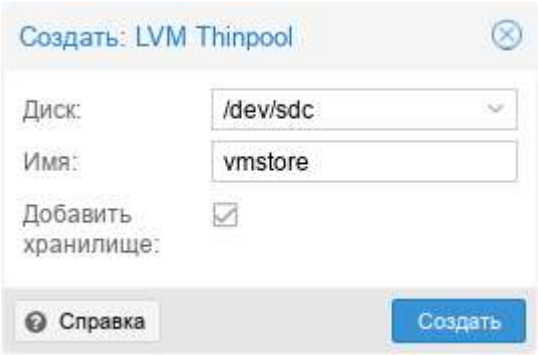


Рис. 101

Для того чтобы внести изменения в настройки LVM-Thin хранилища, следует выбрать «Центр обработки данных» → «Хранилище», затем нужное хранилище и нажать кнопку «Редактировать». В открывшемся окне (Рис. 102) можно изменить тип содержимого контейнера, включить/отключить хранилище.

Редактирование LVM-Thin хранилища

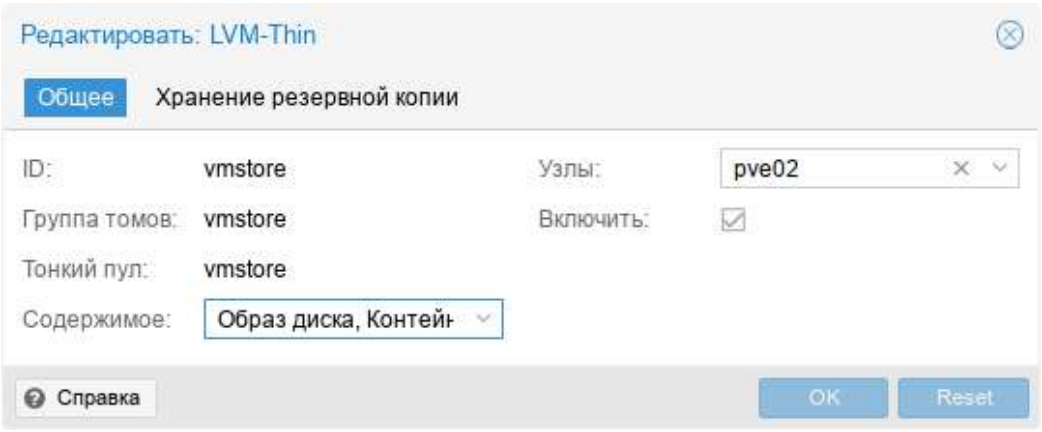


Рис. 102

4.4.4.10.2 Создание хранилища LVM-Thin в командной строке

Для создания и управления пулами LVM-Thin можно использовать инструменты командной строки.

Пул LVM-Thin должен быть создан поверх группы томов.

Команда создания нового тонкого пула LVM (размер 80 ГБ) с именем vmstore (предполагается, что группа томов LVM с именем vg уже существует):

```
# lvcreate -L 80G -T -n vmstore vg
```

Получить список доступных LVM-thin пулов в группе томов vg:

```
# pvesm lvmthinscan vg
vmstore
```

Команда создания LVM-Thin хранилища с именем vmstore на узле pve03:

```
# pvesm add lvmthin vmstore --thinpool vmstore --vgname vg --nodes pve03
```

4.4.4.11 iSCSI

iSCSI (Internet Small Computer System Interface) – широко применяемая технология, используемая для подключения к серверам хранения.

Данное хранилище поддерживает общие свойства (content, nodes, disable) хранилищ, и дополнительно используются следующие свойства:

- portal – IP-адрес или DNS-имя сервера iSCSI;
- target – цель iSCSI.

Пример файла конфигурации (`/etc/pve/storage.cfg`):

```
iscsi: test1-iSCSI
       portal 192.168.0.105
       target iqn.2021-7.local.omv:test
       content images
```

Возможные типы содержимого: `images` (образ виртуального диска в формате raw).

iSCSI является типом хранилища блочного уровня и не предоставляет интерфейса управления. Поэтому обычно лучше экспортировать один большой LUN и установить LVM поверх этого LUN.

Примечание. Если планируется использовать LVM поверх iSCSI, имеет смысл установить:

```
content none
```

В этом случае нельзя будет создавать ВМ с использованием iSCSI LUN напрямую.

Примечание. Для работы с устройством, подключенным по интерфейсу iSCSI, на всех узлах необходимо выполнить команду (должен быть установлен пакет open-iscsi):

```
# systemctl enable --now iscsid
```

На Рис. 103 показано добавление адресата iSCSI с именем test1-iSCSI, который настроен на удаленном узле 192.168.0.105.

*Добавление хранилища «*iSCSI*»*

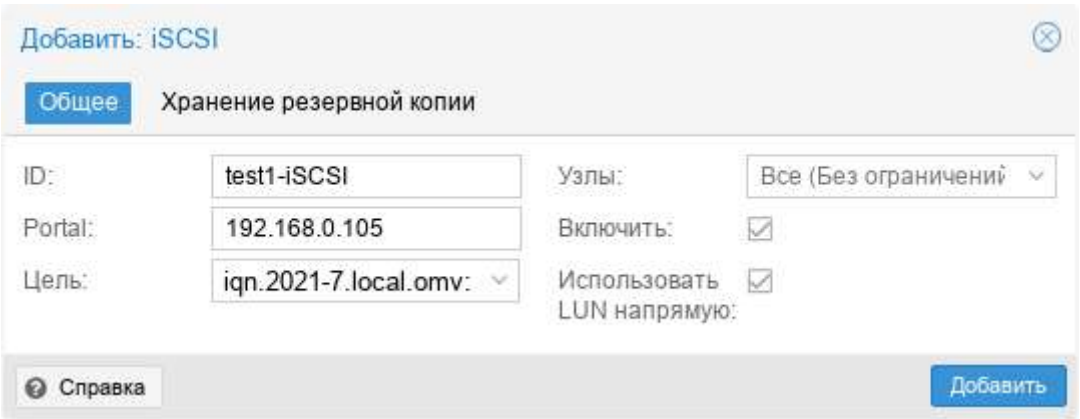


Рис. 103

Для возможности использования LVM на iSCSI необходимо снять отметку с пункта «Использовать LUN напрямую».

Посмотреть доступные для подключения iSCSI цели:

```
# pvesm scan iscsi <IP-адрес сервера[:порт]>
```

Команда создания хранилища iSCSI:

```
# pvesm add iscsi <ID> --portal <Сервер iSCSI> --target <Цель iSCSI> --content none
```

4.4.4.12 iSCSI/ libiscsi

Это хранилище обеспечивает в основном ту же функциональность, что и iSCSI, но использует библиотеку пользовательского уровня. Так как при этом не задействованы драйверы ядра, то это можно рассматривать как оптимизацию производительности. Но поверх такого iSCSI LUN нельзя использовать LVM (управлять распределением пространства необходимо на стороне сервера хранения).

Примечание. Для использования этого хранилища должен быть установлен пакет libiscsi.

Данное хранилище поддерживает все свойства хранилища iscsi.

Пример файла конфигурации (`/etc/pve/storage.cfg`):

```
iscsidirect: test1-iSCSI
      portal 192.168.0.105
      target iqn.2021-7.local.omv:test
```

Возможные типы содержимого: images (образ виртуального диска в формате raw).

4.4.4.13 Ceph RBD

Хранилище RBD (Rados Block Device) предоставляется распределенной системой хранения Ceph. По своей архитектуре Ceph является распределенной системой хранения. Хранилище RBD может содержать только форматы образов `.raw`.

Данное хранилище поддерживает общие свойства (content, nodes, disable) хранилищ, и дополнительно используются следующие свойства:

- monhost – список IP-адресов демона монитора (только если Ceph не работает на кластере PVE);
- pool – название пула Ceph (rbd);
- username – идентификатор пользователя Ceph (только если Ceph не работает на кластере PVE);
- krbd – обеспечивает доступ к блочным устройствам rados через модуль ядра krbd (опционально).

Примечание. Контейнеры будут использовать krbd независимо от значения параметра krbd.

Пример файла конфигурации (`/etc/pve/storage.cfg`):

```
rbd: new
        content images
        krbd 0
        monhost 192.168.0.105
        pool rbd
        username admin
```

Возможные типы содержимого: `rootdir` (данные контейнера), `images` (образ виртуального диска в формате raw).

На Рис. 104 показано добавление хранилища RBD.

*Добавление хранилища «*RBD*»*

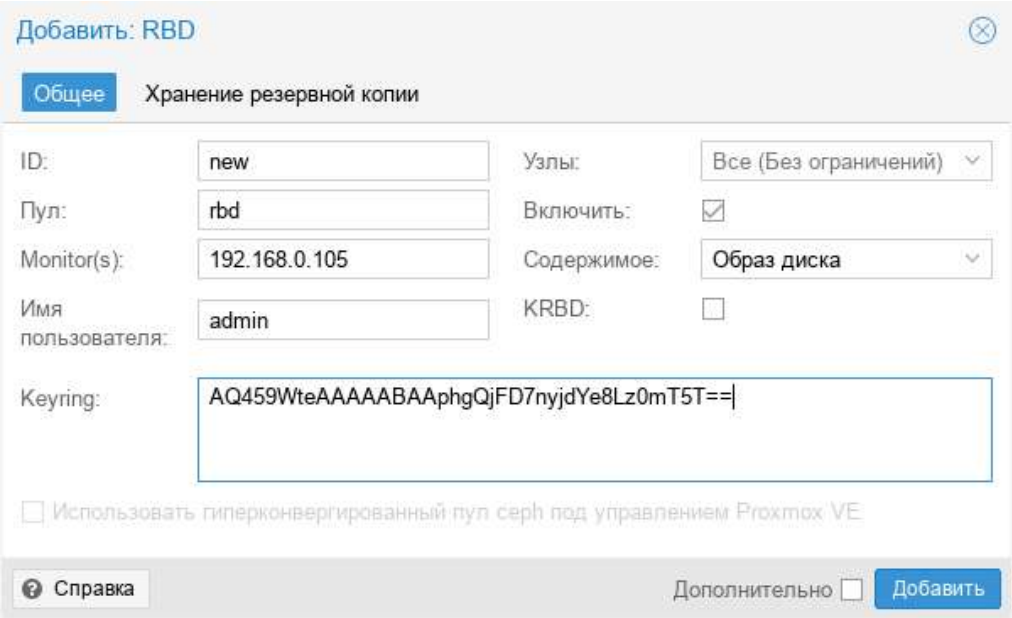


Рис. 104

Если используется аутентификация cephx, которая включена по умолчанию, необходимо предоставить связку ключей из внешнего кластера Ceph.

При настройке хранилища в командной строке, предварительно следует сделать доступным файл, содержащий связку ключей. Один из способов – скопировать файл из внешнего кластера Ceph непосредственно на один из узлов PVE. Например, скопировать файл в каталог /root узла:

```
# scp <external cephserver>:/etc/ceph/ceph.client.admin.keyring /root/rbd.keyring
```

Команда настройки внешнего хранилища RBD:

```
# pvesm add rbd <name> --monhost "10.1.1.20 10.1.1.21 10.1.1.22" \
--content images --keyring /root/rbd.keyring
```

При настройке внешнего хранилища RBD в графическом интерфейсе, связку ключей можно указать в поле «Keyring».

Связка ключей будет храниться в файле `/etc/pve/priv/ceph/<STORAGE_ID>.keyring`.

Добавление хранилища RBD, используещего пул Ceph под управлением PVE (см. «Кластер Ceph») показано на Рис. 105. Связка ключей в этом случае будет скопирована автоматически.

*Добавление хранилища «*RBD*»*

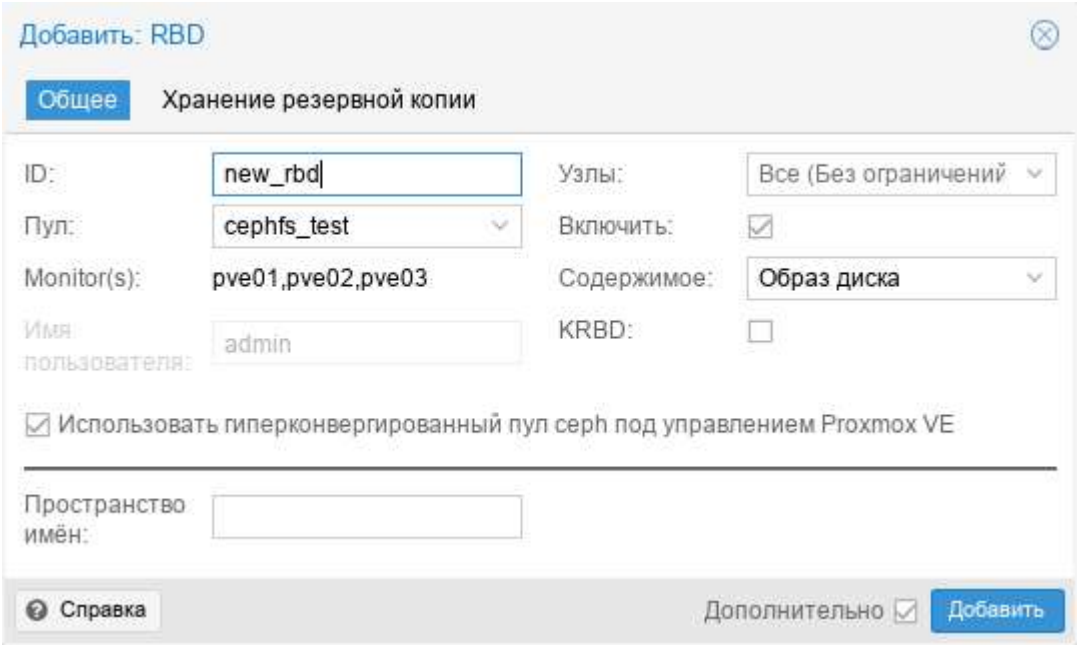


Рис. 105

4.4.4.14 CephFS

CephFS реализует POSIX-совместимую файловую систему, использующую кластер хранения Ceph для хранения своих данных. Поскольку CephFS основывается на Ceph, он разделяет большинство свойств, включая избыточность, масштабируемость, самовосстановление и высокую доступность.

Примечание. PVE может управлять настройками Ceph (см. «Кластер Ceph»), что упрощает настройку хранилища CephFS. Поскольку современное оборудование предлагает большую вычислительную мощность и оперативную память, запуск служб хранения и ВМ на одном узле возможен без существенного влияния на производительность.

Данное хранилище поддерживает общие свойства (content, nodes, disable) хранилищ, и дополнительно используются следующие свойства:

- monhost – список IP-адресов демона монитора (только если Ceph не работает на кластере PVE);
- path – локальная точка монтирования (по умолчанию используется `/mnt/pve/<STORAGE_ID>/`);
- username – идентификатор пользователя (только если Ceph не работает на кластере PVE);
- subdir – подкаталог CephFS для монтирования (по умолчанию /);
- fuse – доступ к CephFS через FUSE (по умолчанию 0).

Пример файла конфигурации (`/etc/pve/storage.cfg`):

```
cephfs: cephfs-external
        content backup,images
        monhost 192.168.0.105
        path /mnt/pve/cephfs-external
        username admin
```

Возможные типы содержимого: vztmpl (шаблон контейнера), iso (ISO-образ), backup (резервная копия), snippets (фрагменты).

На Рис. 106 показано добавление хранилища CephFS.

*Добавление хранилища «*CephFS*»*

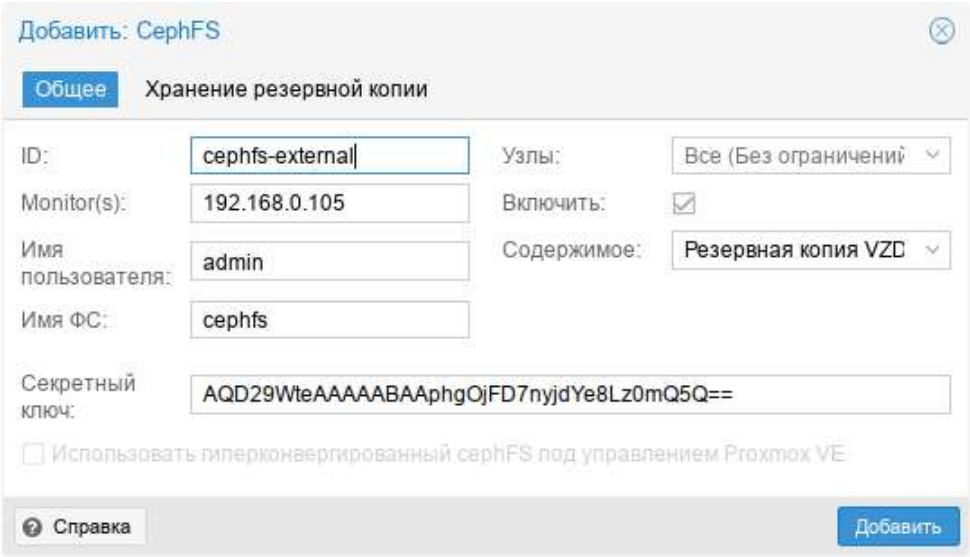


Рис. 106

Примечание. Получить список доступных cephfs, для указания в поле «Имя ФС», можно с помощью команды:

```
# ceph fs ls
```

Если используется аутентификация cephx, которая включена по умолчанию, необходимо указать ключ из внешнего кластера Ceph.

При настройке хранилища в командной строке, предварительно следует сделать файл с ключом доступным. Один из способов – скопировать файл из внешнего кластера Ceph непосредственно на один из узлов PVE. Например, скопировать файл в каталог /root узла:

```
# scp <external cephserver>:/etc/ceph/cephfs.secret /root/cephfs.secret
```

Команда настройки внешнего хранилища CephFS:

```
# pvesm add cephfs <name> --monhost "10.1.1.20 10.1.1.21 10.1.1.22" \
--content backup --keyring /root/cephfs.secret
```

При настройке внешнего хранилища CephFS в графическом интерфейсе, связку ключей можно указать в поле « Секретный ключ».

Связка ключей будет храниться в файле /etc/pve/priv/ceph/<STORAGE_ID>.secret.

Ключ можно получить из кластера Ceph (как администратор Ceph), выполнив команду:

```
# ceph auth get-key client.userid > cephfs.secret
```

4.4.4.15 Proxmox Backup Server

«Proxmox Backup Server» – позволяет напрямую интегрировать сервер резервного копирования Proxmox в PVE.

Данное хранилище поддерживает только резервные копии, они могут быть на уровне блоков (для ВМ) или на уровне файлов (для контейнеров).

Серверная часть поддерживает все общие свойства хранилищ, кроме флага «общее» («shared»), который всегда установлен. Кроме того, для «Proxmox Backup Server» доступны следующие специальные свойства:

- server – IP-адрес или DNS-имя сервера резервного копирования;
- username – имя пользователя на сервере резервного копирования (например, root@pam, backup_u@pbs);
- password – пароль пользователя. Значение будет сохранено в файле `/etc/pve/priv/storage/<STORAGE-ID>.pw`, доступном только суперпользователю;
- datastore– идентификатор хранилища на сервере резервного копирования;
- fingerprint – отпечаток TLS-сертификата API Proxmox Backup Server. Требуется, если сервер резервного копирования использует самоподписанный сертификат. Отпечаток можно получить в веб-интерфейсе сервера резервного копирования или с помощью команды `proxmox-backup-manager cert info`;
- encryption-key – ключ для шифрования резервной копии. В настоящее время поддерживаются только те файлы, которые не защищены паролем (без функции получения ключа (kdf)). Ключ будет сохранен в файле `/etc/pve/priv/storage/<STORAGE-ID>.enc`, доступном только суперпользователю. Опционально;
- master-pubkey – открытый ключ RSA, используемый для шифрования копии ключа шифрования (encryption-key) в рамках задачи резервного копирования. Зашифрованная копия будет добавлена к резервной копии и сохранена на сервере резервного копирования для целей восстановления. Опционально, требуется encryption-key.

Пример файла конфигурации (`/etc/pve/storage.cfg`):

```
pbs: pbs_backup
        datastore store2
        server 192.168.0.123
        content backup
        fingerprint 42:5d:29:20:…:d1:be:bc:c0:c0:a9:9b:b1:a8:1b
        prune-backups keep-all=1
        username root@pam
```

На Рис. 107 показано добавление хранилища «Proxmox Backup Server» с именем pbs_backup с удаленного сервера 192.168.0.123.

Добавление хранилища «Proxmox Backup pve-storage-add-backup»

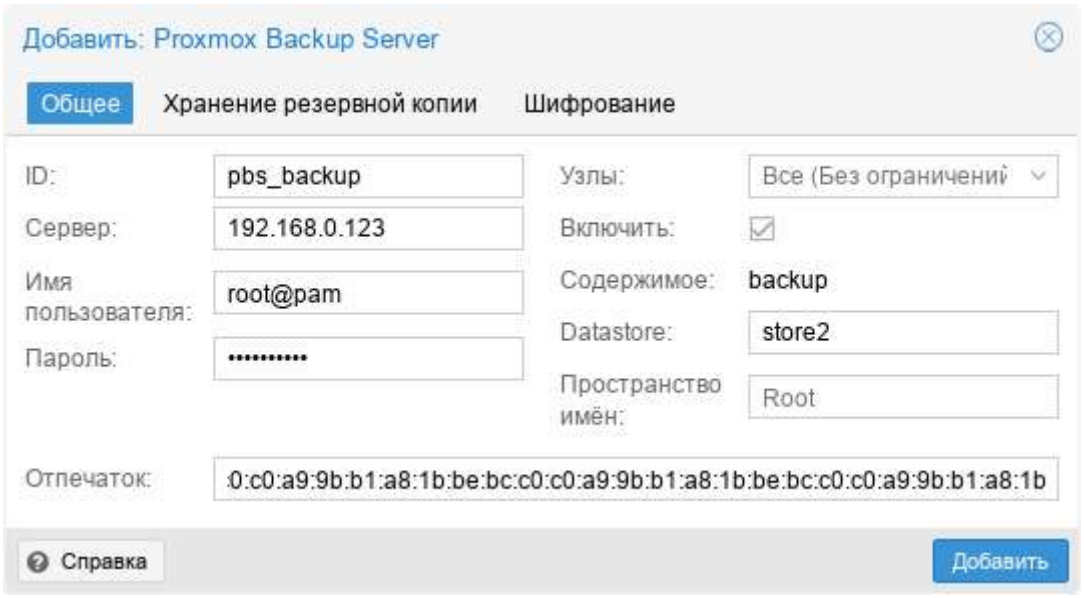


Рис. 107

Добавление хранилища Proxmox Backup в командной строке:

```
# pvesm add pbs pbs_backup --server 192.168.0.123 \
--datastore store2 --username root@pam \
--fingerprint 42:5d:29:…:c0:a9:b1:a8:1b -password
```

4.4.4.15.1 Шифрование

На стороне клиента можно настроить шифрование с помощью AES-256 в режиме GCM. Шифрование можно настроить либо через веб-интерфейс (Рис. 108 – Рис. 110), либо в командной строке с помощью опции `encryption-key`.

Сгенерировать клиентский ключ шифрования

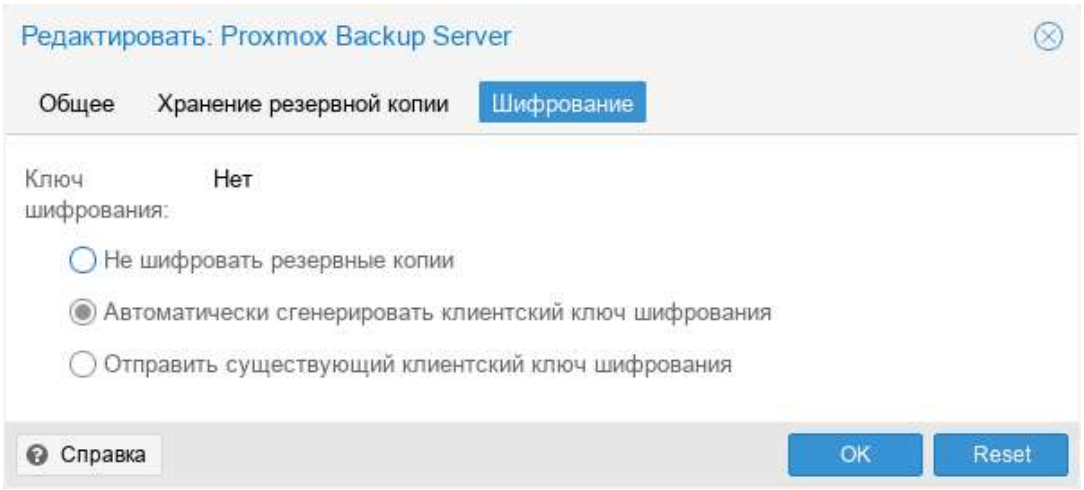


Рис. 108

Сохранить ключ шифрования

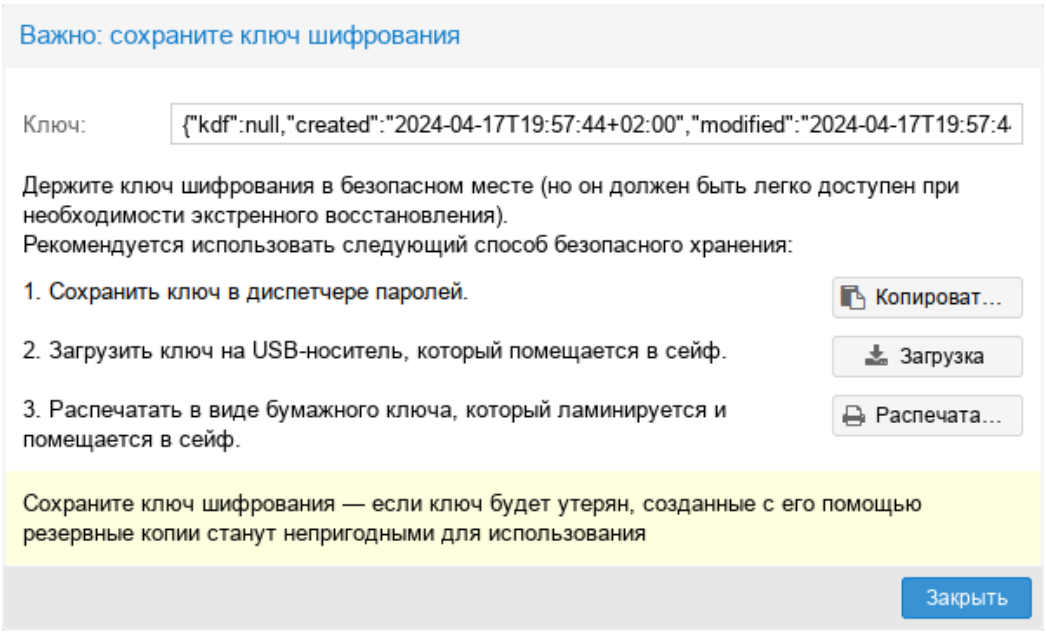


Рис. 109

Ключ будет сохранен в файле /etc/pve/priv/storage/<STORAGE-ID>.enc, который доступен только пользователю root.

Для работы с ключами в командной строке используется команда `proxmox-backup-client key`. Например, сгенерировать ключ:

```
# proxmox-backup-client key create --kdf none <path>
```

Сгенерировать мастер-ключ:

```
# proxmox-backup-client key create-master-key
```

Используется клиентское шифрование

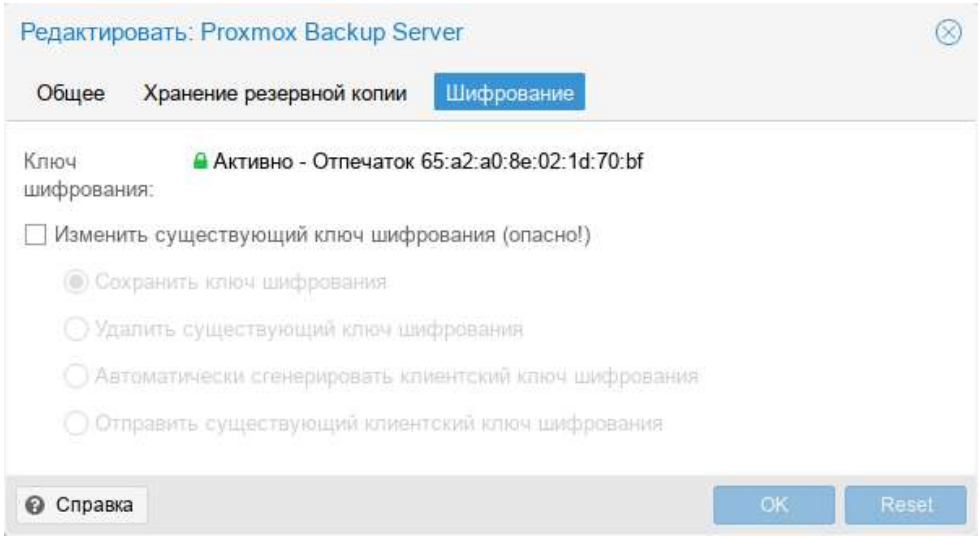


Рис. 110

Примечание. Без ключа шифрования резервные копии будут недоступны. Следует хранить ключи в месте, отдельном от содержимого резервной копии.

Рекомендуется хранить ключ в безопасносном, но при этом легкодоступном месте, чтобы можно было быстро восстановить его после сбоев. Лучшее место для хранения ключа шифрования – менеджер паролей. В качестве резервной копии также следует сохранить ключ на USB-накопитель. Таким образом, ключ будет отсоединен от любой системы, но его по-прежнему легко можно будет восстановить в случае возникновения чрезвычайной ситуации.

Кроме того, можно использовать пару мастер-ключей RSA для целей восстановления ключей: для этого необходимо настроить всех клиентов, выполняющих зашифрованное резервное копирование, на использование одного открытого мастер-ключа, и все последующие зашифрованные резервные копии будут содержать зашифрованную RSA копию использованного ключа шифрования AES. Соответствующий закрытый мастер-ключ позволяет восстановить ключ AES и расшифровать резервную копию, даже если клиентская система больше не доступна.

Примечание. К паре мастер-ключей применяются те же правила хранения, что и к обычным ключам шифрования. Без копии закрытого ключа восстановление невозможно!

Примечание. Не следует использовать шифрование, если от него нет никакой пользы, например, если сервер запущен локально в доверенной сети. Восстановить данные из незашифрованных резервных копий всегда проще.

4.4.5 FC/iSCSI SAN

Данная конфигурация предполагает, что узлы кластера имеют доступ к устройствам хранения (LUN), экспортированным сервером сети хранения данных (SAN) с использованием протокола iSCSI или Fibre Channel (FC).

Соединение узла PVE к хранилищу называется путем. Если в подсистеме хранения данных существует несколько путей к устройству хранения данных (LUN), это называется многопутевым подключением (multipath). Необходимо использовать как минимум два выделенных сетевых адаптера для iSCSI/FC, использующих отдельные сети (и коммутаторы для защиты от сбоев коммутатора).

На узле PVE диск будет виден как локальный диск по пути /dev/mapper/mpathX или /dev/mapper/[WWNID] (в зависимости от настроек в multipath.conf).

Примечание. В данном разделе приведена общая информация. Для получения информации о настройках конкретной СХД следует обратиться к документации производителя хранилища.

4.4.5.1 Особенности подключения СХД по ISCSI

Все необходимые соединения iSCSI должны запускаться во время загрузки узла. Сделать это можно, установив для параметра node.startup значение «automatic». Значение по умолчанию «node.session.timeo.replacement_timeout» составляет 120 секунд. Рекомендуется использовать значение – 15 секунд.

Эти параметры можно указать в файле в `/etc/iscsi/iscsid.conf` (по умолчанию). Если iSCSI target уже подключен, то необходимо изменить настройки по умолчанию для конкретной цели в файле `/etc/iscsi/nodes/<TARGET>/<PORTAL>/default`.

На всех узлах PVE:

1) установить пакет open-iscsi, запустить и добавить в автозагрузку сервис iscsid:

```
# apt-get install open-iscsi
# systemctl enable --now iscsid
```

2) указать следующие параметры в файле /etc/iscsi/iscsid.conf:

```
node.startup = automatic
node.session.timeo.replacement_timeout = 15
```

3) присоединить iSCSI хранилище к кластеру:

```
# iscsiadm -m discovery -t sendtargets -p <iscsi-target-1-ip>
# iscsiadm -m discovery -t sendtargets -p <iscsi-target-2-ip>
# iscsiadm -m node --login
```

4) настроить автоматическое подключение iSCSI-target-ов. Для этого необходимо поменять следующие параметры:

- в файле /etc/iscsi/iscsid.conf:

```
node.startup = automatic
```

- в файлах /var/lib/iscsi/send_targets/<TargetServer>,<Port>/st_config:

```
discovery.sendtargets.use_discoveryd = Yes
```

После перезагрузки должны появиться подключенные устройства, например:

```
# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda 8:0 0 59G 0 disk
sdb 8:16 0 931,3G 0 disk
└─mpatha 253:0 0 931,3G 0 mpath
sdc 8:32 0 931,3G 0 disk
└─mpatha 253:0 0 931,3G 0 mpath
sdd 8:48 0 931,3G 0 disk
└─mpatha 253:0 0 931,3G 0 mpath
sde 8:64 0 931,3G 0 disk
└─mpatha 253:0 0 931,3G 0 mpath
```

В данном примере один LUN на 1000GB виден по четырем путям.

П р и м е ч а н и е . Примеры использования команды `iscsiadm`:

- отключить хранилище (отключить все цели):

```
# iscsiadm -m node --logout
```

- отключить только определенную цель:

```
# iscsiadm -m node --targetname "iscsi-target-1.test.alt:server.target1" --logout
```

- переопросить устройства на ISCSI:

```
# iscsiadm -m node -R
```

- посмотреть все текущие подключения iSCSI:

```
# iscsiadm -m session
```

4.4.5.2 Особенности подключения СХД по FC

Алгоритм подключения:

1) подготовить СХД (создать LUNы);
2) на сервере установить FC HBA, драйверы к ним;
3) настроить сетевое подключение;
4) подключить СХД к серверу;
5) предоставить серверу доступ к СХД по WWPN (провести регистрацию сервера на полке).

Примечание. Для того чтобы узнать глобальные имена портов (WWPN), можно воспользоваться утилитой systool из пакета sysfsutils.

Пакет sysfsutils необходимо установить (из репозитория):

```
# apt-get install sysfsutils
```

Чтобы найти один или несколько WWPN, следует ввести следующую команду:

```
# systool -c fc_host -A port_name
Class = "fc_host"
Class Device = "host1"
port_name = "0x10000090fa59a61a"
Device = "host1"
Class Device = "host16"
port_name = "0x10000090fa59a61b"
Device = "host16"
```

Просмотреть список подключенных устройств можно, например, выполнив команду:

```
# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
sda 8:0 0 59G 0 disk
sdb 8:16 0 931,3G 0 disk
└─mpatha 253:0 0 931,3G 0 mpath
sdc 8:32 0 931,3G 0 disk
└─mpatha 253:0 0 931,3G 0 mpath
sdd 8:48 0 931,3G 0 disk
└─mpatha 253:0 0 931,3G 0 mpath
sde 8:64 0 931,3G 0 disk
└─mpatha 253:0 0 931,3G 0 mpath
```

В данном примере один LUN на 1000GB виден по четырем путям.

4.4.5.3 Настройка multipath

Многопутевой ввод-вывод (Multipath I/O) – технология подключения узлов СХД с использованием нескольких маршрутов. В случае отказа одного из контроллеров, ОС будет использовать другой для доступа к устройству. Это повышает отказоустойчивость системы и позволяет распределять нагрузку.

Multipath устройства объединяются в одно устройство с помощью специализированного программного обеспечения в новое устройство. Multipath обеспечивает выбор пути и переключение на новый маршрут при отказе текущего. Кроме того multipath позволяет увеличить пропускную способность за счет балансировки нагрузки.

На узлах PVE должен быть установлен пакет для multipath:

```
# apt-get install multipath-tools
```

И запущена служба multipathd:

```
# systemctl enable --now multipathd && sleep 5; systemctl status multipathd
```

Примечание. Команда multipath используется для обнаружения и объединения нескольких путей к устройствам. Некоторые параметры команды multipath:

- –l – отобразить текущую multipath-топологию, полученную из sysfs и устройства сопоставления устройств;
- -ll – отобразить текущую multipath-топологию, собранную из sysfs, устройства сопоставления устройств и всех других доступных компонентов системы;
- -f device – удалить указанное multipath-устройство;
- -F – удалить все неиспользуемые multipath-устройства;
- -w device – удалить WWID указанного устройства из файла wwids;
- -W – сбросить файл wwids, чтобы включить только текущие многопутевые устройства;
- -r – принудительная перезагрузка multipath-устройства.

После подключения, устройство хранения данных должно автоматически определиться как multipath-устройство:

```
# multipath -ll
mpatha (3600c0ff00014f56ee9f3cf6301000000) dm-0 HP,P2000 G3 FC
size=931G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=50 status=active
| |- 1:0:0:1 sdb 8:16 active ready running
| `- 16:0:1:1 sde 8:64 active ready running
`-+- policy='service-time 0' prio=10 status=enabled
|- 1:0:1:1 sdc 8:32 active ready running
`- 16:0:0:1 sdd 8:48 active ready runnin
```

Вывод этой команды разделить на три части:

- информация о multipath-устройстве:

  - mpatha (3600c0ff00014f56ee9f3cf6301000000): алиас
  - dm-0: имя устройства dm
  - HP,P2000 G3 FC: поставщик, продукт
  - size=931G: размер
  - features='1 queue_if_no_path': функции
  - hwhandler='01 alua': аппаратный обработчик
  - wp=rw: права на запись
- информация о группе путей:
  - policy='service-time 0': политика планирования
  - prio=50: приоритет группы путей
  - status=active: статус группы путей
- информация о пути:
  - 7:0:1:1: хост:канал:идентификатор:Lun
  - sde: диск
  - 8:80: номера major:minor
  - active: статус dm
  - ready: статус пути
  - running: online статус

Для получения дополнительной информации об используемых устройствах можно выполнить команду:

```
# multipath -v3
```

Настройки multipath содержатся в файле /etc/multipath.conf:

```
defaults {
    find_multipaths         yes
    user_friendly_names     yes
}
```

Если для параметра user_friendly_names установлено значение «no», то для имени multipath-устройства задается значение World Wide Identifier (WWID). Имя устройства будет /dev/mapper/WWID и /dev/dm-X:

```
# ls /dev/mapper/
3600c0ff00014f56ee9f3cf6301000000

# lsblk
NAME                                  MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINTS
sda                                     8:0    0    59G  0 disk
sdb                                     8:16   0 931,3G  0 disk
└─3600c0ff00014f56ee9f3cf6301000000   253:0    0 931,3G  0 mpath
```

```
sdc                                       8:32   0 931,3G  0 disk
└─3600c0ff00014f56ee9f3cf6301000000    253:0    0 931,3G  0 mpath
sdd                                       8:48   0 931,3G  0 disk
└─3600c0ff00014f56ee9f3cf6301000000    253:0    0 931,3G  0 mpath
sde                                       8:64   0 931,3G  0 disk
└─3600c0ff00014f56ee9f3cf6301000000    253:0    0 931,3G  0 mpath
```

Если для параметра user_friendly_names установлено значение «yes», то для имени multipath-устройства задаётся алиас (псевдоним), в форме mpathX. Имя устройства будет /dev/mapper/mpathX и /dev/dm-X:

```
# ls /dev/mapper/
mpatha

# lsblk
NAME             MAJ:MIN RM   SIZE RO TYPE  MOUNTPOINTS
sda                8:0    0    59G  0 disk
sdb                8:16   0 931,3G  0 disk
└─mpatha         253:0    0 931,3G  0 mpath
sdc                8:32   0 931,3G  0 disk
└─mpatha         253:0    0 931,3G  0 mpath
sdd                8:48   0 931,3G  0 disk
└─mpatha         253:0    0 931,3G  0 mpath
sde                8:64   0 931,3G  0 disk
└─mpatha         253:0    0 931,3G  0 mpath
```

Однако не гарантируется, что имя устройства будет одинаковым на всех узлах, использующих это multipath-устройство.

ОС при загрузке определяет пути к устройствам в изменяющейся среде выполнения (например, при новой загрузке в среде выполнения ОС появились новые устройства хранения или исчезли старые, и т.п.) по отношению к предыдущей загрузке или по отношению к заданной ранее конфигурации. Это может приводить к противоречиям при именовании устройств. Для того чтобы избежать такого поведения, рекомендуется:

- сделать явное исключение для устройства (раздела) хранения (например, для 3600c0ff00014f56ee9f3cf6301000000, которое в настоящее время определяется как /dev/mapper/mpatha). Для этого в файл /etc/multipath.conf добавить секции:

```
blacklist {
        wwid .*
}

blacklist_exceptions {
        wwid "3600c0ff00014f56ee9f3cf6301000000"
```

```
}
```

Данная настройка предписывает внести в черный список любые найденные устройства хранения данных, за исключением нужного.

- создать еще одну секцию:

```
multipaths {
  multipath {
        wwid "3600c0ff00014f56ee9f3cf6301000000"
        alias mpatha
  }
}
```

В этом случае устройство всегда будет доступно только по имени /dev/mapper/mpatha. Вместо mpatha можно вписать любое желаемое имя устройства.

Примечание. Получить WWID конкретного устройства можно, выполнив команду (для устройств в одном multipath WWID будут совпадать):

```
# /lib/udev/scsi_id -g -u -d /dev/sdb
```

В файл /etc/multipath.conf может также потребоваться внести рекомендованные производителем СХД параметры.

После внесения изменений в файл /etc/multipath.conf необходимо перезапустить службу multipathd для активации настроек:

```
# systemctl restart multipathd.service
```

Примечание. Проверить файл /etc/multipath.conf на наличие ошибок можно, выполнив команду:

```
# multipath -t
```

4.4.5.4 Multipath в веб-интерфейсе PVE

Диски, подключенные по mulipath, можно увидеть в веб-интерфейсе PVE (Рис. 111).

Multipath в веб-интерфейсе PVE

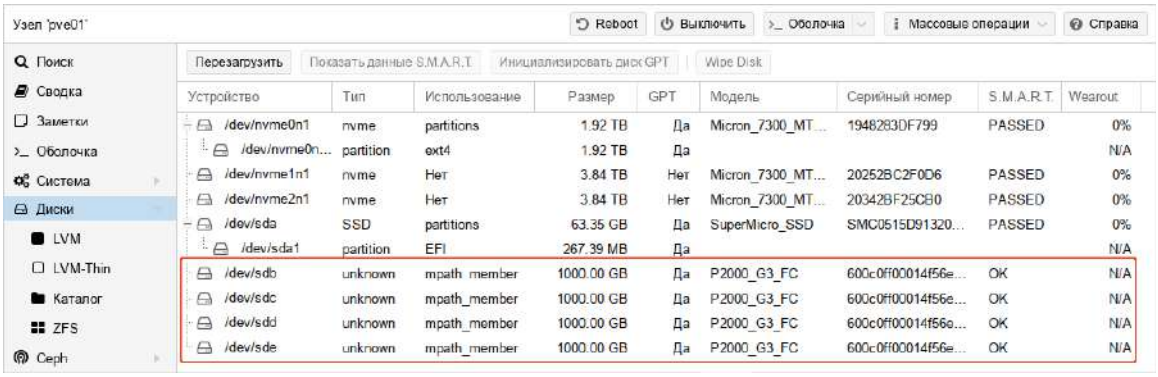


Рис. 111

Поверх хранилища на базе iSCSI или FC можно использовать LVM или хранилище файлового типа (если нужны снапшоты).

4.4.5.5 Файловые системы на multipath

На multipath-устройстве можно создать файловую систему (ФС) и подключить его как хранилище типа «Каталог» в PVE. Можно использовать любую ФС, но при использовании, например, ext4 или xfs, хранилище нельзя будет использовать как общее. Для возможности совместного использования хранилища необходимо использовать кластерную ФС.

Ниже показано создание кластерной ФС ocfs2 на multipath-устройстве и подключение этого устройства.

4.4.5.5.1 Кластерная ФС ocfs2

На всех узлах кластера необходимо установить пакет `ocfs2-tools`:

```
# apt-get install ocfs2-tools
```

Примечание. Основной конфигурационный файл для OCFS2 – /etc/ocfs2/cluster.conf. Этот файл должен быть одинаков на всех узлах кластера, при изменении в одном месте его нужно скопировать на остальные узлы. При добавлении нового узла в кластер, описание этого узла должно быть добавлено на всех остальных узлах до монтирования раздела ocfs2 с нового узла.

Создание кластерной конфигурации возможно с помощью команд или с помощью редактирования файла конфигурации /etc/ocfs2/cluster.conf.

Пример создания кластера из трёх узлов:

- в командной строке:
  - создать кластер с именем mycluster:

```
# o2cb_ctl -C -n mycluster -t cluster -a name=mycluster
```

  - добавить узелы, выполнив команду для каждого:

```
# o2cb_ctl -C -n <имя_узла> -t node -a number=0 -a ip_address=<IP_узла> -a
ip_port=7777 -a cluster=mycluster
```

- редактирование конфигурационного файла /etc/ocfs2/cluster.conf:

```
cluster:
node_count = 3
heartbeat_mode = local
name = mycluster

node:
ip_port = 7777
ip_address = <IP_узла-01>
number = 0
name = <имя_узла-01>
cluster = mycluster

node:
ip_port = 7777
```

```
ip_address = <IP_узла-02>
number = 1
name = <имя_узла-02>
cluster = mycluster

node:
ip_port = 7777
ip_address = <IP_узла-03>
number = 2
name = <имя_узла-03>
cluster = mycluster
```

Примечание. Имя узла кластера должно быть таким, как оно указано в файле /etc/hostname.

Для включения автоматической загрузки сервиса OCFS2 можно использовать скрипт /etc/init.d/o2cb:

```
# /etc/init.d/o2cb configure
```

Для ручного запуска кластера нужно выполнить:

```
# /etc/init.d/o2cb load
checking debugfs...
Loading filesystem "ocfs2_dlmfs": OK
Creating directory '/dlm': OK
Mounting ocfs2_dlmfs filesystem at /dlm: OK
# /etc/init.d/o2cb online mycluster
checking debugfs...
Setting cluster stack "o2cb": OK
Registering O2CB cluster "mycluster": OK
Setting O2CB cluster timeouts : OK
```

Далее на одном из узлов необходимо создать раздел OCFS2, для этого следует выполнить следующие действия:

- создать физический раздел /dev/mapper/mpatha-part1 на диске /dev/mapper/mpatha:

```
# fdisk /dev/mapper/mpatha
```

- отформатировать созданный раздел, выполнив команду:

```
# mkfs.ocfs2 -b 4096 -C 4k -L DBF1 -N 3 /dev/mapper/mpatha-part1
mkfs.ocfs2 1.8.7
Cluster stack: classic o2cb
Label: DBF1
…
mkfs.ocfs2 successful
```

Описание параметров команды mkfs.ocfs2 приведено в табл. 8.

Примечание. Для создания нового раздела может потребоваться предварительно удалить существующие данные раздела на устройстве /dev/mpathX (следует использовать с осторожностью!):

```
# dd if=/dev/zero of=/dev/mapper/mpathX bs=512 count=1 conv=notrunc
```

4.4.5.5.2 OCFS2 в PVE

На каждом узле PVE необходимо смонтировать раздел с ФС OCFS2 (например, в /mnt/ocfs2):

```
# mkdir /mnt/ocfs2
# mount /dev/mapper/mpatha-part1 /mnt/ocfs2
```

Для автоматического монтирования раздела следует добавить в файл /etc/fstab строку (каталог /mnt/ocfs2 должен существовать):

```
/dev/mapper/mpatha-part1 /mnt/ocfs2 ocfs2 _netdev,defaults 0 0
```

Выполнить проверку монтирования:

```
# mount -a
```

Результатом выполнения команды должен быть пустой вывод без ошибок.

Примечание. Опция _netdev позволяет монтировать данный раздел только после успешного старта сетевой подсистемы.

Примечание. Так как имя является символической ссылкой, в некоторых случаях (например, при смене порядка опроса устройств на шине ISCSI) она может меняться (указывая на иное устройство). Поэтому если для устройства хранения не используется алиас, рекомендуется производить автоматическое монтирование этого устройства (раздела) в файле /etc/fstab по его уникальному идентификатору, а не по имени /dev/mapper/mpatha:

```
UUID=<uuid> /<каталог> ocfs2 _netdev,defaults 0 0
```

Например, определить UUID uuid разделов:

```
# blkid
/dev/mapper/mpatha-part1: LABEL="DBF1" UUID="df49216a-a835-47c6-b7c1-6962e9b7dcb6"
BLOCK_SIZE="4096" TYPE="ocfs2" PARTUUID="15f9cd13-01"
```

Добавить монтирование этого UUID в /etc/fstab:

```
UUID=df49216a-a835-47c6-b7c1-6962e9b7dcb6       /mnt/ocfs2     ocfs2 _netdev,defaults
0 0
```

Созданный раздел можно добавить как хранилище в веб-интерфейсе PVE («Центр обработки данных» → «Хранилище», нажать кнопку «Добавить» и в выпадающем меню выбрать пункт «Каталог» Рис. 112) или в командной строке:

```
# pvesm add dir mpath --path /mnt/ocfs2 --shared 1
```

Добавление multipath-устройства

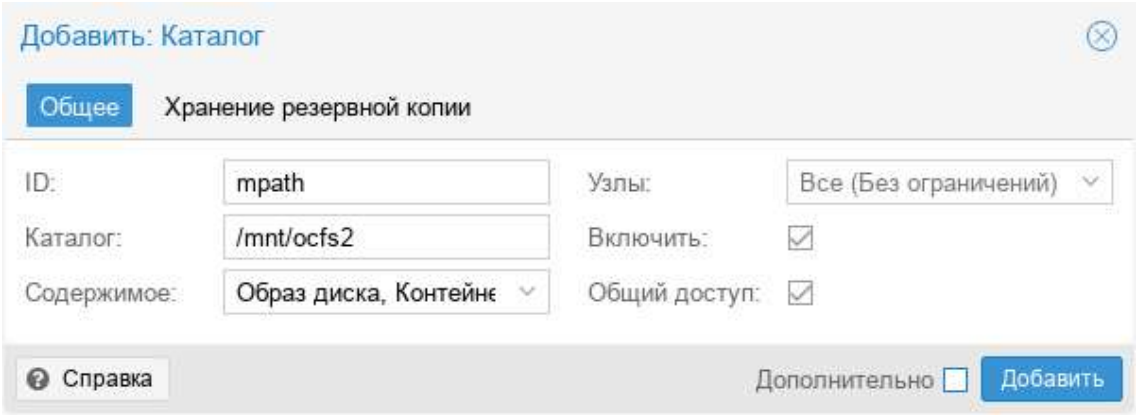


Рис. 112

4.4.5.5.3 LVM/LVM-Thin хранилища на multipath

Примечание. multipath-устройство не отображается в веб-интерфейсе PVE LVM/LVM-Thin, поэтому потребуется использовать командную строку.

Примечание. LVM при запуске сканирует все устройства на предмет поиска конфигурации LVM, и если настроен multipath-доступ, он найдет конфигурацию LVM как на (виртуальных) multipath-устройствах, так и на базовых (физических) дисках. Поэтому рекомендуется создать фильтр LVM для фильтрации физических дисков и разрешить LVM сканировать только multipath-устройства.

Сделать это можно, добавив фильтр в раздел device в файле /etc/lvm/lvm.conf:

```
filter = [ "a|/dev/mapper/|", "a|/dev/sda.*|", "r|.*|" ]
```

В данном примере принимаются только multipath-устройства и /dev/sda.*, все остальные устройства отклоняются:

- a|/dev/mapper/| – принять устройства /dev/mapper (здесь находятся multipath-устройства);
- a|/dev/sda.*| – принять устройство /dev/sda;
- r|.*| – отклонить все остальные устройства.

Пример создания LVM хранилища на multipath:

1) вывести список разделов /dev/mapper/mpatha:

```
# fdisk -l /dev/mapper/mpatha
Disk /dev/mapper/mpatha: 931.32 GiB, 999999995904 bytes, 1953124992 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 1048576 bytes
Disklabel type: dos
Disk identifier: 0x2221951e

Device Boot Start End Sectors Size Id Type
/dev/mapper/mpatha-part1 2048 1953124991 1953122944 931.3G 83 Linux
```

2) создать физический том (PV) на /dev/mapper/mpatha-part1:

```
# pvcreate /dev/mapper/mpatha-part1
Physical volume "/dev/mapper/mpatha-part1" successfully created.
```

3) создать группу томов (VG) с именем VG1:

```
# vgcreate VG1 /dev/mapper/mpatha-part1
Volume group "VG1" successfully created
```

4) показать информацию о физических томах:

```
# pvs
  PV                         VG  Fmt  Attr PSize   PFree
  /dev/mapper/mpatha-part1 VG1 lvm2 a--  931.32g 931.32g
```

5) показать информацию о группах томов:

```
# vgs
  VG  #PV #LV #SN Attr   VSize   VFree
  VG1   1   0   0 wz--n- 931.32g 931.32g
```

4.4.5.5.4 LVM-хранилище

Получить список доступных PVE групп томов:

```
# pvesm lvmscan
VG1
```

Список доступных PVE групп томов можно также увидеть в веб-интерфейсе (Рис. 113).

Список LVM томов

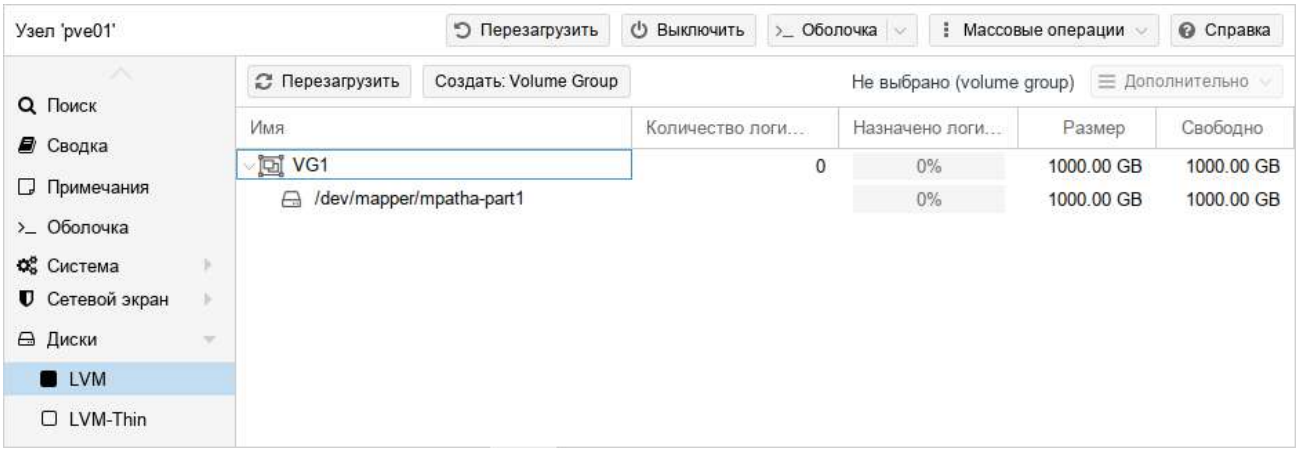


Рис. 113

Пример создания LVM хранилища с именем mpath-lvm:

```
# pvesm add lvm mpath-lvm --vgname VG1 --content images,rootdir
```

Создать LVM хранилище можно в веб-интерфейсе: выбрать «Центр обработки данных» → «Хранилище», нажать кнопку «Добавить» и в выпадающем меню выбрать пункт «LVM» (Рис. 114).

LVM хранилище на multipath

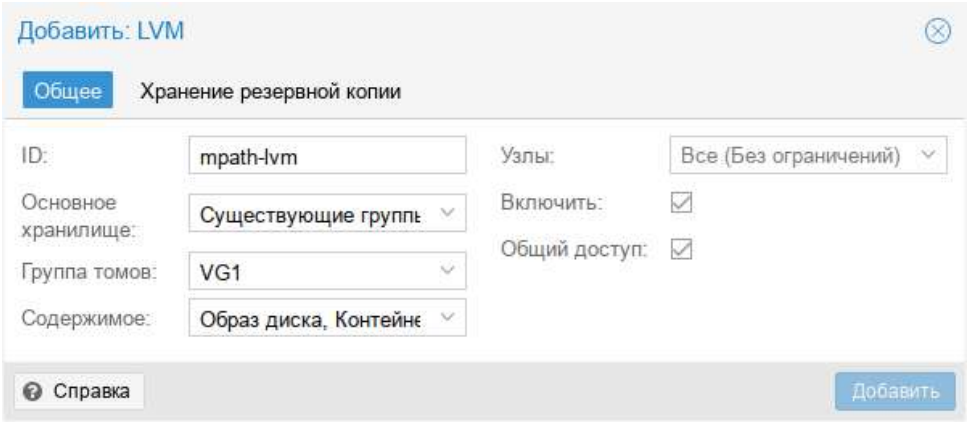


Рис. 114

4.4.5.5.5 LVM-Thin хранилище

Создать тонкий пул LVM на multipath:

1) вывести информацию о физических томах:

```
# pvs
  PV                         VG  Fmt  Attr PSize   PFree
  /dev/mapper/mpatha-part1 VG1 lvm2 a--  931.32g 931.32g
```

2) вывести информацию о группах томов:

```
# vgs
  VG  #PV #LV #SN Attr   VSize   VFree
  VG1   1   0   0 wz--n- 931.32g 931.32g
```

3) создать тонкий пул LVM (размер 100 ГБ) с именем vmstore:

```
# lvcreate -L 100G -T -n vmstore VG1
Logical volume "vmstore" created.
```

Получить список доступных PVE LVM-thin пулов в группе томов VG1:

```
# pvesm lvmthinscan VG1
Vmstore
```

Список доступных PVE LVM-thin пулов можно также увидеть в веб-интерфейсе PVE (Рис. 115).

Список Thinpool

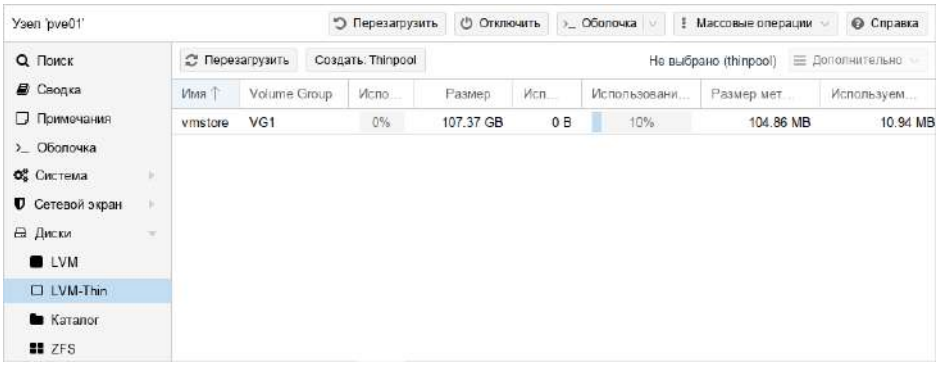


Рис. 115

Пример создания LVM-Thin хранилища с именем mpath-lvmthin:

```
# pvesm add lvmthin mpath-lvmthin --thinpool vmstore --vgname VG1 --nodes pve01
```

Создать LVM-thin хранилище можно в веб-интерфейсе: выбрать «Центр обработки данных» → «Хранилище», нажать кнопку «Добавить» и в выпадающем меню выбрать пункт «LVM-Thin» (Рис. 116).

LVM-Thin хранилище на multipath

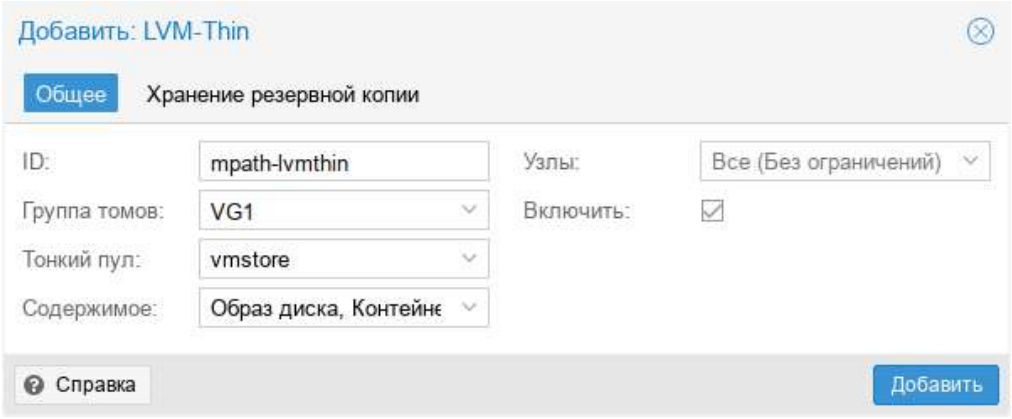


Рис. 116

4.4.5.6 Изменение размера multipath-устройства

Для изменения размера multipath-устройства необходимо:

- изменить размер физического устройства;
- определить пути к номеру логического устройства (LUN):

```
# multipath -l
mpatha (3600c0ff00014f56ee9f3cf6301000000) dm-0 HP,P2000 G3 FC
size=465G features='1 queue_if_no_path' hwhandler='1 alua' wp=rw
|-+- policy='service-time 0' prio=0 status=active
| |- 1:0:1:1  sdc 8:32 active undef running
| `- 16:0:1:1 sde 8:64 active undef running
`-+- policy='service-time 0' prio=0 status=enabled
  |- 1:0:0:1  sdb 8:16 active undef running
  `- 16:0:0:1 sdd 8:48 active undef running
```

- изменить размер путей, выполнив команду:

```
# echo 1 > /sys/block/<path_device>/device/rescan
```

Данную команду необходимо выполнить для каждого диска, входящего в multipath-устройство:

```
# echo 1 > /sys/block/sdb/device/rescan
# echo 1 > /sys/block/sdc/device/rescan
# echo 1 > /sys/block/sdd/device/rescan
# echo 1 > /sys/block/sde/device/rescan
```

- убедиться, что ядро увидело новый размер, выполнив команду (Рис. 117):

```
# dmesg -wHT
```

- изменить размер multipath-устройства:

```
# multipathd -k"resize map 3600c0ff00014f56ee9f3cf6301000000"
```

где 3600c0ff00014f56ee9f3cf6301000000 – WWID multipath-устройства;

- изменить размер блочного устройства (если используется LVM):

```
# pvresize /dev/mapper/mpatha
```

- изменить размер файловой системы (если разделы LVM или DOS не используются):

```
# resize2fs /dev/mapper/mpatha
```

Вывод команды dmesg

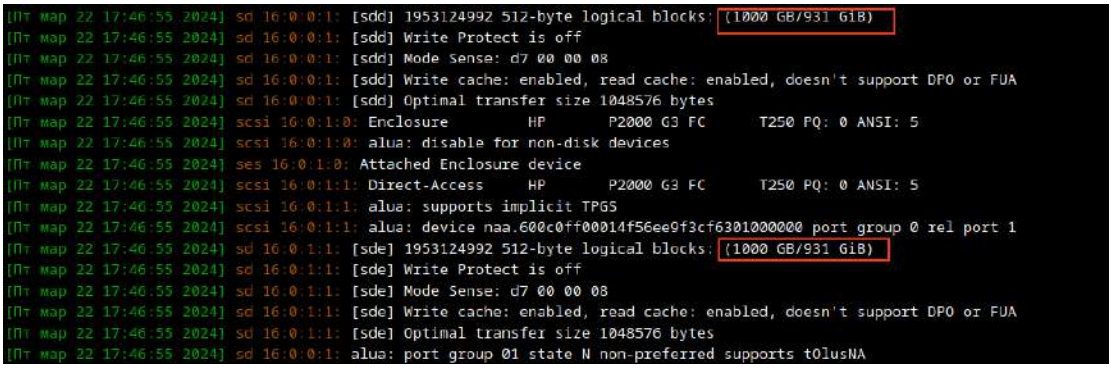


Рис. 117

Примечание. Данную процедуру следует выполнить на каждом PVE-узле, к которому присоединён этот LUN.

4.4.6 Кластер Ceph

PVE позволяет использовать одни и те же физические узлы в кластере как для вычислений (обработка виртуальных машин и контейнеров), так и для реплицированного хранилища.

Ceph – программная объектная отказоустойчивая сеть хранения данных. Она реализует возможность файлового и блочного доступа к данным. Ceph интегрирован в PVE, поэтому запускать и управлять хранилищем Ceph можно непосредственно на узлах гипервизора.

Некоторые преимущества Ceph на PVE:

- простая настройка и управление через веб-интерфейс и командную строку;
- тонкое резервирование;
- поддержка моментальных снимков;
- самовосстановление;
- масштабируемость до уровня эксабайт;
- настройка пулов с различными характеристиками производительности и избыточности;
- данные реплицируются, что делает их отказоустойчивыми;
- работает на стандартном оборудовании;
- нет необходимости в аппаратных RAID-контроллерах;
- открытый исходный код.

pveceph – инструмент для установки и управления службами Ceph на узлах PVE.

Кластерная система хранения данных Ceph состоит из нескольких демонов:

- монитор Ceph (ceph-mon) – хранит информацию о состоянии работоспособности кластера, карту всех узлов и правила распределения данных (карту CRUSH). Мониторы также отвечают за управление аутентификацией между демонами и клиентами. Обычно для обеспечения резервирования и высокой доступности требуется не менее трех мониторов;
- менеджер Ceph (ceph-mgr) – собирает информацию о состоянии со всего кластера. Менеджер Ceph работает вместе с демонами монитора. Он обеспечивает дополнительный мониторинг и взаимодействует с внешними системами мониторинга и управления. Он также включает другие службы. Для обеспечения высокой доступности обычно требуется по крайней мере два менеджера;
- OSD Ceph (ceph-osd; демон хранилища объектов) – демон, управляющий устройствами хранения объектов, которые представляют собой физические или логические единицы хранения (жесткие диски или разделы). Демон дополнительно отвечает за репликацию данных и перебалансировку в случае добавления или удаления узлов. Демоны Ceph OSD взаимодействуют с демонами монитора и предоставляют им состояние других демонов OSD. OSD являются основными демонами кластера, на которые ложится большая часть нагрузки. Для обеспечения резервирования и высокой доступности обычно требуется не менее трех OSD Ceph;
- сервер метаданных Ceph (ceph-mds) – хранит метаданные файловой системы CephFS (блочные устройства Ceph и хранилище объектов Ceph не используют MDS). Разделение метаданных от данных значительно увеличивает производительность кластера. Серверы метаданных Ceph позволяют пользователям файловой системы POSIX выполнять базовые команды (такие как ls, find и т.д.), не создавая огромной нагрузки на кластер хранения Ceph.

4.4.6.1 Системные требования

Чтобы построить гиперконвергентный кластер PVE + Ceph, необходимо использовать не менее трех (предпочтительно) идентичных серверов для настройки.

Требования к оборудованию для Ceph (табл. 13) могут варьироваться в зависимости от размера кластера, ожидаемой нагрузки и других факторов.

Т а б л и ц а 13 – Системные требования к оборудованию для Ceph

Процесс	Критерий	Требования
Монитор Ceph	Процессор	2 ядра минимум, 4 рекомендуется
	ОЗУ	5ГБ+ (большим/производственным кластерам нужно больше)

	Жесткий диск	100 ГБ на демон, рекомендуется SSD
	Сеть	1 Гбит/с (рекомендуется 10+ Гбит/с)
Менеджер Ceph	Процессор	1 ядро
	ОЗУ	1 ГБ
OSD Ceph	Процессор	- 1 ядро минимум, 2 рекомендуется - 1 ядро на пропускную способность 200-500 МБ/с - 1 ядро на 1000-3000 IOPS - SSD OSD, особенно NVMe, выиграют от дополнительных ядер на OSD
	ОЗУ	- 4 ГБ+ на демон (больше – лучше) - 1 ГБ на 1 ТБ данных OSD
	Жесткий диск	1 SSD накопитель на демон
	Сеть	1 Гбит/с (рекомендуется агрегированная сеть 10+ Гбит/с)
Сервер метаданных Ceph	Процессор	2 ядра
	ОЗУ	2 ГБ+ (для производственных кластеров больше)
	Жесткий диск	1 ГБ на демон, рекомендуется SSD
	Сеть	1 Гбит/с (рекомендуется 10+ Гбит/с)

Серверу в целом требуется как минимум сумма потребностей демонов, которые он размещает, а также ресурсы для журналов и других компонентов ОС. Следует также учитывать, что потребность сервера в ОЗУ и хранилище будет больше при запуске и при выходе из строя или добавлении компонентов и перебалансировке кластера.

Дополнительные рекомендации:

Память: в гиперконвергентной настройке необходимо тщательно контролировать потребление памяти. Помимо прогнозируемого использования памяти ВМ и контейнерами, необходимо также учитывать наличие достаточного объема памяти для Ceph.

Сеть: Рекомендуется использовать сеть не менее 10 Гбит/с, которая используется исключительно для Ceph. Объем трафика, особенно во время восстановления, будет мешать другим службам в той же сети и может даже сломать стек кластера PVE. Кроме того, необходимо оценить потребность в пропускной способности. Например, один жесткий диск может не заполнить канал 1 Гбит, несколько жестких дисков OSD на узел могут, а современные твердотельные накопители NVMe быстро заполнят 10 Гбит/с. Развертывание сети, способной обеспечить еще большую пропускную способность, гарантирует, что это не станет узким местом. Возможны 25, 40 или даже 100 Гбит/с.

Жесткие диски: OSD-диски, намного меньшие одного терабайта, используют значительную часть своей емкости для метаданных, а диски меньше 100 гигабайт будут вообще неэффективны. Настоятельно рекомендуется, чтобы SSD были предоставлены как минимум для узлов мониторов Ceph и менеджеров Ceph, а также для пулов метаданных CephFS и пулов индексов Ceph Object Gateway (RGW), даже если жесткие диски должны быть предоставлены для больших объемов данных OSD.

Помимо типа диска, Ceph лучше всего работает с равномерным по размеру и распределенным количеством дисков на узел. Например, 4 диска по 500 ГБ в каждом узле лучше, чем смешанная установка с одним диском на 1 ТБ и тремя дисками по 250 ГБ.

Необходимо также сбалансировать количество OSD и емкость одного OSD. Большая емкость позволяет увеличить плотность хранения, но это также означает, что один сбой OSD заставляет Ceph восстанавливать больше данных одновременно.

Поскольку Ceph обрабатывает избыточность объектов данных и несколько параллельных записей на диски (OSD) самостоятельно, использование RAID-контроллера обычно не улучшает производительность или доступность. Напротив, Ceph разработан для самостоятельной обработки целых дисков, без какой-либо абстракции между ними. RAID-контроллеры не предназначены для рабочей нагрузки Ceph и могут усложнять ситуацию, а иногда даже снижать производительность, поскольку их алгоритмы записи и кэширования могут мешать алгоритмам Ceph.

Репликация: рекомендуется использовать репликацию с коэффициентом 3 для обеспечения высокой доступности и устойчивости к отказам.

Примечание. Приведенные выше рекомендации следует рассматривать как приблизительное руководство по выбору оборудования. Поэтому все равно важно адаптировать его к конкретным потребностям. Следует постоянно тестировать свои настройки и следить за работоспособностью и производительностью.

4.4.6.2 Начальная конфигурация Ceph

Для создания начальной конфигурации Ceph рекомендуется использовать мастер установки PVE Ceph. В случае использования мастер установки PVE Ceph следует перейти в раздел «Центр обработки данных» → «Ceph» и нажать кнопку «Настроить Ceph» (Рис. 118).

Инициализация кластера Ceph

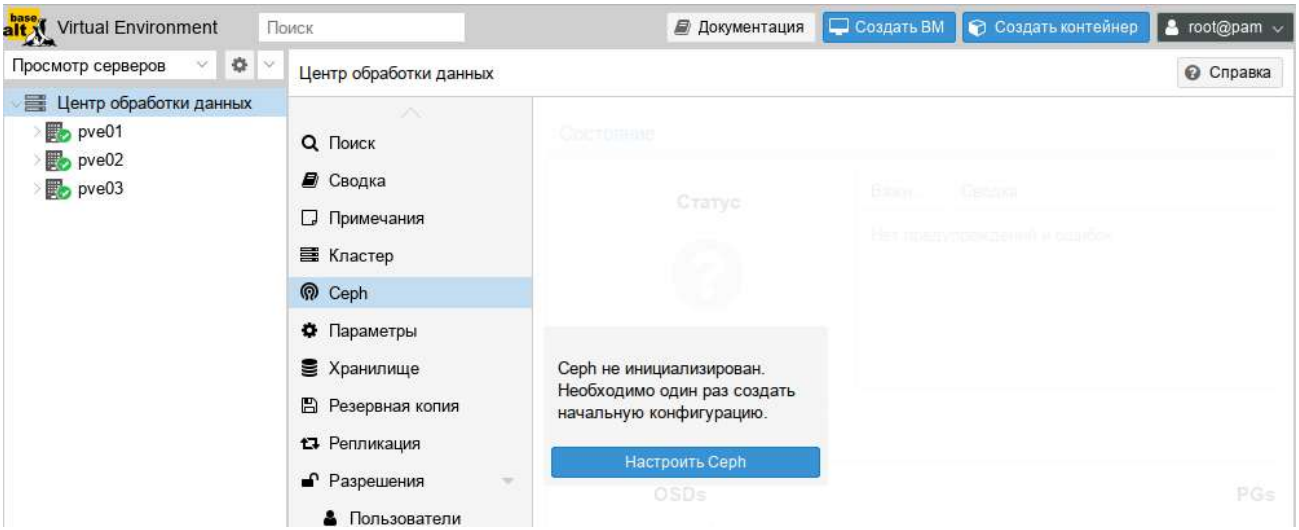


Рис. 118

В открывшемся окне (Рис. 119) выбрать открытую сеть в выпадающем списке «Public Network» и сеть кластера в списке «Cluster Network»:

- «Public Network» – позволяет настроить выделенную сеть для Ceph. Настоятельно рекомендуется выделить трафик Ceph в отдельную сеть;
- «Cluster Network» – позволяет разделить трафик репликации OSD и heartbeat. Это разгрузит общедоступную сеть и может привести к значительному повышению производительности, особенно в больших кластерах.

Примечание. Сеть в поле «Cluster Network» можно не указывать, по умолчанию сеть кластера совпадает с открытой сетью, указанной в поле «Public Network».

Настройка сетевых параметров Ceph

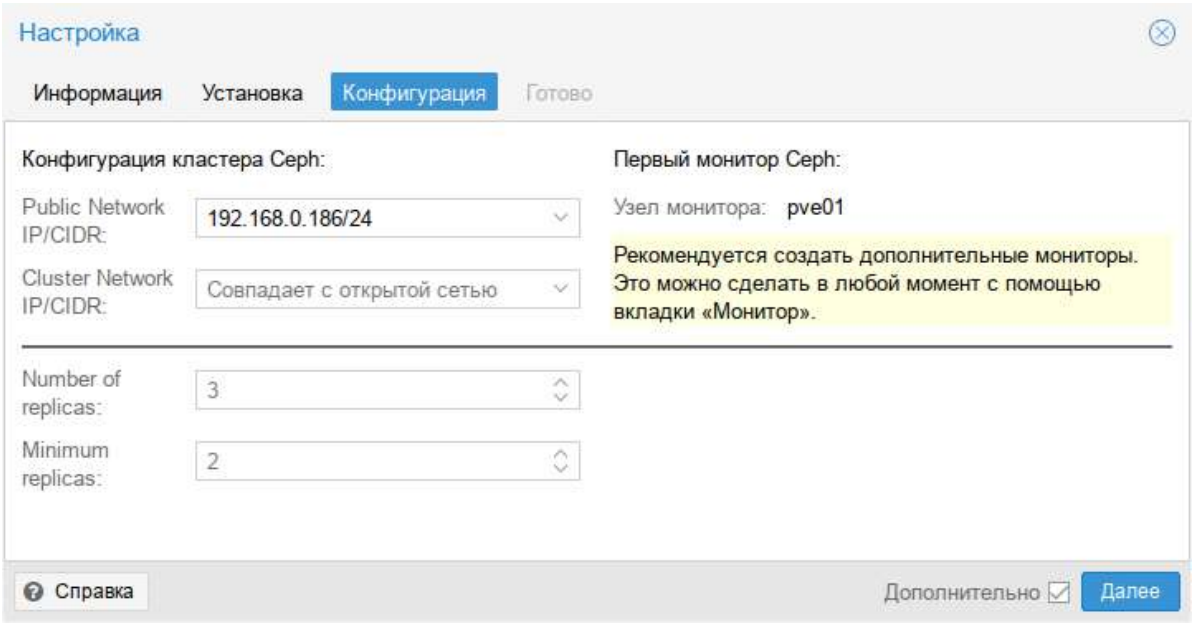


Рис. 119

После нажатия кнопки «Далее» будет выведено сообщение об успешной установке (Рис. 120).

Сообщение об успешной установки Ceph

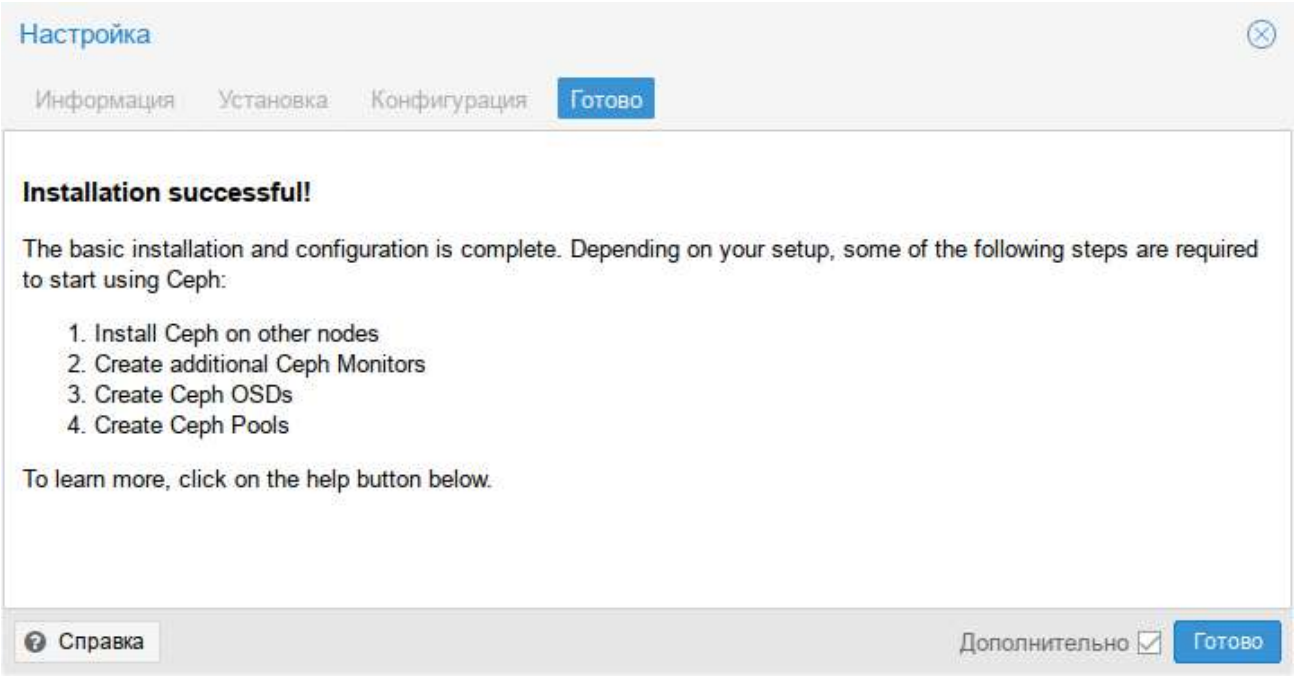


Рис. 120

Создать начальную конфигурацию Ceph также можно, выполнив следующую команду на любом узле PVE:

```
# pveceph init --network 192.168.0.0/24
```

В результате инициализации Ceph будет создана начальная конфигурация в `/etc/pve/ceph.conf` с выделенной сетью для Ceph. Файл `/etc/pve/ceph.conf` автоматически распространяется на все узлы PVE с помощью `pmxcfs`. Будет также создана символическая ссылка `/etc/ceph/ceph.conf`, которая указывает на файл `/etc/pve/ceph.conf`. Таким образом, можно запускать команды Ceph без необходимости указывать файл конфигурации.

При инициализации Ceph будет создан один монитор. Далее необходимо создать несколько дополнительных мониторов, менеджер, OSD и по крайней мере один пул.

4.4.6.3 Монитор Ceph

Монитор Ceph (MON) поддерживает основную копию карты кластера. Для поддержания высокой доступности нужно не менее трёх мониторов. Если использовался мастер установки, один монитор уже будет установлен.

Примечание. Если кластер небольшой или средних размеров, трёх мониторов будет достаточно, больше мониторов требуется только для действительно больших кластеров.

4.4.6.3.1 Создание монитора

Для создания монитора в веб-интерфейсе необходимо на любом узле перейти на вкладку «Хост» → «Ceph» → «Монитор» и создать необходимое количество мониторов на узлах. Для этого нажать кнопку «Создать» в разделе «Монитор» и в открывшемся окне выбрать имя узла, на котором будет создан монитор (Рис. 121).

Создание монитора на узле pve02

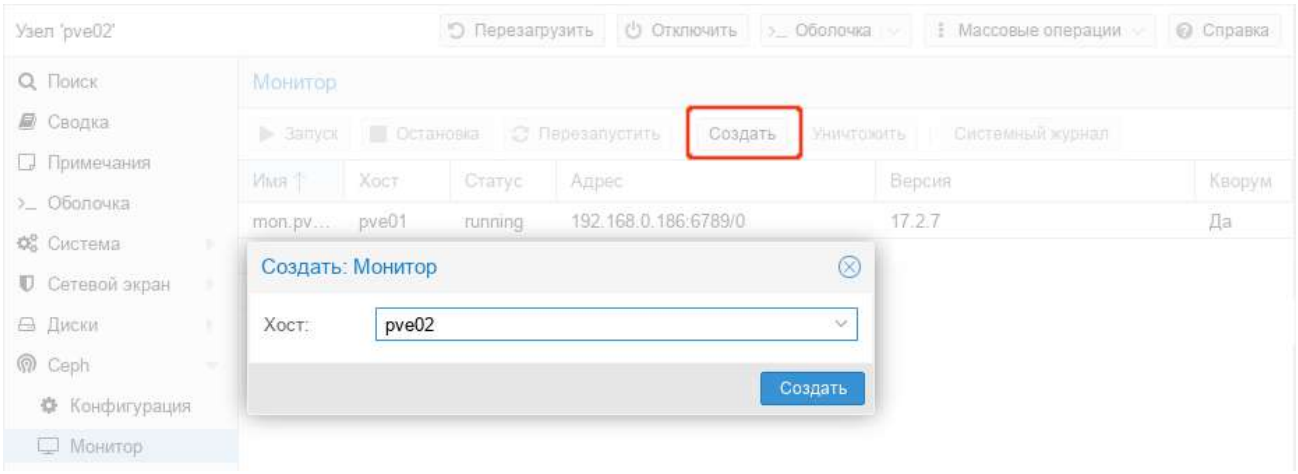


Рис. 121

Для создания монитора в командной строке следует на каждом узле, где нужно разместить монитор, выполнить команду:

```
# pveceph mon create
```

4.4.6.3.2 Удаление монитора

Чтобы удалить монитор в веб-интерфейсе, необходимо выбрать любой узел в дереве, перейти на панель «Ceph» → «Монитор», выбрать монитор и нажать кнопку «Уничтожить».

Для удаления монитора Ceph в командной строке, необходимо подключиться к узлу, на котором запущен монитор, и выполнить команду:

```
# pveceph mon destroy <mon_id>
```

4.4.6.4 Менеджер Ceph

Менеджер Ceph работает вместе с мониторами. Он предоставляет интерфейс для мониторинга кластера. Требуется как минимум один демон ceph-mgr.

Примечание. Рекомендуется устанавливать менеджеры Ceph на тех же узлах, что и мониторы. Для высокой доступности следует установить более одного менеджера, но в любой момент времени будет активен только один менеджер (Рис. 122).

Активный менеджер на узле pve01

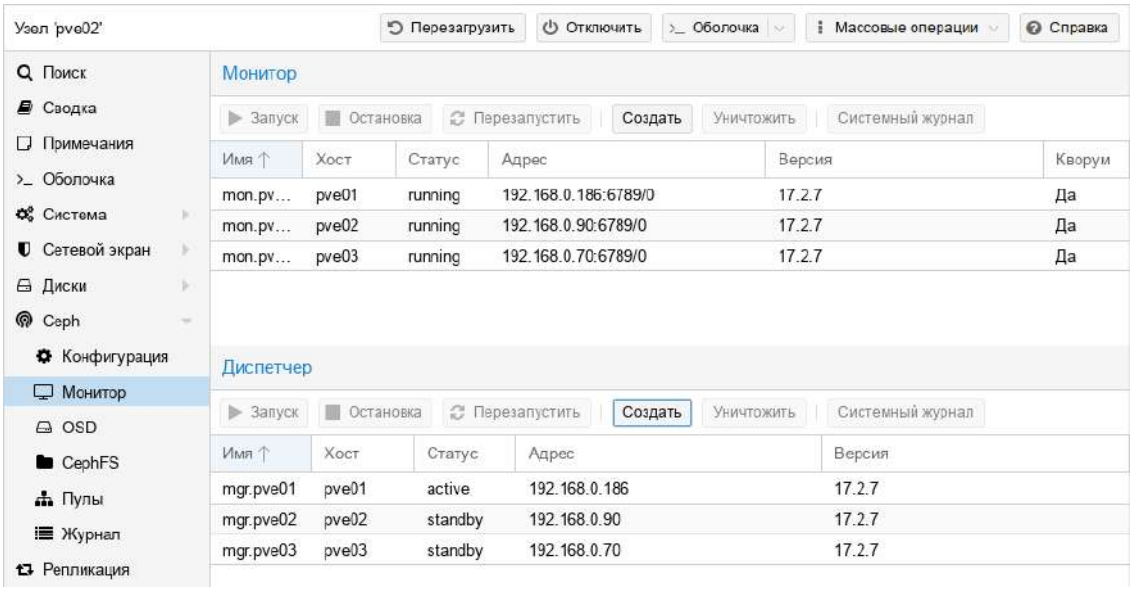


Рис. 122

4.4.6.4.1 Создание менеджера

Для создания менеджера в веб-интерфейсе следует на любом узле перейти на вкладку «Хост» → «Ceph» → «Монитор» и создать необходимое количество менеджеров на узлах. Для этого нажать кнопку «Создать» в разделе «Диспетчер» и в открывшемся окне выбрать имя узла, на котором будет создан менеджер Ceph (Рис. 123).

Создание менеджера на узле pve02

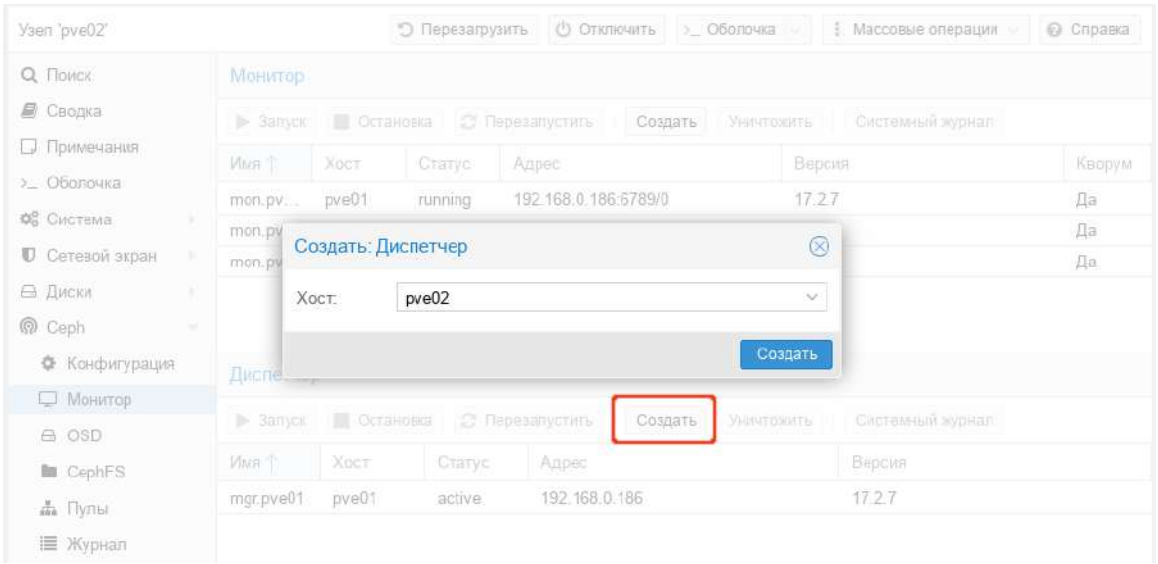


Рис. 123

Для создания менеджера в командной строке следует на каждом узле, где нужно разместить менеджер Ceph, выполнить команду:

```
# pveceph mgr create
```

4.4.6.4.2 Удаление менеджера

Чтобы удалить менеджер в веб-интерфейсе, необходимо выбрать любой узел в дереве, перейти на панель «Ceph» → «Монитор», выбрать менеджер и нажать кнопку «Уничтожить».

Для удаления менеджера Ceph в командной строке необходимо подключиться к узлу, на котором запущен менеджер, и выполнить команду:

```
# pveceph mgr destroy <mgr_id>
```

4.4.6.5 Ceph OSD

Рекомендуется использовать один OSD на каждый физический диск.

4.4.6.5.1 Создание OSD

Рекомендуется использовать кластер Ceph с не менее чем тремя узлами и не менее чем 12 OSD, равномерно распределенными по узлам.

Если диск использовался ранее (например, для ZFS или как OSD), сначала нужно удалить все следы этого использования. Чтобы удалить таблицу разделов, загрузочный сектор и любые другие остатки OSD, можно использовать команду:

```
# ceph-volume lvm zap /dev/[X] --destroy
```

В н и м а н и е ! Эта команда уничтожит все данные на диске!

Для создания OSD в веб-интерфейсе PVE необходимо перейти на вкладку «Хост» → «Ceph» → «OSD» и нажать кнопку «Создать: OSD» (Рис. 124). В открывшемся окне выбрать локальный диск, который будет включен в ceph-кластер. Отдельно можно указать диски для метаданных («Диск базы данных») и журналирования («Диск WAL»).

Создание OSD

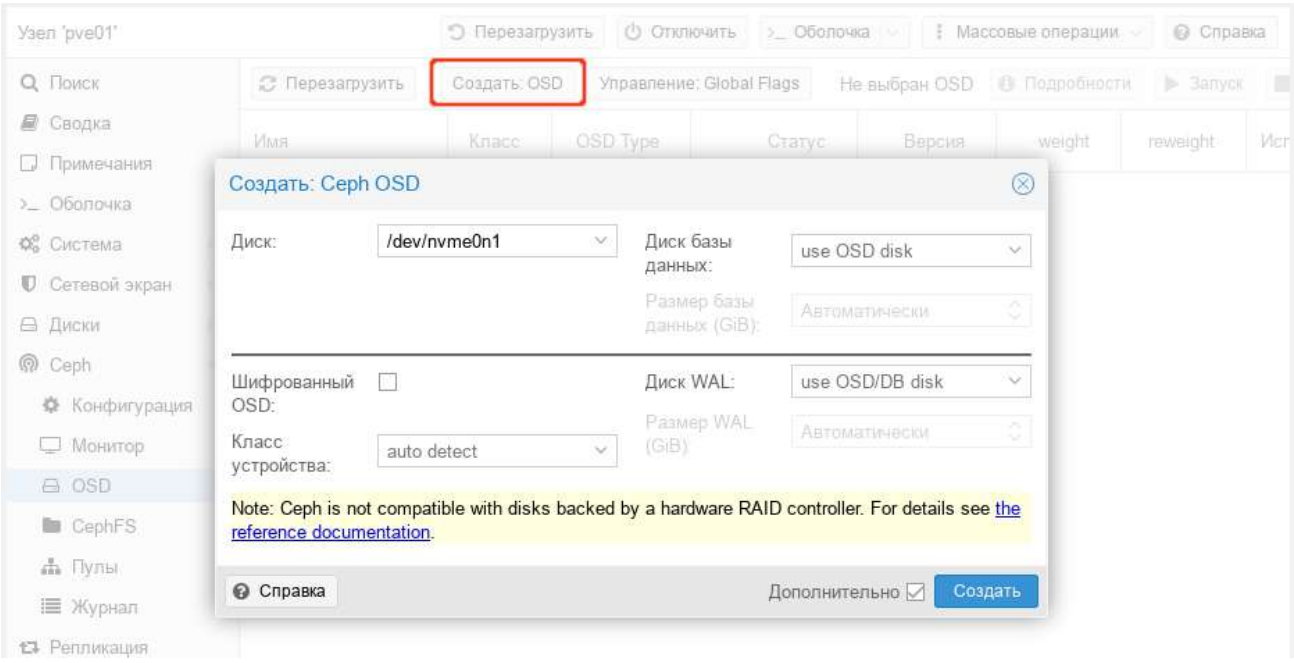


Рис. 124

Для создания OSD в командной строке можно выполнить команду:

```
# pveceph osd create /dev/[X]
```

Указать отдельные устройства для метаданных (DB) и журналирования (WAL) для OSD можно с помощью параметров `-db_dev` и `-wal_dev`:

```
# pveceph osd create /dev/[X] -db_dev /dev/[Y] -wal_dev /dev/[Z]
```

Если диск для журналирования не указан, WAL размещается вместе с DB.

Можно напрямую указать размер WAL и DB с помощью параметров `-db_size` и `-wal_size` соответственно. Если эти параметры не указаны, будут использоваться следующие значения (по порядку):

- `bluestore_block_{db,wal}_size` из конфигурации Ceph:
  - … база данных, раздел `[osd]`;
  - … база данных, раздел `[global]`;
  - … файл, раздел `[osd]`;
  - … файл, раздел `[global]`;
- 10% (DB)/1% (WAL) от размера OSD.

Примечание. В DB хранятся внутренние метаданные BlueStore, а WAL – это внутренний журнал BlueStore или журнал предварительной записи. Для лучшей производительности рекомендуется использовать высокопроизводительные диски.

4.4.6.5.2 Удаление OSD

Процедура удаления OSD в веб-интерфейсе:

1) выбрать узел PVE и перейти на панель «Ceph» → «OSD»;
2) выбрать OSD, который нужно удалить и нажать кнопку «Out»;
3) после того как статус OSD изменится с `in` на `out`, нажать кнопку «Остановка»;
4) после того как статус изменится с `up` на `down`, выбрать раскрывающемся списке «Дополнительно» → «Уничтожить».

Чтобы удалить OSD в консоли, следует выполнить следующие команды:

```
# ceph osd out <ID>
# systemctl stop ceph-osd@<ID>.service
```

Первая команда указывает Ceph не включать OSD в распределение данных. Вторая команда останавливает службу OSD. До этого момента данные не теряются.

Следующая команда уничтожит OSD (можно указать параметр `-cleanup`, чтобы дополнительно уничтожить таблицу разделов):

```
# pveceph osd destroy <ID>
```

Внимание! Эта команда уничтожит все данные на диске!

4.4.6.6 Пулы Ceph

Пул – это логическая группа для хранения объектов. Он содержит набор объектов, известных как группы размещения (PG, pg_num).

4.4.6.6.1 Создание и редактирование пула

Создавать и редактировать пулы можно в командной строке или в веб-интерфейсе любого узла PVE в разделе «Ceph» → «Пулы» (Рис. 125).

Создание пула в веб-интерфейсе

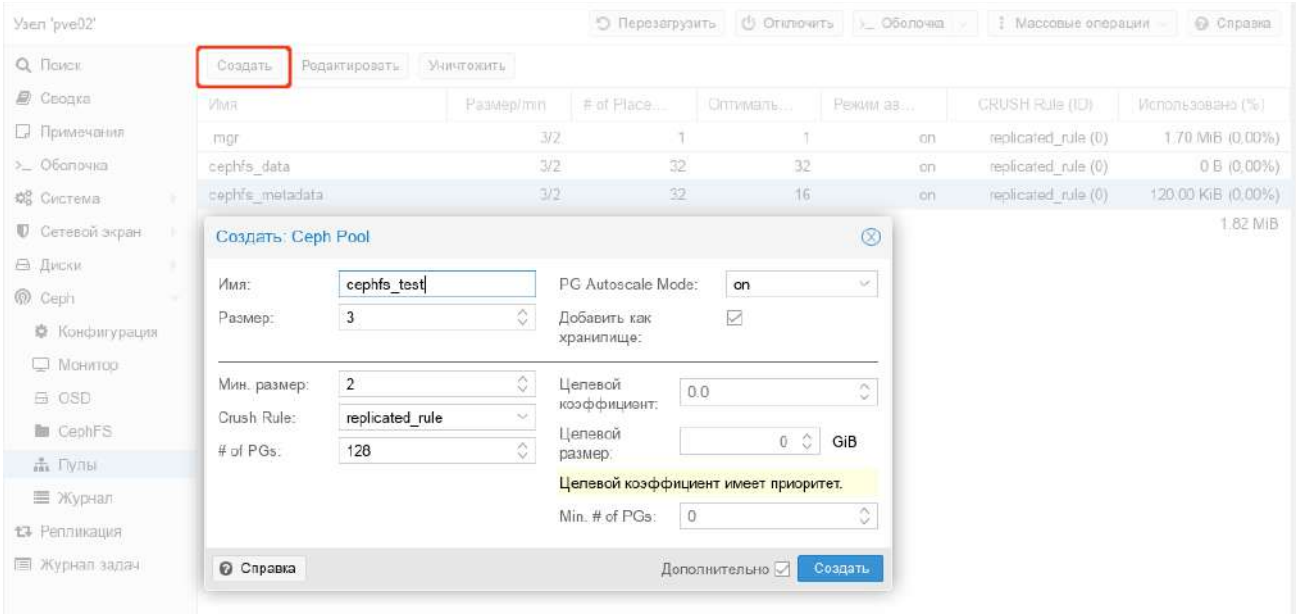


Рис. 125

Следующие параметры доступны при создании пула, а также частично при редактировании пула (в скобках приведены соответствующие опции команды `pveceph pool create`):

- «Имя» – имя пула. Имя должно быть уникальным и не может быть изменено впоследствии;
- «Размер» (-size) – количество реплик на объект. Ceph всегда пытается иметь указанное количество копий объекта (по умолчанию 3);
- «PG Autoscale Mode» («Режим автоматического масштабирования PG») (-pg_autoscale_mode) – автоматический режим масштабирования PG пула. Если установлено значение warn, выводится предупреждающее сообщение, если в пуле неоптимальное количество PG (по умолчанию on);
- «Добавить как хранилище» (-add_storages) – настроить хранилище с использованием нового пула. Доступно только при создании пула (по умолчанию true);
- «Мин. Размер» (-min_size) – минимальное количество реплик для объекта. Ceph отклонит ввод-вывод в пуле, если в PG меньше указанного количества реплик (по умолчанию 2);

- «Crush Rule» («Правило Crush») (-crush_rule) – правило, используемое для сопоставления размещения объектов в кластере. Эти правила определяют, как данные размещаются в кластере;
- «# of PGs» («Количество PG») (-pg_num) – количество групп размещения, которые должен иметь пул в начале (по умолчанию 128);
- «Целевой коэффициент» (-target_size_ratio) – соотношение ожидаемых данных в пуле. Автомасштабирование PG использует соотношение относительно других наборов соотношений. Данный параметр имеет приоритет над целевым размером, если установлены оба;
- «Целевой размер» (-target_size) – предполагаемый объем данных, ожидаемых в пуле. Автомасштабирование PG использует этот размер для оценки оптимального количества PG;
- «Min # of PGs» («Мин. количество PG») (-pg_num_min) – минимальное количество групп размещения. Этот параметр используется для точной настройки нижней границы количества PG для этого пула. Автомасштабирование PG не будет объединять PG ниже этого порогового значения.

Примечание. Не следует устанавливать `min_size` равным 1. Реплицированный пул с `min_size` равным 1 разрешает ввод-вывод для объекта, при наличии только одной реплики, что может привести к потере данных, неполным PG или ненайденным объектам.

Рекомендуется либо включить режим автоматического масштабирования PG (PG autoscaler), либо рассчитать количество PG на основе ваших настроек.

PG autoscaler может автоматически масштабировать количество PG для пула в фоновом режиме. Установка параметров «Целевой размер» или «Целевой коэффициент» помогает PG-Autoscaler принимать более обоснованные решения.

Команда создания пула в командной строке:

```
# pveceph pool create <pool-name> -add_storages
```

4.4.6.6.2 Пулы EC

Erasure coding (EC) – это метод коррекции ошибок, используемый для обеспечения надежности и восстановления данных в системах хранения. Основная цель EC – повысить доступность данных, минимизировав их избыточное копирование. Пулы EC могут предложить больше полезного пространства по сравнению с реплицированными пулами, но они делают это за счет производительности.

В классических реплицированных пулах хранится несколько реплик данных (`size`), тогда как в пуле EC данные разбиваются на `k` фрагментов данных с дополнительными `m` фрагментами кодирования (проверки). Эти фрагменты кодирования можно использовать для воссоздания данных, если фрагменты данных отсутствуют.

Количество фрагментов кодирования m определяет, сколько OSD может быть потеряно без потери данных. Общее количество хранимых объектов равно k + m.

Пулы EC можно создавать с помощью команды pveceph. При планировании пула EC необходимо учитывать тот факт, что они работают иначе, чем реплицированные пулы.

По умолчанию значение min_size для пула EC зависит от параметра m. Если m = 1, значение min_size для пула EC будет равно k. Если m > 1, значение min_size будет равно k + 1. В документации Ceph рекомендуется использовать консервативное значение min_size, равное k + 2.

Если доступно меньше, чем min_size OSD, любой ввод-вывод в пул будет заблокирован до тех пор, пока снова не будет достаточно доступных OSD.

Примечание. При планировании пула EC необходимо следить за min_size, так как он определяет, сколько OSD должно быть доступно. В противном случае ввод-вывод будет заблокирован.

Например, пул EC с k = 2 и m = 1 будет иметь size = 3, min_size = 2 и останется работоспособным, если один OSD выйдет из строя. Если пул настроен с k = 2, m = 2, будет иметь size = 4 и min_size = 3 и останется работоспособным, если один OSD будет потерян.

Команда для создания пула EC:

```
# pveceph pool create <pool-name> --erasure-coding k=<integer> ,m=<integer> [,device-class=<class>] [,failure-domain=<domain>] [,profile=<profile>]
```

В результате выполнения этой команды будет создан новый пул EC для RBD с сопутствующим реплицированным пулом для хранения метаданных (<pool name>-data и <pool name>-metada). По умолчанию также будет создано соответствующее хранилище. Если такое поведение нежелательно, отключить создание хранилища можно, указав параметр --add_storages 0. При настройке конфигурации хранилища вручную следует иметь в виду, что необходимо задать параметр data-pool, только тогда пул EC будет использоваться для хранения объектов данных.

Примечание. Необязательные параметры --size, --min_size и --crush_rule будут использоваться для реплицированного пула метаданных, но не для пула данных EC. Если нужно изменить min_size в пуле данных, это можно будет сделать позже. Параметры size и crush_rule нельзя изменить в пулах EC.

Изменить настройки профиля EC нельзя, в этом случае нужно создать новый пул с новым профилем.

Если необходимо дополнительно настроить профиль EC, можно создать его напрямую с помощью инструментов Ceph и указать профиль в параметре profile. Например:

```
# pveceph pool create <pool-name> --erasure-coding profile=<profile-name>
```

Существующий пул EC можно добавить в качестве хранилища в PVE:

```
# pvesm add rbd <storage-name> --pool <replicated-pool> --data-pool <ec-pool>
```

Примечание. Для любых внешних кластеров Ceph, не управляемых локальным кластером PVE, также следует указывать параметры `keyring` и `monhost`.

4.4.6.6.3 Удаление пула

Чтобы удалить пул в веб-интерфейсе, необходимо в разделе «Хост» → «Ceph» → «Пулы» выбрать пул, который нужно удалить и нажать кнопку «Уничтожить». Для подтверждения уничтожения пула, нужно в открывшемся диалоговом окне ввести имя пула и нажать кнопку «Удалить».

Команда для удаления пула:

```
# pveceph pool destroy <name>
```

Чтобы также удалить связанное хранилище следует указать опцию `-remove_storages`.

Примечание. Удаление пула выполняется в фоновом режиме и может занять некоторое время.

4.4.6.6.4 Автомасштабирование PG

Автомасштабирование PG позволяет кластеру учитывать объем (ожидаемых) данных, хранящихся в каждом пуле, и автоматически выбирать соответствующие значения `pg_num`.

Примечание. Может потребоваться активировать модуль pg_autoscaler:

```
# ceph mgr module enable pg_autoscaler
```

Список запущенных модулей можно посмотреть, выполнив команду:

```
# ceph mgr module ls
```

Автомасштабирование настраивается для каждого пула и имеет следующие режимы:

- warn – предупреждение о работоспособности выдается, если предлагаемое значение `pg_num` слишком сильно отличается от текущего значения;
- on – `pg_num` настраивается автоматически без необходимости ручного вмешательства;
- off – автоматические корректировки `pg_num` не производятся, и предупреждение не выдается, если количество PG не является оптимальным.

Коэффициент масштабирования можно настроить с помощью параметров `target_size`, `target_size_ratio` и `pg_num_min`.

4.4.6.7 Ceph CRUSH и классы устройств

В основе Ceph лежит алгоритм CRUSH (Controlled Replication Under Scalable Hashing). CRUSH вычисляет, где хранить и откуда извлекать данные. Этот алгоритм позволяет однозначно определить местоположение объекта на основе хеша имени объекта и определенной карты (Рис. 126), которая формируется исходя из физической и логической структур. Карта не включает в себя

информацию о местонахождении данных. Путь к данным каждый клиент определяет сам, с помощью CRUSH-алгоритма и актуальной карты, которую он предварительно спрашивает у монитора. При добавлении диска или падении сервера карта обновляется.

Карта Crush

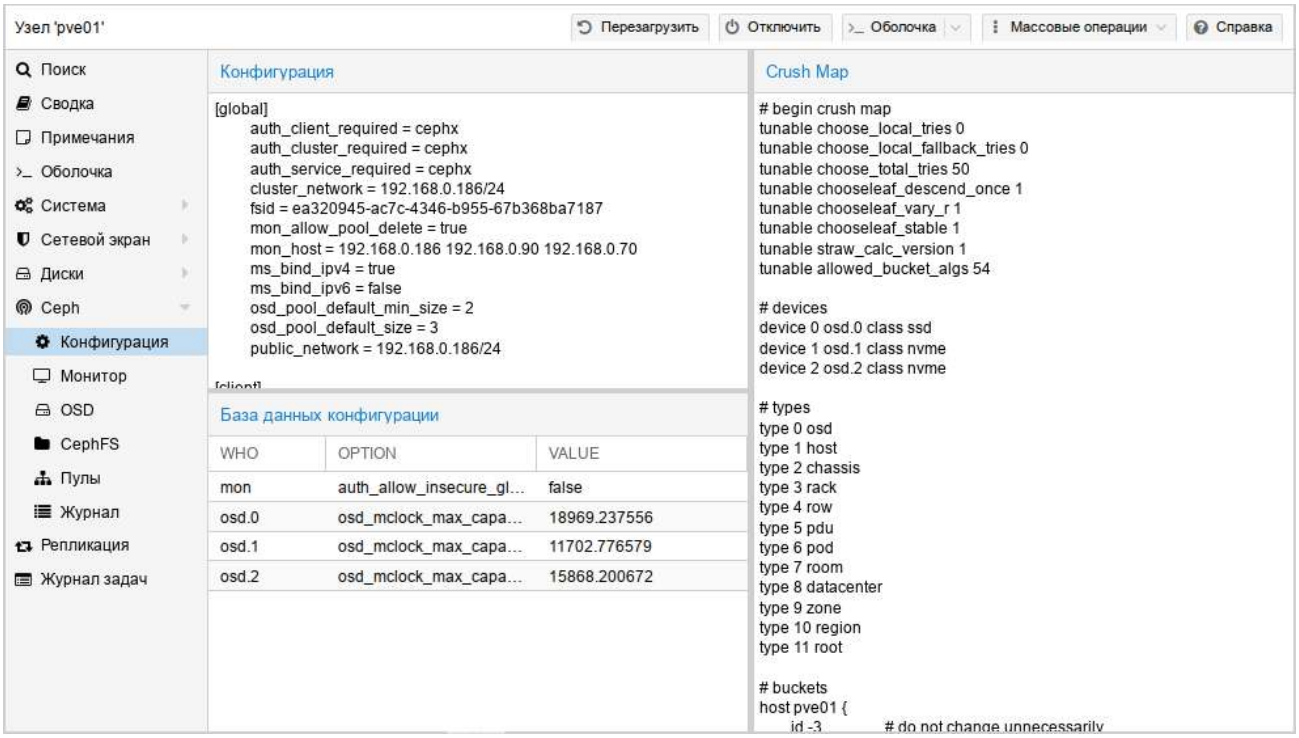


Рис. 126

Карту можно изменить, чтобы она отражала различные иерархии репликации. Реплики объектов можно разделить (например, домены отказов), сохраняя при этом желаемое распределение.

Классы устройств можно увидеть в выходных данных дерева OSD ceph. Эти классы представляют свой собственный корневой контейнер, который можно увидеть, выполнив команду:

```
# ceph osd crush tree --show-shadow
ID   CLASS  WEIGHT   TYPE NAME
 -6   nvme  0.09760  root default~nvme
 -5   nvme        0      host pve01~nvme
 -9   nvme  0.04880      host pve02~nvme
  1   nvme  0.04880          osd.1
-12   nvme  0.04880      host pve03~nvme
  2   nvme  0.04880          osd.2
 -2    ssd  0.04880  root default~ssd
 -4    ssd  0.04880      host pve01~ssd
  0    ssd  0.04880          osd.0
 -8    ssd        0      host pve02~ssd
-11    ssd        0      host pve03~ssd
```

```
 -1         0.14639  root default
 -3         0.04880      host pve01
  0    ssd  0.04880          osd.0
 -7         0.04880      host pve02
  1   nvme  0.04880          osd.1
-10         0.04880      host pve03
  2   nvme  0.04880          osd.2
```

Чтобы указать пулу распределять объекты только на определенном классе устройств, сначала необходимо создать политику для класса устройств:

```
# ceph osd crush rule create-replicated <rule-name> <root> <failure-domain> <class>
```

Где:

- rule-name – имя политики;
- root – корень отказа (значение default – корень Ceph);
- failure-domain – домен отказа, на котором должны распределяться объекты (обычно host);
- class – какой тип хранилища резервных копий OSD использовать (например, nvme, ssd).

Пример создания политики replicated_nvme для реплицированных пулов, данные будут иметь домен отказа host, а размещаться – на nvme:

```
# ceph osd crush rule create-replicated my_rule default host nvme
```

Посмотреть настройки политик можно, выполнив команду:

```
# ceph osd crush rule dump
```

После того как политика будет создана в карте CRUSH, можно указать пулу использовать набор правил:

```
# ceph osd pool set <pool-name> crush_rule <rule-name>
```

Примечание. Если пул уже содержит объекты, их необходимо переместить соответствующим образом. В зависимости от настроек это может оказать большое влияние на производительность кластера. Либо можно создать новый пул и переместить диски по отдельности.

4.4.6.8 Клиент Ceph

Пулы Ceph можно использовать для создания хранилищ RBD (см. раздел Ceph RBD).

Для внешнего кластера Ceph необходимо скопировать связку ключей в предопределенное место. Если Ceph установлен на узлах PVE, то это будет сделано автоматически.

Примечание. Имя файла должно быть в формате <storage_id>.keyring, где <storage_id> – идентификатор хранилища rbd.

4.4.6.9 CephFS

Ceph предоставляет файловую систему, которая работает поверх того же хранилища объектов, что и блочные устройства RADOS. Сервер метаданных (MDS) используется для сопоставления поддерживаемых RADOS объектов с файлами и каталогами, что позволяет Ceph предоставлять POSIX-совместимую, реплицированную файловую систему. Это позволяет легко настраивать

кластерную, высокодоступную, общую файловую систему. Серверы метаданных Ceph гарантируют, что файлы равномерно распределены по всему кластеру Ceph. В результате даже случаи высокой нагрузки не перегрузят один хост, что может быть проблемой при традиционных подходах к общим файловым системам, например, NFS.

PVE поддерживает как создание гиперконвергентной CephFS, так и использование существующей CephFS в качестве хранилища для хранения резервных копий, ISO-файлов и шаблонов контейнеров.

4.4.6.9.1 Сервер метаданных (MDS)

В кластере можно создать несколько серверов метаданных, но по умолчанию только один может быть активным. Если MDS или его узел перестает отвечать, другой резервный MDS становится активным. Ускорить передачу данных между активным и резервным MDS можно, используя параметр hotstandby при создании сервера, или, после его создания, установив в соответствующем разделе MDS в `/etc/pve/ceph.conf` параметр:

```
mds standby replay = true
```

Если этот параметр включен, указанный MDS будет находиться в состоянии warm, опрашивая активный, чтобы в случае возникновения каких-либо проблем быстрее взять на себя управление.

Примечание. Этот активный опрос оказывает дополнительное влияние на производительность системы и активного MDS.

Для работы CephFS требуется настроить и запустить по крайней мере один сервер метаданных. MDS можно создать в веб-интерфейсе PVE («Хост» → «Ceph» → «CephFS» и нажать кнопку «Создать» (Рис. 127)) или из командной строки, выполнив команду:

```
# pveceph mds create
```

Создание сервера метаданных Ceph

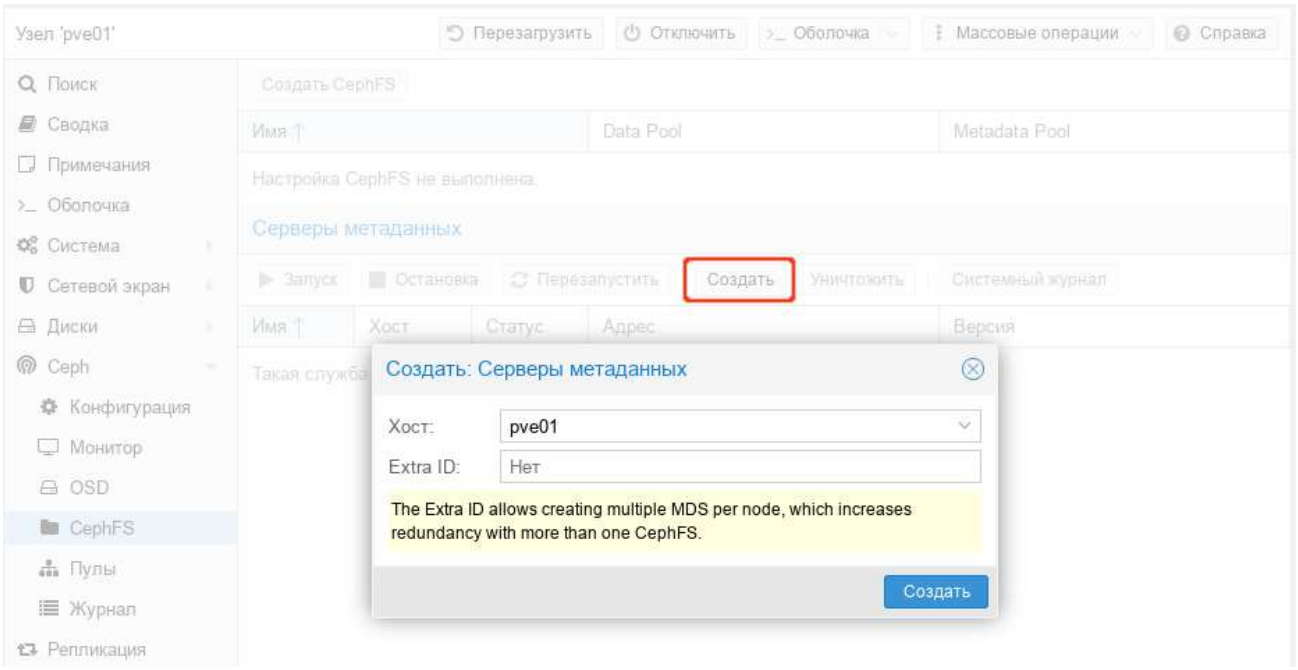


Рис. 127

4.4.6.9.2 Создание CephFS

Создать CephFS можно либо в веб-интерфейсе PVE («Хост» → «Ceph» → «CephFS», нажав кнопку «Создать CephFS» (Рис. 128)) или с помощью инструмента командной строки `pveceph`, например:

```
# pveceph fs create --pg_num 128 --add-storage
```

Создание CephFS

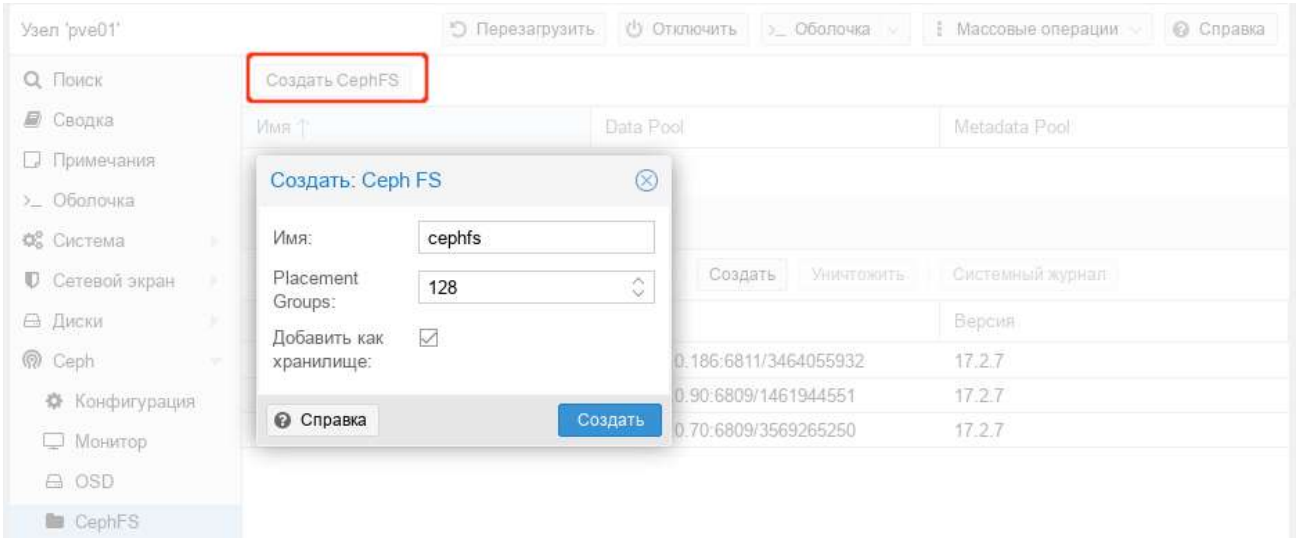


Рис. 128

В результате будет создана CephFS с именем cephfs (Рис. 129), пул данных cephfs_data со 128 группами размещения и пул метаданных cephfs_metadata с одной четвертью групп размещения пула данных (32). Параметр `--add-storage` (опция «Добавить как хранилище») добавит CephFS в конфигурацию хранилища PVE.

CephFS

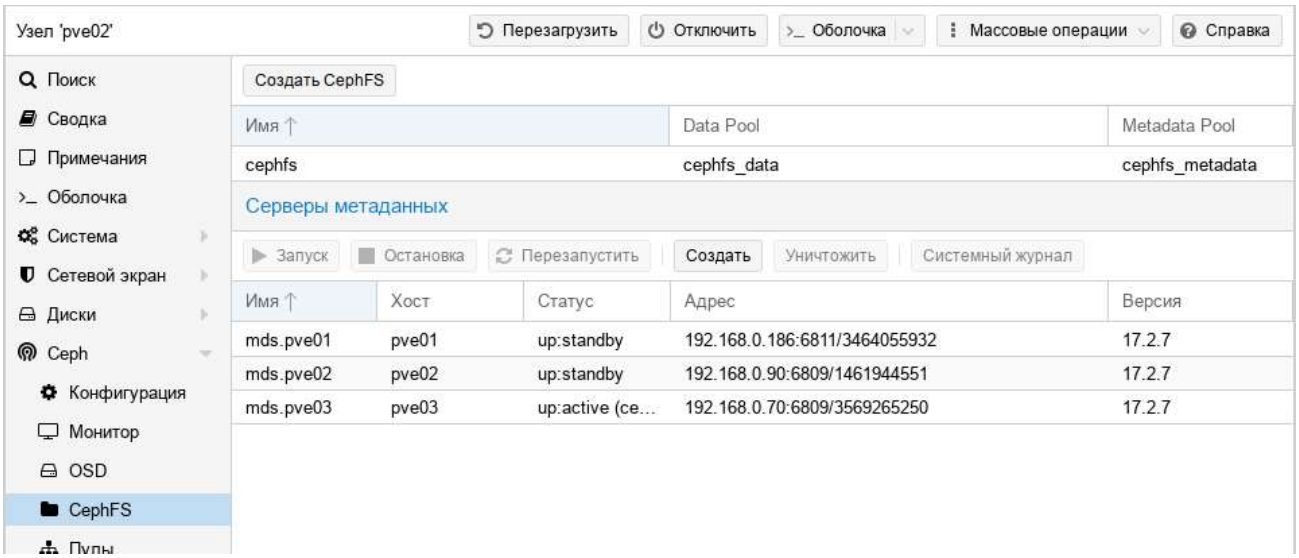


Рис. 129

4.4.6.9.3 Удаление CephFS

П р е д у п р е ж д е н и е . Уничтожение CephFS сделает все ее данные непригодными для использования. Это действие нельзя отменить!

Чтобы полностью и корректно удалить CephFS, необходимо выполнить следующие шаги:

1) отключить всех клиентов, не являющихся PVE (например, размонтировать CephFS в гостевых системах);
2) отключить все связанные записи хранилища CephFS PVE (чтобы предотвратить автоматическое монтирование);
3) удалить все используемые ресурсы из гостевых систем (например, ISO-образы), которые находятся на CephFS, которую нужно уничтожить;
4) вручную размонтировать хранилища CephFS на всех узлах кластера с помощью команды:

```
# umount /mnt/pve/<STORAGE-NAME>
```

где <STORAGE-NAME> – имя хранилища CephFS в PVE.

5) убедиться, что для этого CephFS не запущен ни один сервер метаданных (MDS), остановив или уничтожив их. Это можно сделать в веб-интерфейсе или в командной строке, выполнив команду:

```
# pveceph stop --service mds.NAME
```

чтобы остановить их, или команду:

```
# pveceph mds destroy NAME
```

чтобы уничтожить их.

Следует обратить внимание, что резервные серверы будут автоматически повышены до активных при остановке или удалении активного MDS, поэтому лучше сначала остановить все резервные серверы.

6) теперь можно уничтожить CephFS, выполнив команду:

```
# pveceph fs destroy NAME --remove-storages --remove-pools
```

Это автоматически уничтожит базовые пулы Ceph, а также удалит хранилища из конфигурации PVE.

После этих шагов CephFS должен быть полностью удален, и при наличии других экземпляров CephFS, остановленные серверы метаданных можно снова запустить для работы в качестве резервных.

4.4.6.10 Техническое обслуживание Ceph

4.4.6.10.1 Замена OSD

Одной из наиболее распространенных задач по техническому обслуживанию Ceph является замена диска OSD. Если диск уже находится в состоянии сбоя, можно выполнить шаги, указанные в разделе «Удаление OSD». Ceph воссоздаст копии на оставшихся OSD, если это возможно. Перебалансировка начнется, как только будет обнаружен сбой OSD или если OSD будет остановлен.

Примечание. При значениях `size/min_size` по умолчанию (3/2) восстановление начнется только при наличии узлов `size + 1`. Причина этого в том, что балансировщик объектов Ceph CRUSH по умолчанию использует полный узел в качестве «домена отказа».

Чтобы заменить работающий диск из веб-интерфейса, следует выполнить шаги, указанные в разделе «Удаление OSD». Единственное дополнение – дождаться, пока кластер не покажет HEALTH_OK, прежде чем останавливать OSD для его уничтожения.

Для замены работающего диска в командной строке, следует выполнить следующие действия:

1) выполнить команду:

```
# ceph osd out osd.<id>
```

2) проверить можно ли безопасно удалить OSD:

```
# ceph osd safe-to-destroy osd.<id>
```

3) после того как проверка покажет, что можно безопасно удалить OSD, выполнить команды:

```
# systemctl stop ceph-osd@<id>.service
# pveceph osd destroy <id>
```

Далее следует заменить старый диск новым и использовать ту же процедуру, что описана в разделе «Создание OSD».

4.4.6.10.2 Trim/Discard

Рекомендуется регулярно запускать fstrim (discard) на ВМ и контейнерах. Это освобождает блоки данных, которые файловая система больше не использует. В результате снижается нагрузка на ресурсы. Большинство современных ОС регулярно отправляют такие команды discard своим дискам. Нужно только убедиться, что ВМ включают опцию disk discard.

4.4.6.10.3 Очистка (scrubing)

Ceph обеспечивает целостность данных, очищая группы размещения. Ceph проверяет каждый объект в PG на предмет его работоспособности. Существует две формы очистки: ежедневные проверки метаданных и еженедельные глубокие проверки данных. Еженедельная глубокая очистка считывает объекты и использует контрольные суммы для обеспечения целостности данных. Если запущенная очистка мешает бизнес-потребностям (производительности), можно настроить время выполнения очисток.

4.4.6.11 Мониторинг и устранение неполадок Ceph

Важно постоянно контролировать работоспособность Ceph, либо с помощью инструментов Ceph, либо путем доступа к статусу через API PVE.

Следующие команды Ceph можно использовать для проверки работоспособности кластера (HEALTH_OK), наличия предупреждений (HEALTH_WARN) или ошибок (HEALTH_ERR):

```
# ceph -s # однократный вывод
# ceph -w # непрерывный вывод изменений статуса
```

Эти команды также предоставляют обзор действий, которые необходимо предпринять, если кластер находится в неработоспособном состоянии.

Чтобы получить более подробную информацию можно воспользоваться файлами журнала Ceph в /var/log/ceph/.

4.4.7 Репликация хранилища

Репликация хранилища в PVE позволяет синхронизировать данные между двумя или более узлами кластера. Репликация хранилища обеспечивает избыточность для гостевых систем, использующих локальное хранилище, и сокращает время миграции. Репликация работает на уровне блоков, что делает её эффективной для синхронизации больших объёмов данных.

Инфраструктурой репликации хранилища PVE управляет инструмент командной строки `pvesr`.

Репликация использует моментальные снимки для минимизации трафика, отправляемого по сети. Поэтому новые данные отправляются только пошагово после первоначальной полной синхронизации.

Репликация выполняется автоматически с настраиваемыми интервалами. Минимальный интервал репликации составляет одну минуту, максимальный – раз в неделю. Формат, используемый для указания этих интервалов, является подмножеством событий календаря systemd, см. «Формат расписания».

Можно реплицировать гостевую систему на несколько целевых узлов, но не дважды на один и тот же целевой узел.

Чтобы избежать перегрузки хранилища или сервера можно ограничить пропускную способность репликации.

Если ВМ/СТ мигрирует на узел, на который он уже реплицирован, необходимо переносить только изменения с момента последней репликации (так называемые дельты). Это значительно сокращает время миграции. Направление репликации автоматически переключается, если гостевая система мигрировала на целевой узел репликации.

Например, если VM100 находилась на узле pve02 и реплицировалась на узел pve03, то после миграции VM100 на узел pve03, ВМ будет автоматически реплицироваться с узла pve03 на узел pve02.

В случае если ВМ/СТ мигрирует на узел, на которую гостевая система не реплицируется, после миграции задание репликации продолжает реплицировать эту гостевую систему на настроенные узлы.

Примечание. Высокая доступность допускается в сочетании с репликацией хранилища, но может быть некоторая потеря данных между последним временем синхронизации и временем отказа узла.

Для репликации поддерживается тип хранилища zfspool – ZFS (локальный).

Для использования функции репликации должны быть выполнены следующие условия:

- исходный и целевой узлы должны находиться в одном кластере;
- все диски ВМ или контейнера должны храниться в хранилище ZFS;
- на исходном и целевом узле должно быть настроено хранилище ZFS с одинаковым именем;
- целевой узел должен иметь достаточно места для хранения.

4.4.7.1 Управление заданиями

Управлять заданиями репликации можно как в веб-интерфейсе, так и с помощью инструмента командной строки (CLI) `pvesr`.

Панель репликации можно найти на всех уровнях (центр обработки данных, узел, ВМ/СТ). В зависимости от уровня на панели репликации отображаются все задания (центр обработки данных) или задания, специфичные для узла (Рис. 130) или гостевой системы.

Список заданий репликации узла

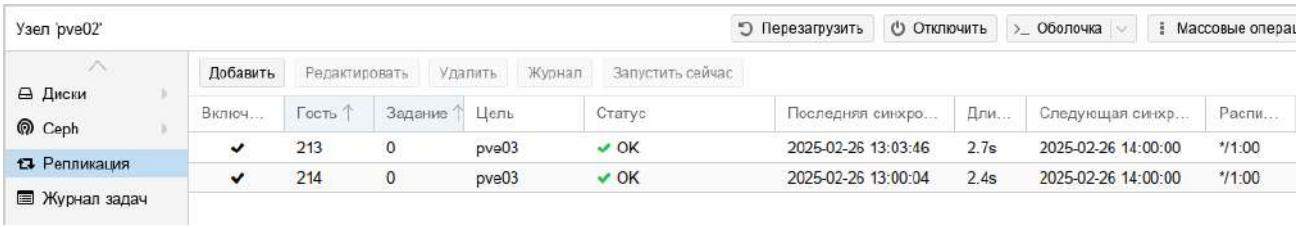


Рис. 130

Добавление нового задания репликации в веб-интерфейсе:

1) перейти в раздел «Репликация» (центра обработки данных, узла, ВМ или контейнера, для которых настраивается репликация;
2) нажать кнопку «Добавить»;
3) в открывшемся окне (Рис. 131) указать:
   - «CT/VM ID» – ID гостевой системы (если он еще не выбран);
   - «Цель» – узел, на который будут реплицироваться данные;
   - «Расписание» – расписание репликации;
   - «Ограничение скорости (MB/s)» – ограничение скорости репликации (если нужно);
4) нажать кнопку «Создать».

Создание задания репликации

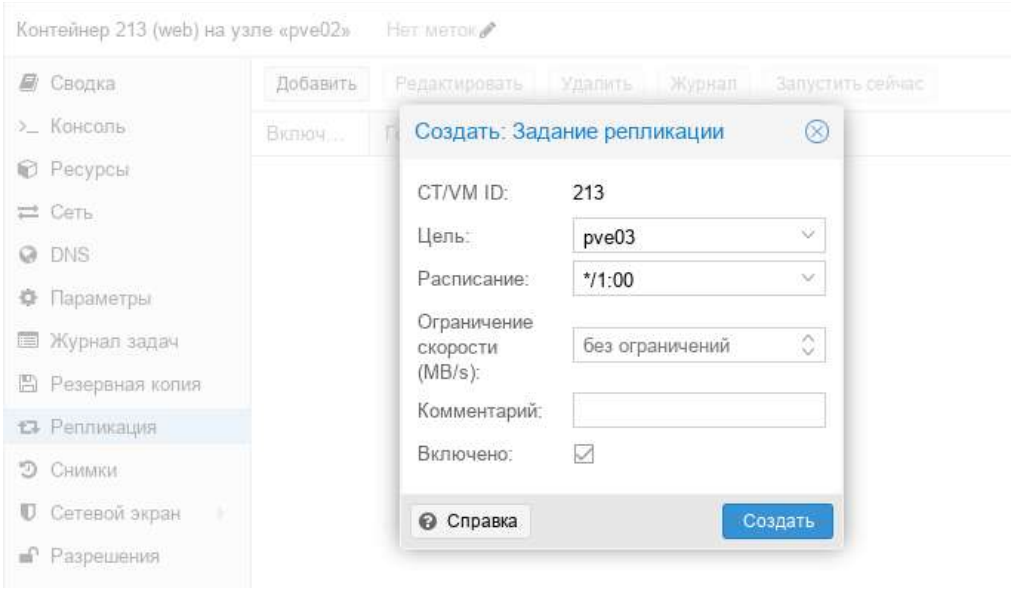


Рис. 131

Задание репликации имеет уникальный идентификатор. Идентификатор состоит из VMID и номера задания. Идентификатор необходимо указывать вручную только при использовании инструмента CLI.

Примеры работы с заданиями репликации в командной строке:

- создать задание репликации, которое запускается каждые 10 минут с ограниченной пропускной способностью 10 Мб/с для ВМ с идентификатором 214:

```
# pvesr create-local-job 214-0 pve03 --schedule "*/10" --rate 10
```

- отключить активное задание с идентификатором 214-0:

```
# pvesr disable 214-0
```

- просмотреть список заданий репликации:

```
# pvesr list
JobID                Target                 Schedule  Rate  Enabled
213-0                local/pve02              */1:00     -      yes
214-0                local/pve03                */10    10       no
```

- проверить статус репликации:

```
# pvesr status
JobID  Enabled  Target                LastSync             NextSync
Duration  FailCount State
213-0  Yes      local/pve02  2025-02-26_10:10:48  2025-02-26_11:00:00
2.455654          0 OK
```

- включить деактивированное задание с идентификатором 214-0:

```
# pvesr enable 214-0
```

- изменить интервал расписания задания с идентификатором 214-0 на один раз в час:

```
# pvesr update 214-0 --schedule '*/1:00'
```

4.4.7.2 Обработка ошибок

Если задание репликации сталкивается с проблемами, оно переводится в состояние ошибки. В этом состоянии настроенные интервалы репликации временно приостанавливаются. Неудачная репликация повторяется с интервалом в 30 минут. После успешного выполнения исходное расписание активируется снова.

Некоторые из наиболее распространенных проблем репликации:

- проблемы с сетью;
- недостаточно места в целевом хранилище репликации;
- на целевом узле не доступно хранилище с тем же идентификатором хранилища.

Примечание. Чтобы выяснить, что является причиной проблемы можно использовать журнал репликации.

4.4.7.3 Миграция гостевой системы в случае ошибки

В случае серьезной ошибки гостевая система может застрять на неисправном узле. В этом случае её нужно вручную переместить на рабочий узел.

Предположим, что есть две гостевые системы (VM 100 и CT 200), работающие на узле pve02 и реплицирующихся на узел pve03. Узел pve02 вышел из строя и не может вернуться в сеть. Нужно вручную перенести гостевые системы на узел pve03. Для этого необходимо:

1) подключиться к узлу pve03 по SSH или открыть его консоль в веб-интерфейсе;
2) проверить, является ли кластер кворумным:

```
# pvecm status
```

Если кворума нет, настоятельно рекомендуется сначала восстановить кворум и снова сделать узел работоспособным. Только если в данный момент это невозможно, можно использовать следующую команду для принудительной установки кворума на текущем узле:

```
# pvecm expected 1
```

Примечание. Следует любой ценой избегать изменений, которые влияют на кластер, если установлено ожидаемое количество голосов (например, добавление/удаление узлов, хранилищ, ВМ/CT). Этот режим можно использовать только для повторного запуска и работы важных гостевых систем или для решения самой проблемы кворума.

3) переместить оба файла конфигурации с исходного узла pve02 на узел pve03:

```
# mv /etc/pve/nodes/pve02/qemu-server/100.conf /etc/pve/nodes/pve03/qemu-server/100.conf
# mv /etc/pve/nodes/pve02/lxc/200.conf /etc/pve/nodes/pve03/lxc/200.conf
```

4) запустить гостевые системы:

```
# qm start 100
# pct start 200
```

4.5 Сетевая подсистема

PVE использует сетевой стек Linux, что обеспечивает большую гибкость в настройке сети на узлах PVE. Настройку сети можно выполнить либо через графический интерфейс («Хост»→«Система»→«Сеть», Рис. 132), либо вручную, редактируя файлы в каталоге `/etc/net/ifaces`.

Сетевые интерфейсы узла pve01

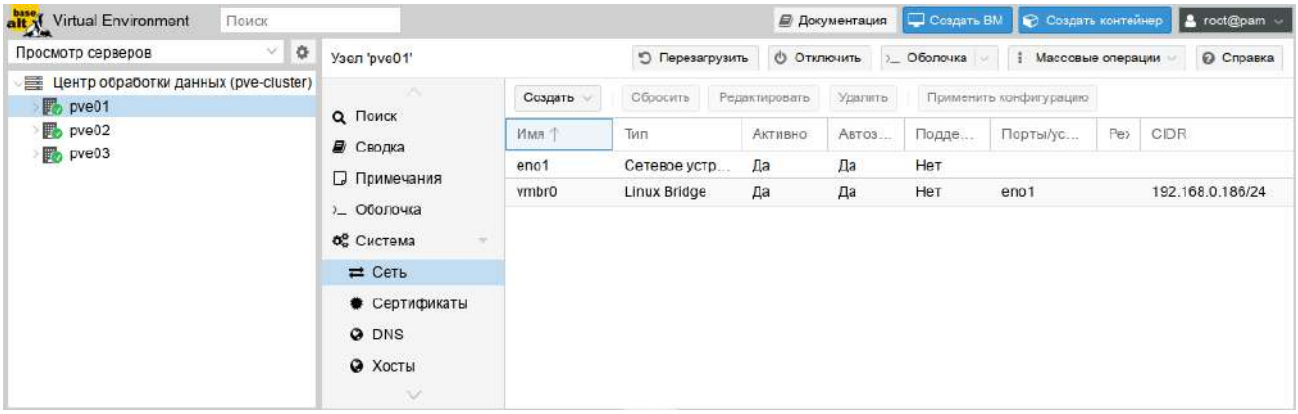


Рис. 132

Примечание. Интерфейс vmbr0 необходим для подключения гостевых систем к базовой физической сети. Это мост Linux, который можно рассматривать как виртуальный коммутатор, к которому подключены гостевые системы и физические интерфейсы.

Виды сетевых соединений в PVE (Рис. 133):

- «Linux Bridge» – способ соединения двух сегментов Ethernet на канальном уровне;
- «Linux Bond» – реализация агрегации нескольких сетевых интерфейсов в единый логический bonded интерфейс на базе ядра Linux;
- «Linux VLAN» – реализация VLAN на базе ядра Linux;
- «OVS Bridge» – реализация моста на базе Open vSwitch. Мосты Open vSwitch могут содержать необработанные устройства Ethernet, а также виртуальные интерфейсы OVSBonds или OVSIntPorts. Эти мосты могут нести несколько vlan и быть разбиты на «внутренние порты» для использования в качестве интерфейсов vlan на хосте. Все интерфейсы, входящие в мост, должны быть перечислены в опции `ovs_ports`;
- «OVS Bond» – реализация агрегации сетевых интерфейсов на базе Open vSwitch. Отличается от реализованной в ядре Linux режимами балансировки нагрузки;
- «OVS IntPort» – виртуальный сетевой интерфейс, предназначенный для взаимодействия узла PVE с определённой VLAN через OVS-мост.

Новый сетевой интерфейс

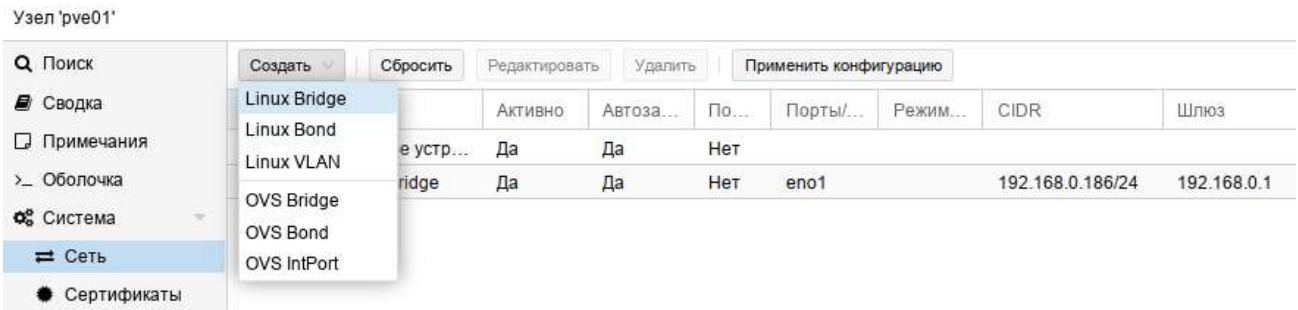


Рис. 133

Мосты, VLAN и агрегированные интерфейсы Open vSwitch и Linux не должны смешиваться. Например, не нужно добавлять Linux Bond к OVSBridge или наоборот.

4.5.1 Применение изменений сетевых настроек

Все изменения конфигурации сети, сделанные в веб-интерфейсе PVE, сначала записываются во временный файл, что позволяет сделать несколько связанных изменений одновременно. Это также позволяет убедиться, что изменения сделаны верно, так как неправильная конфигурация сети может сделать узел недоступным.

Для применения изменений сетевых настроек, сделанных в веб-интерфейсе PVE, следует нажать кнопку «Применить конфигурацию». В результате изменения будут применены в реальном времени.

Еще один способ применить новую сетевую конфигурацию – перезагрузить узел.

4.5.2 Имена сетевых устройств

В PVE используются следующие соглашения об именах устройств:

- устройства Ethernet: en*, имена сетевых интерфейсов systemd;
- мосты: vmbr[N], где $0 \le N \le 4094$ (vmbr0 — vmbr4094);
- сетевые объединения: bond[N], где $0 \le N$ (bond0, bond1, …);
- VLAN: можно просто добавить номер VLAN к имени устройства, отделив точкой (eno1.50, bond1.30).

Systemd использует префикс `en` для сетевых устройств Ethernet. Следующие символы зависят от драйвера устройства и того факта, какая схема подходит первой:

- o<index>[n<phys_port_name>|d<dev_port>] – встроенные устройства;
- s<slot>[f<function>][n<phys_port_name>|d<dev_port>] – устройства по идентификатору горячего подключения;
- [P<domain>]p<bus>s<slot>[f<function>][n<phys_port_name>|d<dev_port>] – устройства по идентификатору шины;
- x<MAC> – устройство по MAC-адресу.

Наиболее распространенные шаблоны:

- eno1 – первая сетевая карта;
- enp0s3 — сетевая карта в слоте 3 шины pcibus 0.

4.5.3 Конфигурация сети с использованием моста

Мосты похожи на физические сетевые коммутаторы, реализованные в программном обеспечении. Все виртуальные системы могут использовать один мост, также можно создать несколько мостов для отдельных сетевых доменов. На каждом хосте можно создать до 4094 мостов.

По умолчанию после установки на каждом узле PVE есть единственный мост (vmbr0), который подключается к первой плате Ethernet (Рис. 134).

Узлы PVE с мостом vmbr0

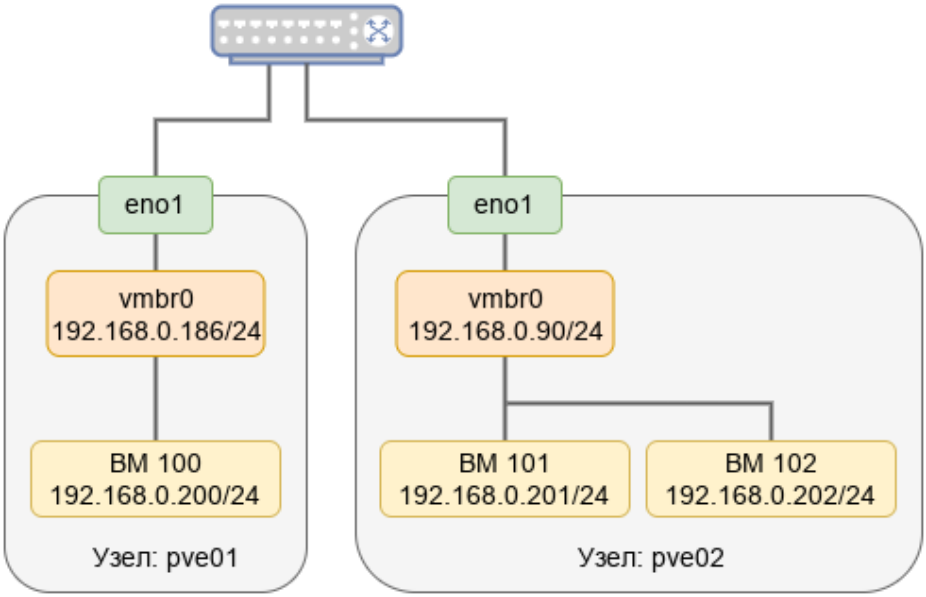


Рис. 134

При этом ВМ ведут себя так, как если бы они были напрямую подключены к физической сети. Каждая ВМ видна в сети со своим MAC-адресом.

4.5.3.1 Внутренняя сеть для ВМ

Если необходимо несколько ВМ объединить в локальную сеть без доступа во внешний мир, можно создать новый мост.

4.5.3.1.1 Настройка в веб-интерфейсе PVE

Для того чтобы создать мост, в разделе «Сеть» необходимо нажать кнопку «Создать» и в выпадающем меню выбрать пункт «Linux Bridge» или «OVS Bridge» (Рис. 133).

В открывшемся окне (Рис. 135) в поле «Имя» следует указать имя моста и нажать кнопку «Создать».

PVE. Создание Linux Bridge

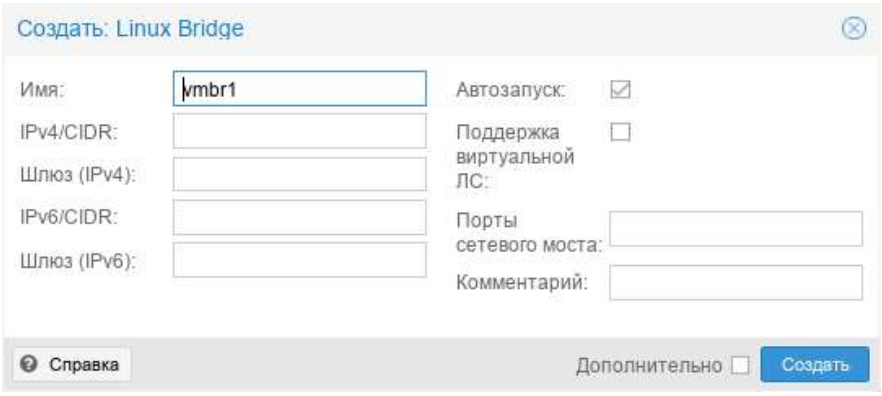


Рис. 135

Создание моста Open vSwitch (Рис. 136) отличается возможностью указания дополнительных параметров Open vSwitch (поле «Параметры OVS»).

PVE. Создание OVS Bridge

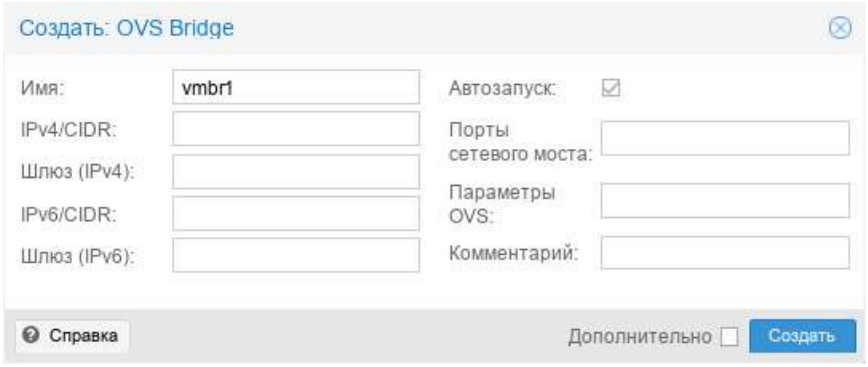


Рис. 136

Примечание. Адрес интерфейса можно не указывать, настроенные на подключение к интерфейсу ВМ будут использовать его как обычный коммутатор. Если же указать IPv4 и/или IPv6-адрес, то он будет доступен извне на интерфейсах, перечисленных в поле «Порты сетевого моста».

Для применения изменений следует нажать кнопку «Применить конфигурацию».

Теперь мост vmbr1 можно назначать ВМ (Рис. 137).

PVE. Назначение моста vmbr1 ВМ

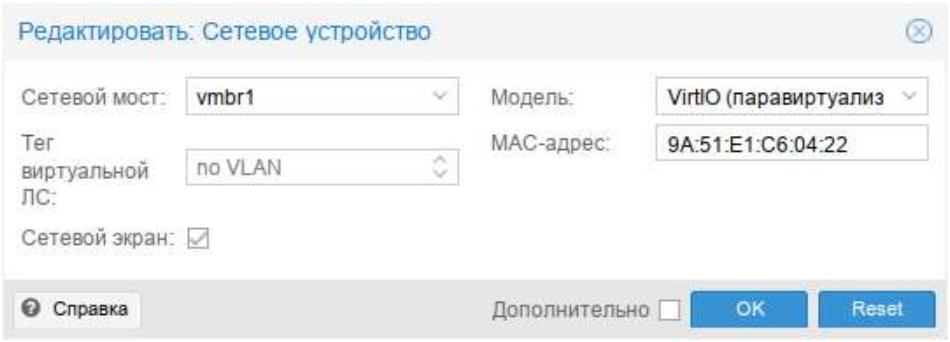


Рис. 137

4.5.3.1.2 Настройка в командной строке

Для настройки Linux bridge с именем vmbr1, следует выполнить следующие команды:

```
# mkdir /etc/net/ifaces/vmbr1
# cat <<EOF > /etc/net/ifaces/vmbr1/options
BOOTPROTO=static
CONFIG_IPV4=yes
HOST=
ONBOOT=yes
TYPE=bri
EOF
```

Примечание. Если в мост будут входить интерфейсы, которые до этого имели IP-адрес, то этот адрес должен быть удалён. Интерфейсы, которые будут входить в мост, должны быть указаны в опции HOST. Пример настройки моста vmbr1 на интерфейсе enp0s8 (IP-адрес для интерфейса vmbr1 будет взят из /etc/net/ifaces/enp0s8/ipv4address):

```
# mkdir /etc/net/ifaces/vmbr1
# cp /etc/net/ifaces/enp0s8/* /etc/net/ifaces/vmbr1/
# rm -f /etc/net/ifaces/enp0s8/{i,r}*
# cat <<EOF > /etc/net/ifaces/vmbr1/options
BOOTPROTO=static
CONFIG_WIRELESS=no
CONFIG_IPV4=yes
HOST='enp0s8'
ONBOOT=yes
TYPE=bri
EOF
```

Для настройки OVS bridge с именем vmbr1, следует выполнить следующие команды:

```
# mkdir /etc/net/ifaces/vmbr1
# cat <<EOF > /etc/net/ifaces/vmbr1/options
BOOTPROTO=static
CONFIG_IPV4=yes
ONBOOT=yes
TYPE=ovsbr
EOF
```

Пример настройки OVS bridge с именем vmbr1 на интерфейсе enp0s8:

```
# mkdir /etc/net/ifaces/vmbr1
# cp /etc/net/ifaces/enp0s8/* /etc/net/ifaces/vmbr1/
# rm -f /etc/net/ifaces/enp0s8/{i,r}*
# cat <<EOF > /etc/net/ifaces/vmbr1/options
BOOTPROTO=static
CONFIG_IPV4=yes
ONBOOT=yes
HOST='enp0s8'
TYPE=ovsbr
EOF
```

Для применения изменений необходимо перезапустить службу network:

```
# systemctl restart network
```

или перезагрузить узел:

```
# reboot
```

4.5.4 Объединение/агрегация интерфейсов

Объединение/агрегация интерфейсов (bonding) – это объединение двух и более сетевых интерфейсов в один логический интерфейс для достижения отказоустойчивости или увеличения пропускной способности. Поведение такого логического интерфейса зависит от выбранного режима работы.

Если на узлах PVE есть несколько портов Ethernet, можно распределить точки отказа, подключив сетевые кабели к разным коммутаторам, и в случае проблем с сетью агрегированное соединение переключится с одного кабеля на другой. Агрегация интерфейсов может сократить задержки выполнения миграции в реальном времени и повысить скорость репликации данных между узлами кластера PVE.

Кластерную сеть (Corosync) рекомендуется настраивать с несколькими сетями. Corosync не нуждается в агрегации для резервирования сети, поскольку может сам переключаться между сетями.

4.5.4.1 Параметры Linux Bond

В табл. 14 приведены режимы агрегации Linux Bond.

Т а б л и ц а 14 – Режимы агрегации Linux Bond

Режим	Название	Описание	Отказоустойчивость	Балансировка нагрузки
balance-rr или mode=0	Round-robin	Режим циклического выбора активного интерфейса для трафика. Пакеты последовательно передаются и принимаются через каждый интерфейс один за другим. Данный режим не требует применения специальных коммутаторов	Да	Да
active-backup или mode=1	Active Backup	В этом режиме активен только один интерфейс, остальные находятся в режиме горячей замены. Если активный интерфейс выходит из строя, его заменяет резервный. MAC-адрес интерфейса виден извне только на одном сетевом адаптере, что предотвращает путаницу в сетевом коммутаторе. Это самый простой режим, работает с любым оборудованием, не требует применения специальных коммутаторов	Да	Нет
balance-xor или mode=2	XOR	Один и тот же интерфейс работает с определённым получателем. Передача пакетов распределяется между интерфейсами на основе формулы ((MAC-адрес источника) XOR	Да	Да

Режим	Название	Описание	Отказоустойчивость	Балансировка нагрузки
		(MAC-адрес получателя)) % число интерфейсов. Режим не требует применения специальных коммутаторов. Этот режим обеспечивает балансировку нагрузки и отказоустойчивость		
broadcast или mode=3	Широковещательный	Трафик идёт через все интерфейсы одновременно	Да	Нет
LACP (802.3ad) или mode=4	Агрегирование каналов по стандарту IEEE 802.3ad	В группу объединяются одинаковые по скорости и режиму интерфейсы. Все физические интерфейсы используются одновременно в соответствии со спецификацией IEEE 802.3ad. Для реализации этого режима необходима поддержка на уровне драйверов сетевых карт и коммутатор, поддерживающий стандарт IEEE 802.3ad (коммутатор требует отдельной настройки)	Да	Да
balance-tlb или mode=5	Адаптивная балансировка нагрузки при передаче	Исходящий трафик распределяется в соответствии с текущей нагрузкой (с учетом скорости) на интерфейсах (для данного режима необходима его поддержка в драйверах сетевых карт). Входящие пакеты принимаются только активным сетевым интерфейсом	Да	Да (исходящий трафик)
balance-alb или mode=6	Адаптивная балансировка нагрузки	Включает в себя балансировку исходящего трафика, плюс балансировку на приём (rlb) для IPv4 трафика и не требует применения специальных коммутаторов (балансировка на приём достигается на уровне протокола ARP, перехватом ARP ответов локальной системы и перезаписью физического адреса на адрес одного из сетевых интерфейсов, в зависимости от загрузки)	Да	Да

В табл. 15 приведены алгоритмы выбора каналов (распределения пакетов между физическими каналами, входящими в многоканальное соединение) для режимов balance-alb, balance-tlb, balance-xor, 802.3ad (значение параметра `xmit_hash_policy`).

Т а б л и ц а 15 – Режимы выбора каналов при организации балансировки нагрузки

Режим	Описание
layer2	Канал для отправки пакета однозначно определяется комбинацией MAC-адреса источника и MAC-адреса назначения. Весь трафик между определённой парой узлов всегда идёт

	по определённому каналу. Алгоритм совместим с IEEE 802.3ad. Этот режим используется по умолчанию
layer2+3	Канал для отправки пакета определяется по совокупности MAC- и IP-адресов источника и назначения. Трафик между определённой парой IP-хостов всегда идёт по определённому каналу (обеспечивается более равномерная балансировка трафика, особенно в случае, когда большая его часть передаётся через промежуточные маршрутизаторы). Для протоколов 3 уровня, отличных от IP, данный алгоритм равносилен layer2. Алгоритм совместим с IEEE 802.3ad
layer3+4	Канал для отправки пакета определяется по совокупности IP-адресов и номеров портов источника и назначения (трафик определённого узла может распределяться между несколькими каналами, но пакеты одного и того же TCP/UDP-соединения всегда передаются по одному и тому же каналу). Для фрагментированных пакетов TCP и UDP, а также для всех прочих протоколов 4 уровня, учитываются только IP-адреса. Для протоколов 3 уровня, отличных от IP, данный алгоритм равносилен layer2. Алгоритм не полностью совместим с IEEE 802.3ad

Для создания агрегированного bond-интерфейса средствами etcnet необходимо создать каталог для интерфейса (например, bond0) с файлами `options`, `ipv4address`. В файле `options` в переменной TYPE следует указать тип интерфейса bond, в переменной HOST перечислить родительские интерфейсы, которые будут входить в агрегированный интерфейс, в переменной BONDMODE указать режим (по умолчанию 0), а опции для модуля ядра bonding – в BONDOPTIONS.

Примечание. Агрегированный bond-интерфейс можно создать в веб-интерфейсе ЦУС, подробнее см. «Объединение сетевых интерфейсов».

4.5.4.2 Параметры OVS Bond

В табл. 16 приведены параметры OVS Bond.

Т а б л и ц а 16 – Параметры OVS Bond

Параметр	Описание
bond_mode=active-backup	В этом режиме активен только один интерфейс, остальные находятся в режиме горячей замены. Если активный интерфейс выходит из строя, его заменяет резервный. MAC-адрес интерфейса виден извне только на одном сетевом адаптере, что предотвращает путаницу в сетевом коммутаторе. Этот режим не требует какой-либо специальной настройки на коммутаторах
bond_mode=balance-slb	Режим простой балансировки на основе MAC и VLAN. В этом режиме нагрузка трафика на интерфейсы постоянно измеряется, и если один из интерфейсов сильно загружен, часть трафика перемещается на менее загруженные интерфейсы. Параметр bond-rebalance-interval определяет, как часто OVS должен выполнять измерение нагрузки трафика (по умолчанию 10 секунд). Этот режим не требует какой-либо специальной настройки на коммутаторах
bond_mode=balance-tcp	Этот режим выполняет балансировку нагрузки, принимая во внимание данные уровней 2-4 (например, MAC-адрес, IP -адрес и порт TCP). На коммутаторе должен быть настроен LACP. Этот режим похож на режим mode=4 Linux Bond. Всегда, когда это возможно, рекомендуется использовать этот режим
lacp=[active|passive|off]	Управляет поведением протокола управления агрегацией каналов (LACP). На коммутаторе должен быть настроен протокол LACP. Если коммутатор не поддерживает LACP, необходимо использовать bond_mode=balance-slb или bond_mode=active-backup
other-config:lacp-fall-back-ab=true	Устанавливает поведение LACP для переключения на bond_mode=active-backup в качестве запасного варианта
other_config:lacp-time=[fast|slow]	Определяет, с каким интервалом управляющие пакеты LACPDU отправляются по каналу LACP: каждую секунду (fast) или каждые 30 секунд (slow). По умолчанию slow
other_config:bond-detect-mode=[miimon|carrier]	Режим определения состояния канала. По умолчанию carrier
other_config:bond-miimon-interval=100	Устанавливает периодичность MII мониторинга в миллисекундах
other_config:bond_updelay=1000	Задает время задержки в миллисекундах, перед тем как поднять линк при обнаружении восстановления канала
other_config:bond-rebalance-interval=10000	Устанавливает периодичность выполнения измерения нагрузки трафика в миллисекундах (по умолчанию 10 секунд)

4.5.4.2.1 Агрегированный bond-интерфейс с фиксированным IP-адресом

Конфигурация с агрегированным bond-интерфейсом с фиксированным IP-адресом может использоваться как распределенная/общая сеть хранения. Преимущество будет заключаться в том, что вы получите больше скорости, а сеть будет отказоустойчивой (Рис. 138).

Агрегированный bond-интерфейс с фиксированным IP-адресом

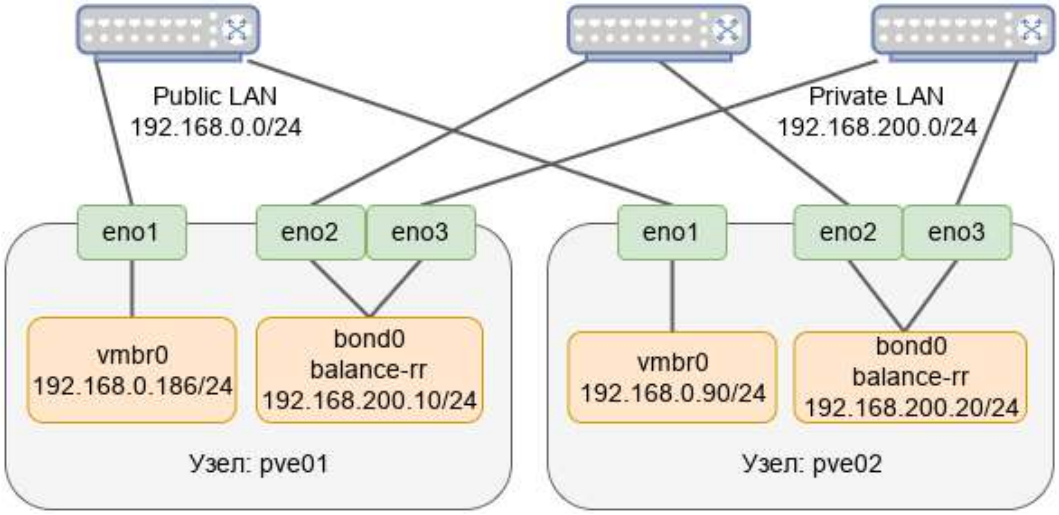


Рис. 138

4.5.4.2.1.1 Настройка в веб-интерфейсе

Для настройки Linux Bond необходимо выполнить следующие действия:

1) перейти в раздел «Сеть», нажать кнопку «Создать» и в выпадающем меню выбрать пункт «Linux Bond» (Рис. 133);

2) в открывшемся окне указать имя агрегированного соединения, в выпадающем списке «Режим» выбрать режим агрегации (в примере balance-rr), в поле «Устройства» указать сетевые интерфейсы, которые будут входить в объединение, в поле «IPv4/CIDR» ввести IP-адрес объединения и нажать кнопку «Создать» (Рис. 139);

Примечание. В зависимости от выбранного режима агрегации будут доступны разные поля.

3) для применения изменений нажать кнопку «Применить конфигурацию»;

4) получившаяся конфигурация показана на Рис. 140.

Редактирование параметров объединения bond0

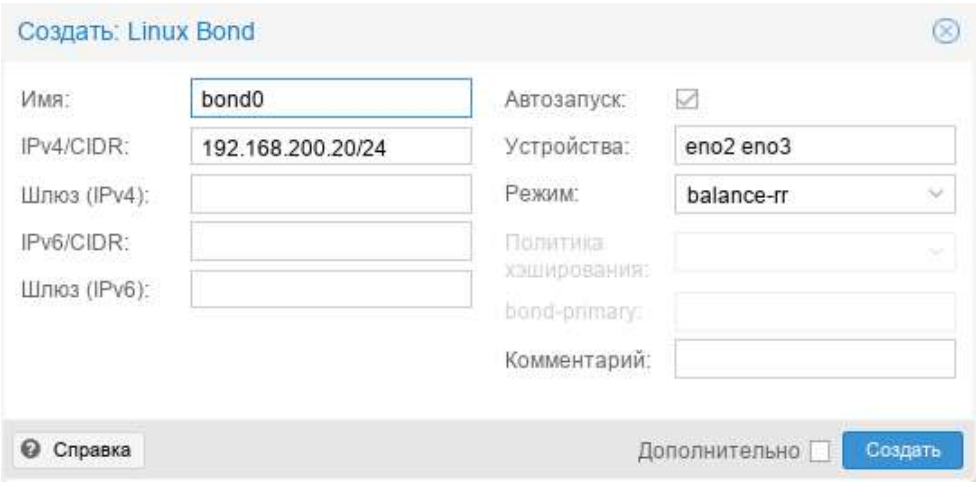


Рис. 139

Агрегированный интерфейс с фиксированным IP-адресом

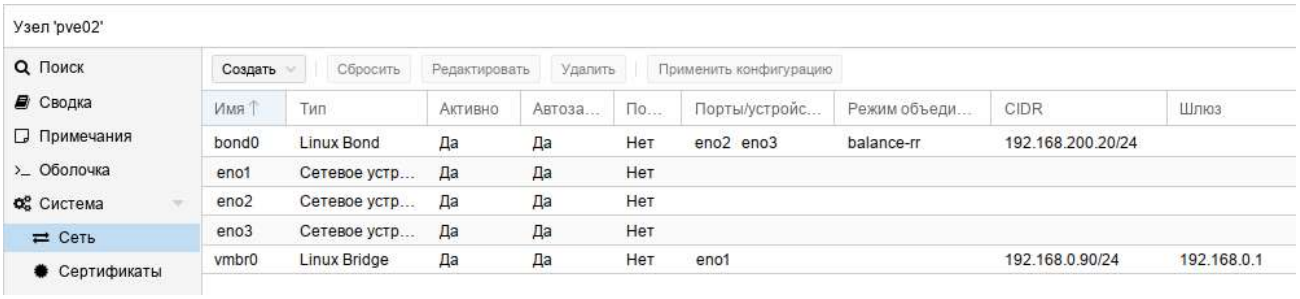


Рис. 140

4.5.4.2.1.2 Настройка в командной строке

Для создания такой конфигурации необходимо выполнить следующие действия:

1) создать Linux Bond bond0 на интерфейсах eno1 и eno2, выполнив следующие команды:

```
# mkdir /etc/net/ifaces/bond0
# rm -f /etc/net/ifaces/eno1/{i,r}*
# rm -f /etc/net/ifaces/eno2/{i,r}*
# cat <<EOF > /etc/net/ifaces/bond0/options
BOOTPROTO=static
CONFIG_WIRELESS=no
CONFIG_IPV4=yes
HOST='eno1 eno2'
ONBOOT=yes
TYPE=bond
BONDOPTIONS='miimon=100'
BONDMODE=0
EOF
```

где:

- BONDMODE=1 – режим агрегации Round-robin;
- HOST='eno1 eno2' – интерфейсы, которые будут входить в объединение;
- miimon=100 – определяет, как часто производится мониторинг MII (Media Independent Interface).

2) в файле `/etc/net/ifaces/bond0/ipv4address`, указать IP-адрес для интерфейса bond0:

```
# echo "192.168.200.20/24" > /etc/net/ifaces/bond0/ipv4address
```

3) перезапустить службу network, чтобы изменения вступили в силу:

```
# systemctl restart network
```

4.5.4.2.2 Агрегированный bond-интерфейс в качестве порта моста

Чтобы сделать гостевую сеть отказоустойчивой можно использовать bond напрямую в качестве порта моста (Рис. 141).

Агрегированный bond-интерфейс в качестве порта моста

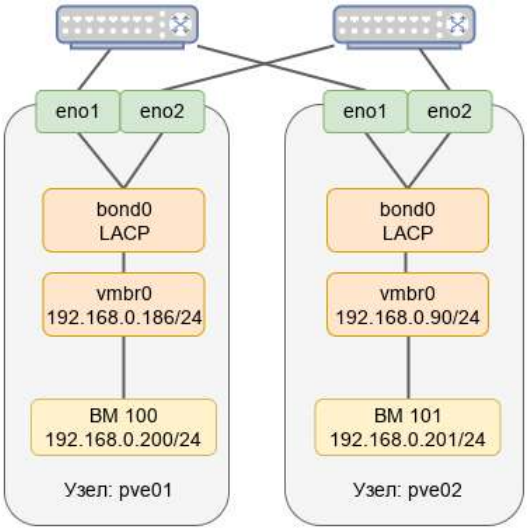


Рис. 141

4.5.4.2.2.1 Настройка в веб-интерфейсе

Для настройки Linux Bond необходимо выполнить следующие действия:

1) перейти в раздел «Сеть», выбрать существующий мост vmbr0 и нажать кнопку «Редактировать» (Рис. 142);

2) в открывшемся окне (Рис. 143) удалить содержимое поля «Порты сетевого моста» и нажать кнопку «ОК»;

3) нажать кнопку «Создать» (Рис. 133) и в выпадающем меню выбрать пункт «Linux Bond».

4) в открывшемся окне в выпадающем списке «Режим» выбрать режим агрегации (в примере LACP), в поле «Устройства» указать сетевые интерфейсы, которые будут входить в объединение, в выпадающем списке «Политика хэширования» выбрать политику хэширования и нажать кнопку «Создать» (Рис. 144);

5) выбрать мост vmbr0 и нажать кнопку «Редактировать».

6) в открывшемся окне в поле «Порты сетевого моста» вписать значение bond0 и нажать кнопку ОК (Рис. 145);

7) для применения изменений нажать кнопку «Применить конфигурацию».

8) получившаяся конфигурация показана на Рис. 146.

Мост vmbr0

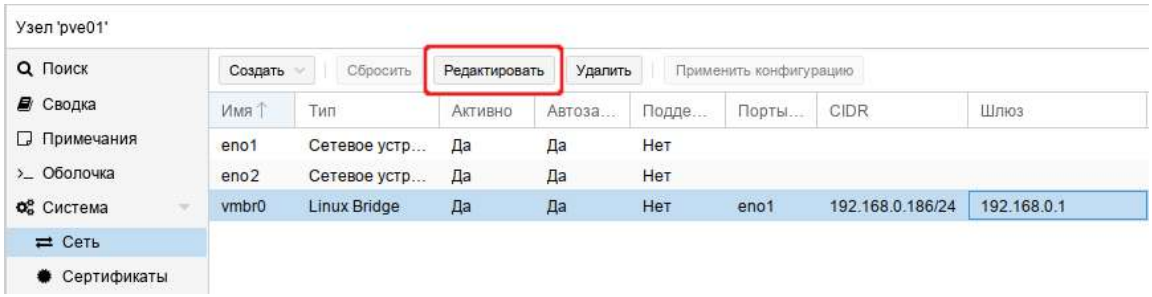


Рис. 142

Редактирование параметров моста vmbr0

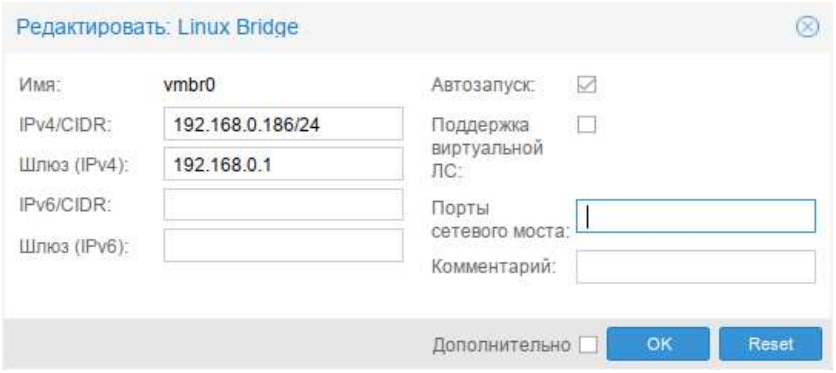


Рис. 143

Редактирование параметров объединения bond0

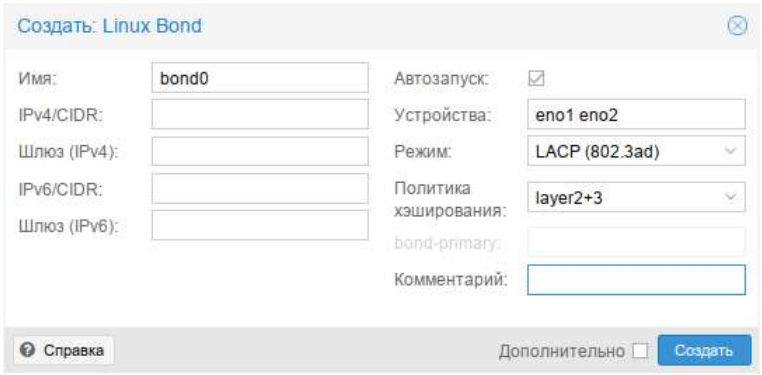


Рис. 144

Редактирование параметров моста vmbr0

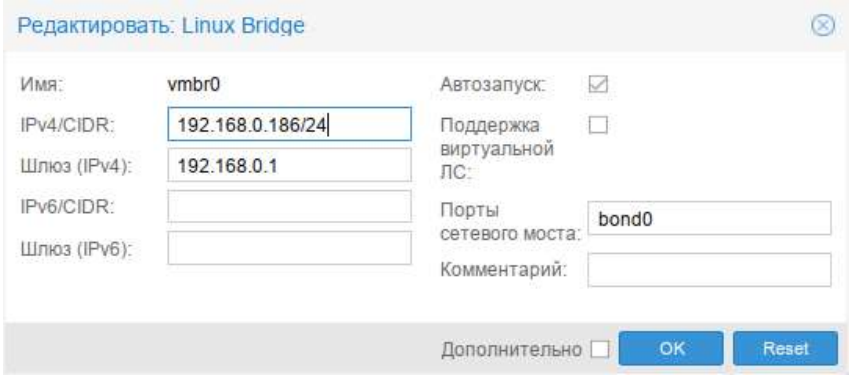


Рис. 145

Агрегированный bond-интерфейс в качестве порта моста

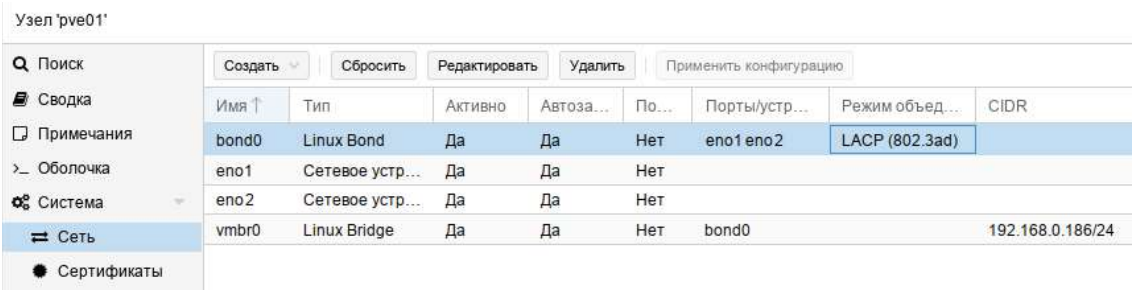


Рис. 146

Для настройки OVS Bond необходимо выполнить следующие действия:

1) перейти в раздел «Сеть», выбрать существующий мост vmbr0 и нажать кнопку «Редактировать» (Рис. 147);

2) в открывшемся окне удалить содержимое поля «Порты сетевого моста» и нажать кнопку «ОК» (Рис. 148);

3) нажать кнопку «Создать» (Рис. 133) и в выпадающем меню выбрать пункт «OVS Bond».

4) в открывшемся окне указать имя агрегированного интерфейса, в выпадающем списке «Режим» выбрать режим агрегации, в поле «Устройства» указать сетевые интерфейсы, которые будут входить в объединение, в выпадающем списке «OVS Bridge» выбрать мост, в который должен добавиться созданный интерфейс и нажать кнопку «Создать» (Рис. 149);

5) для применения изменений нажать кнопку «Применить конфигурацию»;

6) получившаяся конфигурация показана на Рис. 150.

Мост vmbr0

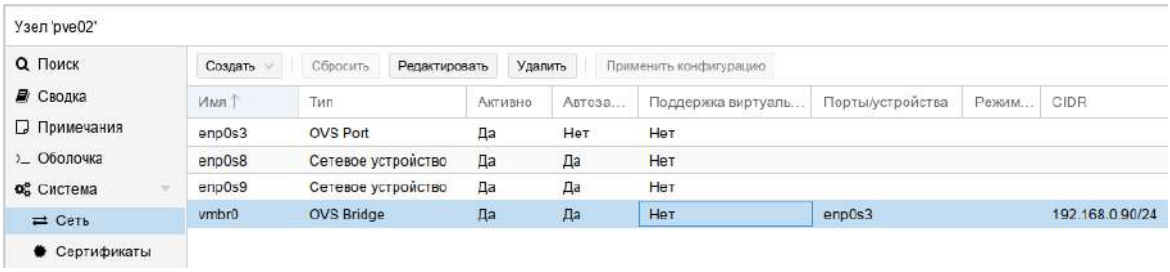


Рис. 147

Редактирование параметров моста vmbr0

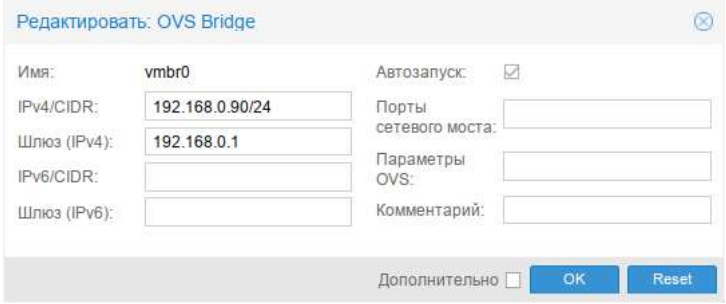


Рис. 148

Редактирование параметров объединения bond0

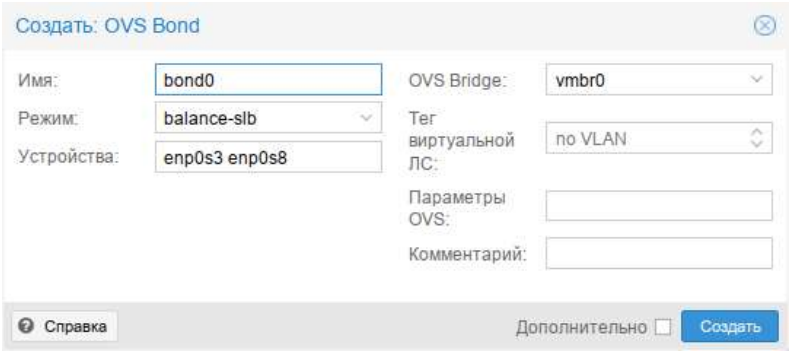


Рис. 149

Агрегированный bond-интерфейс в качестве порта моста

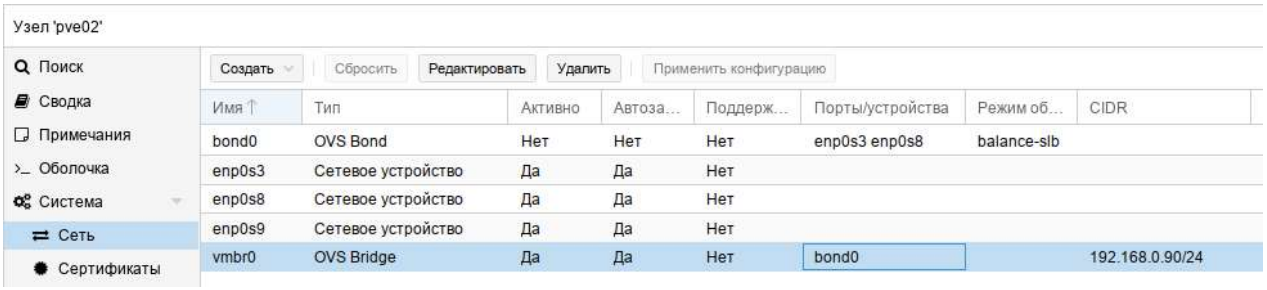


Рис. 150

4.5.4.2.2.2 Настройка в командной строке

Исходное состояние: мост vmbr0 на интерфейсе enp0s3. Необходимо создать агрегированный интерфейс bond0 (enp0s3 и enp0s8) и включить его в мост vmbr0.

Для создания Linux Bond необходимо выполнить следующие действия:

1) создать агрегированный интерфейс bond0 на интерфейсах enp0s3 и enp0s8, выполнив следующие команды:

```
# mkdir /etc/net/ifaces/bond0
# cat <<EOF > /etc/net/ifaces/bond0/options
BOOTPROTO=static
CONFIG_WIRELESS=no
CONFIG_IPV4=yes
HOST='enp0s3 enp0s8'
ONBOOT=yes
TYPE=bond
BONDOPTIONS='xmit_hash_policy=layer2+3 lacp_rate=1 miimon=100'
BONDMODE=4
EOF
```

где:

- BONDMODE=4 – режим агрегации LACP (802.3ad);
- HOST='enp0s3 enp0s8' – интерфейсы, которые будут входить в объединение;

- xmit_hash_policy=layer2+3 – определяет режим выбора каналов;
- lacp_rate=1 – определяет, что управляющие пакеты LACPDU отправляются по каналу LACP каждую секунду;
- miimon=100 – определяет, как часто производится мониторинг MII (Media Independent Interface).

2) в настройках Ethernet-моста vmbr0 (файл `/etc/net/ifaces/vmbr0/options`) в опции HOST указать интерфейс bond0:

```
BOOTPROTO=static
BRIDGE_OPTIONS="stp_state 0"
CONFIG_IPV4=yes
HOST='bond0'
ONBOOT=yes
TYPE=bri
```

3) перезапустить службу network, чтобы изменения вступили в силу:

```
# systemctl restart network
```

Для создания OVS Bond необходимо выполнить следующие действия:

1) начальная конфигурации:

```
# ovs-vsctl show
6b1add02-fb20-48e6-b925-260bf92fa889
    Bridge vmbr0
        Port enp0s3
            Interface enp0s3
        Port vmbr0
            Interface vmbr0
                type: internal
    ovs_version: "2.17.6"
```

2) привести настройки сетевого интерфейса enp0s3 (файл `/etc/net/ifaces/enp0s3/options`) к виду:

```
TYPE=eth
CONFIG_WIRELESS=no
BOOTPROTO=static
CONFIG_IPV4=yes
```

3) создать агрегированный интерфейс bond0 на интерфейсах enp0s3 и enp0s8, выполнив следующие команды:

```
# mkdir /etc/net/ifaces/bond0
# cat <<EOF > /etc/net/ifaces/bond0/options
BOOTPROTO=static
BRIDGE=vmbr0
CONFIG_IPV4=yes
```

```
HOST='enp0s3 enp0s8'
OVS_OPTIONS='other_config:bond-miimon-interval=100 bond_mode=balance-slb'
TYPE=ovsbond
EOF
```

где:

- bond_mode=balance-slb – режим агрегации;
- HOST='enp0s3 enp0s8' – интерфейсы, которые будут входить в объединение;
- BRIDGE=vmbr0 – мост, в который должен добавиться созданный интерфейс;
- other_config:bond-miimon-interval=100 – определяет, как часто производится мониторинг MII (Media Independent Interface).

4) В опции HOST настроек моста vmbr0 (файл `/etc/net/ifaces/vmbr0/options`) указать bond0:

```
BOOTPROTO=static
CONFIG_IPV4=yes
HOST=bond0
ONBOOT=yes
TYPE=ovsbr
```

5) перезапустить службу network, чтобы изменения вступили в силу:

```
# systemctl restart network
```

6) проверка конфигурации:

```
# ovs-vsctl show
6b1add02-fb20-48e6-b925-260bf92fa889
    Bridge vmbr0
        Port bond0
            Interface enp0s3
            Interface enp0s8
        Port vmbr0
            Interface vmbr0
                type: internal
    ovs_version: "2.17.6"
```

4.5.5 Настройка VLAN

Виртуальные локальные сети (VLAN) – это сетевой стандарт IEEE 802.1Q, позволяющий создавать логически изолированные сегменты сети на одном физическом интерфейсе для разделения трафика между разными сетями.

Примечание. На стороне физического коммутатора порт должен быть настроен как trunk, от него должен приходить тегированный трафик 802.1Q. Если на коммутаторе сделана агрегация портов (Portchannel или Etherchannel), то параметр `Trunk` выставляется на это новом интерфейсе.

4.5.5.1 Мост с поддержкой VLAN

Если используется Linux Bridge, то для возможности использования тегов VLAN в настройках ВМ, необходимо включить поддержку VLAN для моста. Для этого в веб-интерфейсе в настройках моста следует установить отметку в поле «Поддержка виртуальной ЛС» (Рис. 151).

Настройки моста Linux Bridge

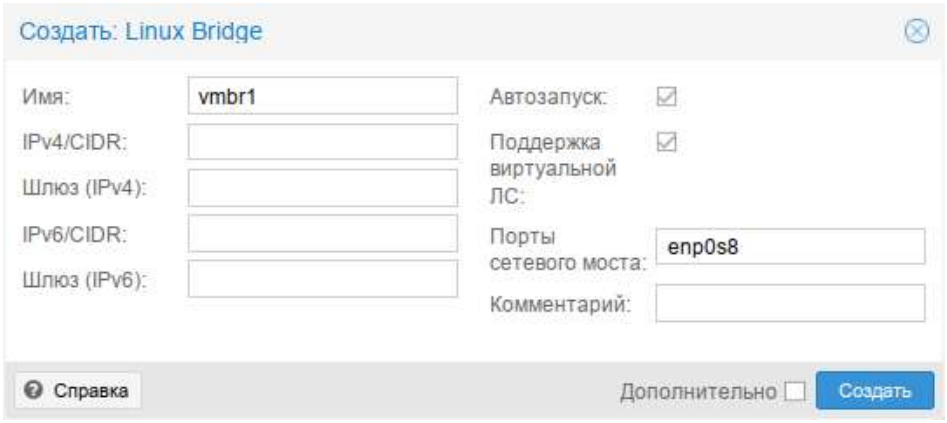


Рис. 151

Если используется OVS Bridge, то никаких дополнительных настроек не требуется.

Тег VLAN можно указать в настройках сетевого интерфейса при создании ВМ (Рис. 152), либо отредактировав параметры сетевого устройства.

Настройки сетевого интерфейса ВМ

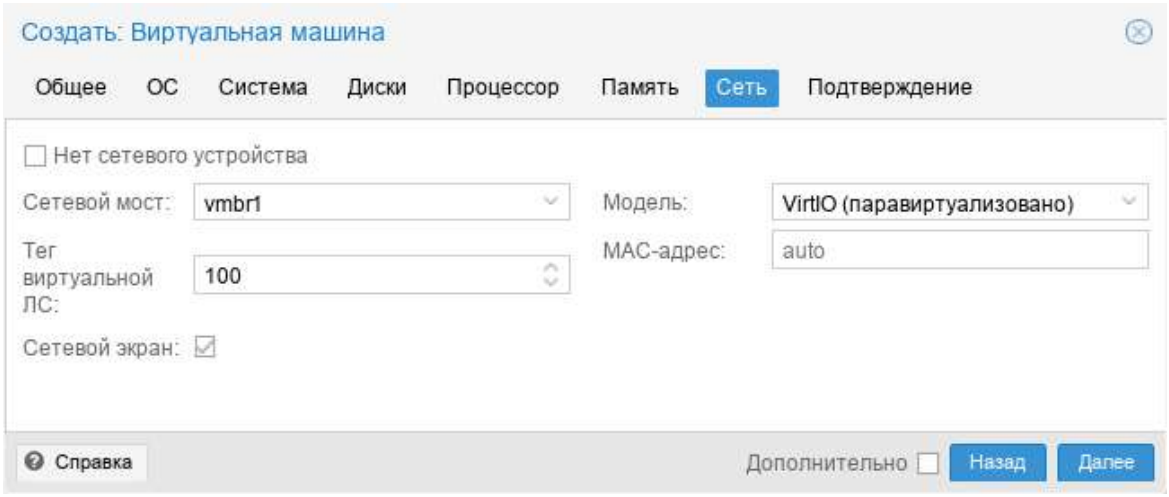


Рис. 152

4.5.5.2 Мост на VLAN

Можно создать конфигурацию VLAN <интерфейс>.<vlan tag> (например, enp0s8.100), этот VLAN включить в мост Linux Bridge и указывать этот мост в настройках сетевого интерфейса ВМ.

Для создания такой конфигурации необходимо выполнить следующие действия:

1) настроить VLAN с ID 100 на интерфейсе enp0s8, выполнив следующие команды (в опции HOST нужно указать тот интерфейс, на котором будет настроен VLAN):

```
# mkdir /etc/net/ifaces/enp0s8.100
# cat <<EOF > /etc/net/ifaces/enp0s8.100/options
BOOTPROTO=static
CONFIG_WIRELESS=no
CONFIG_IPV4=yes
HOST=enp0s8
ONBOOT=yes
TYPE=vlan
VID=100
EOF
```

2) настроить Ethernet-мост vmbr1, выполнив следующие команды (в опции HOST нужно указать VLAN-интерфейс):

```
# mkdir /etc/net/ifaces/vmbr1
# cat <<EOF > /etc/net/ifaces/vmbr1/options
BOOTPROTO=static
CONFIG_WIRELESS=no
CONFIG_IPV4=yes
HOST='enp0s8.100'
ONBOOT=yes
TYPE=bri
EOF
```

3) в файле `/etc/net/ifaces/vmbr1/ipv4address`, если это необходимо, можно указать IP-адрес для интерфейса моста:

```
# echo "192.168.10.3/24" > /etc/net/ifaces/vmbr1/ipv4address
```

4) перезапустить службу network, чтобы изменения вступили в силу:

```
# systemctl restart network
```

Теперь в настройках сетевого интерфейса ВМ можно указать сетевой мост vmbr1 (Рис. 153). Трафик через этот интерфейс будет помечен тегом 100.

Настройки сетевого интерфейса ВМ

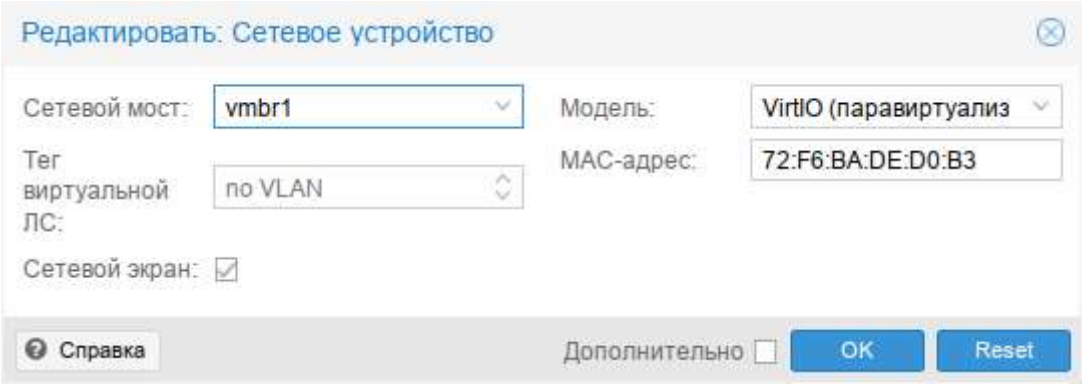


Рис. 153

4.6 Управление ISO-образами и шаблонами LXC

Для загрузки ISO-образов и шаблонов LXC в хранилище PVE следует выполнить следующие шаги:

1) выбрать хранилище;

2) перейти на вкладку «ISO-образы» для загрузки ISO-образов (Рис. 154) или на вкладку «Шаблоны контейнеров» для загрузки шаблонов LXC;

3) нажать кнопку «Шаблоны». В открывшемся окне нажать кнопку «Выбрать файл», выбрать файл с ISO-образом и нажать кнопку «Отправить» (Рис. 155). Здесь же можно выбрать алгоритм и указать контрольную сумму. В этом случае после загрузки образа будет проверена его контрольная сумма;

4) для загрузки образа (шаблона) с сервера следует нажать кнопку «Загрузить по URL-адресу». В открывшемся окне указать ссылку на образ (шаблон), нажать кнопку «Запрос URL-адреса», для того чтобы получить метаинорфмацию о файле, нажать кнопку «Загрузка» для старта загрузки файла в хранилище (Рис. 156).

Локальное хранилище. Вкладка «ISO-образы»

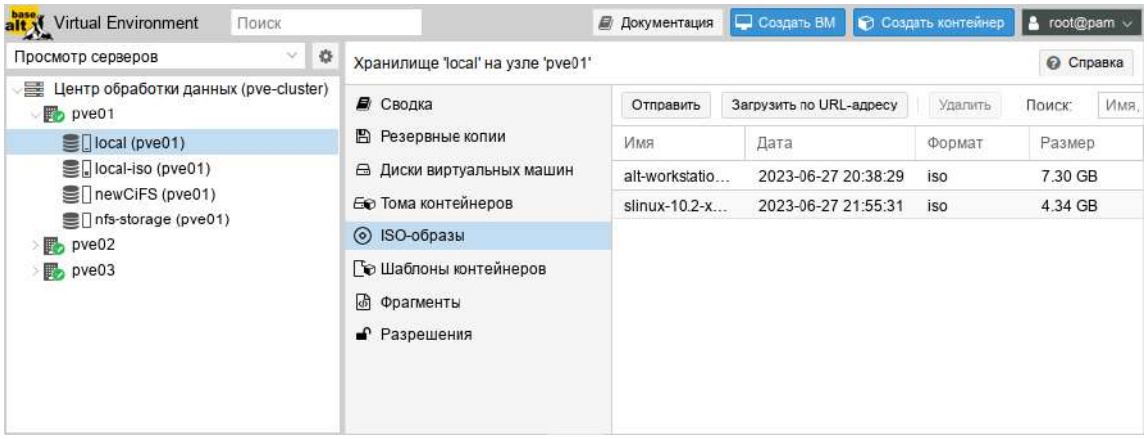


Рис. 154

Выбор образа

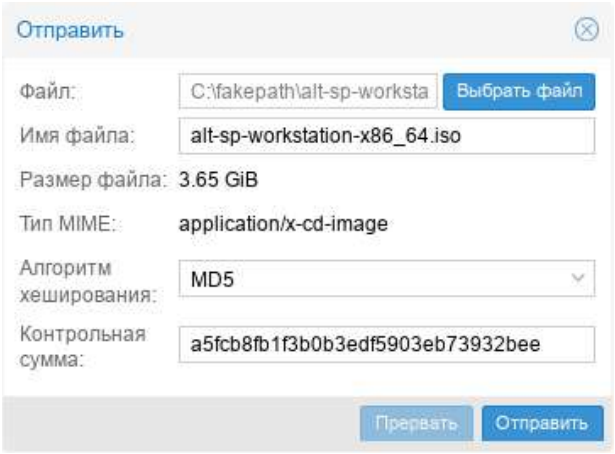


Рис. 155

Выбор образа для загрузки файла с сервера

Рис. 156

Для удаления ISO-образа или шаблона LXC следует выбрать файл из списка в хранилище (Рис. 154) и нажать кнопку «Удалить».

PVE предоставляет базовые шаблоны для некоторых дистрибутивов Linux. Эти шаблоны можно загрузить в веб-интерфейсе (кнопка «Шаблоны») или в командной строке (утилита `pveam`).

Загрузка базового шаблона в веб-интерфейсе:

1) запустить обновление списка доступных шаблонов (например, на вкладке «Оболочка»):

```
# pveam update
```

2) выбрать хранилище;

3) перейти на вкладку «Шаблоны контейнеров» и нажать кнопку «Шаблоны» (Рис. 157);

4) в открывшемся окне выбрать шаблон и нажать кнопку «Загрузка» (Рис. 158).

Вкладка «Шаблоны контейнеров»

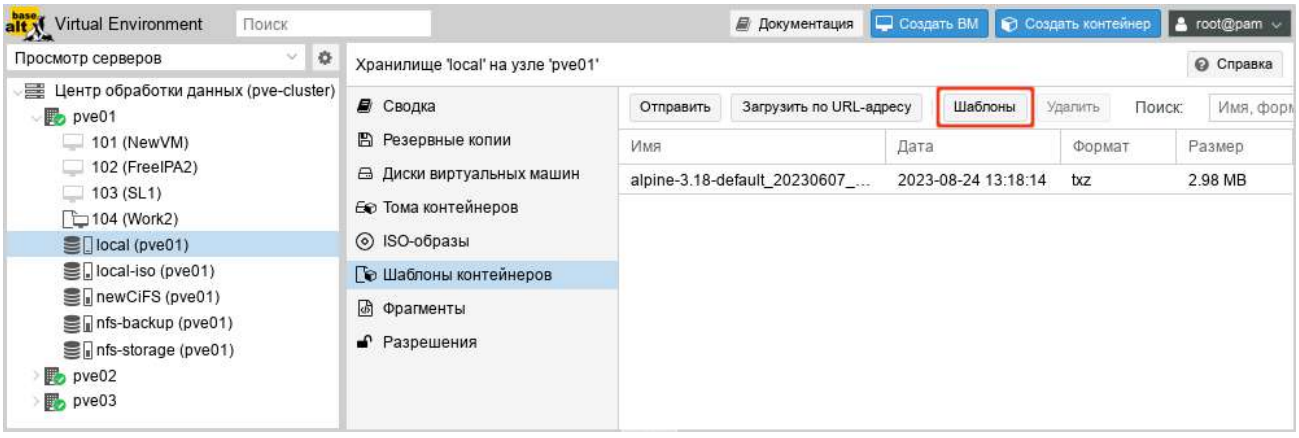


Рис. 157

Выбор шаблона для загрузки

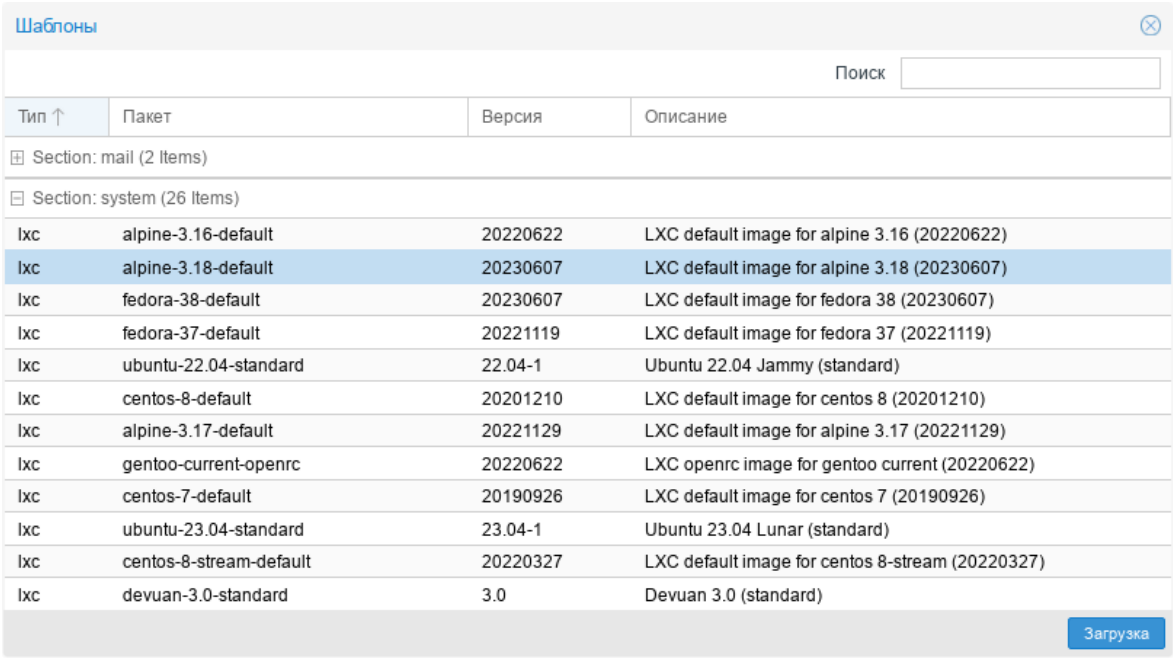


Рис. 158

Загрузка базового шаблона в консоли:

1) запустить обновление списка доступных шаблонов:

```
# pveam update
update successful
```

2) просмотреть список доступных шаблонов:

```
# pveam available
mail              proxmox-mailgateway-7.3-standard_7.3-1_amd64.tar.zst
mail              proxmox-mailgateway-8.0-standard_8.0-1_amd64.tar.zst
system            almalinux-8-default_20210928_amd64.tar.xz
system            almalinux-9-default_20221108_amd64.tar.xz
system            alpine-3.16-default_20220622_amd64.tar.xz
…
```

3) загрузить шаблон в хранилище `local`:

```
# pveam download local almalinux-9-default_20221108_amd64.tar.xz
```

4) просмотреть список загруженных шаблонов в хранилище `local`:

```
# pveam list local
NAME                                                         SIZE
local:vztmpl/almalinux-9-default_20221108_amd64.tar.xz       97.87MB
```

Если используются только локальные хранилища, то ISO-образы и шаблоны необходимо загрузить на все узлы в кластере. Если есть общее хранилище, то можно хранить все образы в одном месте, таким образом, сохраняя пространство локальных хранилищ.

В таблице 17 показаны каталоги для локального хранилища. В таблице 18 показаны каталоги для всех других хранилищ.

Т а б л и ц а 17 – Каталоги локального хранилища

Каталог	Тип шаблона
`/var/lib/vz/template/iso`	ISO-образы
`/var/lib/vz/template/cache`	Шаблоны контейнеров LXC

Т а б л и ц а 18 – Каталоги общих хранилищ

Каталог	Тип шаблона
`/mnt/pve/<storage_name>/template/iso`	ISO-образы
`/mnt/pve/<storage_name>/template/cache`	Шаблоны контейнеров LXC

4.7 Виртуальные машины на базе KVM

4.7.1 Создание виртуальной машины на базе KVM

Прежде чем создать в интерфейсе PVE виртуальную машину (ВМ), необходимо определиться со следующими моментами:

- откуда будет загружен инсталлятор ОС, которая будет установлена внутрь ВМ;
- на каком физическом узле будет выполняться процесс гипервизора kvm;
- в каком хранилище будут располагаться образы дисков ВМ.

Все остальные параметры ВМ относятся к конфигурации виртуального компьютера и могут быть определены по ходу процесса создания ВМ (PVE пытается выбрать разумные значения по умолчанию для ВМ).

Чтобы установить ОС на ВМ, расположенную на этом узле, нужно обеспечить возможность загрузки инсталлятора на этой ВМ. Для этого необходимо загрузить ISO-образ инсталлятора в хранилище данных выбранного физического узла или общее хранилище. Это можно сделать через веб-интерфейс (Рис. 154).

Для создания ВМ необходимо нажать кнопку «Создать ВМ», расположенную в правом верхнем углу веб-интерфейса PVE (Рис. 159).

Кнопка «Создать ВМ»

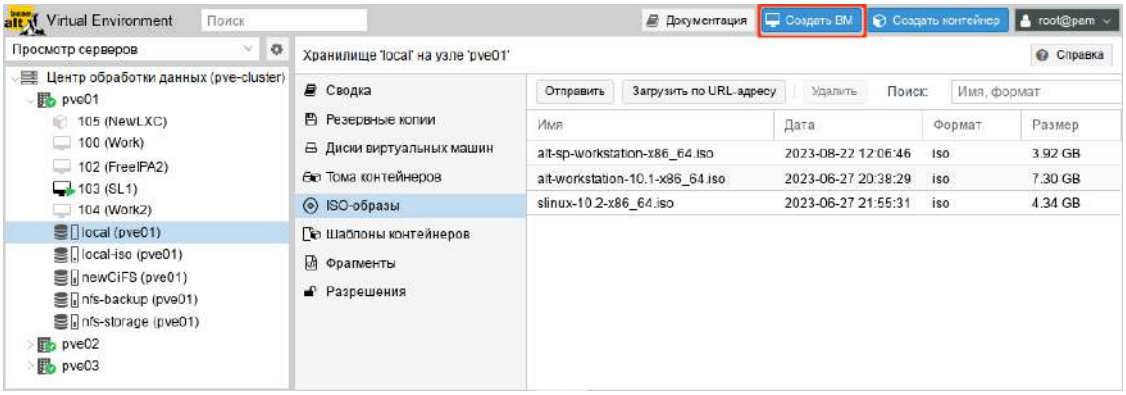


Рис. 159

Процесс создания ВМ оформлен в виде «мастера», привычного для пользователей систем управления ВМ.

На вкладке «Общее» необходимо указать (Рис. 160):

- «Узел» – физический сервер, на котором будет работать ВМ;
- «VM ID» – идентификатор ВМ в численном выражении. Одно и то же значение идентификатора не может использоваться более чем для одной машины. Поле идентификатора ВМ заполняется автоматически инкрементально: первая созданная ВМ, по умолчанию будет иметь VM ID со значением 100, следующая 101 и так далее;
- «Имя» – текстовая строка названия ВМ;
- «Пул ресурсов» – логическая группа ВМ. Чтобы иметь возможность выбора, пул должен быть предварительно создан.

Вкладка «Общее»

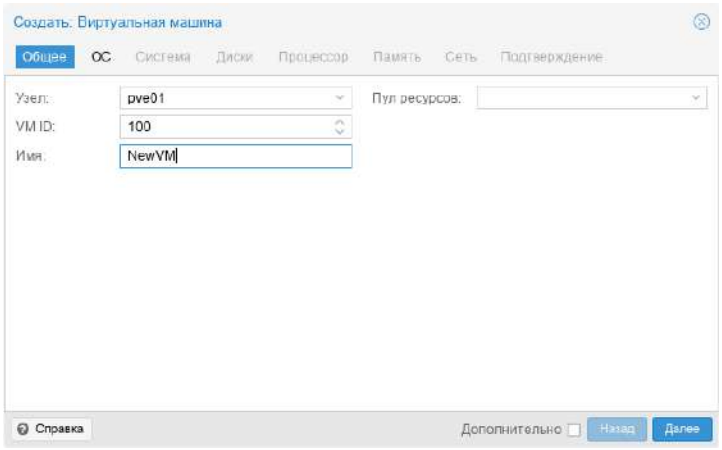


Рис. 160

Примечание. Настроить диапазон, из которого выбираются новые VM ID при создании ВМ или контейнера можно, выбрав на вкладке «Центр обработки данных» → «Параметры» пункт «Следующий свободный диапазон ID виртуальных машин» (Рис. 161). Установка нижнего значения («Нижний предел») равным верхнему («Верхний предел») полностью отключает автоподстановку VM ID.

Настройка диапазона VM ID

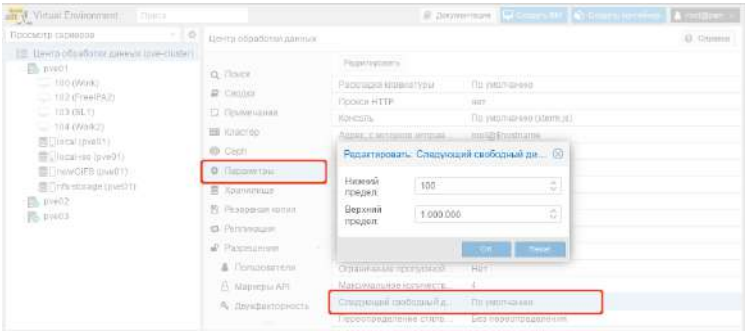


Рис. 161

На вкладке «ОС» необходимо указать источник установки ОС и тип ОС (Рис. 162).

Вкладка «ОС»

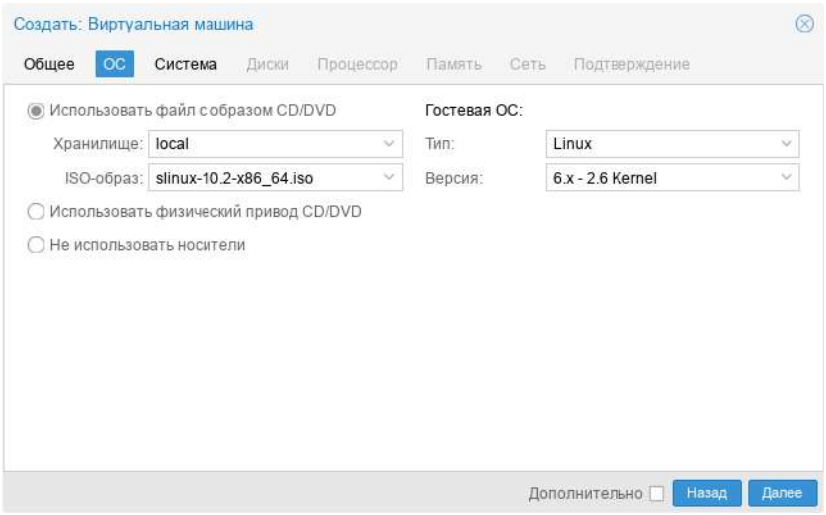


Рис. 162

В качестве источника установки ОС можно указать:

- «Использовать файл с образом CD/DVD»– использовать уже загруженный в хранилище ISO-образ (Рис. 163);
- «Использовать физический привод CD/DVD» – использовать физический диск хоста PVE;
- «Не использовать носители» – не использовать ISO-образ или физический носитель.

Выбор типа гостевой ОС при создании ВМ позволяет PVE оптимизировать некоторые параметры низкого уровня.

Выбор ISO-образа

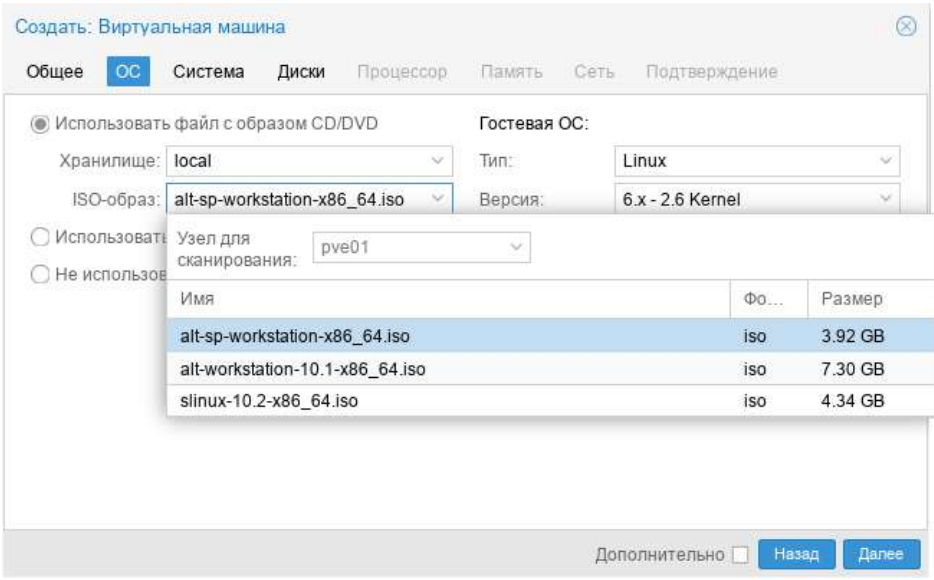


Рис. 163

На следующем этапе (вкладка «Система») можно выбрать видеокарту, контроллер SCSI, указать нужно ли использовать агент QEMU (Рис. 164).

Вкладка «Система»

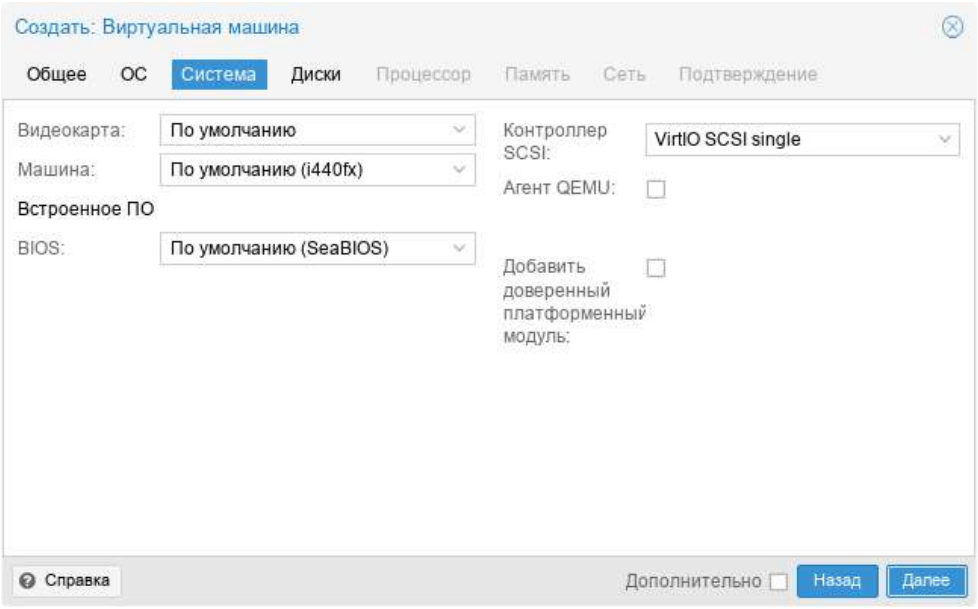


Рис. 164

Подробнее о выборе видеокарты см. «Настройки дисплея».

PVE позволяет загружать ВМ с разными прошивками (SeaBIOS и OVMF). Прошивку OVMF следует выбирать, если планируется использовать канал PCIe. При выборе прошивки OVMF (Рис. 165) для сохранения порядка загрузки, должен быть добавлен диск EFI (см. «BIOS и UEFI»).

Выбор прошивки OVMF

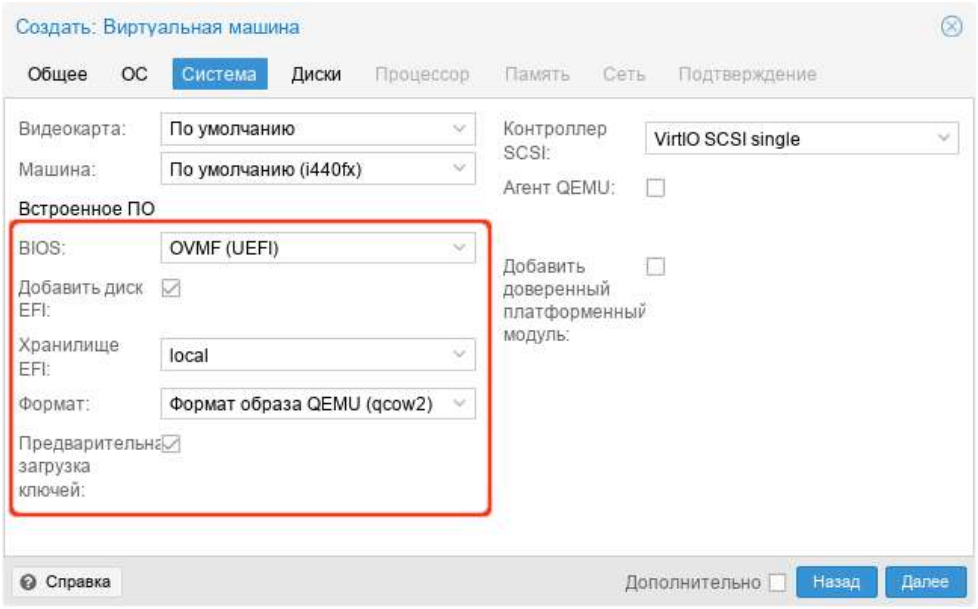


Рис. 165

Тип машины ВМ определяет аппаратную компоновку виртуальной материнской платы ВМ. Доступно два варианта набора микросхем: Intel 440FX (по умолчанию) и Q35 (предоставляет виртуальную шину PCIe).

Вкладка «Диски» содержит следующие настройки (Рис. 166):

- «Шина/Устройство» – тип устройства виртуального диска. Допустимые значения: «IDE», «SATA», «VirtIO Block» и «SCSI» (по умолчанию). Можно также указать идентификатор устройства;
- «Хранилище» – выбор хранилища для размещения виртуального диска (выбор хранилища определяет возможный формат образа диска);
- «Размер диска» (GiB) – размер виртуального диска в гигабайтах;
- «Формат» – выбирается формат образа виртуального диска. Доступные значения: «Несжатый образ диска (raw)», «Формат образа QEMU (qcow2)» и «Формат образа Vmware (vmdk)». Формат образа RAW является полностью выделяемым (thick-provisioned), т.е. выделяется сразу весь объем образа. QEMU и VMDK поддерживают динамичное выделение пространства (thin-provisioned), т.е. объем растет по мере сохранения данных на виртуальный диск;
- «Кэш» – выбор метода кэширования виртуальной машины. По умолчанию выбирается работа без кэширования. Доступные значения: «Direct sync», «Write through», «Write back», «Writeback (не безопасно)» и «Нет кэша»;
- «Отклонить» – если эта опция активирована и если гостевая ОС поддерживает TRIM, то это позволит очищать неиспользуемое пространство образа виртуального диска и соответственно сжимать образ диска.

Вкладка «Жесткий диск»

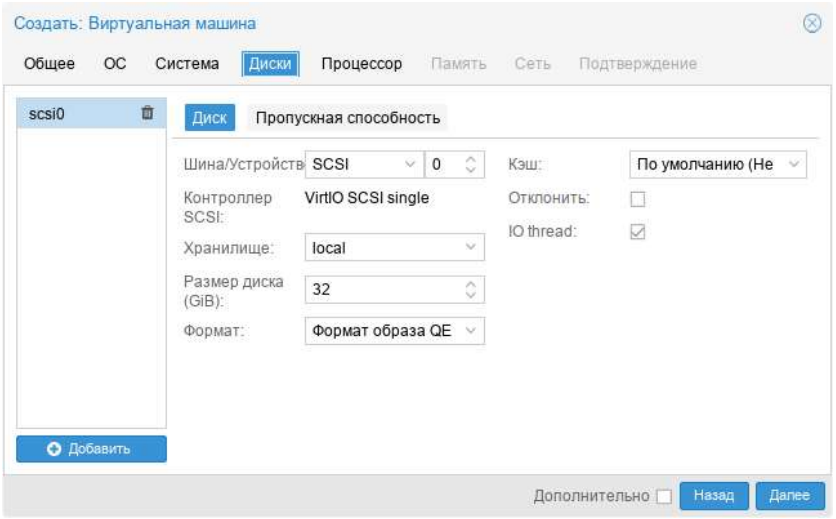


Рис. 166

В мастере создания ВМ можно добавить несколько дисков (Рис. 167) (кнопка «Добавить»). Максимально можно добавить: 31 диск SCSI, 16 – VirtIO, 6 – SATA, 4 – IDE.

Вкладка «Жесткий диск». Создание нескольких дисков

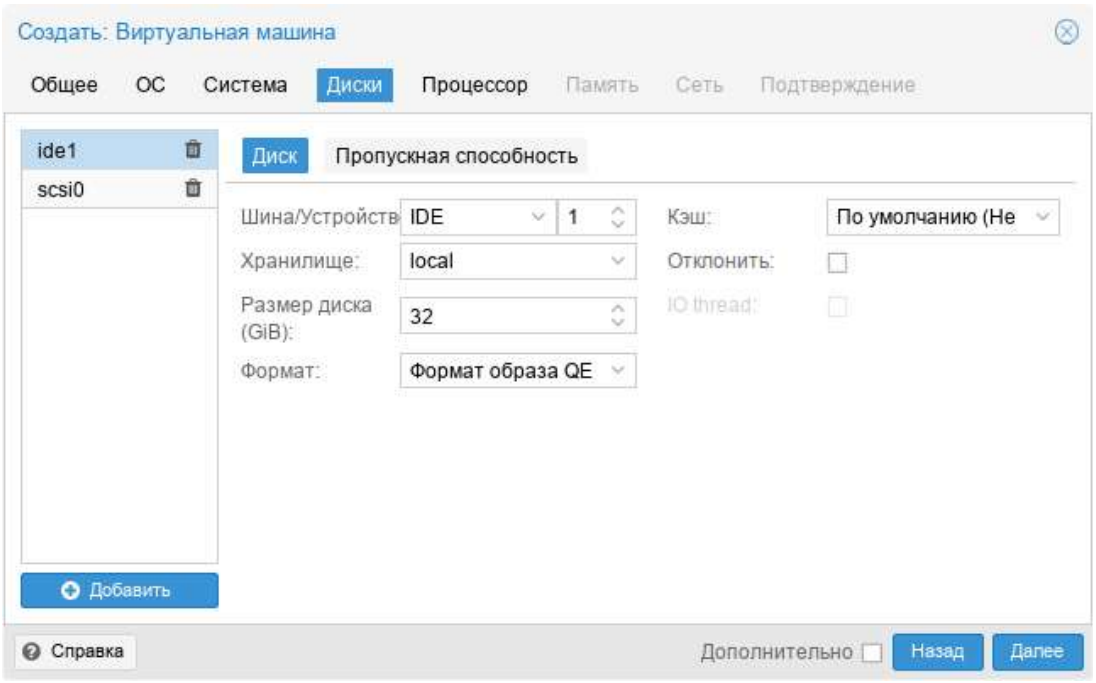


Рис. 167

В разделе «Пропускная способность» (Рис. 168) можно задать максимальную скорость чтения/записи с диска (в мегабайтах в секунду или в операциях в секунду).

Скорость чтения/записи с диска

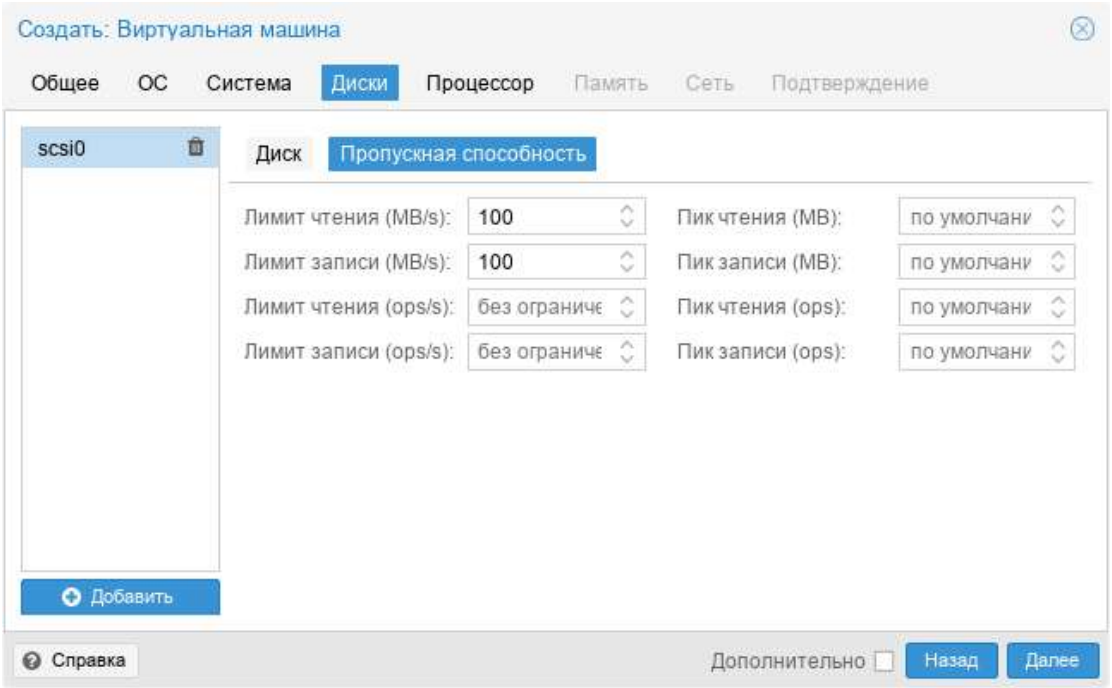


Рис. 168

Примечание. SCSI и VirtIO дискам может быть добавлен атрибут read-only (Рис. 169) (отметка «Только для чтения»).

Отметка «Только для чтения»

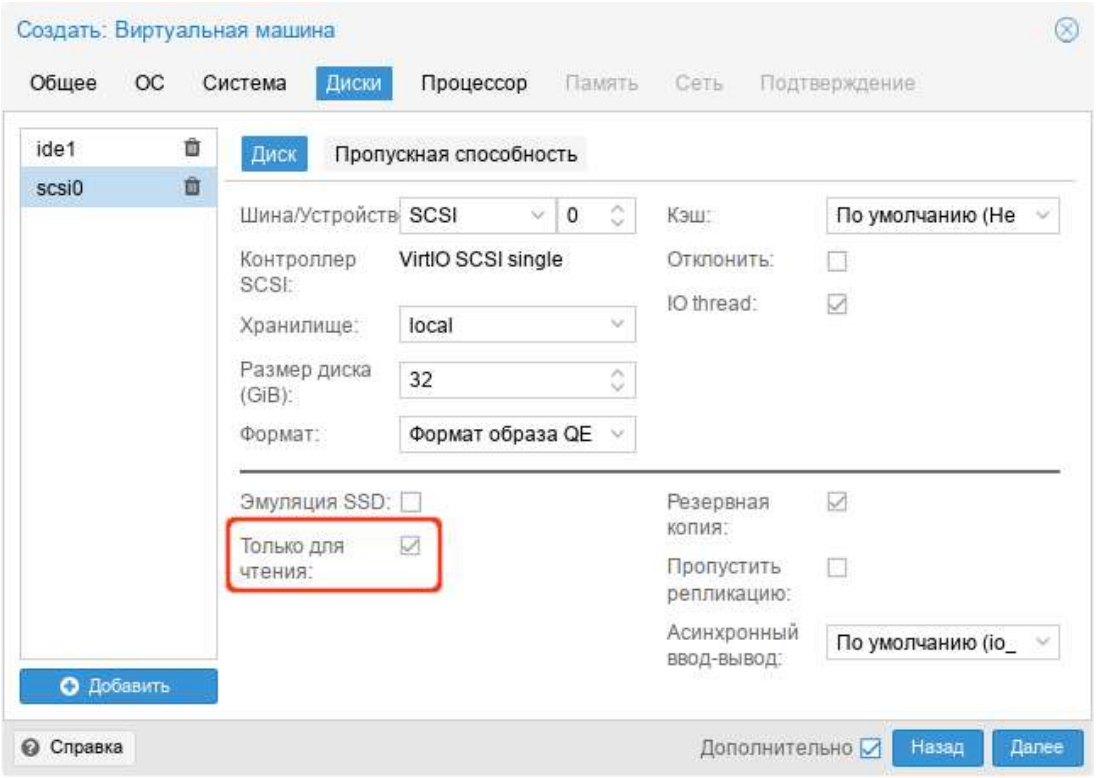


Рис. 169

На следующем этапе настраивается процессор (CPU) (Рис. 170):

- «Сокеты» – число сокетов ЦПУ для ВМ;
- «Ядра» – число ядер для ВМ;
- «Тип» – тип процессора.

Вкладка «Процессор»

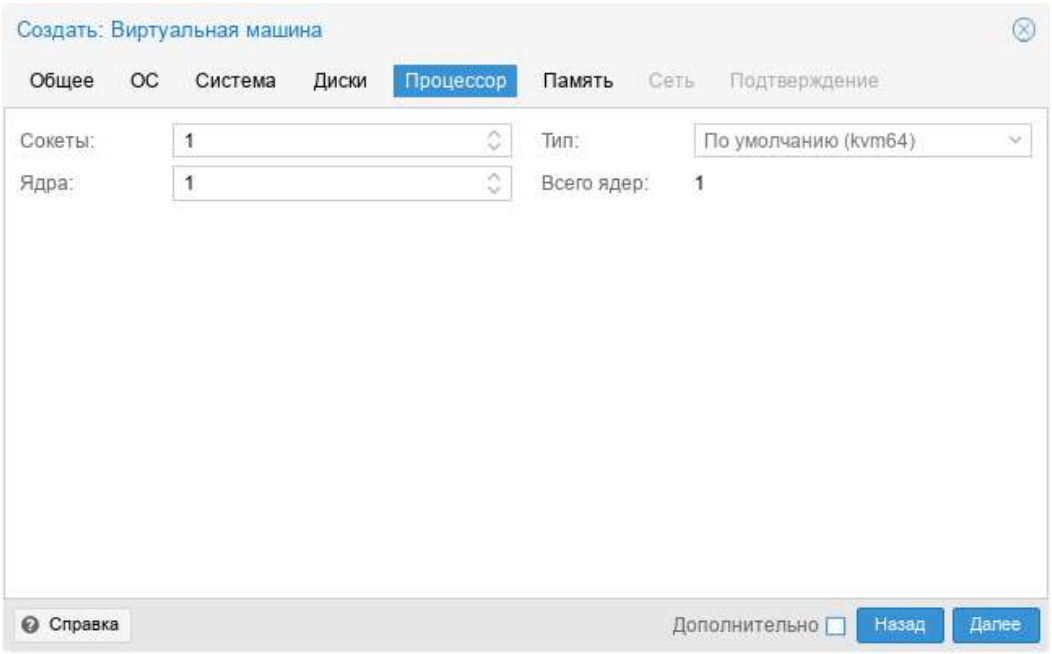


Рис. 170

На вкладке «Память» (Рис. 171) необходимо указать объем оперативной памяти выделяемой ВМ.

Вкладка «Память»

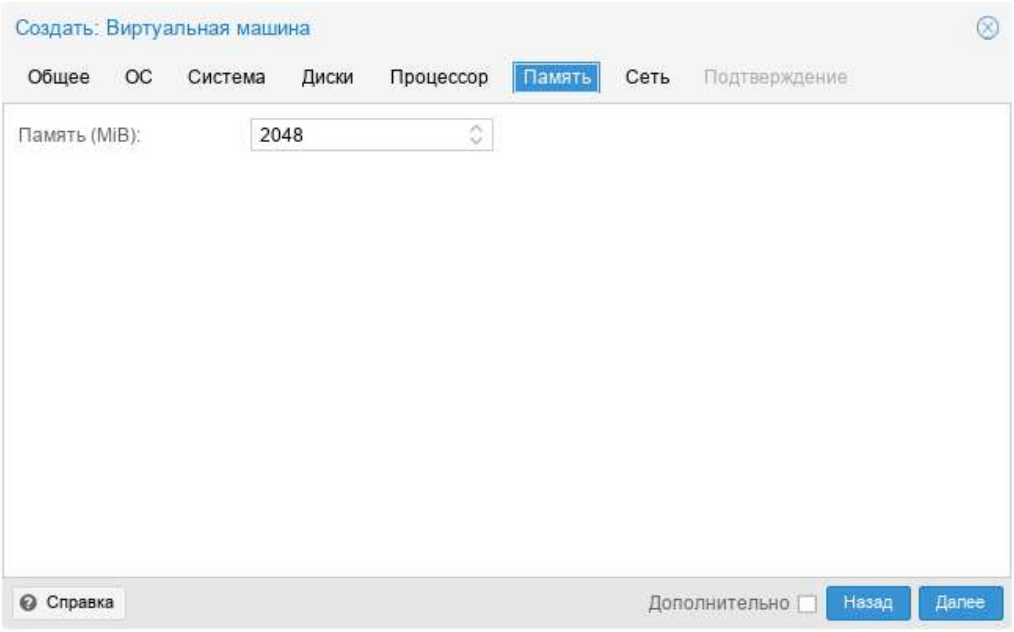


Рис. 171

Вкладка «Сеть» содержит следующие настройки (Рис. 172):

- «Нет сетевого устройства» – выбор данного параметра пропускает шаг настройки сетевой среды;
- «Сетевой мост» – установка сетевого интерфейса в режиме моста. Это предпочтительный параметр для сетевой среды ВМ. В этом режиме возможно создание множества мостов с виртуальными сетями для создания изолированных сетей в одной и той же платформе, поскольку ВМ не имеют прямого доступа к реальной локальной сетевой среде;
- «Тег виртуальной ЛС» – применяется для установки идентификатора VLAN для данного виртуального интерфейса;
- «Сетевой экран» – разрешает использование для ВМ встроенных межсетевых экранов;
- «Модель» – тип драйвера сетевого устройства. Для максимальной сетевой производительности ВМ следует выбрать пункт «VirtIO (паравиртуализированно)»;
- «MAC-адрес» – по умолчанию PVE автоматически создает уникальный MAC-адрес для сетевого интерфейса. Если есть такая необходимость, можно ввести пользовательский MAC-адрес вручную.

Вкладка «Сеть»

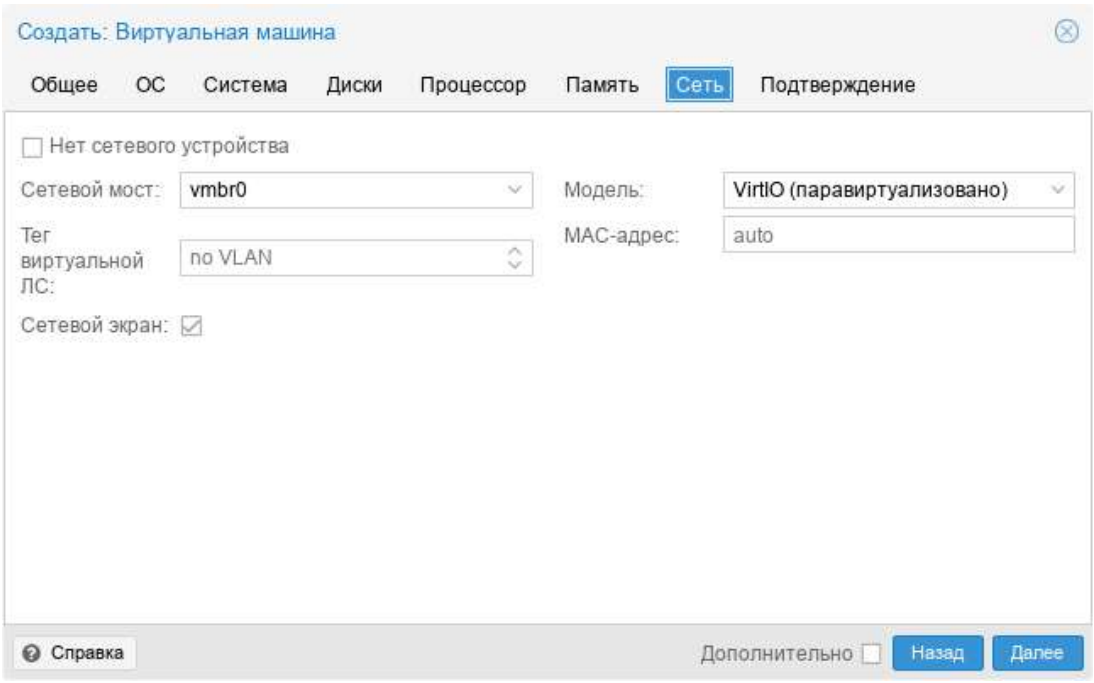


Рис. 172

Последняя вкладка «Подтверждение» отобразит все введенные или выбранные значения для ВМ (Рис. 173). Для создания ВМ следует нажать кнопку «Готово». Если необходимо внести изменения в параметры ВМ, можно перейти по вкладкам назад. Если отметить пункт «Запуск после создания» ВМ будет запущена сразу после создания.

Вкладка «Подтверждение»

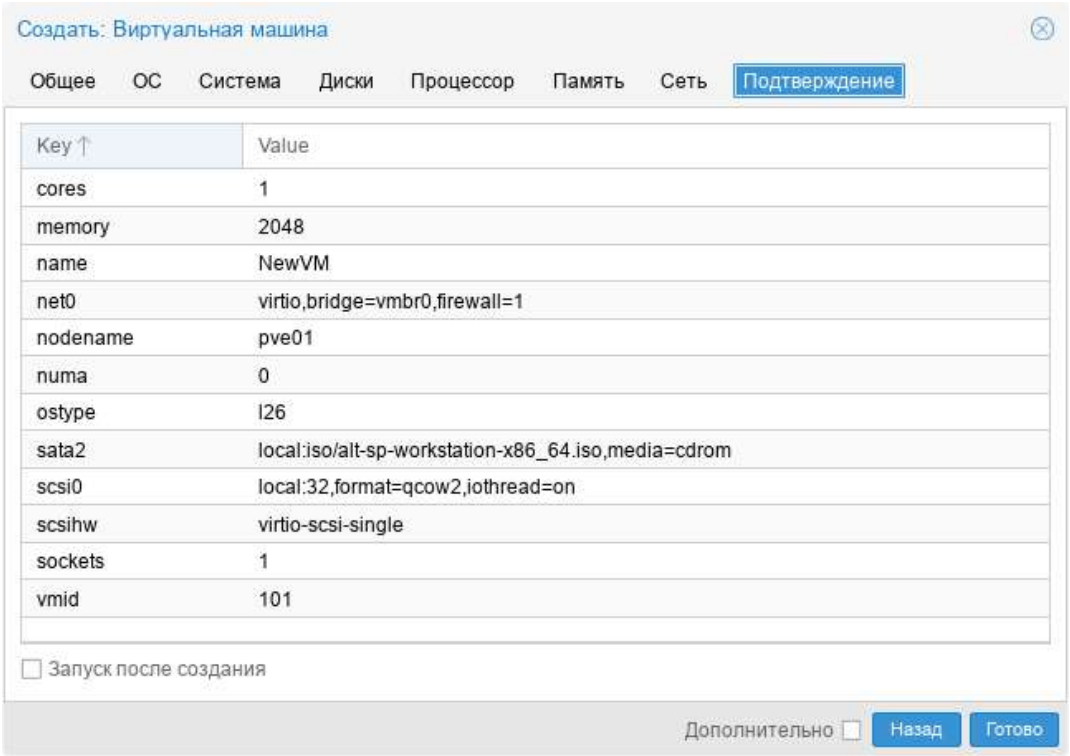


Рис. 173

4.7.2 Запуск и остановка ВМ

4.7.2.1 Изменение состояния ВМ в веб-интерфейсе

Запустить ВМ можно, выбрав в контекстном меню ВМ пункт «Запуск» (Рис. 174), либо нажав на кнопку «Запуск» («Start») (Рис. 175).

Запущенная ВМ будет обозначена зеленой стрелкой на значке ВМ.

Контекстное меню ВМ

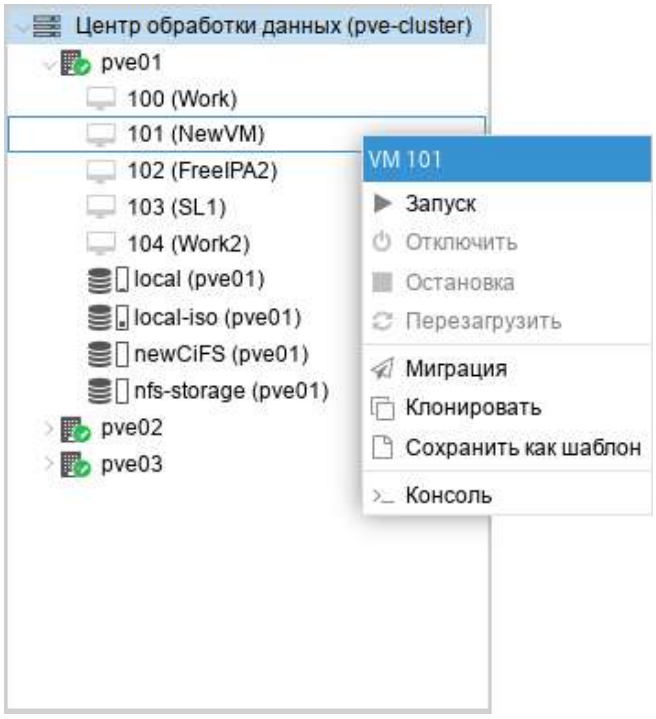


Рис. 174

Кнопки управления состоянием ВМ

Рис. 175

Запустить ВМ также можно, нажав кнопку «Start Now» в консоли гостевой машины (Рис. 176).

Для запущенной ВМ доступны следующие действия (Рис. 177):

- «Приостановить» – перевод ВМ в спящий режим;
- «Гибернация» – перевод ВМ в ждущий режим;
- «Отключить» – выключение ВМ;
- «Остановка» – остановка ВМ, путем прерывания ее работы;
- «Перезагрузить» – перезагрузка ВМ.

Кнопка «Start Now» в консоли ВМ

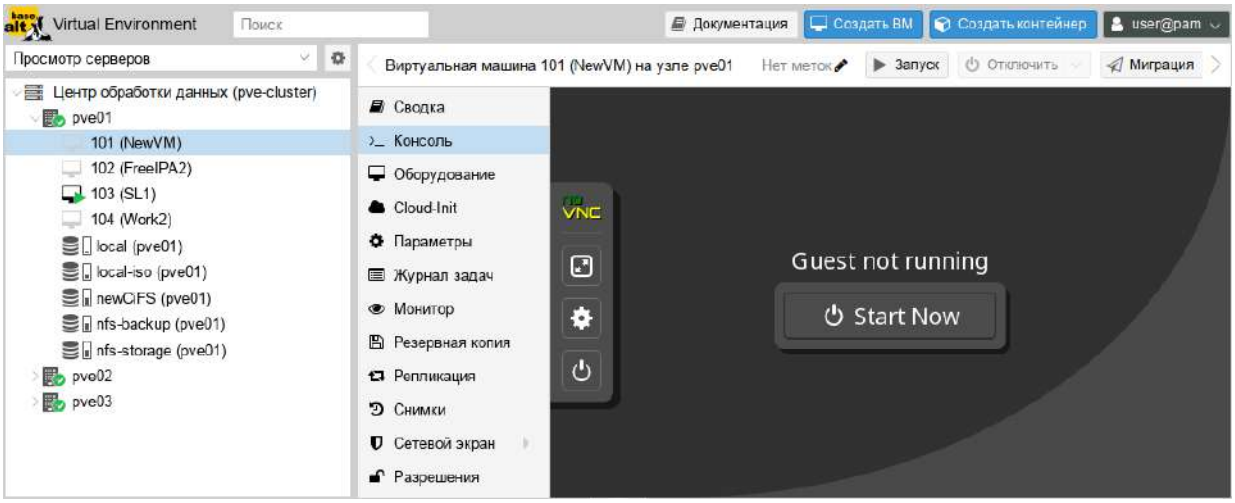


Рис. 176

Контекстное меню запущенной ВМ

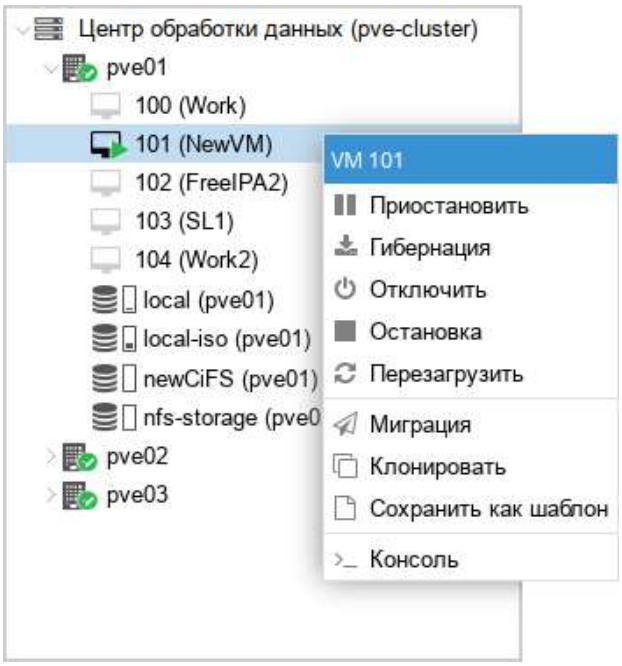


Рис. 177

4.7.2.2 Автоматический запуск ВМ

Для того чтобы ВМ запускалась автоматически при загрузке хост-системы, необходимо отметить опцию «Запуск при загрузке» на вкладке «Параметры» ВМ в веб-интерфейсе или установить ее с помощью команды:

```
# qm set <vmid> -onboot 1
```

Иногда необходимо точно настроить порядок загрузки ВМ, например, если одна из ВМ обеспечивает межсетевой экран или DHCP для других гостевых систем. Для настройки порядка запуска ВМ можно использовать следующие параметры (Рис. 178) (опция «Порядок запуска и отключения» на вкладке «Параметры» требуемой ВМ):

- «Порядок запуска и отключения» – определяет приоритет порядка запуска. Для того чтобы ВМ запускалась первой, необходимо установить этот параметр в значение 1. Для выключения используется обратный порядок: ВМ, с порядком запуска 1, будет выключаться последней. Если несколько хостов имеют одинаковый порядок, определенный на хосте, они будут дополнительно упорядочены в порядке возрастания VMID;
- «Задержка запуска» – определяет интервал (в секундах) между запуском этой ВМ и последующими запусками ВМ;
- «Задержка отключения» – определяет время в секундах, в течение которого PVE должен ожидать, пока ВМ не перейдет в автономный режим после команды выключения. Значение по умолчанию – 180, т.е. PVE выдаст запрос на завершение работы и подождет 180 секунд, пока машина перейдет в автономный режим. Если после истечения тайм-аута машина все еще находится в сети, она будет принудительно остановлена.

Настройка порядка запуска и выключения ВМ

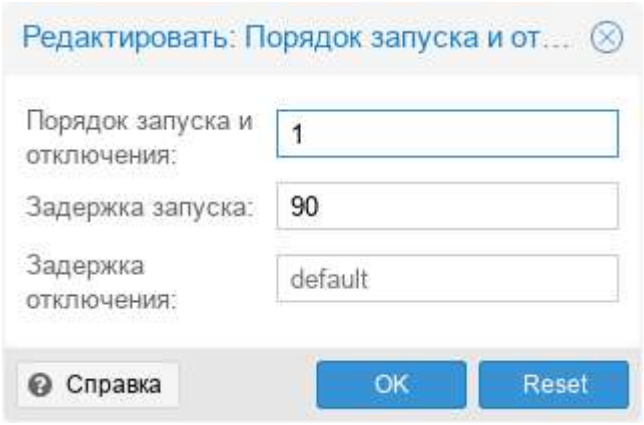


Рис. 178

Примечание. Виртуальные машины, управляемые стеком HA, не поддерживают опции запуска при загрузке и порядок загрузки. Запуск и остановку таких ВМ обеспечивает диспетчер HA.

ВМ без установленного параметра «Порядок запуска и отключения» всегда будут запускаться после тех, для которых этот параметр установлен. Кроме того, этот параметр может применяться только для ВМ, работающих на одном хосте, а не в масштабе кластера.

4.7.2.3 Массовый запуск и остановка ВМ

Для массового запуска или остановки ВМ, необходимо в контекстном меню узла выбрать «Массовый запуск» или «Массовое отключение» соответственно (Рис. 179). В окрывшемся окне необходимо отметить нужные ВМ и нажать кнопку «Запуск»/«Отключить». Для массового отключения можно также переопределить параметры «Время ожидания» и «Принудительная остановка» (Рис. 180).

Контекстное меню узла

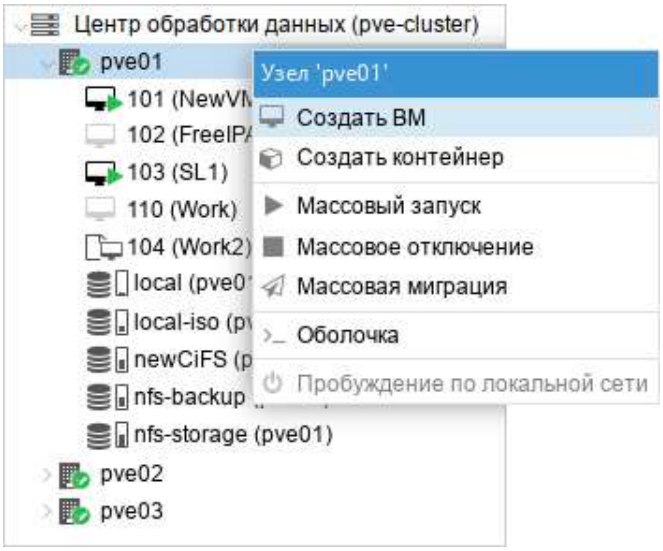


Рис. 179

Массовое отключение ВМ

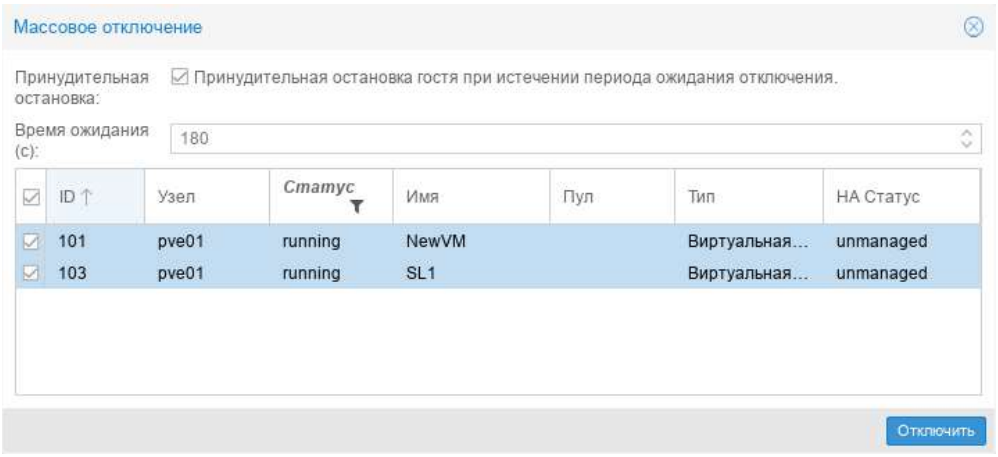


Рис. 180

4.7.3 Управление ВМ с помощью qm

Если веб-интерфейс PVE недоступен, можно управлять ВМ в командной строке (используя сеанс SSH, из консоли noVNC, или зарегистрировавшись на физическом хосте).

`qm` – это инструмент для управления ВМ Qemu/KVM в PVE. Утилиту `qm` можно использовать для создания/удаления ВМ, для управления работой ВМ (запуск/остановка/приостановка/возобновление), для установки параметров в соответствующем конфигурационном файле, а также для создания виртуальных дисков.

Синтаксис команды:

```
qm <КОМАНДА> [АРГУМЕНТЫ] [ОПЦИИ]
```

Чтобы просмотреть доступные для управления ВМ команды можно выполнить следующую команду:

```
# qm help
```

Некоторые команды qm приведены в табл. 19. В командах vmid – идентификатор ВМ, может принимать значение из диапазона 100 – 999999999.

Т а б л и ц а 19 – Команды qm

Команда	Описание
`qm agent`	Псевдоним для `qm guest cmd`
`qm block <vmid>`	Заблокировать ВМ
`qm cleanup <vmid> <clean-shutdown> <guest-requested>`	Очищает ресурсы, такие как сенсорные устройства, vgpu и т.д. Вызывается после выключения, сбоя ВМ и т. д. - vmid – идентификатор ВМ; - clean-shutdown – указывает, корректно ли была завершена работа qemu; - guest-requested – указывает, было ли завершение работы запрошено гостем или через qmp
`qm clone <vmid> <newid> [ОПЦИИ]`	Создать копию ВМ/шаблона. - vmid – идентификатор ВМ; - newid – VMID для клона (100 – 999999999); - --bwlimit <целое число> – переопределить ограничение пропускной способности ввода-вывода (в КиБ/с) (0 – N); - --description <строка> – описание новой ВМ; - --format <qcow2 | raw | vmdk> – целевой формат хранения файлов (действительно только для полного клона); - --full <логическое значение> – создать полную копию всех дисков (используется по умолчанию при клонировании ВМ). Для шаблонов ВМ по умолчанию пытается создать связанный клон; - --name <строка> – имя новой ВМ; - --pool <строка> – пул, в который будет добавлена ВМ; - --snapname <строка> – имя снимка; - --storage <строка> – целевое хранилище для полного клона; - --target <строка> – целевой узел (доступно в случае, если исходная ВМ находится в общем хранилище)
`qm cloudinit dump <vmid> <type>`	Получить автоматически сгенерированную конфигурацию cloud-init. - vmid – идентификатор ВМ; - type – тип конфигурации (meta | network | user)
`qm cloudinit pending <vmid>`	Получить конфигурацию cloud-init с текущими и ожидающими значениями
`qm cloudinit update <vmid>`	Восстановить и изменить диск конфигурации cloud-init
`qm config <vmid> <ОПЦИИ>`	Вывести конфигурацию ВМ с применёнными ожидающими изменениями конфигурации. Для вывода текущей конфигурации следует указать параметр current. - vmid – идентификатор ВМ; - --current <логическое значение> – вывести текущие значения вместо ожидающих (по умолчанию 0); - --snapshot <строка> – вывести значения конфигурации из данного снимка
`qm create <vmid> <ОПЦИИ>`	Создать или восстановить ВМ. Некоторые опции: - vmid – идентификатор ВМ; - --acpi <логическое значение> – включить/отключить ACPI (по умолчанию 1); - --affinity <строка> – список ядер хоста, используемых для выполнения

Команда	Описание
	гостевых процессов, например: 0,5,8-11; - --agent [enabled=]<1|0> [,freeze-fs-on-backup=<1|0>] [,fstrim_cloned_disks=<1|0>] [,type=<virtio|isa>] – включить/отключить связь с гостевым агентом QEMU; - --arch <aarch64 | x86_64> – архитектура виртуального процессора; - --archive <строка> – создать ВМ из архива. Указывается либо путь к файлу .tar или .vma, либо идентификатор тома резервной копии хранилища PVE; - --args <строка> – передача произвольных аргументов в KVM; - --audio0 device=<ich9-intel-hda|intel-hda|AC97> [,driver=<spice |none>] – настройка аудиоустройства; - --balloon <целое число> – объём целевой оперативной памяти для ВМ в МиБ (0 отключает Balloon Driver); - --bios <ovmf | seabios> – реализация BIOS (по умолчанию seabios); - --boot [order=<устройство[;устройство...]>] – порядок загрузки ВМ; - --bwlimit <целое число> – переопределить ограничение пропускной способности ввода-вывода (в КиБ/с); - --cdrom <volume> – псевдоним опции ide2; - --cicustom [meta=<volume>] [,network=<volume>] [,user=<volume>] [,vendor=<volume>] – cloud-init: указать пользовательские файлы для замены автоматически созданных; - --cipassword <пароль> – cloud-init: пароль для пользователя. Рекомендуется использовать ключи SSH вместо пароля; - --citype <configdrive2 | nocloud | opennebula> – формат конфигурации cloud-init; - --ciupgrade <логическое значение> – cloud-init: выполнить автоматическое обновление пакета после первой загрузки (по умолчанию 1); - --ciuser <строка> – cloud-init: имя пользователя для изменения пароля и ключей SSH вместо настроенного пользователя по умолчанию; - --cores <целое число> – количество ядер на сокет (по умолчанию 1); - --cpu <тип> – эмулируемый тип процессора; - --cpulimit <целое число (0–128)> – ограничение использования процессора (по умолчанию 0); - --cpuunits <целое число (1–262144)> – вес ЦП для ВМ будет ограничен значением [1, 10000] в cgroup v2 (по умолчанию cgroup v1: 1024, cgroup v2: 100); - --description <строка> – описание ВМ; - --efidisk0 [file=]<volume> [,efitype=<2m|4m>] [,format=<enum>] [,import-from=<source volume>] [,pre-enrolled-keys=<1|0>] [,size=<DiskSize>] – диск для хранения переменных EFI; - --force <логическое значение> – разрешить перезапись существующей ВМ (требуется опция --archive); - --freeze <логическое значение> – заморозить процессор при запуске; - --hookscript <строка> – скрипт, который будет выполняться на разных этапах жизненного цикла ВМ; - --ide[n] <описание> – использовать в качестве жёсткого диска IDE или компакт-диск (n от 0 до 3). Чтобы выделить новый том используется синтаксис STORAGE_ID:SIZE_IN_GiB. Для импорта из существующего тома используется STORAGE_ID:0 и параметр import-from; - --ipconfig[n] [gw=<GatewayIPv4>] [,gw6=<GatewayIPv6>] [,ip=<IPv4Format/CIDR>] [,ip6=<IPv6Format/CIDR>] – cloud-init: указать IP-адрес и шлюз для соответствующего интерфейса; - --kvm <логическое значение> – включить/отключить аппаратную виртуализацию KVM (по умолчанию 1); - --live-restore <логическое значение> – запустить ВМ из резервной копии

Команда	Описание
	и восстановить её в фоновом режиме (только PBS). Требуется опция --archive; - --localtime <логическое значение> – установить часы реального времени (RTC) на местное время; - --lock <backup | clone | create | migrate | rollback | snapshot | snapshot-delete | suspended | suspending> – заблокировать/разблокировать ВМ; - --machine <тип> – тип машины QEMU; - --memory [current=]<целое число> – свойства памяти; - --migrate_downtime <число> – максимально допустимое время простоя (в секундах) для миграции (по умолчанию 0,1); - --migrate_speed <целое число> – максимальная скорость (в МБ/с) для миграции (по умолчанию 0 – не ограничивать скорость); - --name <строка> – имя ВМ; - --nameserver <строка> – cloud-init: устанавливает IP-адрес DNS-сервера для контейнера; - --net <сеть> – сетевые устройства; - --numa <логическое значение> – включить/отключить NUMA (по умолчанию 0); - --numa n <топология> – топология NUMA; - --onboot <логическое значение> – запускать ВМ во время загрузки системы (по умолчанию 0); - --ostype <l24 | l26 | other | solaris | w2k | w2k3 | w2k8 | win10 | win11 | win7 | win8 | wvista | wxp> – гостевая ОС; - --pool <строка> – добавить ВМ в указанный пул; - --protection <логическое значение> – установить флаг защиты ВМ (по умолчанию 0). Флаг защиты отключит возможность удаления ВМ и удаления дисковых операций; - --reboot <логическое значение> – разрешить перезагрузку (по умолчанию 1). Если установлено значение 0, ВМ завершит работу при перезагрузке; - --rng0 [source=] </dev/urandom|/dev/random|/dev/hwrng> [,max_bytes=<целое число>] [,period=<целое число>] – настройть генератор случайных чисел на основе VirtIO; - --sata[n] <описание> – использовать в качестве жёсткого диска SATA или компакт-диск (n от 0 до 5). Чтобы выделить новый том используется синтаксис STORAGE_ID:SIZE_IN_GiB. Для импорта из существующего тома используется STORAGE_ID:0 и параметр import-from; - --scsi[n] <описание> – использовать в качестве жёсткого диска SCSI или компакт-диск (n от 0 до 30). Чтобы выделить новый том используется синтаксис STORAGE_ID:SIZE_IN_GiB. Для импорта из существующего тома используется STORAGE_ID:0 и параметр import-from; - --scsihw <lsi | lsi53c810 | megasas | pvscsi | virtio-scsi-pci | virtio-scsi-single> – модель контроллера SCSI (по умолчанию lsi); - searchdomain <строка> – cloud-init: устанавить домены поиска DNS для контейнера; - serial[n] (/dev/.+|socket) – последовательное устройство внутри ВМ (n от 0 до 3); - --scsihw <модель> – модель контроллера SCSI (по умолчанию lsi); - --shares <целое число (0–50000)> – объем разделяемой памяти (по умолчанию 1000); - --sockets <целое число> – количество сокетов процессора (по умолчанию 1); - --spice_enhancements [foldersharing=<1|0>] [,videostreaming=<off|all|filter>] – настройки для SPICE; - --sshkeys <путь к файлу> – cloud-init: настройка публичных ключей SSH

Команда	Описание
	(по одному ключу в строке, формат OpenSSH); - --start <логическое значение> – запустить ВМ после создания (по умолчанию 0); - --startup `[[order=]\d+] [,up=\d+] [,down=\d+] ` – поведение при запуске и выключении. order – неотрицательное число, определяющее общий порядок запуска. Выключение выполняется в обратном порядке. up/down – задержка включения/выключения в секундах; - --storage <строка> – хранилище; - --tablet <логическое значение> – включить/отключить USB-планшет (по умолчанию 1); - --tags <строка> — теги ВМ; - --template <логическое значение> – включить/отключить шаблон (по умолчанию 0); - --tpmstate0 <диск> – настроить диск для хранения состояния TPM. Формат фиксированный – raw; - --unique <логическое значение> – назначить уникальный случайный адрес Ethernet; - --usb[n] [[host=]<HOSTUSBDEVICE|spice>] [,mapping=<mapping-id>] [,usb3=<1|0>] – настройка USB-устройства (n — от 0 до 4, для версии машины >= 7.1 и ostype l26 или windows > 7, n может достигать 14); - --vcpus <целое число> – количество виртуальных процессоров с горячим подключением; - --vga [[type=]<enum>] [,clipboard=<vnc>] [,memory=<целое число>] – настройка VGA; - virtio[n] <описание> — использовать жёсткий диск VIRTIO (n от 0 до 15); - vmgenid <UUID> – установить идентификатор поколения ВМ (по умолчанию 1 – генерировать автоматически); - --vmstatestorage <строка> – хранилище по умолчанию для томов/файлов состояния ВМ; - --watchdog [[model=]<i6300esb|ib700>] [,action=<enum>] – создать сторожевое устройство виртуального оборудования
`qm delsnapshot <vmid> <snapname> <ОПЦИИ>`	Удалить снимок ВМ. - vmid – идентификатор ВМ; - snapshot – имя снимка; - --force <логическое значение> – удалить из файла конфигурации, даже если удаление снимков диска не удалось
`qm destroy <vmid> [ОПЦИИ]`	Уничтожить ВМ и все её тома (будут удалены все разрешения, специфичные для ВМ). - vmid – идентификатор ВМ; - --destroy-unreferenced-disks <логическое значение> – дополнительно уничтожить все диски, не указанные в конфигурации, но с совпадающим VMID из всех включенных хранилищ (по умолчанию 0); - --purge <логическое значение> – удалить VMID из конфигураций резервного копирования и высокой доступности; - --skiplock <логическое значение> – игнорировать блокировки (может использовать только root)
`qm disk import <vmid> <source> <storage> [ОПЦИИ]`	Импортировать образ внешнего диска в неиспользуемый диск ВМ. Формат образа должен поддерживаться qemu-img. - vmid – идентификатор ВМ; - source – путь к образу диска; - storage – идентификатор целевого хранилища; - --format <qcow2 | raw | vmdk> – целевой формат
`qm disk move <vmid>`	Переместить том в другое хранилище или в другую ВМ.

Команда	Описание
`<disk> <storage> [ОПЦИИ]`	- vmid – идентификатор ВМ; - disk – диск, который необходимо переместить (например, scsi1); - storage – целевое хранилище; - --bwlimit <целое число> – переопределить ограничение пропускной способности ввода-вывода (в КиБ/с); - --delete <логическое значение> – удалить исходный диск после успешного копирования. По умолчанию исходный диск сохраняется как неиспользуемый (по умолчанию 0); - --digest <строка> – запретить изменения, если текущий файл конфигурации имеет другой SHA1 дайджест (можно использовать для предотвращения одновременных изменений); - --format <qcow2 | raw | vmdk> – целевой формат; - --target-digest <строка> – запретить изменения, если текущий файл конфигурации целевой ВМ имеет другой SHA1 дайджест (можно использовать для обнаружения одновременных модификаций); - --target-disk <efidisk0 | ide0 | ide1| …| virtio9 > – ключ конфигурации, в который будет перемещен диск на целевой ВМ (например, ide0 или scsi1). По умолчанию используется ключ исходного диска; - --target-vmid <целое число> – идентификатор целевой ВМ
`qm disk rescan [ОПЦИИ]`	Пересканировать все хранилища и обновить размеры дисков и неиспользуемые образы дисков. - --dryrun <логическое значение> – не записывать изменения в конфигурацию ВМ (по умолчанию 0); - --vmid <целое число> – идентификатор ВМ
`qm disk resize <vmid> <disk> <size> [ОПЦИИ]`	Увеличить размер диска. - vmid – идентификатор ВМ; - disk – диск, размер которого необходимо увеличить (например, scsi1); - size – новый размер. Со знаком «+» значение прибавляется к фактическому размеру тома, а без него значение принимается как абсолютное. Уменьшение размера диска не поддерживается; - --digest <строка> – запретить изменения, если текущий файл конфигурации имеет другой SHA1 дайджест (можно использовать для предотвращения одновременных изменений); - --skiplock <логическое значение> – игнорировать блокировки (может использовать только root)
`qm disk unlink <vmid> --idlist <строка> [ОПЦИИ]`	Отсоединить/удалить образы дисков. - vmid – идентификатор ВМ; - --idlist <строка> – список идентификаторов дисков, которые необходимо удалить; - --force <логическое значение> – принудительное физическое удаление (иначе диск будет удалён из файла конфигурации и будет создана дополнительная запись конфигурации с именем unused[n], которая содержит идентификатор тома)
`qm guest cmd <vmid> <команда>`	Выполнить команды гостевого агента QEMU. - vmid – идентификатор ВМ; - команда – команда QGA (fsfreeze-freeze | fsfreeze-status | fsfreeze-thaw | fstrim | get-fsinfo | get-host-name | get-memory-block-info | get-memory-blocks | get-osinfo | get-time | get-timezone | get-users | get-vcpus | info | network-get-interfaces | ping | shutdown | suspend-disk | suspend-hybrid | suspend-ram)
`qm guest exec <vmid> [<extra-args>] [ОПЦИИ]`	Выполнить данную команду через гостевой агент. - vmid – идентификатор ВМ; - extra-args – дополнительные аргументы в виде массива; - --pass-stdin <логическое значение> – если установлено, читать STDIN

Команда	Описание
	до EOF и пересылать гостевому агенту через входные данные (по умолчанию 0). Допускается максимум 1 МБ; - --synchronous <логическое значение> – если установлено значение 0, возвращает pid немедленно, не дожидаясь завершения команды или тайм-аута (по умолчанию 1); - --timeout <целое число> – максимальное время синхронного ожидания завершения команды. Если достигнуто, возвращается pid. Для отключения следует установить значение 0 (по умолчанию 30)
`qm guest exec-status <vmid> <pid>`	Получить статус данного pid, запущенного гостевым агентом. - vmid – идентификатор ВМ; - pid – PID для запроса
`qm guest passwd <vmid> <username> [ОПЦИИ]`	Установить пароль для данного пользователя. - vmid – идентификатор ВМ; - username – пользователь, для которого устанавливается пароль; - crypted <логическое значение> – если пароль уже был передан через crypt(), следует установить значение 1 (по умолчанию 0)
`qm help [extra-args] [ОПЦИИ]`	Показать справку по указанной команде. - extra-args – показать справку по конкретной команде; - --verbose <логическое значение> – подробный формат вывода
`qm importdisk`	Псевдоним для `qm disk import`
`qm importovf <vmid> <manifest> <storage> [ОПЦИИ]`	Создать новую ВМ, используя параметры, считанные из манифеста OVF. - vmid – идентификатор ВМ; - manifest – путь до файла ovf; - storage – идентификатор целевого хранилища; - --format <qcow2 | raw | vmdk> – целевой формат
`qm list [ОПЦИИ]`	Вывести список ВМ узла. - --full <логическое значение> – определить полный статус активных ВМ
`qm listsnapshot <vmid>`	Вывести список снимков ВМ
`qm migrate <vmid> <target> [ОПЦИИ]`	Перенос ВМ. Создаёт новую задачу миграции. - vmid – идентификатор ВМ; - target – целевой узел; - --bwlimit <целое число> – переопределить ограничение пропускной способности ввода-вывода (в КиБ/с); - --force <логическое значение> – разрешить миграцию ВМ, использующих локальные устройства (может использовать только root); - --migration_network <строка> – CIDR (под)сети, которая используется для миграции; - --migration_type <insecure | secure> – трафик миграции по умолчанию шифруется с использованием SSH-туннеля. В безопасных сетях эту функцию можно отключить для повышения производительности; - --online <логическое значение> – использовать онлайн-/живую миграцию, если ВМ запущена (игнорируется, если ВМ остановлена); - --targetstorage <строка> – сопоставление исходных и целевых хранилищ. Предоставление только одного идентификатора хранилища сопоставляет все исходные хранилища с этим хранилищем. Если указать специальное значение 1, каждое исходное хранилище будет сопоставлено самому себе; - --online <логическое значение> – включить живую миграцию хранилища для локального диска
`qm monitor <vmid>`	Войти в интерфейс монитора QEMU

Команда	Описание
qm move-disk	Псевдоним для qm disk move
qm move_disk	Псевдоним для qm disk move
qm nbdstop <vmid>	Остановить встроенный сервер NBD
qm pending <vmid>	Получить конфигурацию ВМ с текущими и ожидающими значениями
qm reboot <vmid> [ОПЦИИ]	Перезагрузить ВМ. Применяет ожидающие изменения. - vmid – идентификатор ВМ; - --timeout <целое число> – максимальное время ожидания для выключения
qm remote-migrate <vmid> [<target-vmid>] <target-endpoint> --target-bridge <строка> --target-storage <строка> [ОПЦИИ]	Перенос ВМ в удалённый кластер. Создаёт новую задачу миграции. ЭКСПЕРИМЕНТАЛЬНАЯ функция! - vmid – идентификатор ВМ; - target-vmid – идентификатор целевой ВМ; - target-endpoint – удалённая целевая конечная точка. Удалённая точка указывается в формате: apitoken=<API-токен PVE, включая секретное значение> ,host=<имя или IP удалённого узла> [,fingerprint=<отпечаток сертификата удаленного хоста, если ему не доверяет системное хранилище>] [,port=<целое число>]; - --bwlimit <целое число> – переопределить ограничение пропускной способности ввода-вывода (в КиБ/с); - --delete <логическое значение> – удалить исходную ВМ и связанные с ней данные после успешной миграции (по умолчанию 0). По умолчанию исходная ВМ остается в исходном кластере в остановленном состоянии; - --online <логическое значение> – использовать онлайн-/живую миграцию, если ВМ запущена (игнорируется, если ВМ остановлена); - --target-bridge <строка> – сопоставление исходных и целевых мостов. Предоставление только одного идентификатора моста сопоставляет все исходные мосты с этим мостом. Предоставление специального значения 1 сопоставит каждый исходный мост с самим собой; - --target-storage <строка> – сопоставление исходных и целевых хранилищ. Предоставление только одного идентификатора хранилища сопоставляет все исходные хранилища с этим хранилищем. Если указать специальное значение 1, каждое исходное хранилище будет сопоставлено самому себе; - --online <логическое значение> – включить живую миграцию хранилища для локального диска
qm rescan	Псевдоним для qm disk rescan
qm reset <vmid> [ОПЦИИ]	Сбросить ВМ. - vmid – идентификатор ВМ; - --skiplock <логическое значение> – игнорировать блокировки (может использовать только root)
qm resize	Псевдоним для qm disk resize
qm resume	Возобновить работу ВМ. - vmid – идентификатор ВМ; - --skiplock <логическое значение> – игнорировать блокировки (может использовать только root)
qm rollback <vmid> <snapname> [ОПЦИИ]	Откат состояния ВМ до указанного снимка. - vmid – идентификатор ВМ; - snapname – имя снимка; - --start <логическое значение> – запустить ВМ после отката (по умолчанию 0). ВМ будут запускаться автоматически, если снимок включает

Команда	Описание
	ОЗУ
`qm sendkey <vmid> <ключ> [ОПЦИИ]`	- Послать нажатия клавиш на ВМ. - vmid – идентификатор ВМ; - ключ – ключ (в кодировке qemu monitor, например, ctrl-shift); - --skiplock <логическое значение> – игнорировать блокировки (может использовать только root)
`qm set <vmid> [ОПЦИИ]`	Установить параметры ВМ. Некоторые опции: - vmid – идентификатор ВМ; - --acpi <логическое значение> – включить/отключить ACPI (по умолчанию 1); - --affinity <строка> – список ядер хоста, используемых для выполнения гостевых процессов, например: 0,5,8-11; - --agent [enabled=]<1|0> [,freeze-fs-on-backup=<1|0>] [,fstrim_cloned_disks=<1|0>] [,type=<virtio|isa>] – включить/отключить связь с гостевым агентом QEMU; - --arch <aarch64 | x86_64> – архитектура виртуального процессора; - --args <строка> – передача произвольных аргументов в KVM; - --audio0 device=<ich9-intel-hda|intel-hda|AC97> [,driver=<spice|none>] – настройка аудиоустройства; - --balloon <целое число> – объём целевой оперативной памяти для ВМ в МиБ (0 отключает Balloon Driver); - --bios <ovmf | seabios> – реализация BIOS (по умолчанию seabios); - --boot [order=<устройство[;устройство...]>] – порядок загрузки ВМ; - --cdrom <volume> – псевдоним опции ide2; - --cicustom [meta=<volume>] [,network=<volume>] [,user=<volume>] [,vendor=<volume>] – cloud-init: указать пользовательские файлы для замены автоматически созданных; - --cipassword <пароль> – cloud-init: пароль для пользователя. Рекомендуется использовать ключи SSH вместо пароля; - --citype <configdrive2 | nocloud | opennebula> – формат конфигурации cloud-init; - --ciupgrade <логическое значение> – cloud-init: выполнить автоматическое обновление пакета после первой загрузки (по умолчанию 1); - --ciuser <строка> — cloud-init: имя пользователя для изменения пароля и ключей SSH вместо настроенного пользователя по умолчанию; - --cores <целое число> – количество ядер на сокет (по умолчанию 1); - --cpu <тип> – эмулируемый тип процессора; - --cpulimit <целое число (0–128)> – ограничение использования процессора (по умолчанию 0); - --cpuunits <целое число (1–262144)> – вес ЦП для ВМ будет ограничен значением [1, 10000] в cgroup v2 (по умолчанию cgroup v1: 1024, cgroup v2: 100); - --delete <строка> – список настроек, которые необходимо удалить; - --description <строка> – описание ВМ; - --digest <строка> – запретить изменения, если текущий файл конфигурации имеет другой дайджест SHA1 (можно использовать для предотвращения одновременных изменений); - --efidisk0 [file=]<volume> [,efitype=<2m|4m>] [,format=<enum>] [,import-from=<source volume>] [,pre-enrolled-keys=<1|0>] [,size=<DiskSize>] – диск для хранения переменных EFI; - --force <логическое значение> – разрешить перезапись существующей ВМ (требуется опция --archive); - --freeze <логическое значение> – заморозить процессор при запуске; - --hookscript <строка> – описание ВМ;

Команда	Описание
	- --hostpci[n] [описание] – сопоставить PCI-устройства хоста с гостевыми устройствами; - --ide[n] <описание> – использовать в качестве жёсткого диска IDE или компакт-диск (n от 0 до 3). Чтобы выделить новый том используется синтаксис STORAGE_ID:SIZE_IN_GiB. Для импорта из существующего тома используется STORAGE_ID:0 и параметр import-from; - --ipconfig[n] [gw=<GatewayIPv4>] [,gw6=<GatewayIPv6>] [,ip=<IPv4Format/CIDR>] [,ip6=<IPv6Format/CIDR>] – cloud-init: указать IP-адрес и шлюз для соответствующего интерфейса; - --kvm <логическое значение> – включить/отключить аппаратную виртуализацию KVM (по умолчанию 1); - --localtime <логическое значение> – установите часы реального времени (RTC) на местное время; - --lock <backup | clone | create | migrate | rollback | snapshot | snapshot-delete | suspended | suspending> – заблокировать/разблокировать ВМ; - --machine <тип> – тип машины QEMU; - --memory [current=]<целое число> – свойства памяти; - --migrate_downtime <число> – максимально допустимое время простоя (в секундах) для миграции (по умолчанию = 0,1); - --migrate_speed <целое число> – максимальная скорость (в МБ/с) для миграции (по умолчанию = 0 – не ограничивать); - --name <строка> – имя ВМ; - --nameserver <строка> – cloud-init: устанавливает IP-адрес DNS-сервера для контейнера; - --net <сеть> – сетевые устройства; - --numa <логическое значение> – включить/отключить NUMA (по умолчанию 0); - --numa n <топология> – топология NUMA; - --onboot <логическое значение> – запускать ВМ во время загрузки системы (по умолчанию 0); - --ostype <l24 | l26 | other | solaris | w2k | w2k3 | w2k8 | win10 | win11 | win7 | win8 | wvista | wxp> – гостевая ОС; - --protection <логическое значение> – установить флаг защиты ВМ (по умолчанию 0). Флаг защиты отключит возможность удаления ВМ и удаления дисковых операций; - --reboot <логическое значение> – разрешить перезагрузку (по умолчанию 1). Если установлено значение 0, ВМ завершит работу при перезагрузке; - --revert <строка> – отменить ожидающее изменение; - --rng0 [source=] </dev/urandom|/dev/random|/dev/hwrng> [,max_bytes=<целое число>] [,period=<целое число>] – настройть генератор случайных чисел на основе VirtIO; - --sata[n] <описание> – использовать в качестве жёсткого диска SATA или компакт-диск (n от 0 до 5). Чтобы выделить новый том используется синтаксис STORAGE_ID:SIZE_IN_GiB. Для импорта из существующего тома используется STORAGE_ID:0 и параметр import-from; - --scsi[n] <описание> – использовать в качестве жёсткого диска SCSI или компакт-диск (n от 0 до 30). Чтобы выделить новый том используется синтаксис STORAGE_ID:SIZE_IN_GiB. Для импорта из существующего тома используется STORAGE_ID:0 и параметр import-from; - --scsihw <модель> – модель контроллера SCSI (по умолчанию lsi); - searchdomain <строка> – cloud-init: устанавить домены поиска DNS для контейнера; - serial[n] (/dev/.+|socket) – последовательное устройство внутри ВМ (n от 0 до 3);

Команда	Описание
	- --shares <целое число (0–50000)> – объем разделяемой памяти (по умолчанию 1000); - --skiplock <логическое значение> – игнорировать блокировки (только root может использовать эту опцию); - --sockets <целое число> – количество сокетов процессора (по умолчанию 1); - --spice_enhancements [foldersharing=<1|0>] [,videostreaming=<off|all|filter>] – настройки для SPICE; - --sshkeys <путь к файлу> – cloud-init: настройка общедоступных ключей SSH (по одному ключу в строке, формат OpenSSH).; - --startup `[[порядок=]\d+] [,up=\d+] [,down=\d+] ` – поведение при запуске и выключении. Порядок – неотрицательное число, определяющее общий порядок запуска. Выключение выполняется в обратном порядке. up/down – задержка включения/выключения в секундах; - --tablet <логическое значение> – включить/отключить USB-планшет (по умолчанию 1); - --tags <строка> – теги ВМ; - --template <логическое значение> – включить/отключить шаблон (по умолчанию 0); - --tpmstate0 <диск> – настроить диск для хранения состояния TPM. Формат фиксированный – raw; - --usb[n] [[host=]<HOSTUSBDEVICE|spice>] [,mapping=<mapping-id>] [,usb3=<1|0>] – настройка USB-устройства (n – от 0 до 4, для версии машины >= 7.1 и ostype l26 или windows > 7, n может достигать 14); - --vcpus <целое число> – количество виртуальных процессоров с горячим подключением; - --vga [[type=]<enum>] [,clipboard=<vnc>] [,memory=<целое число>] – настройка VGA; - --virtio[n] <описание> – использовать жёсткий диск VIRTIO (n от 0 до 15); - --vmgenid <UUID> – установить идентификатор поколения ВМ (по умолчанию 1 – генерировать автоматически); - --vmstatestorage <строка> – хранилище по умолчанию для томов/файлов состояния ВМ; - --watchdog [[model=]<i6300esb|ib700>] [,action=<enum>] – создать сторожевое устройство виртуального оборудования
`qm showcmd <vmid> [ОПЦИИ]`	Показать командную строку, которая используется для запуска ВМ (информация для отладки). - vmid – идентификатор ВМ; - --pretty <логическое значение> – поместить каждый параметр на новой строке; - --snapshot <строка> – получить значения конфигурации из данного снимка
`qm shutdown <vmid> [ОПЦИИ]`	Выключение ВМ (эмуляция нажатия кнопки питания). Гостевой ОС будет отправлено событие ACPI. - vmid — идентификатор ВМ; - --forceStop <логическое значение> – убедиться, что ВМ остановлена (по умолчанию 0); - --keepActive <логическое значение> – не деактивировать тома хранения (по умолчанию 0); - --skiplock <логическое значение> – игнорировать блокировки (может использовать только root); - --timeout <целое число> – максимальный таймаут в секундах
`qm snapshot <vmid>`	Сделать снимок ВМ.

Команда	Описание
<snapname> [ОПЦИИ]	- vmid – идентификатор ВМ; - snapname – имя снимка; - --description <строка> – описание или комментарий; - --vmstate <логическое значение> – учитывать ОЗУ
qm start <vmid> [ОПЦИИ]	Запустить ВМ. - vmid – идентификатор ВМ; - --force-cpu <строка> – переопределить cpu QEMU заданной строкой; - --machine <тип> – указывает тип компьютера QEMU (например, pc+pve0); - --migratedfrom <строка> – имя узла кластера; - --migration_network <строка> – CIDR (под)сети, которая используется для миграции; - --migration_type <insecure | secure> – трафик миграции по умолчанию шифруется с использованием SSH-туннеля. В безопасных сетях эту функцию можно отключить для повышения производительности; - --keepActive <логическое значение> – не деактивировать тома хранения (по умолчанию 0); - --skiplock <логическое значение> – игнорировать блокировки (может использовать только root); - --stateuri <строка> – некоторые команды сохраняют/восстанавливают состояние из этого места; - --targetstorage <строка> – сопоставление исходных и целевых хранилищ. Предоставление только одного идентификатора хранилища сопоставляет все исходные хранилища с этим хранилищем. Если указать специальное значение 1, каждое исходное хранилище будет сопоставлено самому себе; - --timeout <целое число> – максимальный таймаут в секундах (по умолчанию max(30, память ВМ в ГБ))
qm status <vmid> [ОПЦИИ]	Показать статус ВМ. - vmid – идентификатор ВМ; - --verbose <логическое значение> – подробный вывод
qm stop <vmid> [ОПЦИИ]	Останов ВМ (эмуляция выдергивания вилки). Процесс qemu немедленно завершается. - vmid – идентификатор ВМ; - --keepActive <логическое значение> – не деактивировать тома хранения (по умолчанию 0); - --migratedfrom <строка> – имя узла кластера; - --skiplock <логическое значение> – игнорировать блокировки (может использовать только root); - --timeout <целое число> – максимальный таймаут в секундах
qm suspend <vmid> [ОПЦИИ]	Приостановить ВМ. - vmid – идентификатор ВМ; - --skiplock <логическое значение> – игнорировать блокировки (может использовать только root); - --statestorage <строка>хранилище состояния ВМ (должна быть указана опция --todisk); - --todisk <логическое значение> – приостанавливает работу ВМ на диск. Будет возобновлено при следующем запуске ВМ (по умолчанию 0)
qm template <vmid> [ОПЦИИ]	Создать шаблон. - vmid – идентификатор ВМ; - --disk <диск> – если в базовый образ нужно преобразовать только один диск (например, sata1)
qm terminal <vmid> [ОПЦИИ]	Открыть терминал с помощью последовательного устройства. На ВМ должно быть настроено последовательное устройство, например, Serial0:

Команда	Описание
	Socket. - vmid – идентификатор ВМ; - --escape <строка> – escape-символ (по умолчанию ^O); - --iface <serial0 | serial1 | serial2 | serial3> – последовательное устройство (по умолчанию используется первое подходящее устройство)
`qm unlink`	Псевдоним для `qm disk unlink`
`qm unlock <vmid>`	Разблокировать ВМ
`qm vncproxy <vmid>`	Проксировать VNC-трафик ВМ на стандартный ввод/вывод
`qm wait <vmid>` `[ОПЦИИ]`	Подождать, пока ВМ не будет остановлена. - vmid – идентификатор ВМ; - --timeout <целое число> – максимальный таймаут в секундах (по умолчанию – не ограничено)

Примеры использования утилиты `qm`:

- создать ВМ, используя ISO-файл, загруженный в локальное хранилище, с диском IDE 21 ГБ, в хранилище local-lvm:

```
# qm create 300 -ide0 local-lvm:21 -net0 e1000 -cdrom local:iso/alt-server-9.1-x86_64.iso
```

- запуск ВМ с VM ID 109:

```
# qm start 109
```

- отправить запрос на отключение, и дождаться остановки ВМ:

```
# qm shutdown 109 && qm wait 109
```

- отправить сочетание клавиш <CTRL>+<SHIFT> на ВМ:

```
# qm sendkey 109 ctrl-shift
```

- войти в интерфейс монитора QEMU и вывести список доступных команд:

```
# qm monitor 109
qm> help
```

4.7.4 Скрипты-ловушки (hookscripts)

Скрипты-ловушки позволяют выполнить скрипт на узле виртуализации при запуске или остановке ВМ или контейнера. Скрипт может вызываться на разных этапах жизни ВМ: до запуска (pre-start), после запуска (post-start), до остановки (pre-stop), после остановки (post-stop). Скрипты-ловушки должны находиться в хранилище, поддерживающем «фрагменты» (сниппеты).

Для возможности использовать данную функцию необходимо создать скрипт в каталоге сниппетов (например, для хранилища local по умолчанию это /var/lib/vz/snippets) и добавить его к ВМ или контейнеру.

Примечание. При переносе ВМ на другой узел, следует убедиться, что скрипт ловушки также доступен на целевом узле (хранилище со сниппетами, должно быть доступно на всех узлах, на которые будет выполняться миграция).

Добавить скрипт-ловушку к ВМ или контейнеру можно с помощью свойства hookscript:

```
# qm set <vmid> --hookscript <storage>:snippets/<script_file>
# pct set <vmid> --hookscript <storage>:snippets/<script_file>
```

где <script_file> – исполняемый файл скрипта.

Например:

```
# qm set 103 --hookscript snippet:snippets/guest-hookscript.pl
update VM 103: -hookscript snippet:snippets/guest-hookscript.pl
```

Примечание. В настоящее время добавить скрипт-ловушку можно только в командной строке. В веб-интерфейсе можно только просмотреть список скриптов в хранилище (Рис. 181) и скрипт, добавленный к ВМ (Рис. 182).

Хранилище snippet на узле pve01

Рис. 181

Скрипт-ловушка для ВМ 103

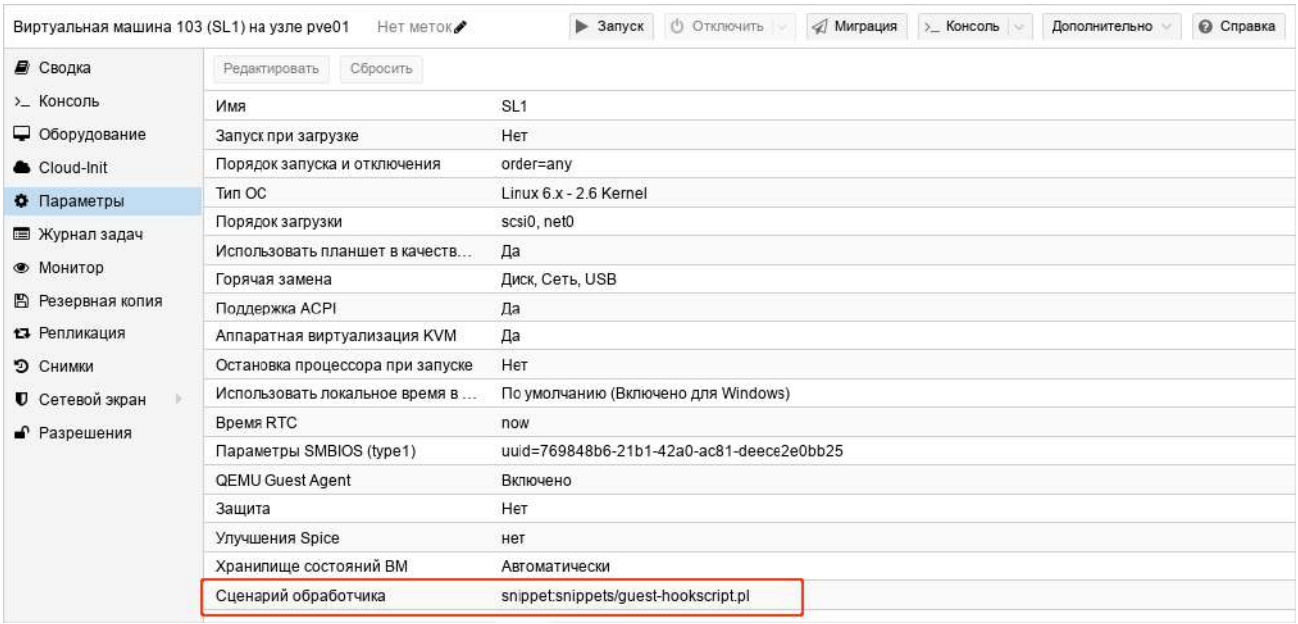


Рис. 182

Пример скрипта-ловушки на Perl (файл guest-hookscript.pl):

```
#!/usr/bin/perl

# Example hook script for PVE guests (hookscript config option)
```

```
use strict;
use warnings;

print "GUEST HOOK: " . join(' ', @ARGV). "\n";

# First argument is the vmid

my $vmid = shift;

# Second argument is the phase

my $phase = shift;

if ($phase eq 'pre-start') {

    # Первый этап 'pre-start' будет выполнен до запуска ВМ
    # Выход с code != 0 отменит старт ВМ

    print "$vmid is starting, doing preparations.\n";

    # print "preparations failed, aborting."
    # exit(1);

} elsif ($phase eq 'post-start') {

    # Второй этап 'post-start' будет выполнен после успешного
    # запуска ВМ
    system("/root/date.sh $vmid");
    print "$vmid started successfully.\n";

} elsif ($phase eq 'pre-stop') {

    # Третий этап 'pre-stop' будет выполнен до остановки ВМ через API
    # Этап не будет выполнен, если ВМ остановлена изнутри,
    # например, с помощью 'poweroff'

    print "$vmid will be stopped.\n";
} elsif ($phase eq 'post-stop') {

    # Последний этап 'post-stop' будет выполнен после остановки ВМ
    # Этап должен быть выполнен даже в случае сбоя или неожиданной остановки ВМ
```

```
    print "$vmid stopped. Doing cleanup.\n";

} else {
    die "got unknown phase '$phase'\n";
}

exit(0);
```

Функция system() используется для вызова сценария bash, которому передается VMID в качестве аргумента. Текст отладки выводится в «консоль»/stdout. Текст будет помещен в журналы задач ВМ (Рис. 183) и узла PVE. Сообщения pre-start, post-start и pre-stop будут опубликованы в обоих журналах. Сообщения post-stopt будут публиковаться только в журналах истории задач узла PVE (поскольку ВМ уже остановлена).

Выполнение скрипта guest-hookscript.pl при запуске ВМ

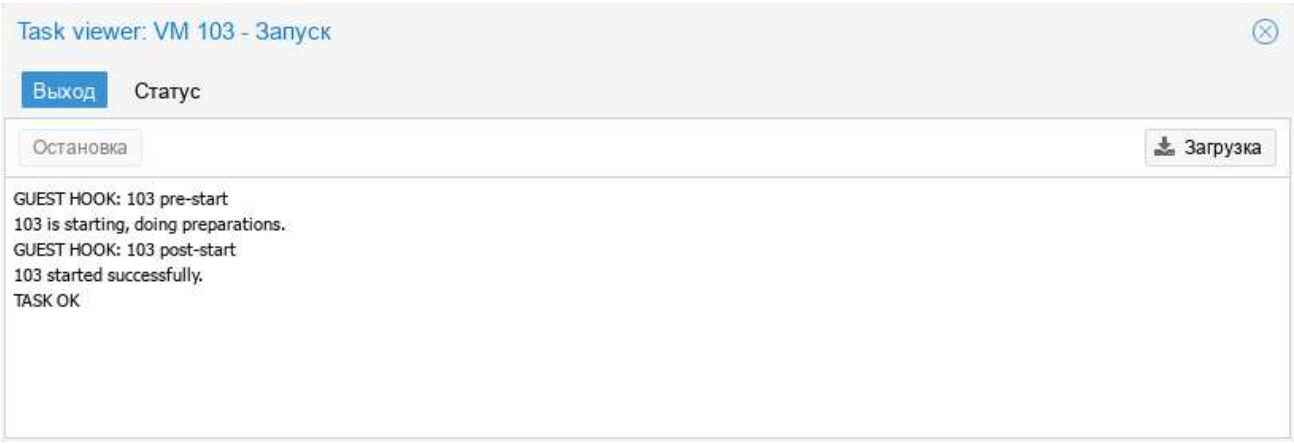


Рис. 183

Пример скрипта-ловушки на bash:

```
#!/bin/bash
if [ $2 == "pre-start" ]
then
echo "Запуск ВМ $1" >> /root/test.txt
date >> /root/test.txt
fi
```

4.7.5 Доступ к ВМ

По умолчанию PVE предоставляет доступ к ВМ через noVNC и/или SPICE. Рекомендуется использовать их, когда это возможно.

Использование протокола SPICE позволяет задействовать множество возможностей, в том числе, проброс USB, смарт-карт, принтеров, звука, получить более тесную интеграцию с окном

гостевой системы (бесшовную работу мыши, клавиатуры, динамическое переключение разрешения экрана, общий с гостевой системой буфер обмена для операций копирования/вставки). Для возможности использования SPICE:

- на хосте, с которого происходит подключение, должен быть установлен клиент SPICE (например, пакет virt-viewer):
- для параметра «Экран» ВМ должно быть установленно значение VirtIO, SPICE (qxl) (см. «Настройки дисплея»).

При подключении к ВМ с использованием noVNC, консоль открывается во вкладке браузера (не нужно устанавливать клиентское ПО).

Для доступа к ВМ следует выбрать её в веб-интерфейсе, нажать кнопку «Консоль» и в выпадающем меню выбрать нужную консоль (Рис. 184).

Кнопка «Консоль»

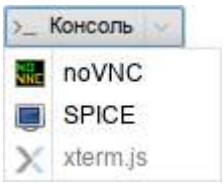


Рис. 184

Консоль noVNC также можно запустить, выбрав вкладку «Консоль» для ВМ (Рис. 185).

Консоль noVNC

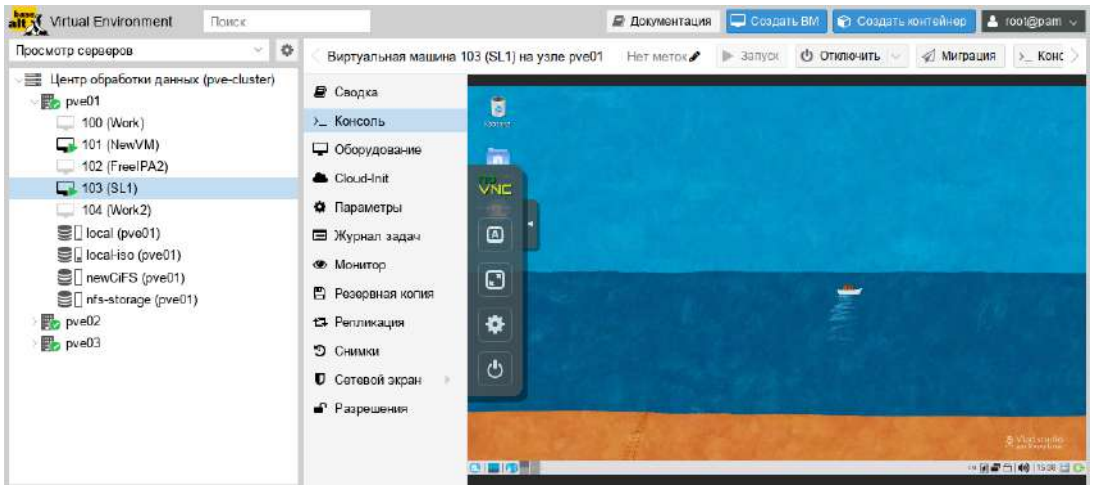


Рис. 185

Если нужен независимый от браузера доступ, можно также использовать внешний клиент VNC. Для этого в файл конфигурации ВМ `/etc/pve/qemu-server/<VMID>.conf` необходимо добавить строку с указанием номера дисплея VNC (в примере – 55):

```
args: -vnc 0.0.0.0:55
```

Или, чтобы включить защиту паролем:

```
args: -vnc 0.0.0.0:55,password=on
```

Если была включена защита паролем, необходимо установить пароль (после запуска ВМ). Пароль можно установить на вкладке «Монитор», выполнив команду:

```
set_password vnc newvnc -d vnc2
```

В данном примере, при подключении будет запрашиваться пароль: newvnc. Максимальная длина пароля VNC: 8 символов. После перезапуска ВМ указанную выше команду необходимо повторить, чтобы снова установить пароль.

Примечание. Номер дисплея VNC можно выбрать произвольно, но каждый номер должен встречаться только один раз. Служба VNC прослушивает порт 5900+номер_дисплея. Соединения noVNC используют номер дисплея 0 и последующие, поэтому во избежание конфликтов рекомендуется использовать более высокие номера.

Для подключения клиента VNC следует указать IP-адрес хоста с ВМ и порт (в приведенном выше примере – 5955).

4.7.6 Внесение изменений в ВМ

Вносить изменения в конфигурацию ВМ можно и после ее создания. Для того чтобы внести изменения в конфигурацию ВМ, необходимо выбрать ВМ и перейти на вкладку «Оборудование» (Рис. 186). На этой вкладке следует выбрать ресурс и нажать кнопку «Редактировать» для выполнения изменений.

Оборудование ВМ

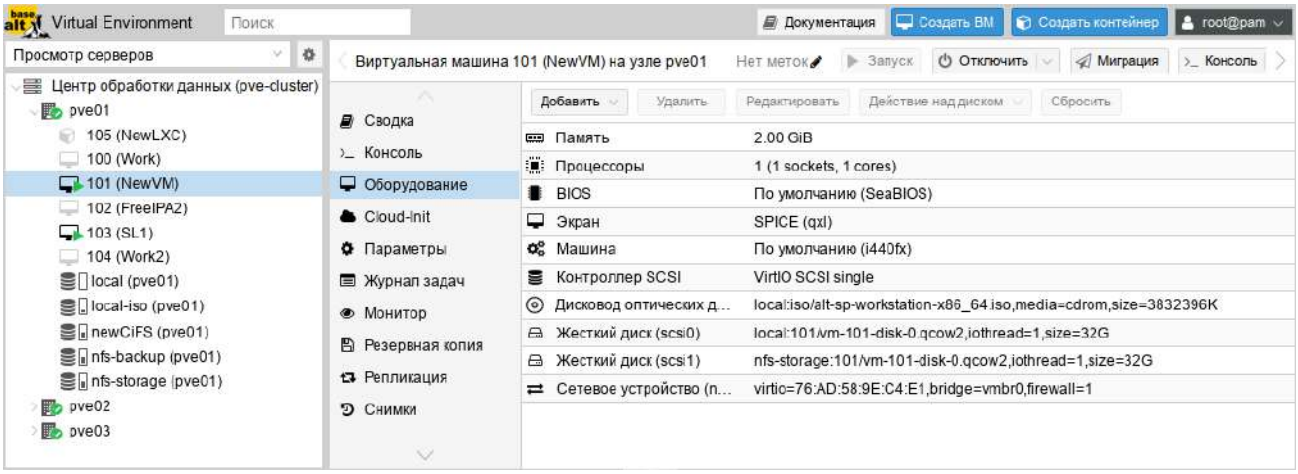


Рис. 186

Примечание. В случаях, когда изменение не может быть выполнено в горячем режиме, оно будет зарегистрировано как ожидающее изменение (Рис. 187) (выделяется цветом). Такие изменения будут применены только после перезагрузки ВМ.

Изменения, которые будут применены после перезапуска ВМ

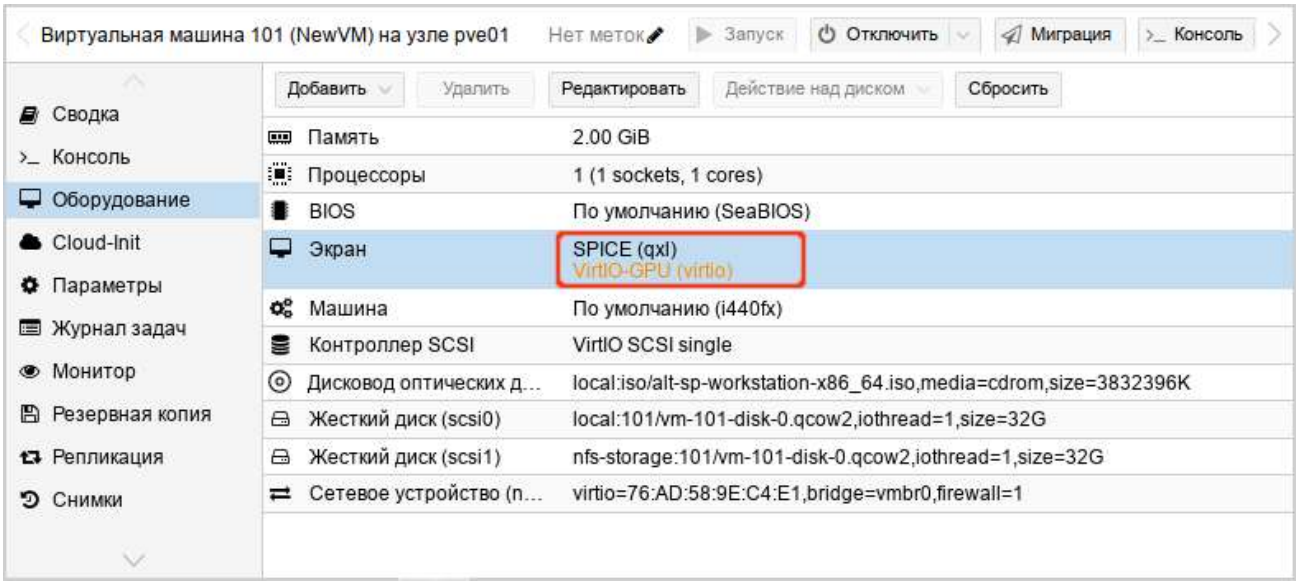


Рис. 187

4.7.6.1 Управление образами виртуальных дисков

Образ виртуального диска является файлом или группой файлов, в которых ВМ хранит свои данные.

`qemu-img` – утилита для манипулирования с образами дисков машин QEMU. `qemu-img` позволяет выполнять операции по созданию образов различных форматов, конвертировать файлы-образы между этими форматами, получать информацию об образах и объединять снимки ВМ для тех форматов, которые это поддерживают (подробнее см. раздел «Утилита qemu-img»).

Примеры, использования утилиты `qemu-img`:

- преобразование (конвертация) vmdk-образа виртуального накопителя VMware под названием test в формат qcow2:

```
# qemu-img convert -f vmdk test.vmdk -O qcow2 test.qcow2
```

- создание образа test в формате RAW, размером 40 ГБ:

```
# qemu-img create -f raw test.raw 40G
```

- изменение размера виртуального диска:

```
# qemu-img resize -f raw test.raw 80G
```

- просмотр информации об образе:

```
# qemu-img info test.raw
```

Для управления образами виртуальных дисков в веб-интерфейсе PVE необходимо выбрать ВМ и перейти на вкладку «Оборудование». После выбора образа диска станут доступными кнопки (Рис. 188): «Добавить», «Отключить», «Редактировать», «Изменить размер», «Переназначить владельца», «Переместить хранилище».

Вкладка «Оборудование». Управление образом виртуального диска

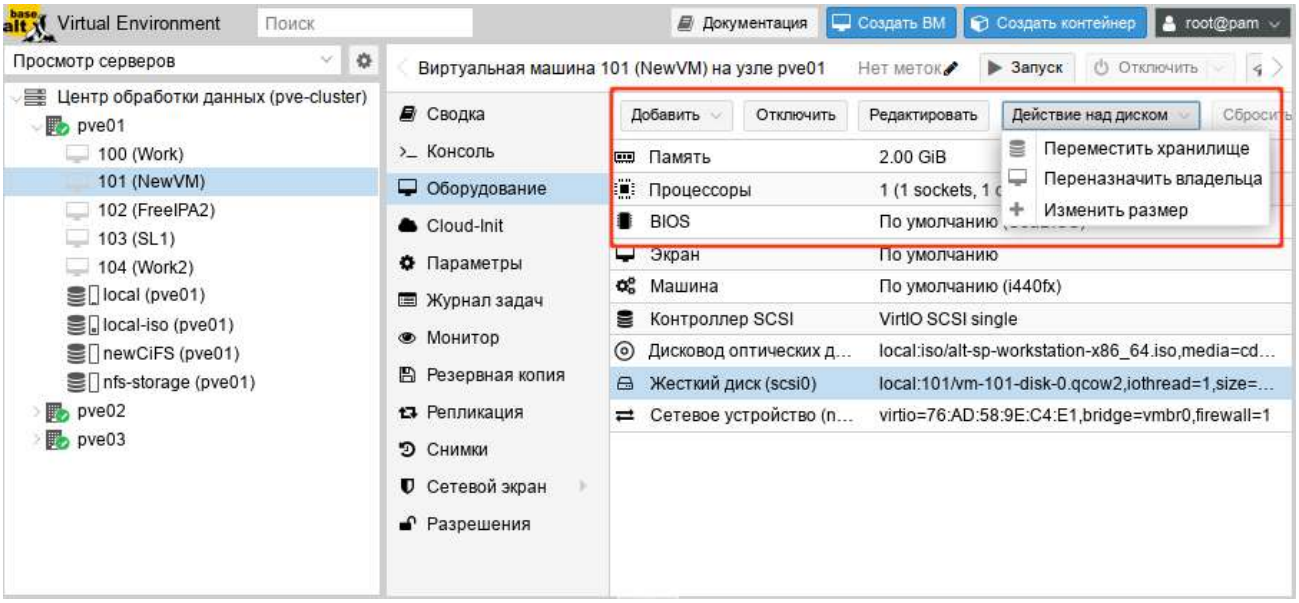


Рис. 188

4.7.6.1.1 Добавление виртуального диска в ВМ

Для добавления образа виртуального диска к ВМ необходимо:

1) перейти на вкладку «Оборудование» (Рис. 188);

2) нажать кнопку «Добавить» и выбрать в выпадающем списке пункт «Жесткий диск» (Рис. 189);

3) указать параметры жесткого диска (Рис. 190) и нажать кнопку «Добавить».

Кнопка «Добавить»→«Жесткий диск»

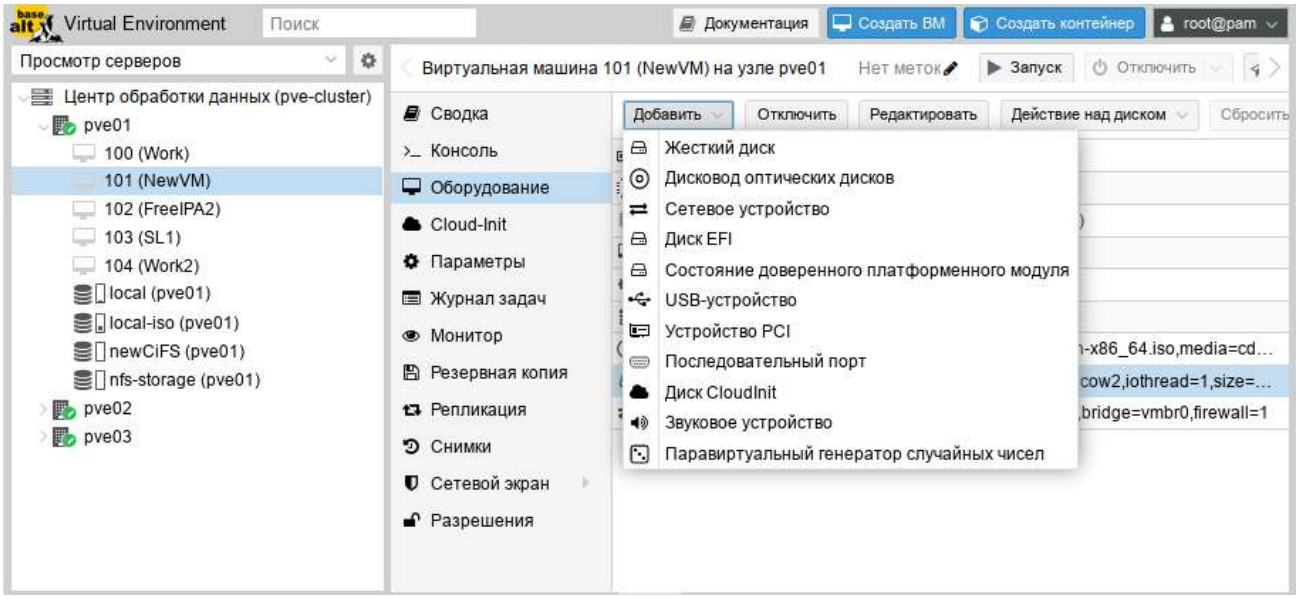


Рис. 189

Опции добавления жесткого диска

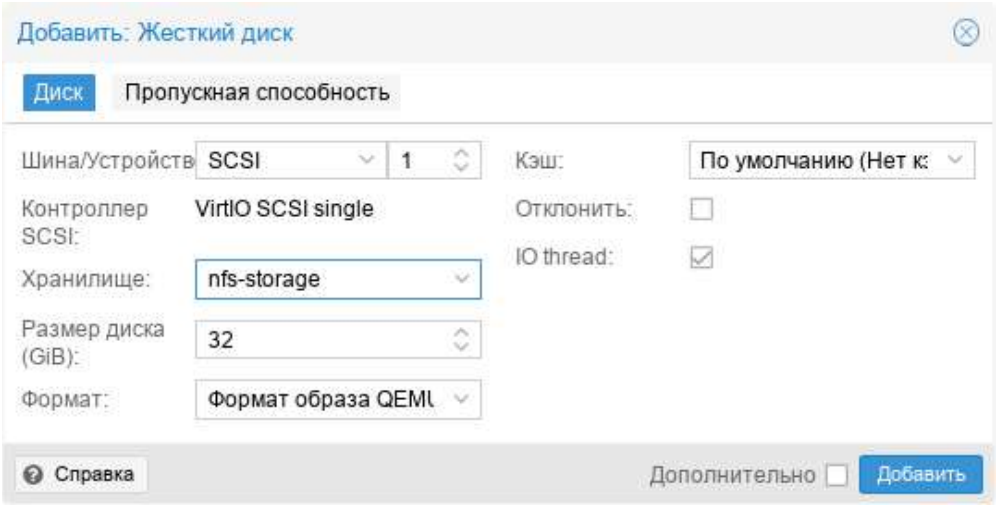


Рис. 190

4.7.6.1.2 Удаление образа виртуального диска

Для удаления образа виртуального диска необходимо:

1) перейти на вкладку «Оборудование» (Рис. 188);

2) выбрать образ диска ВМ;

3) нажать кнопку «Отключить»;

4) в окне подтверждения нажать кнопку «Да» для подтверждения действия. При этом виртуальный диск будет отсоединен от ВМ, но не удален полностью. Он будет присутствовать в списке оборудования ВМ как «Неиспользуемый диск» (Рис. 191).

«Неиспользуемый диск»

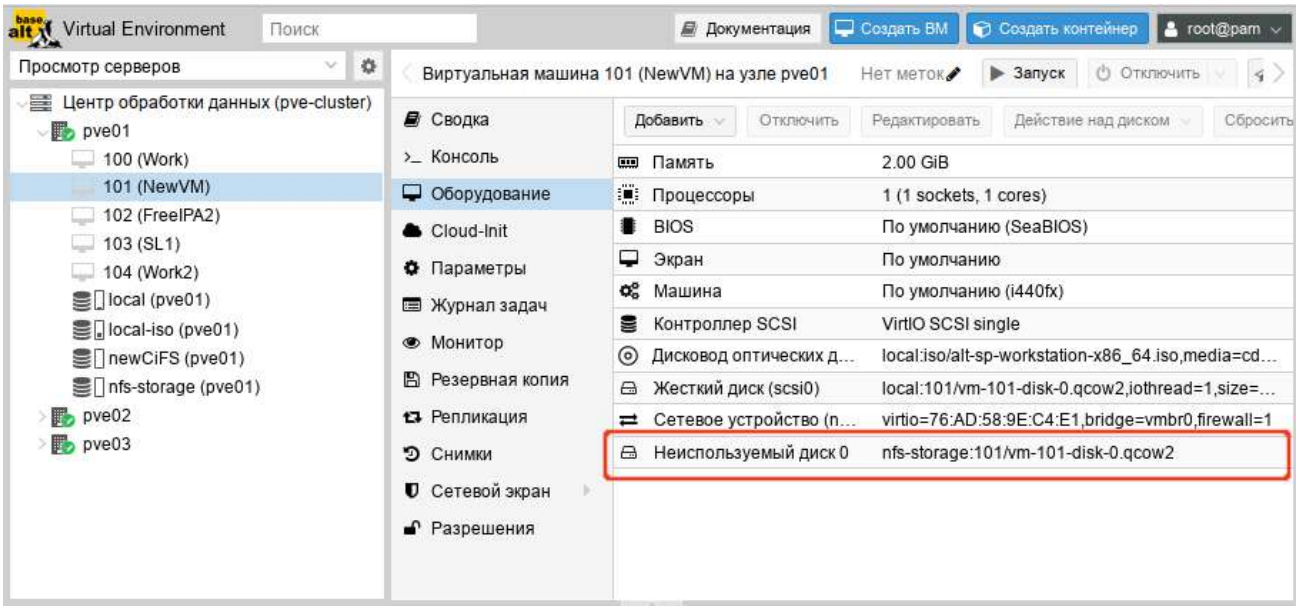


Рис. 191

Чтобы удалить образ диска окончательно, следует выбрать неиспользуемый диск и нажать кнопку «Удалить».

Если образ диска был отключен от ВМ по ошибке, можно повторно подключить его к ВМ, выполнив следующие действия:

5) выбрать неиспользуемый диск;

6) нажать кнопку «Редактировать»;

7) в открывшемся диалоговом окне (Рис. 192) изменить, если это необходимо, параметры «Шина/Устройство».

8) нажать кнопку «Добавить» для повторного подключения образа диска.

Подключение неиспользуемого диска

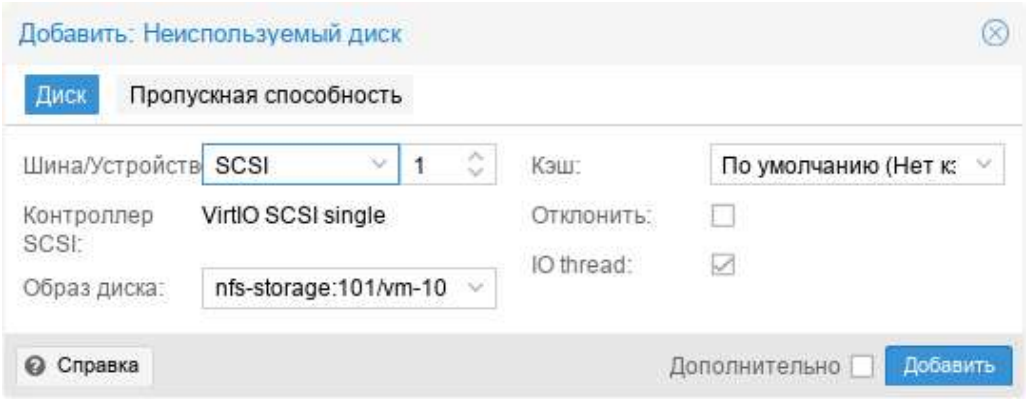


Рис. 192

4.7.6.1.3 Изменение размера диска

Функция изменения размера поддерживает только увеличение размера файла образа виртуального диска.

При изменении размера образа виртуального диска изменяется только размер файла образа виртуального диска. После изменения размера файла, разделы жесткого диска должны быть изменены внутри самой ВМ.

Для изменения размера виртуального диска необходимо:

1) перейти на вкладку «Оборудование» (Рис. 188);

2) выбрать образ виртуального диска.

3) нажать кнопку «Действие над диском» → «Изменить размер»;

4) в открывшемся диалоговом окне в поле «Увеличение размера (GiB)» ввести значение, на которое необходимо увеличить размер диска. Например, если размер существующего диска составляет 20 ГБ, для изменения размера диска до 30 ГБ следует ввести число 10 (Рис. 193);

5) нажать кнопку «Изменить размер диска» для завершения изменения размера.

Изменение размера диска

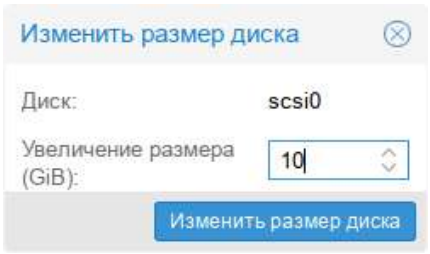


Рис. 193

Команда изменения размера виртуального диска:

```
# qm resize <vm_id> <virtual_disk> [+]<size>
```

Примечание. Если указать размер диска со знаком «+», то данное значение добавится к реальному размеру тома, без знака «+» указывается абсолютное значение. Уменьшение размера диска не поддерживается. Например, изменить размер виртуального диска до 80 ГБ:

```
# qm resize 100 scsi1 80G
```

4.7.6.1.4 Перемещение диска в другое хранилище

Образы виртуального диска могут перемещаться с одного хранилища на другое в пределах одного кластера.

Для перемещения образа диска необходимо:

1) перейти на вкладку «Оборудование» (Рис. 188);
2) выбрать образ диска, который необходимо переместить;
3) нажать кнопку «Действие над диском» → «Переместить хранилище»;
4) в открывшемся диалоговом окне (Рис. 194) в выпадающем меню «Целевое хранилище» выбрать хранилище-получатель, место, куда будет перемещен образ виртуального диска;
5) в выпадающем меню «Формат» выбрать формат образа диска. Этот параметр полезен для преобразования образа диска из одного формата в другой;
6) отметить, если это необходимо, пункт «Удалить источник» для удаления образа диска из исходного хранилища после его перемещения в новое хранилище;
7) нажать кнопку «Переместить диск».

Команда перемещения образа диска в другое хранилище:

```
# qm move-disk <vm_id> <virtual_disk> <storage>
```

Диалоговое окно перемещения диска

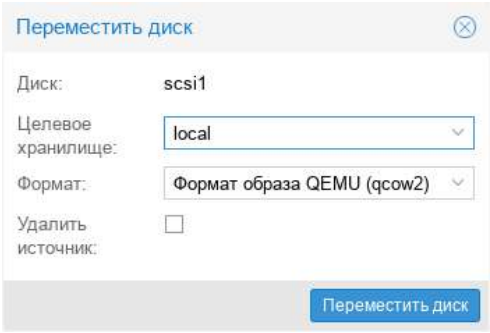


Рис. 194

4.7.6.1.5 Переназначение диска другой ВМ

При переназначении образа диска другой ВМ, диск будет удалён из исходной ВМ и подключен к целевой ВМ.

Для переназначения образа диска другой ВМ необходимо:

1) перейти на вкладку «Оборудование» (Рис. 188);
2) выбрать образ диска, который необходимо переназначить;
3) нажать кнопку «Действие над диском» → «Переназначить владельца»;
4) в открывшемся диалоговом окне (Рис. 195) в выпадающем «Целевой гость» выбрать целевую ВМ (место, куда будет перемещен образ виртуального диска);
5) выбрать нужные параметры в выпадающем меню «Шина/Устройство»;
6) нажать кнопку «Переназначить диск».

Диалоговое окно переназначения диска

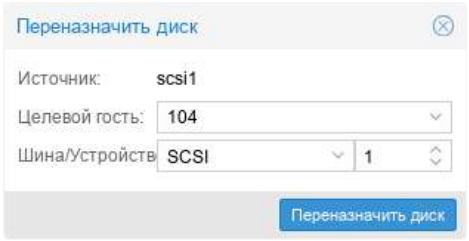


Рис. 195

Команда переназначения образа диска другой ВМ:

```
# qm move-disk <vm_id> <virtual_disk> --target-vmid <vm_id> --target-disk <virtual_disk>
```

Пример удаления образа диска scsi0 из ВМ 107 и подключение его как scsi1 к ВМ 10007:

```
# qm move-disk 107 scsi0 --target-vmid 10007 --target-disk scsi1
```

4.7.6.2 Настройки дисплея

QEMU может виртуализировать разные типы оборудования VGA (Рис. 196), например:

- std («Стандартный VGA») – эмулирует карту с расширениями Bochs VBE;
- vmware («Совместимый с VMware») – адаптер, совместимый с VMWare SVGA-II;

- qxl («SPICE») – паравиртуализированная видеокарта QXL. Выбор этого параметра включает SPICE (протокол удаленного просмотра) для ВМ;
- virtio («VirtIO-GPU») – стандартный драйвер графического процессора virtio;
- virtio-gl («VirGL GPU») – виртуальный 3D-графический процессор для использования внутри ВМ, который может переносить рабочие нагрузки на графический процессор хоста.

PVE. Настройки дисплея

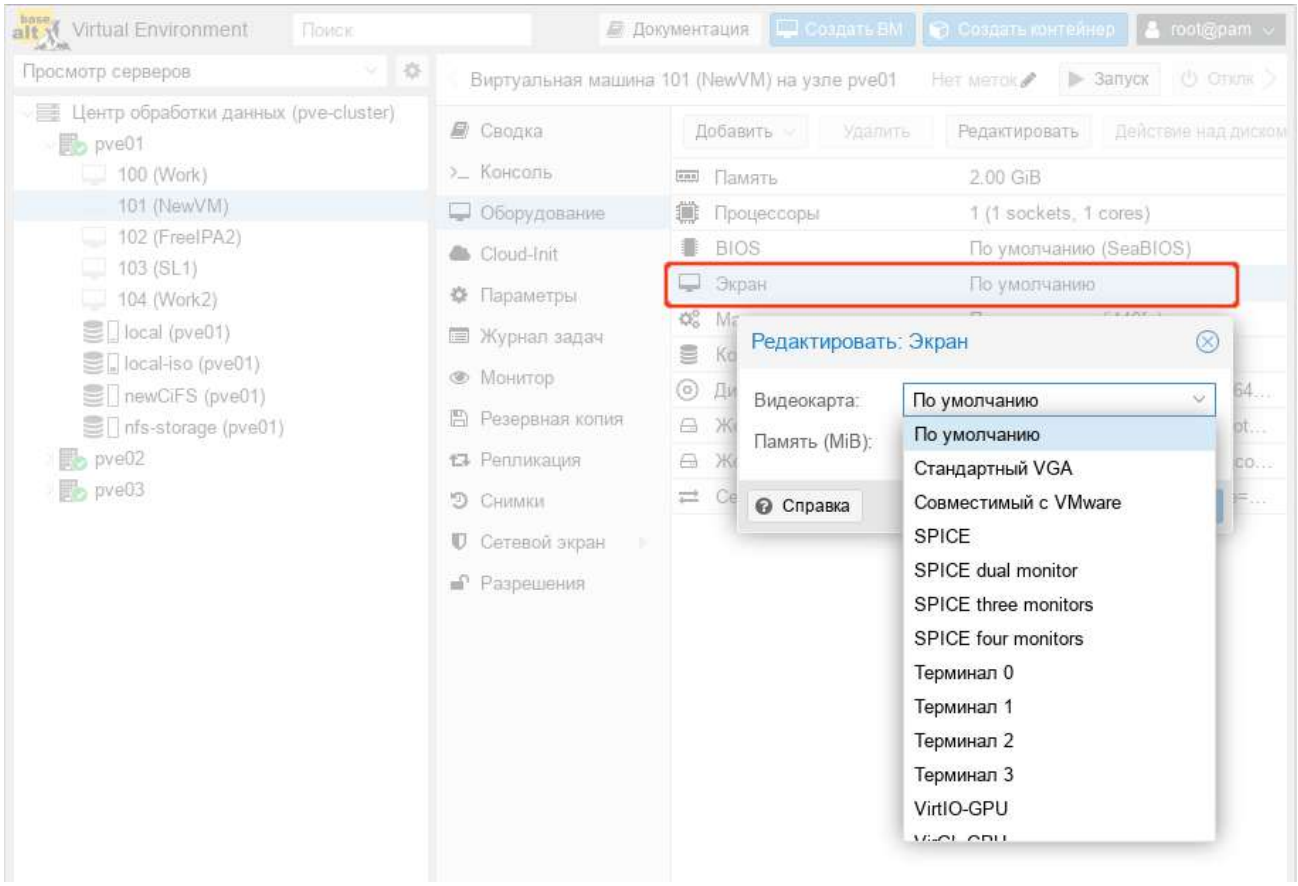


Рис. 196

Примечание. Для типов дисплеев «VirtIO» и «VirGL» по умолчанию включена поддержка SPICE.

Примечание. Для подключения к SPICE-серверу может использоваться любой SPICE-клиент (например, remote-viewer из пакета virt-viewer).

Можно изменить объем памяти, выделяемый виртуальному графическому процессору (поле «Память (MiB)»). Увеличение объема памяти может обеспечить более высокое разрешение внутри ВМ, особенно при использовании SPICE/QXL.

Поскольку память резервируется устройством дисплея, выбор режима нескольких мониторов для SPICE (например, qxl2 для двух мониторов) имеет некоторые последствия:

- ВМ с ОС Windows требуется устройство для каждого монитора. Поэтому PVE предоставляет ВМ дополнительное устройство для каждого монитора. Каждое устройство получает указанный объем памяти;
- ВМ с ОС Linux всегда могут включать больше виртуальных мониторов, но при выборе режима нескольких мониторов, объём памяти, предоставленный устройству, умножается на количество мониторов.

Выбор serialX («Терминал X») в качестве типа дисплея, отключает выход VGA и перенаправляет веб-консоль на выбранный последовательный порт. В этом случае настроенный параметр памяти дисплея игнорируется.

4.7.6.3 Дополнительные функции SPICE

Дополнительно в PVE можно включить две дополнительные функции SPICE:

- общий доступ к папкам – доступ к локальной папке из ВМ;
- потоковое видео – области быстрого обновления кодируются в видеопоток.

Включение дополнительных функций SPICE:

- в веб-интерфейсе (Рис. 197) (пункт «Улучшения SPICE» в разделе «Параметры» ВМ);
- в командной строке:

```
# qm set VMID -spice_enhancements foldersharing=1,videostreaming=all
```

Примечание. Чтобы использовать дополнительные функции SPICE, для параметра «Экран»ВМ должно быть установленно значение SPICE (qxl).

4.7.6.3.1 Общий доступ к папкам (Folder Sharing)

Для возможности получения доступа к локальной папке, внутри ВМ должен быть установлен пакет `spice-webdavd`. В этом случае общая папка будет доступна через локальный сервер WebDAV по адресу http://localhost:9843.

Примечание. Чтобы открыть общий доступ к папке, следует в меню virt-viewer выбрать пункт «Настройки» («Preferences»), в открывшемся окне установить отметку «Общая папка» и выбрать папку для перенаправления (Рис. 198).

PVE. Дополнительные функции SPICE

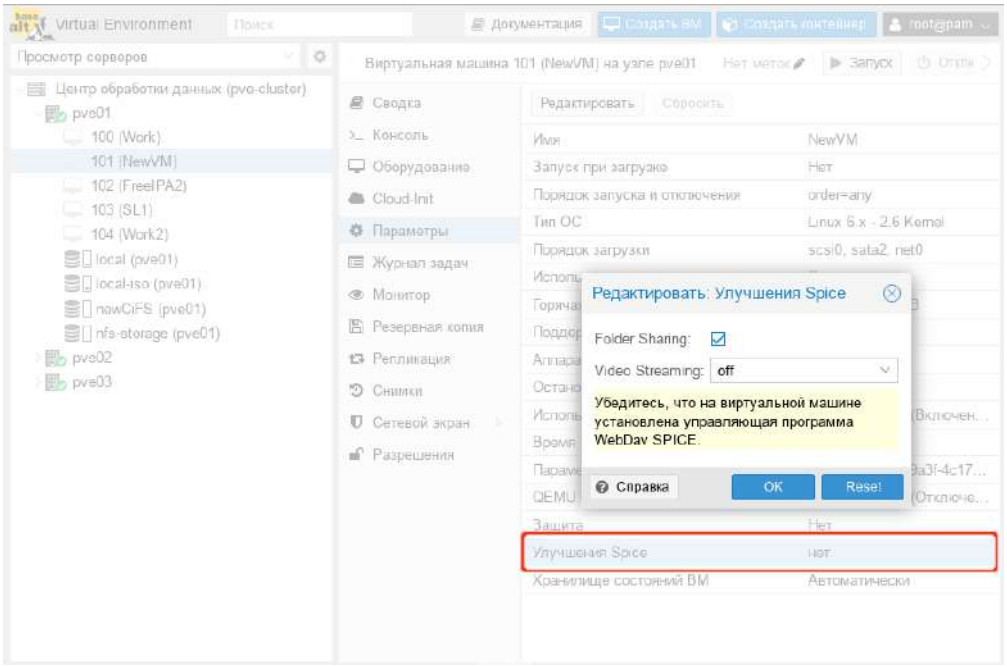


Рис. 197

Совместный доступ к папке

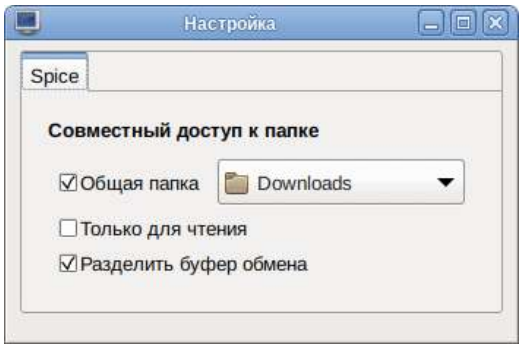


Рис. 198

Если в ВМ общая папка не отображается, следует проверить, что служба WebDAV (spice-webdavd) запущена. Может также потребоваться перезапустить сеанс SPICE.

Для возможности доступа к общей папке из файлового менеджера, а не из браузера, внутри ВМ должен быть установлен пакет davfs2.

П р и м е ч а н и е . Для доступа к общей папке из файлового менеджера:

- «Dolphin» – выбрать пункт «Сеть»→«Сетевые службы»→«Сетевой каталог WebDav»→«Spice client folder»;
- «Thunar» – в адресной строке ввести адрес с указанием протокола dav или davs (dav://localhost:9843/).

4.7.6.3.2 Потоковое видео (Video Streaming)

Если потоковое видео включено, доступны две опции:

- «all» – все области быстрого обновления кодируются в видеопоток;
- «filter» – для принятия решения о том, следует ли использовать потоковое видео, используются дополнительные фильтры.

4.7.6.4 Проброс USB

Для проброса USB-устройства в ВМ необходимо:

1) перейти на вкладку «Оборудование» (Рис. 188);

2) нажать кнопку «Добавить» и выбрать в выпадающем списке пункт «USB-устройство» (Рис. 199);

Кнопка «Добавить»→«Устройство USB»

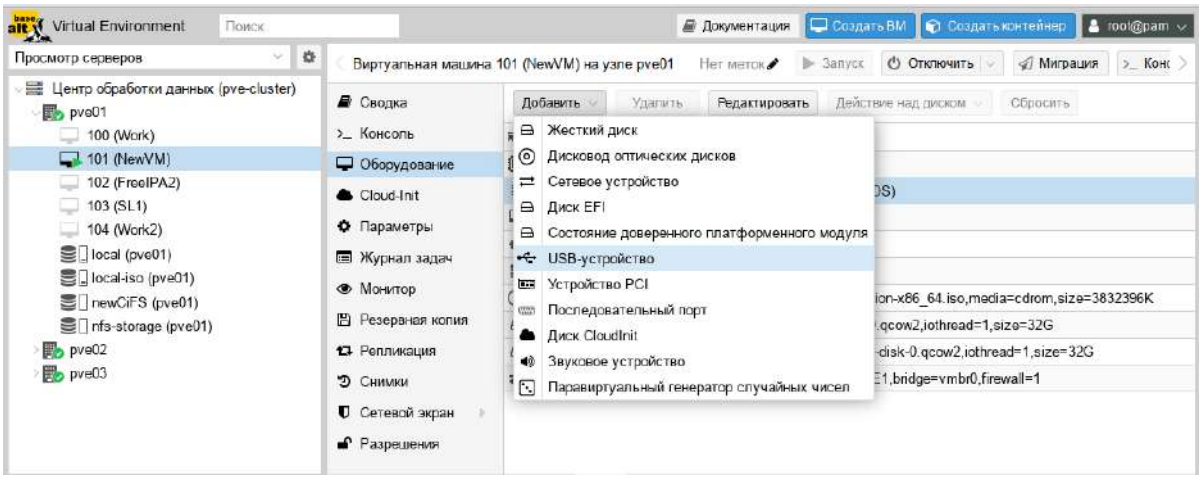


Рис. 199

3) откроется окно добавления устройства, в котором можно выбрать режим проброса:

- «Порт Spice» – сквозная передача SPICE USB (Рис. 200) (позволяет пробросить USB-устройство с клиента SPICE);
- «Использовать устройство USB по номеру» – проброс в ВМ конкретного USB-устройства (Рис. 201). USB-устройство можно выбрать в выпадающем списке «Выберите устройство» или ввести вручную, указав *<ID-производителя>:<ID-устройства>* (можно получить из вывода команды `lsusb`).
- «Использовать порт USB» – проброс конкретного порта (Рис. 202) (в ВМ будет проброшено любое устройство, вставленное в этот порт). USB-порт можно выбрать в выпадающем списке «Выберите порт» или указать вручную, указав *<Номер_шины>:<Путь_к_порту>* (можно получить из вывода команды `lsusb -t`).

4) нажать кнопку «Добавить»;

5) остановить и запустить ВМ (перезагрузки недостаточно).

Порт Spice

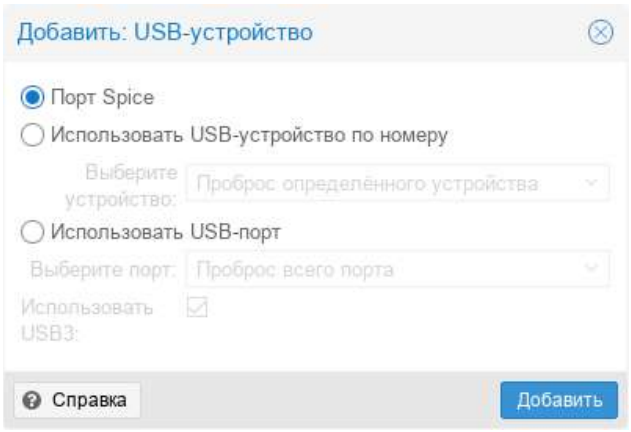


Рис. 200

Использовать устройство USB по номеру

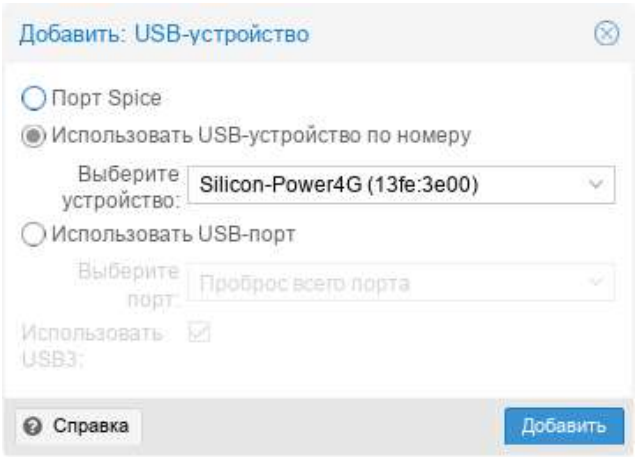


Рис. 201

Использовать порт USB

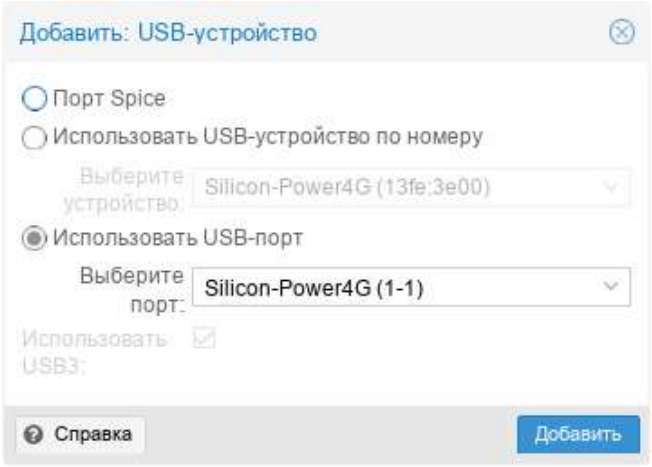


Рис. 202

Примечание. Список подключенных к ВМ и хосту USB-устройств можно получить, введя на вкладке «Монитор» соответственно команды `info usb` или `info usbhost` (Рис. 203).

Список подключенных к ВМ и хосту USB-устройств

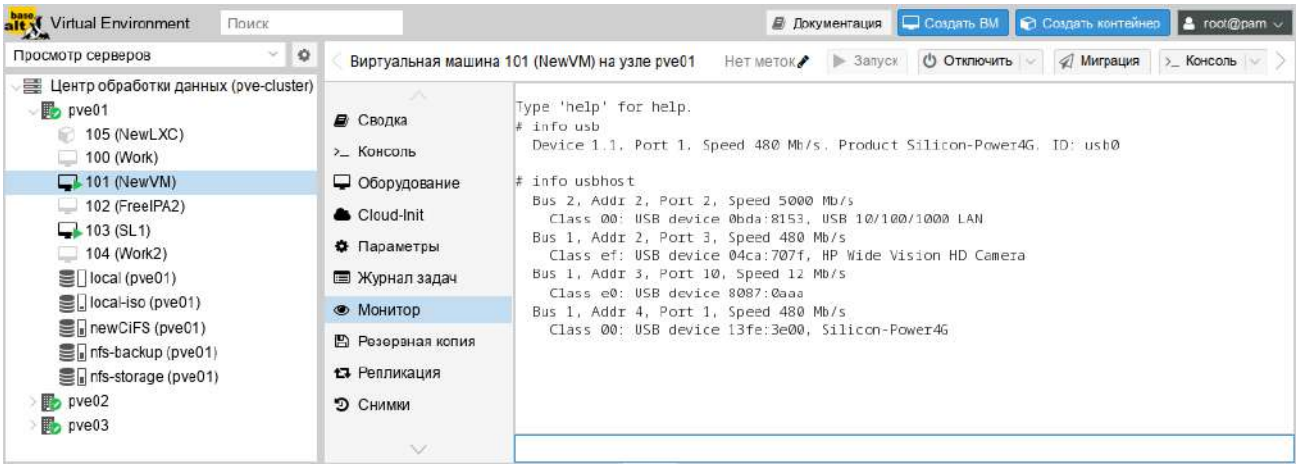


Рис. 203

Если USB-устройство присутствует в конфигурации ВМ (и для него указаны «Использовать устройство USB по номеру» или «Использовать порт USB») при запуске ВМ, но отсутствует на хосте, ВМ будет загружена без проблем. Как только устройство/порт станет доступным на хосте, оно будет проброшено в ВМ.

Примечание. Использование проброса типа «Использовать устройство USB по номеру» или «Использовать порт USB» не позволит переместить ВМ на другой хост, поскольку оборудование доступно только на хосте, на котором в данный момент находится ВМ.

4.7.6.5 BIOS и UEFI

По умолчанию, в качестве прошивки, используется SeaBIOS, который эмулирует BIOS x86. Можно также выбрать OVMF, который эмулирует UEFI.

При использовании OVMF, необходимо учитывать несколько моментов:

- для сохранения порядка загрузки, должен быть добавлен диск EFI (этот диск будет включен в резервные копии и моментальные снимки, и может быть только один);
- при использовании OVMF с виртуальным дисплеем (без проброса видеокарты в ВМ) необходимо установить разрешение клиента в меню OVMF (которое можно вызвать нажатием кнопки ESC во время загрузки) или выбрать SPICE в качестве типа дисплея.

Пример изменения прошивки ВМ на UEFI:

1) поменять тип прошивки на UEFI (Рис. 204);
2) добавить в конфигурацию ВМ диск EFI (Рис. 205).

PVE. Настройка BIOS

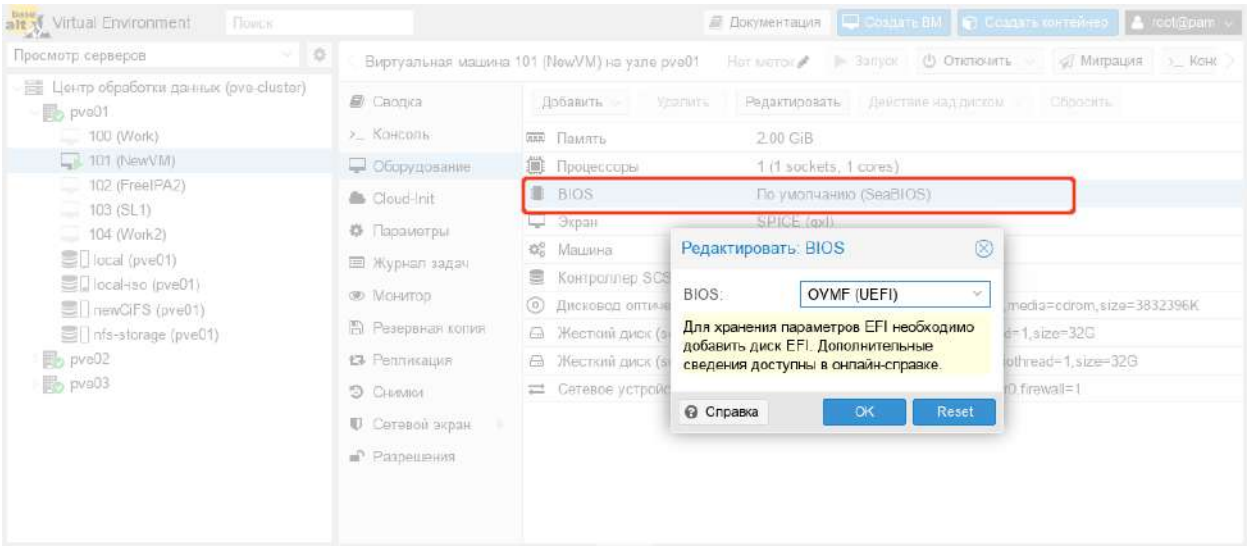


Рис. 204

PVE. Добавление диска EFI

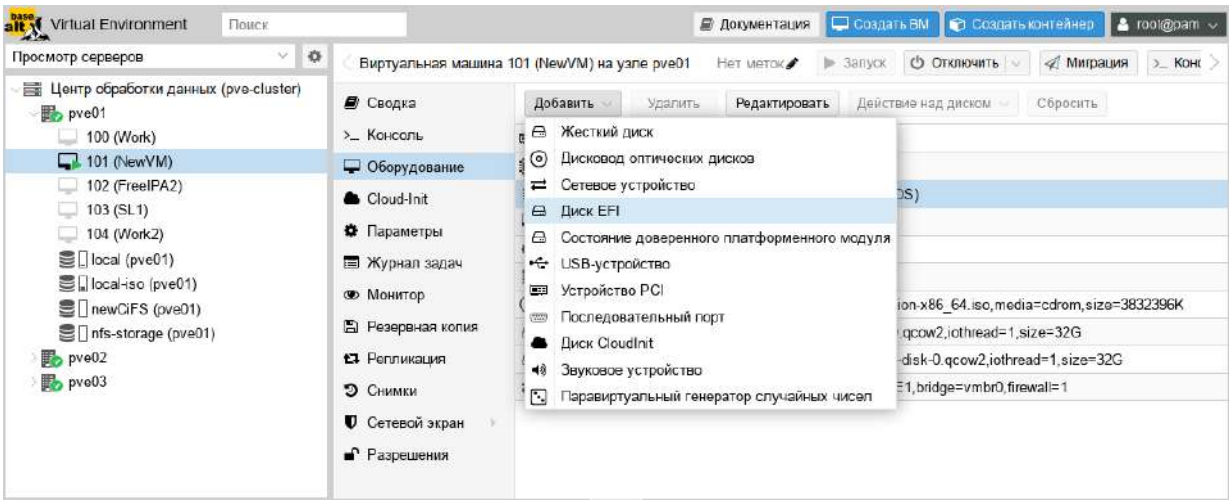


Рис. 205

Команда создания диска EFI:

```
# qm set <vm_id> -efidisk0 <storage>:1,format=<format>,efitype=4m,pre-enrolled-keys=1
```

где:

- <storage> – хранилище, в котором будет размещён диск;
- <format> – формат, поддерживаемый хранилищем;
- efitype – указывает, какую версию микропрограммы OVMF следует использовать. Для новых ВМ необходимо указывать 4м (это значение по умолчанию в графическом интерфейсе);
- pre-enroll-keys – указывает, должен ли efidisk поставляться с предварительно загруженными ключами безопасной загрузки для конкретного дистрибутива и Microsoft Standard Secure Boot. Включает безопасную загрузку по умолчанию.

4.7.6.6 Доверенный платформенный модуль (TPM)

TPM (англ. Trusted Platform Module) – спецификация, описывающая криптопроцессор, в котором хранятся криптографические ключи для защиты информации, а также обобщённое наименование реализаций указанной спецификации, например, в виде «чипа TPM» или «устройства безопасности TPM» (Dell).

Доверенный платформенный модуль можно добавить на этапе создания ВМ (вкладка «Система») или для уже созданной ВМ.

Добавление TPM в веб-интерфейсе («Добавить» → «Состояние доверенного платформенного модуля») показано на Рис. 206.

Команда добавления TRM:

```
# qm set <vm_id> -tpmstate0 <storage>:1,version=<version>
```

где:

- <storage> – хранилище, в которое будет помещён модуль;
- <version> – версия (v1.2 или v2.0).

PVE. Добавление TPM в веб-интерфейсе

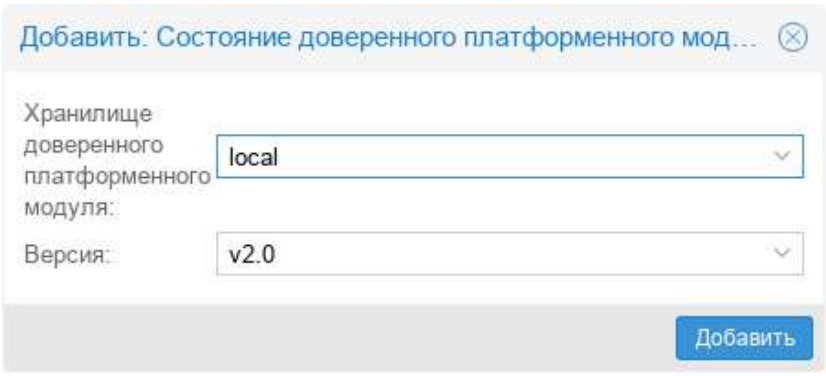


Рис. 206

4.7.6.7 Проброс PCI(e)

Проброс PCI(e) – это механизм, позволяющий ВМ управлять устройством PCI(e) хоста.

Примечание. Если устройство передано на ВМ, его нельзя будет использовать на хосте или в любой другой ВМ.

Поскольку пробросPCI(e) – это функция, требующая аппаратной поддержки, необходимо убедиться, что ваше оборудование (ЦП и материнская плата) поддерживает IOMMU (I/O Memory Management Unit).

Если оборудование поддерживает проброс, необходимо выполнить следующую настройку:

1) включить поддержку IOMMU в BIOS/UEFI;

2) для процессоров Intel – передать ядру параметр `intel_iommu=on` (для процессоров AMD он должен быть включен автоматически);

3) убедиться, что следующие модули загружены (этого можно добиться, добавив их в файл /etc/modules):

```
vfio
vfio_iommu_type1
vfio_pci
vfio_virqfd
```

4) перезагрузить систему, чтобы изменения вступили в силу, и убедиться, что проброс действительно включен:

```
# dmesg | grep -e DMAR -e IOMMU -e AMD-Vi
```

Наиболее часто используемый вариант проброса PCI(e) – это проброс всей карты PCI(e), например, GPU или сетевой карты. В этом случае хост не должен использовать карту. Этого можно добиться двумя методами:

- передать идентификаторы устройств в параметры модулей vfio-pci, добавив, например, в файл /etc/modprobe.d/vfio.conf строку:

```
options vfio-pci ids=1234:5678,4321:8765
```

где 1234:5678 и 4321:8765 – идентификаторы поставщика и устройства.

Посмотреть идентификаторы поставщика и устройства можно в выводе команды:

```
# lspci -nn
```

- занести на хосте драйвер в черный список, для этого добавить в файл /etc/modprobe.d/blacklist.conf:

```
blacklist DRIVERNAME
```

Для применения изменений необходимо перезагрузить систему.

Добавления устройства PCI ВМ:

- в веб-интерфейсе («Добавить» → «Устройство PCI» в разделе «Оборудование») (Рис. 207). В веб-интерфейсе можно назначить ВМ до 16 устройств PCI(e).
- в командной строке:

```
# qm set VMID -hostpci0 00:02.0
```

Если устройство имеет несколько функций (например, «00:02.0» и «00:02.1»), можно передать их с помощью сокращенного синтаксиса «00:02». Это эквивалентно установке отметки «Все функции» в веб-интерфейсе.

Идентификаторы поставщика и устройства PCI могут быть переопределены для сквозной записи конфигурации, и они необязательно должны соответствовать фактическим идентификаторам физического устройства. Доступные параметры: vendor-id, device-id, sub-vendor-id и sub-device-id. Можно установить любой или все из них, чтобы переопределить идентификаторы устройства по умолчанию:

```
# qm set VMID -hostpci0 02:00,device-id=0x10f6,sub-vendor-id=0x0000
```

PVE. Добавление устройства PCI

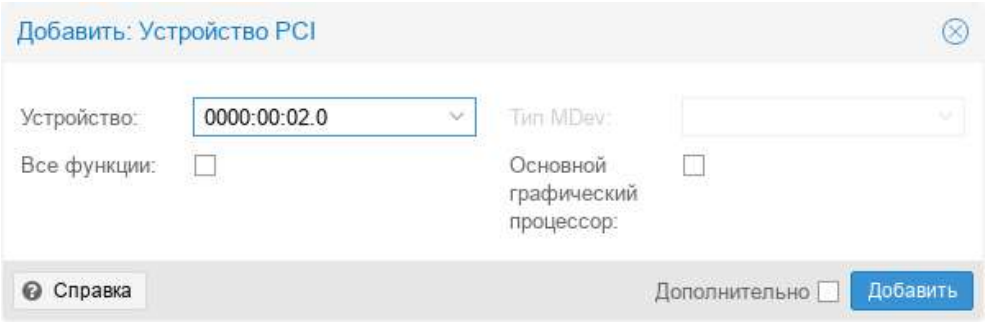


Рис. 207

4.7.7 Гостевой агент QEMU

Гостевой агент QEMU (QEMU Guest Agent) – это служба, которая работает внутри ВМ, обеспечивая канал связи между узлом и гостевой системой. Гостевой агент QEMU обеспечивает выполнение команд на ВМ и обмен информацией между ВМ и узлом кластера. Например, IP-адреса на панели сводки ВМ извлекаются с помощью гостевого агента.

Для правильной работы гостевого агента QEMU необходимо выполнить следующие действия:

- установить агент в гостевой системе и убедиться, что он запущен;
- включить связь гостевого агента с PVE.

Установка гостевой агент QEMU в ВМ с ОС «Альт»:

1) установить пакет qemu-guest-agent:

```
# apt-get install qemu-guest-agent
```

2) добавить агент в автозапуск и запустить его:

```
# systemctl enable --now qemu-guest-agent
```

Установка гостевого агента QEMU в ВМ с ОС «Windows»:

1) скачать и установить на ВМ драйверы Virtio;
2) скачать и установить на ВМ ПО QEMU Guest Agent;
3) убедиться, что в списке запущенных служб есть QEMU Guest Agent.

Связь PVE с гостевым агентом QEMU можно включить на вкладке «Параметры» требуемой ВМ в веб-интерфейсе (Рис. 208) или в командной строке:

```
# qm set <vmid> --agent 1
```

Для вступления изменений в силу необходим перезапуск ВМ.

Если включена опция «Выполнять комнду «trim»…», PVE выдаст команду `trim` гостевой системе после следующих операций, которые могут записать нули в хранилище:

- перемещение диска в другое хранилище;
- живая миграция ВМ на другой узел с локальным хранилищем.

PVE. Включить связь с гостевым агентом QEMU

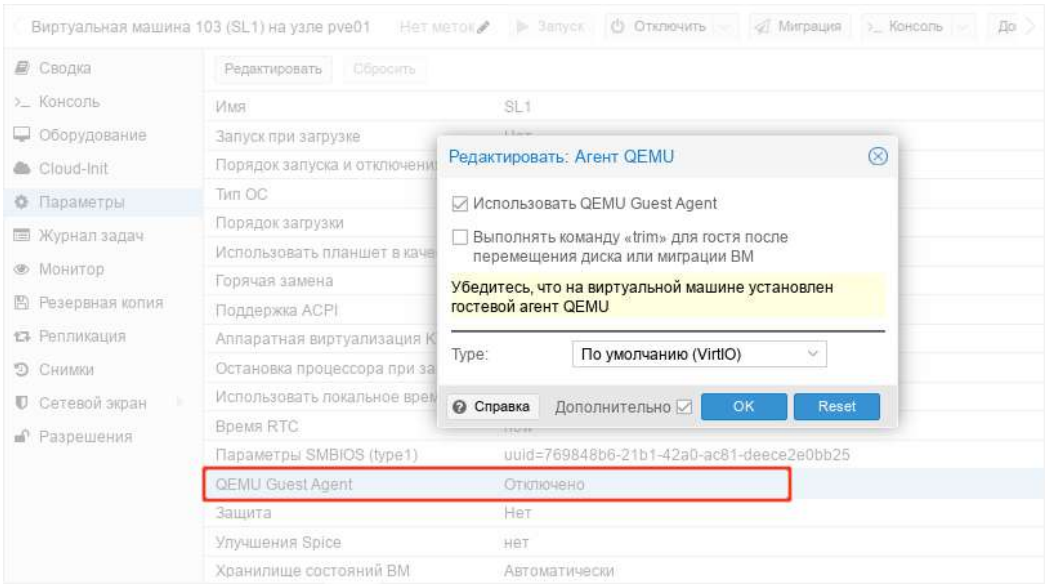


Рис. 208

В хранилище с тонким выделением ресурсов это может помочь освободить неиспользуемое пространство.

Связь с гостевым агентом QEMU происходит через UNIX-сокет, расположенный в /var/run/qemu-server/<my_vmid>.qga. Проверить связь с агентом можно помощью команды:

```
# qm agent <vmid> ping
```

Если гостевой агент правильно настроен и запущен в ВМ, вывод команды будет пустой.

4.7.8 Файлы конфигурации ВМ

Файлы конфигурации ВМ хранятся в файловой системе кластера PVE (`/etc/pve/qemu-server/<VMID>.conf`). Как и другие файлы, находящиеся в `/etc/pve/`, они автоматически реплицируются на все другие узлы кластера.

Примечание. VMID < 100 зарезервированы для внутренних целей. VMID должны быть уникальными для всего кластера.

Пример файла конфигурации:

```
boot: order=scsi0;scsi7;net0
cores: 1
memory: 2048
name: newVM
net0: virtio=3E:E9:24:FF:85:D9,bridge=vmbr0,firewall=1
numa: 0
ostype: l26
sata2: local-iso:iso/slinux-10.1-x86_64.iso,media=cdrom,size=4586146K
scsi0: local:100/vm-100-disk-0.qcow2,size=42G
scsihw: virtio-scsi-pci
smbios1: uuid=ee9db068-5427-4934-bf7a-5895c377b5af
```

```
sockets: 1
vmgenid: dfec8e3b-d391-40cb-8983-b4938461b79a
```

Файлы конфигурации ВМ используют простой формат: разделенные двоеточиями пары ключ/значение (пустые строки игнорируются, строки, начинающиеся с символа #, рассматриваются как комментарии и также игнорируются):

```
OPTION: value
```

Для применения изменений, которые напрямую вносились в файл конфигурации, необходимо перезапустить ВМ. По этой причине рекомендуется использовать команду `qm` для генерации и изменения этих файлов, либо выполнять такие действия в веб-интерфейсе.

При создании снимка ВМ, конфигурация ВМ во время снимка, сохраняется в этом же файле конфигурации в отдельном разделе. Например, после создания снимка «snapshot» файл конфигурации будет выглядеть следующим образом:

```
boot: order=scsi0;scsi7;net0
…
parent: snapshot
…
vmgenid: f631f900-b5b3-4802-a300-7bfad377cd3a

[snapshot]
boot: order=scsi0;sata2;net0
cores: 1
memory: 2048
meta: creation-qemu=7.2.0,ctime=1692701248
name: NewVM
net0: virtio=76:AD:58:9E:C4:E1,bridge=vmbr0,firewall=1
numa: 0
ostype: l26
runningcpu: kvm64,enforce,+kvm_pv_eoi,+kvm_pv_unhalt,+lahf_lm,+sep
runningmachine: pc-i440fx-7.2+pve0
sata2: none,media=cdrom
scsi0: local:101/vm-101-disk-0.qcow2,iothread=1,size=32G
scsi1: nfs-storage:101/vm-101-disk-0.qcow2,iothread=1,size=32G
scsihw: virtio-scsi-single
smbios1: uuid=547b268e-9a3f-4c17-8dff-b0dc20c39e58
snaptime: 1692774060
sockets: 1
spice_enhancements: foldersharing=1
vga: qxl
vmgenid: f631f900-b5b3-4802-a300-7bfad377cd3a
vmstate: nfs-storage:101/vm-101-state-first.raw
```

Свойство `parent` при этом используется для хранения родительских/дочерних отношений между снимками, а `snaptime` – это отметка времени создания снимка (эпоха Unix).

4.8 Создание и настройка контейнера LXC

4.8.1 Создание контейнера в графическом интерфейсе

Перед созданием контейнера можно загрузить шаблоны LXC в хранилище.

Для создания контейнера необходимо нажать кнопку «Создать контейнер», расположенную в правом верхнем углу веб-интерфейса PVE (Рис. 209). Будет запущен диалог «Создать: Контейнер LXC» (Рис. 210), который предоставляет графический интерфейс для настройки контейнера.

Кнопка «Создать контейнер»

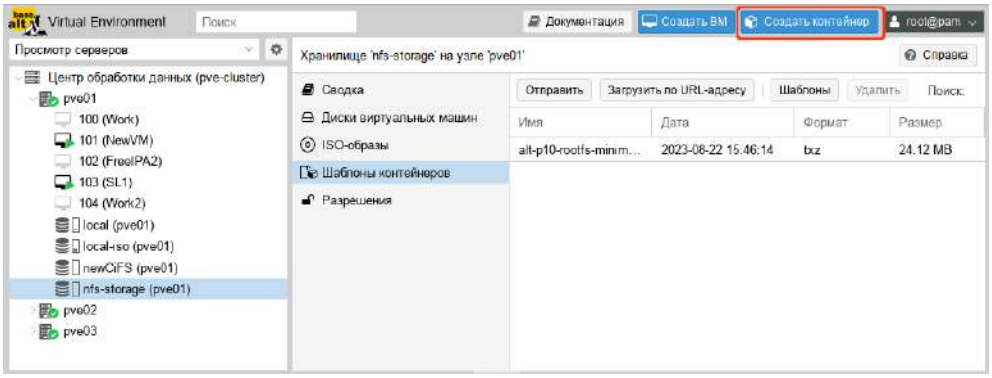


Рис. 209

Вкладка «Общее» диалога создания контейнера

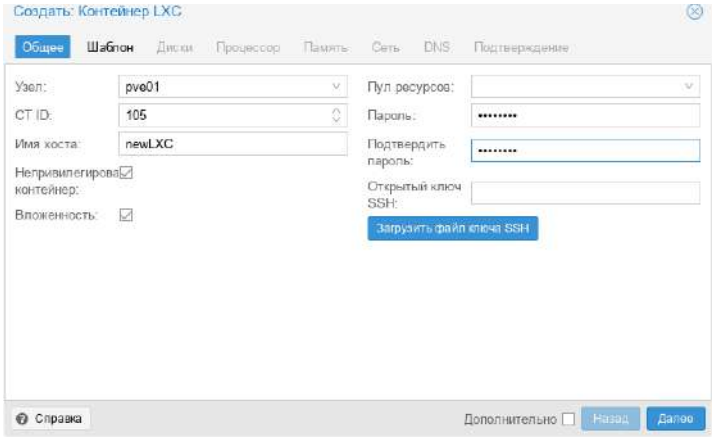


Рис. 210

На первой вкладке «Общее» необходимо указать (Рис. 210):

- «Узел» – узел назначения для данного контейнера;
- «CT ID» – идентификатор контейнера в численном выражении;
- «Имя хоста» – алфавитно-цифровая строка названия контейнера;
- «Непривилегированный контейнер» – определяет, как будут запускаться процессы контейнера (если процессам внутри контейнера не нужны полномочия администратора, то необходимо снять отметку с этого пункта);

- «Вложенность» – определяет возможность запуска контейнера в контейнере;
- «Пул ресурсов» – логическая группа контейнеров. Чтобы иметь возможность выбора, пул должен быть предварительно создан;
- «Пароль» – пароль для данного контейнера;
- «Открытый SSH ключ» – SSH ключ.

На вкладке «Шаблон» следует выбрать (Рис. 211):

- «Хранилище» – хранилище в котором хранятся шаблоны LXC;
- «Шаблон» – шаблон контейнера.

Вкладка «Шаблон» диалога создания контейнера

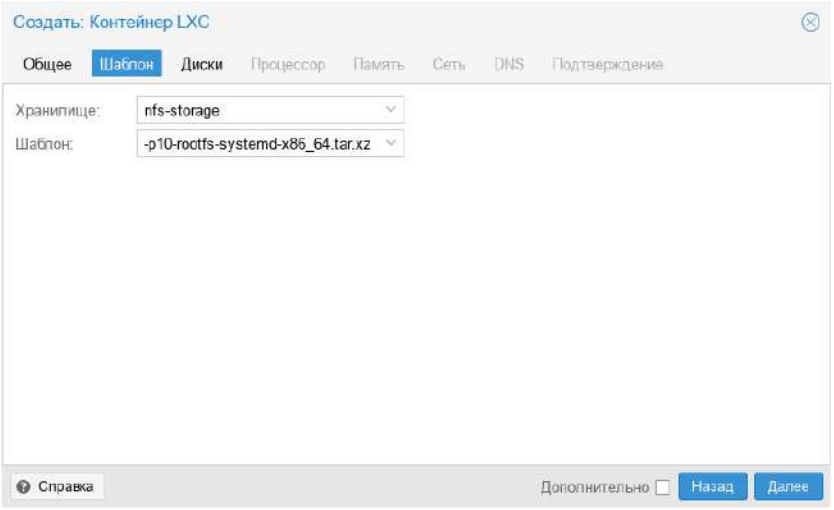


Рис. 211

На вкладке «Диски» определяется хранилище, где будут храниться диски контейнера (Рис. 212). Здесь также можно определить размер виртуальных дисков (не следует выбирать размер диска менее 4 ГБ).

Вкладка «Диски» диалога создания контейнера

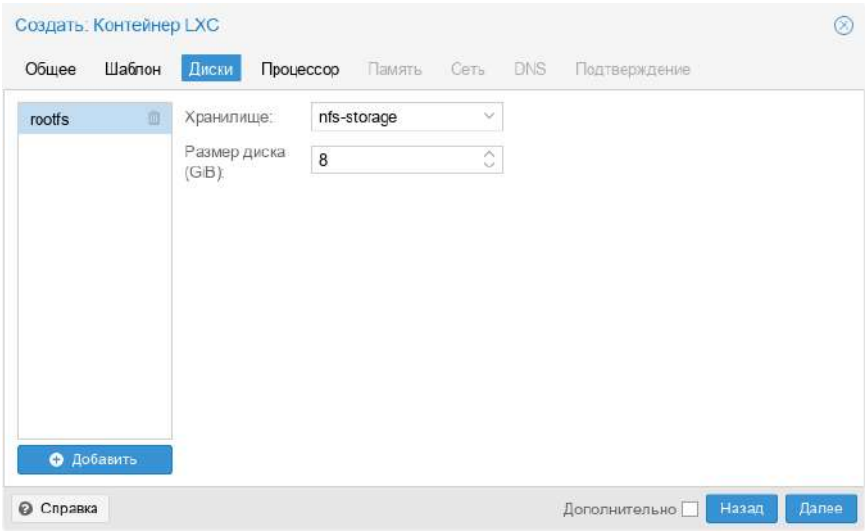


Рис. 212

На вкладке «Процессор» определяется количество ядер процессора, которые будут выделены контейнеру (Рис. 213).

Вкладка «Процессор» диалога создания контейнера

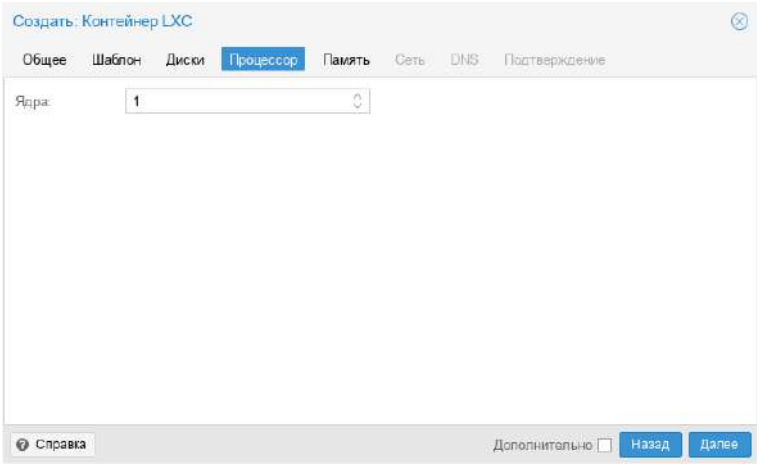


Рис. 213

На вкладке «Память» настраиваются (Рис. 214):

- «Память» (MiB) – выделяемая память в мегабайтах;
- «Подкачка» (MiB) – выделяемое пространство подкачки в мегабайтах.

Вкладка «Память» диалога создания контейнера

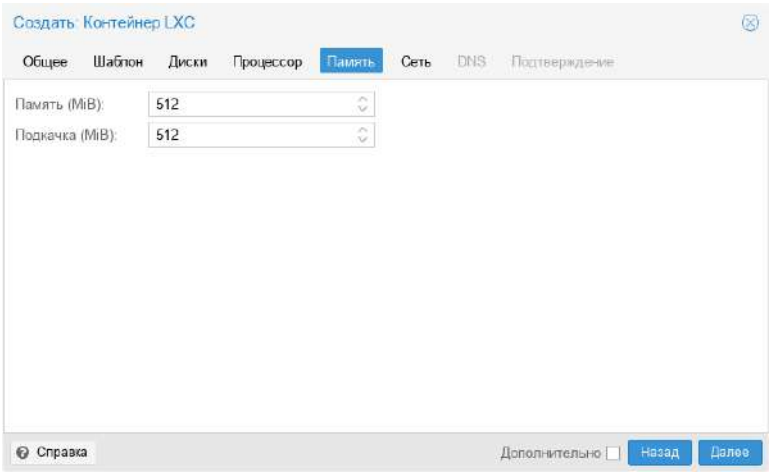


Рис. 214

Вкладка «Сеть» включает следующие настройки (Рис. 215):

- «Имя» – определяет, как будет именоваться виртуальный сетевой интерфейс внутри контейнера (по умолчанию eth0);
- «MAC-адрес» – можно задать определенный MAC-адрес, необходимый для приложения в данном контейнере (по умолчанию все MAC-адреса для виртуальных сетевых интерфейсов назначаются автоматически);
- «Сетевой мост» – выбор виртуального моста, к которому будет подключаться данный интерфейс (по умолчанию vmbr0);

- «Тег виртуальной ЛС» – применяется для установки идентификатора VLAN для данного виртуального интерфейса;
- «Сетевой экран» – поддержка межсетевого экрана (если пункт отмечен, применяются правила хоста);
- «IPv4/IPv6» – можно настроить и IPv4, и IPv6 для виртуального сетевого интерфейса. IP-адреса можно устанавливать вручную или разрешить получать от DHCP-сервера для автоматического назначения IP. IP-адрес должен вводиться в нотации CIDR (например, 192.168.0.30/24).

Вкладка «Сеть» диалога создания контейнера

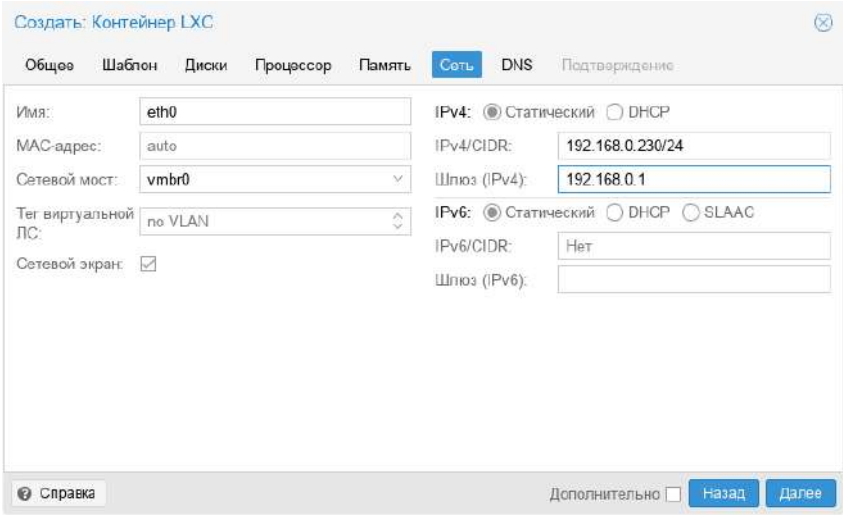


Рис. 215

Вкладка «DNS» содержит настройки (Рис. 216):

- «Домен DNS» – имя домена (по умолчанию используются параметры хост системы);
- «Серверы DNS» – IP-адреса серверов DNS (по умолчанию используются параметры хост системы).

Вкладка «DNS» диалога создания контейнера

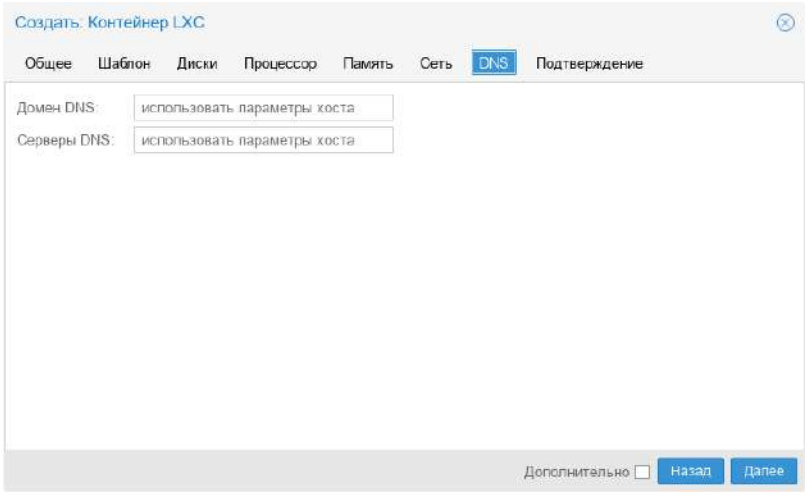


Рис. 216

Во вкладке «Подтверждение» отображаются все введенные или выбранные значения для данного контейнера (Рис. 217). Для создания контейнера необходимо нажать кнопку «Готово». Если необходимо внести изменения в параметры контейнера, можно перейти по вкладкам назад.

Вкладка «Подтверждение» диалога создания контейнера

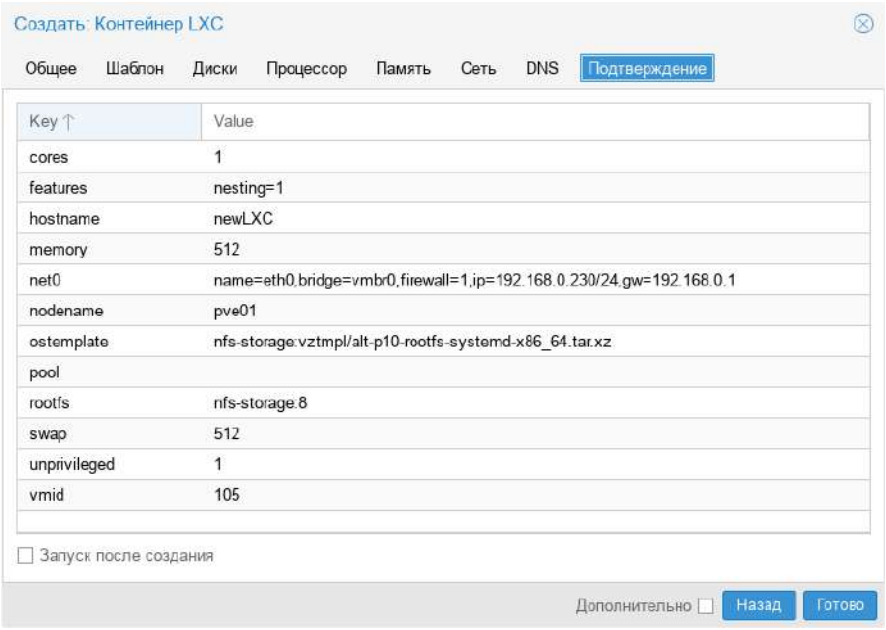


Рис. 217

Если отметить пункт «Запуск после создания» контейнер будет запущен сразу после создания.

После нажатия кнопки «Готово» во вкладке «Подтверждение», диалог настройки закрывается и в браузере открывается новое окно, которое предлагает возможность наблюдать за построением PVE контейнера LXC из шаблона (Рис. 218).

Создание контейнера

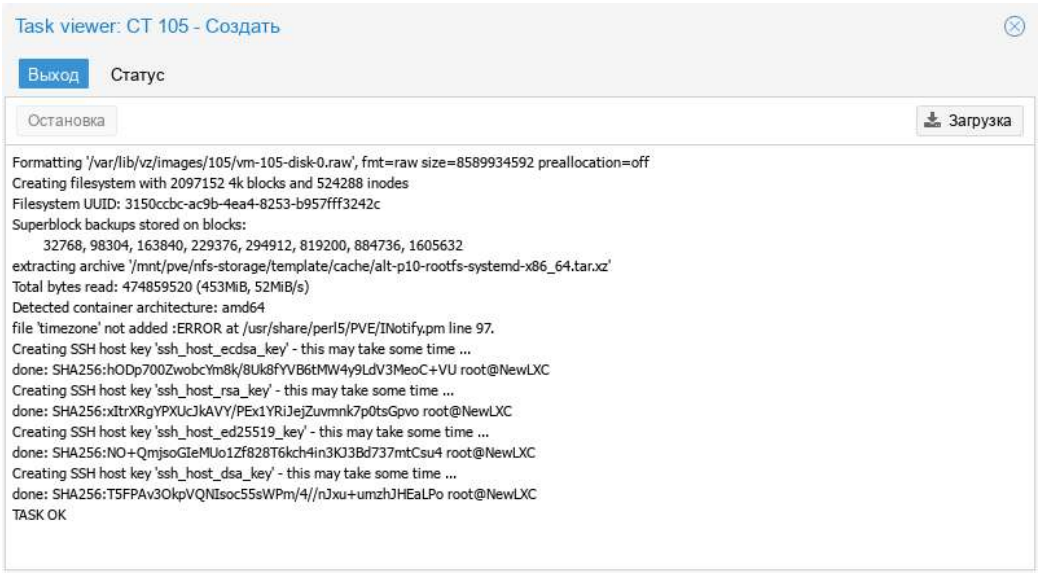


Рис. 218

4.8.2 Создание контейнера из шаблона в командной строке

Контейнер может быть создан из шаблона в командной строке хоста.

Следующий bash-сценарий иллюстрирует применение команды `pct` для создания контейнера:

```
#!/bin/bash
#### Set Variables ####
hostname="pve01"
vmid="104"
template_path="/var/lib/vz/template/cache"
storage="local"
description="alt-p10"
template="alt-p10-rootfs-systemd-x86_64.tar.xz"
ip="192.168.0.93/24"
nameserver="8.8.8.8"
ram="1024"
rootpw="password"
rootfs="4"
gateway="192.168.0.1"
bridge="vmbr0"
if="eth0"
#### Execute pct create using variable substitution ####
pct create $vmid \
  $template_path/$template \
  -description $description \
  -rootfs $rootfs \
  -hostname $hostname \
  -memory $ram \
  -nameserver $nameserver \
  -storage $storage \
  -password $rootpw \
  -net0 name=$if,ip=$ip,gw=$gateway,bridge=$bridge
```

4.8.3 Изменение настроек контейнера

Изменения в настройки контейнера можно вносить и после его создания. При этом изменения сразу же вступают в действие, без необходимости перезагрузки контейнера.

Есть три способа, которыми можно регулировать выделяемые контейнеру ресурсы:

- веб-интерфейс PVE;
- командная строка;
- изменение файла конфигурации.

4.8.3.1 Изменение настроек в веб-интерфейсе

В большинстве случаев изменение настроек контейнера и добавление виртуальных устройств может быть выполнено в веб-интерфейсе.

Для изменения настроек контейнера можно использовать вкладки (Рис. 219):

- «Ресурсы» (оперативная память, подкачка, количество ядер ЦПУ, размер диска);
- «Сеть»;
- «DNS»;
- «Параметры».

Изменений настроек контейнера в веб-интерфейсе PVE

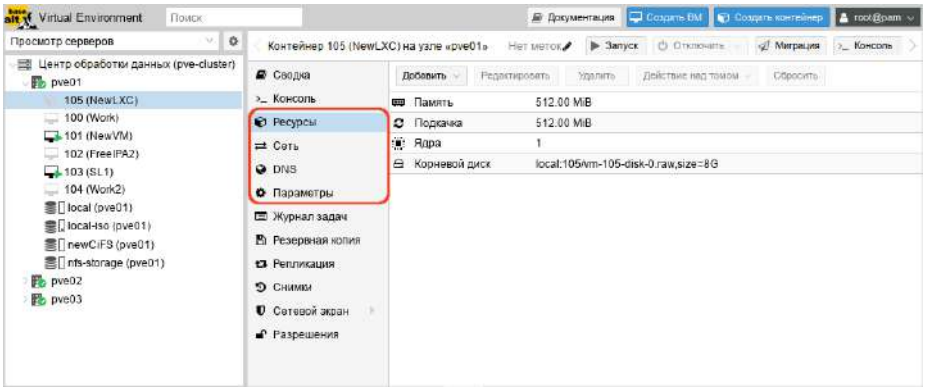


Рис. 219

Для редактирования ресурсов следует выполнить следующие действия:

1) в режиме просмотра по серверам выбрать контейнер;
2) перейти на вкладку «Ресурсы»;
3) выбрать элемент для изменения: «Память», «Подкачка» или «Ядра», и нажать кнопку «Редактировать»;
4) в открывшемся диалоговом окне ввести нужные значения и нажать кнопку «OK».

Если необходимо изменить размер диска контейнера, например, увеличить до 18 ГБ вместо предварительно созданного 8 ГБ, нужно выбрать элемент «Корневой диск», нажать кнопку «Действие над томом» → «Изменить размер», в открывшемся диалоговом окне ввести значение увеличения размера диска (Рис. 220) и нажать кнопку «Изменить размер диска».

Изменение размера диска

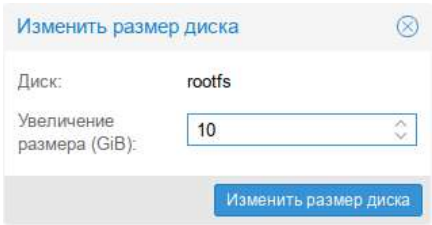


Рис. 220

Для перемещения образа диска в другое хранилище нужно выбрать элемент «Корневой диск», нажать кнопку «Действие над томом» → «Переместить хранилище», в открывшемся окне (Рис. 221) в выпадающем меню «Целевое хранилище» выбрать хранилище-получатель, отметить, если это необходимо, пункт «Удалить источник» для удаления образа диска из исходного хранилища и нажать кнопку «Переместить том».

Диалоговое окно перемещения тома

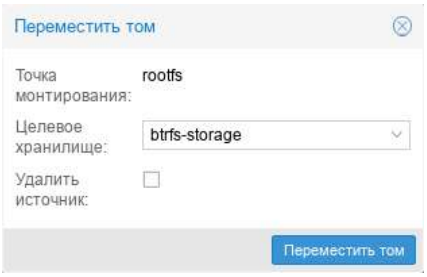


Рис. 221

Для изменения сетевых настроек контейнера необходимо:

1) в режиме просмотра по серверам выбрать контейнер;

2) перейти на вкладку «Сеть». На экране отобразятся все настроенные для контейнера виртуальные сетевые интерфейсы (Рис. 222);

3) выбрать интерфейс и нажать кнопку «Редактировать» (Рис. 223);

4) после внесения изменений нажать кнопку «OK».

Виртуальные сетевые интерфейсы контейнера

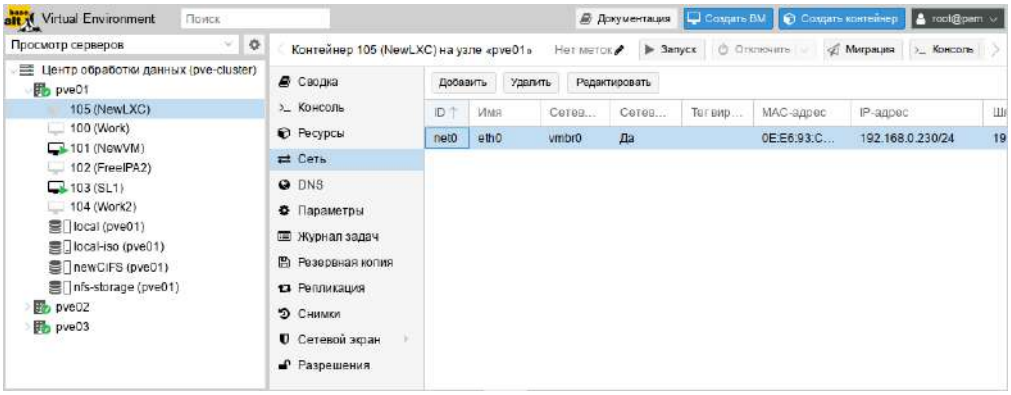


Рис. 222

На вкладке «Параметры» можно отредактировать разные настройки контейнера (Рис. 224), например, «Режим консоли»:

- «tty» – открывать соединение с одним из доступных tty-устройств (по умолчанию);
- «shell» – вызывать оболочку внутри контейнера (без входа в систему);
- «/dev/console» – подключаться к /dev/console.

Изменение сетевых настроек контейнера

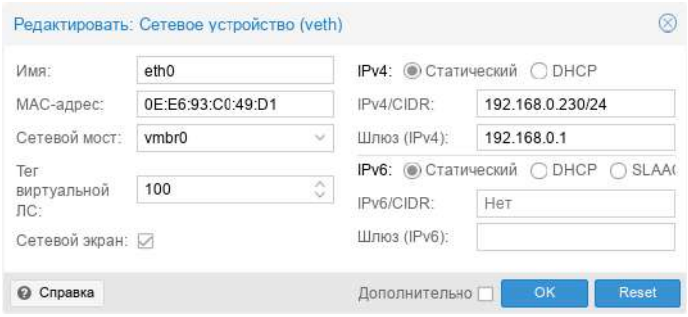


Рис. 223

Изменение настроек контейнера. Вкладка «Параметры»

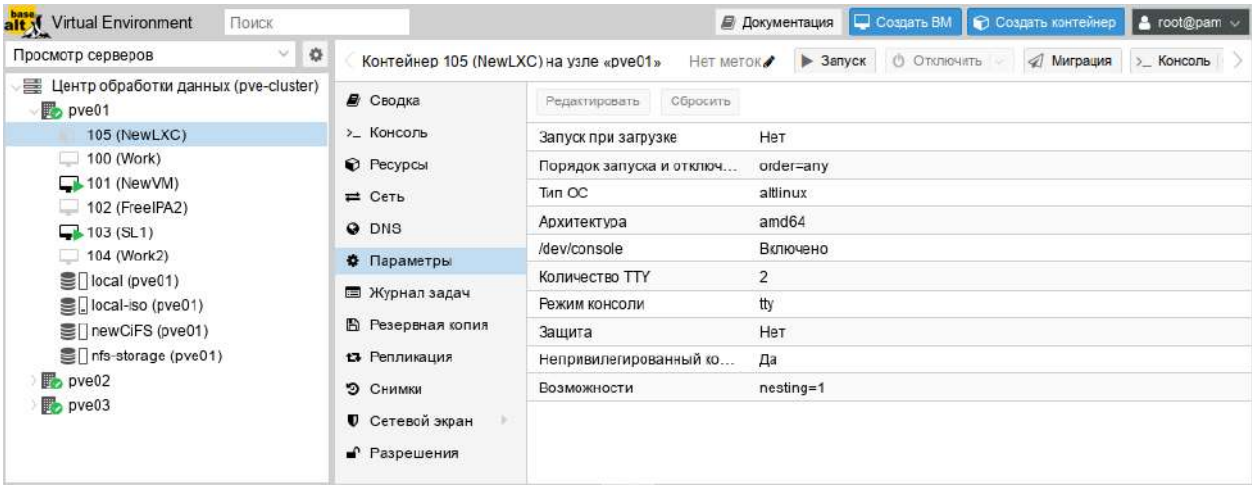


Рис. 224

Примечание. В случаях, когда изменение не может быть выполнено в горячем режиме, оно будет зарегистрировано как ожидающее изменение (выделяется цветом, см. Рис. 225). Такие изменения будут применены только после перезапуска контейнера.

Изменения, которые будут применены после перезапуска контейнера

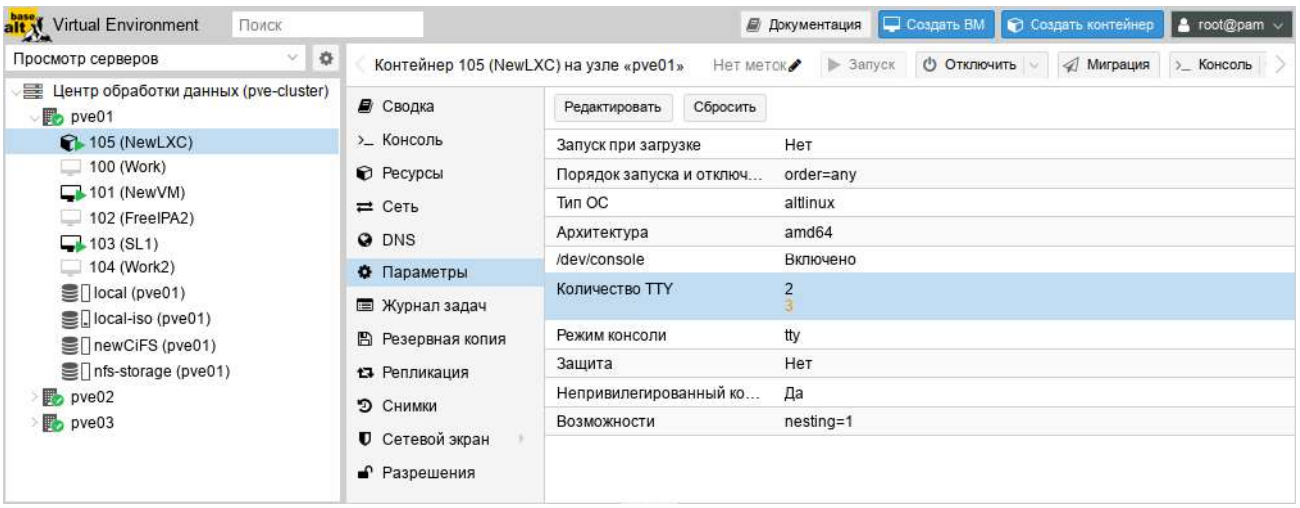


Рис. 225

4.8.3.2 Настройка ресурсов в командной строке

Если веб-интерфейс PVE недоступен, можно управлять контейнером в командной строке (либо через сеанс SSH, либо из консоли noVNC, или зарегистрировавшись на физическом хосте).

`pct` – утилита управления контейнерами LXC в PVE. Чтобы просмотреть доступные для контейнеров команды PVE, можно выполнить следующую команду:

```
# pct help
```

Формат использования команды для изменения ресурсов контейнера:

```
# pct set <ct_id> [options]
```

Например, изменить IP-адрес контейнера #101:

```
# pct set 101 –net0 name=eth0,bridge=vmbr0,ip=192.168.0.17/24,gw=192.168.0.1
```

Изменить количество выделенной контейнеру памяти:

```
# pct set <ct_id> -memory <int_value>
```

Команда изменения размера диска контейнера:

```
# pct set <ct_id> -rootfs <volume>,size=<int_value for GB>
```

Например, изменить размер диска контейнера #101 до 10 ГБ:

```
# pct set 101 -rootfs local:101/vm-101-disk-0.raw,size=10G
```

Показать конфигурацию контейнера:

```
pct config <ct_id>
```

Разблокировка заблокированного контейнера:

```
# pct unlock <ct_id>
```

Список контейнеров LXC данного узла:

```
# pct list
VMID       Status     Lock         Name
101        running                 newLXC
102        stopped                 pve01
103        stopped                 LXC2
```

Запуск и остановка контейнера LXC из командной строки:

```
# pct start <ct_id>
# pct stop <ct_id>
```

4.8.3.3 Настройка ресурсов прямым изменением

В PVE файлы конфигурации контейнеров находятся в каталоге `/etc/pve/lxc`, а файлы конфигураций ВМ – в `/etc/pve/qemu-server/`.

У контейнеров LXC есть большое число параметров, которые не могут быть изменены в веб-интерфейсе или с помощью утилиты `pct`. Эти параметры могут быть настроены только путем изменений в файл конфигурации с последующим перезапуском контейнера.

Пример файла конфигурации контейнера `/etc/pve/lxc/102.conf`:

```
arch: amd64
```

```
cmode: shell
console: 0
cores: 1
features: nesting=1
hostname: newLXC
memory: 512
net0:
name=eth0,bridge=vmbr0,firewall=1,gw=192.168.0.1,hwaddr=C6:B0:3E:85:03:C9,ip=192.168.
0.30/24,type=veth
ostype: altlinux
rootfs: local:101/vm-101-disk-0.raw,size=8G
swap: 512
tty: 3
unprivileged: 1
```

4.8.4 Запуск и остановка контейнеров

4.8.4.1 Изменение состояния контейнера в веб-интерфейсе

Для запуска контейнера следует выбрать его в левой панели; его иконка должна быть серого цвета, обозначая, что контейнер не запущен (Рис. 226).

Запустить контейнер можно, выбрав в контекстном меню контейнера пункт «Запуск» (Рис. 226), либо нажав кнопку «Запуск» (Рис. 227).

Запущенный контейнер будет обозначен зеленой стрелкой на значке контейнера.

Контекстное меню контейнера

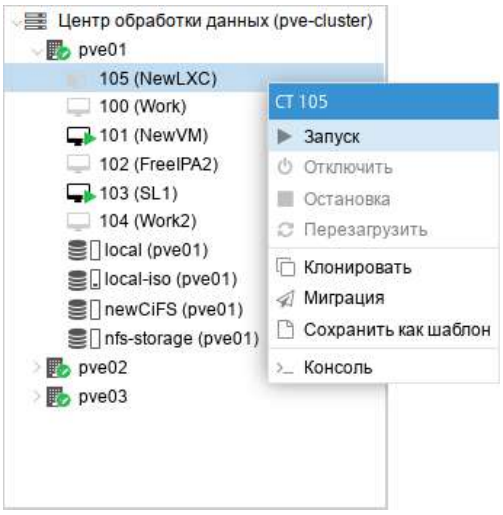


Рис. 226

Кнопки управления состоянием контейнера

Рис. 227

Для запущенного контейнера доступны следующие действия (Рис. 228):

- «Отключить» – остановка контейнера;
- «Остановка» – остановка контейнера, путем прерывания его работы;
- «Перезагрузить» – перезапуск контейнера.

Контекстное меню запущенного контейнера

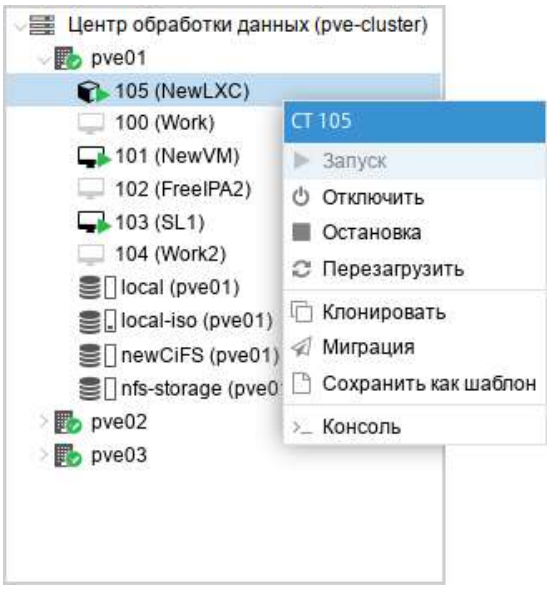


Рис. 228

4.8.4.2 Изменение состояний контейнера в командной строке

Состоянием контейнера можно управлять из командной строки PVE (либо через сеанс SSH, либо из консоли noVNC, или зарегистрировавшись на физическом хосте).

Для запуска контейнера с VM ID 102 необходимо ввести команду:

```
# pct start 102
```

Этот же контейнер может быть остановлен при помощи команды:

```
# pct stop 102
```

4.8.5 Доступ к LXC контейнеру

Способы доступа к LXC контейнеру:

- консоль: noVNC, SPICE или xterm.js;
- SSH;
- интерфейс командной строки PVE.

Можно получить доступ к контейнеру из веб-интерфейса при помощи консоли noVNC. Это почти визуализированный удаленный доступ к экземпляру.

Для доступа к запущенному контейнеру в консоли следует выбрать в веб-интерфейсе нужный контейнер, а затем нажать кнопку «Консоль» («Console») и в выпадающем меню выбрать нужную консоль (Рис. 229).

Кнопка «Консоль»

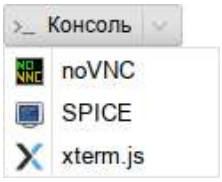


Рис. 229

Консоль также можно запустить, выбрав вкладку «Консоль» для контейнера (Рис. 230).

Консоль контейнера

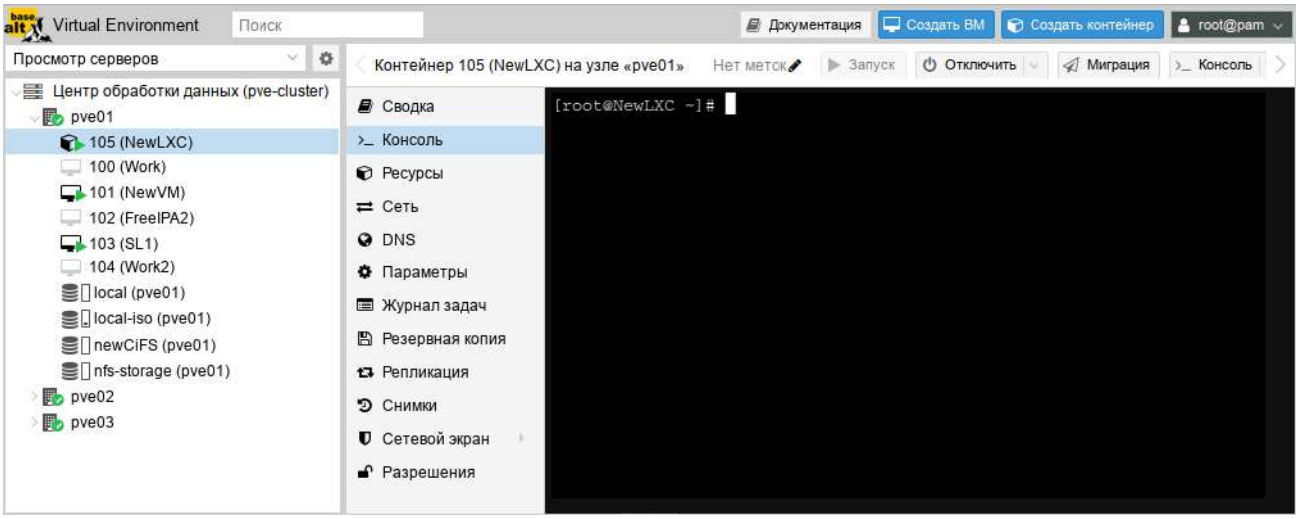


Рис. 230

Одной из функций LXC контейнера является возможность прямого доступа к оболочке контейнера через командную строку его узла хоста. Команда для доступа к оболочке контейнера LXC:

```
# pct enter <ct_id>
```

Данная команда предоставляет прямой доступ на ввод команд внутри указанного контейнера:

```
[root@pve01 ~]# pct enter 105
[root@newLXC ~]#
```

Таким образом был получен доступ к контейнеру LXC с именем newLXC на узле pve01. При этом для входа в контейнер не был запрошен пароль. Так как контейнер работает под пользователем root, можно выполнять внутри этого контейнера любые задачи. Завершив их, можно просто набрать `exit`.

Примечание. При возникновении ошибки:

```
Insecure $ENV{ENV} while running with...
```

необходимо в файле `/root/.bashrc` закомментировать строку: `"ENV=$HOME/.bashrc"` и выполнить команду:

```
# unset ENV
```

Команды можно выполнять внутри контейнера без реального входа в такой контейнер:

```
# pct exec <ct_id> -- <command>
```

Например, создать каталог внутри контейнера и проверить, что этот каталог был создан:

```
# pct exec 105 mkdir /home/demouser
# pct exec 105 ls /home
demouser
```

Для выполнения внутри контейнера команды с параметрами необходимо изменить команду `pct`, добавив `--` после идентификатора контейнера:

```
# pct exec 101 -- df -H /
Файловая система Размер Использовано  Дост Использовано% Смонтировано в
/dev/loop0         8,4G         516M  7,4G            7% /
```

4.9 Миграция виртуальных машин и контейнеров

В случае, когда PVE управляет не одним физическим узлом, а кластером физических узлов, должна обеспечиваться возможность миграции ВМ с одного физического узла на другой. Миграция представляет собой заморозку состояния ВМ на одном узле, перенос данных и конфигурации на другой узел и разморозку состояния ВМ на новом месте. Возможные сценарии, при которых может возникнуть необходимость миграции:

- отказ физического узла;
- необходимость перезагрузки узла после применения обновлений или обслуживания технических средств;
- перемещение ВМ с узла с низкой производительностью на высокопроизводительный узел.

Есть два механизма миграции:

- онлайн-миграция (Live Migration);
- офлайн-миграция.

Примечание. Миграция контейнеров без перезапуска в настоящее время не поддерживается. При выполнении миграции запущенного контейнера, контейнер будет выключен, перемещен, а затем снова запущен на целевом узле. Поскольку контейнеры легковесные, то это обычно приводит к простою в несколько сотен миллисекунд.

Для возможности онлайн-миграции ВМ должны выполняться следующие условия:

- у ВМ нет локальных ресурсов;
- хосты находятся в одном кластере PVE;
- между хостами имеется надежное сетевое соединение;
- на целевом хосте установлены такие же или более высокие версии пакетов PVE.

Миграция в реальном времени обеспечивает минимальное время простоя ВМ, но, в то же время занимает больше времени. При миграции в реальном времени (без выключения питания)

процесс должен скопировать все содержимое оперативной памяти ВМ на новый узел. Чем больше объем выделенной ВМ памяти, тем дольше будет происходить ее перенос.

Если образ виртуального диска ВМ хранится в локальном хранилище узла PVE миграция в реальном времени невозможна. В этом случае ВМ должна быть перед миграцией выключена. В процессе миграции ВМ, хранящейся локально, PVE скопирует виртуальный диск на узел получателя с применением `rsync`.

Запустить процесс миграции можно как в графическом интерфейсе PVE, так в интерфейсе командной строки.

4.9.1 Миграция с применением графического интерфейса

Для миграции ВМ или контейнера необходимо выполнить следующие шаги:

1) выбрать ВМ или контейнер для миграции и нажать кнопку «Миграция» (Рис. 231);

2) в открывшемся диалоговом окне (Рис. 232) выбрать узел назначения, на который будет осуществляться миграция, и нажать кнопку «Миграция».

Выбор ВМ или контейнера для миграции

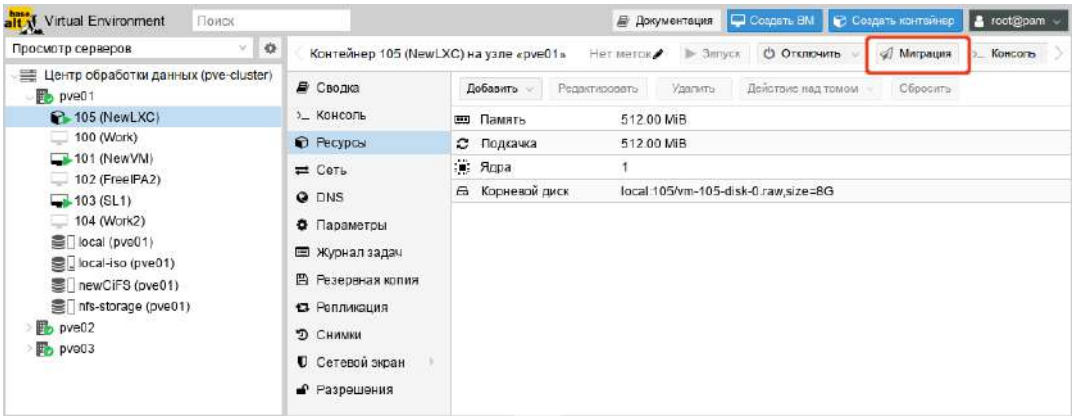


Рис. 231

Примечание. Режим миграции будет выбран автоматически (Рис. 232, Рис. 233, Рис. 234) в зависимости от состояния ВМ/контейнера (запущен/остановлен).

Миграция контейнера с перезапуском

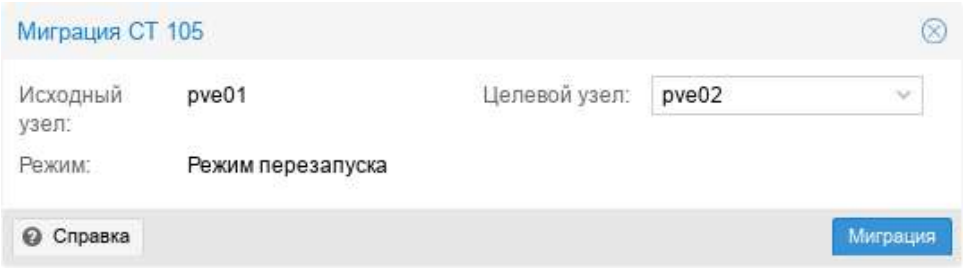


Рис. 232

Миграция ВМ Онлайн

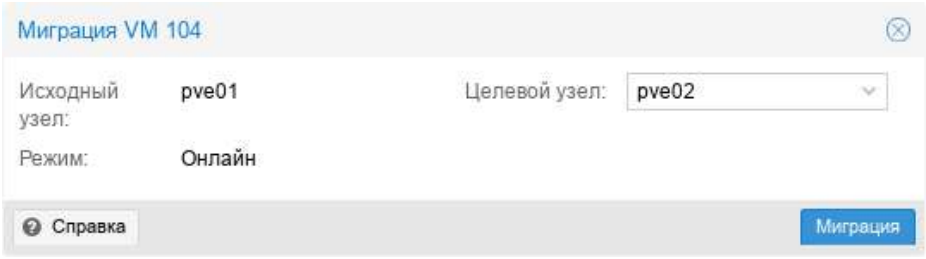


Рис. 233

Миграция ВМ Офлайн

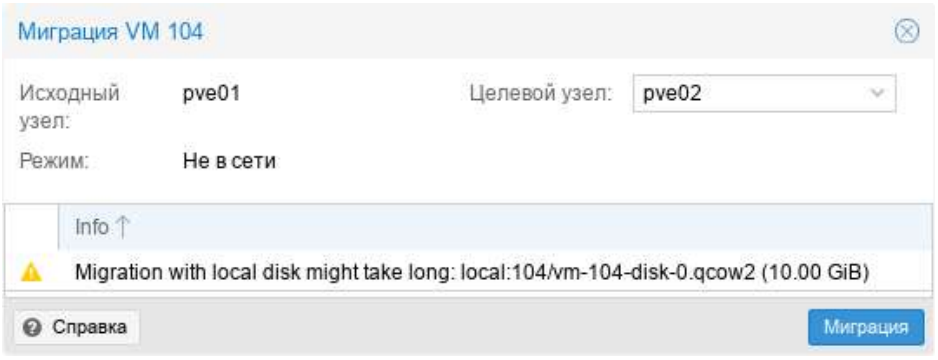


Рис. 234

4.9.2 Миграция с применением командной строки

Чтобы осуществить миграцию ВМ необходимо выполнить следующую команду:

```
# qm migrate <vmid> <target> [OPTIONS]
```

Для осуществления миграции ВМ в реальном времени следует использовать параметр `--online`.

Чтобы осуществить миграцию контейнера необходимо выполнить следующую команду:

```
# pct migrate <ctid> <target> [OPTIONS]
```

Поскольку миграция контейнеров в реальном времени невозможна, можно выполнить миграцию работающего контейнера с перезапуском, добавив параметр `--restart`. Например:

```
# pct migrate 101 pve02 --restart
```

4.9.3 Миграция ВМ из внешнего гипервизора

Экспорт ВМ из внешнего гипервизора обычно заключается в переносе одного или нескольких образов дисков с файлом конфигурации, описывающим настройки ВМ (ОЗУ, количество ядер). Образы дисков могут быть в формате vmdk (VMware или VirtualBox) или qcow2 (KVM). Наиболее популярным форматом конфигурации для экспорта ВМ является стандарт OVF.

Примечание. Для ВМ Windows необходимо также установить паравиртуализированные драйверы Windows.

4.9.3.1 Миграция KVM ВМ в PVE

В данном разделе рассмотрен процесс миграции ВМ из OpenNebula в PVE.

Выключить ВМ на хосте источнике. Найти путь до образа жесткого диска, который используется в ВМ (в данной команде 14 – id образа диска ВМ):

```
$ oneimage show 14
IMAGE 14 INFORMATION
ID             : 14
NAME           : ALT Linux p9
USER           : oneadmin
GROUP          : oneadmin
LOCK           : None
DATASTORE      : default
TYPE           : OS
REGISTER TIME  : 04/30 11:00:42
PERSISTENT     : Yes
SOURCE         : /var/lib/one//datastores/1/f811a893808a9d8f5bf1c029b3c7e905
FSTYPE         : save_as
SIZE           : 12G
STATE          : used
RUNNING_VMS    : 1

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

IMAGE TEMPLATE
DEV_PREFIX="vd"
DRIVER="qcow2"
SAVED_DISK_ID="0"
SAVED_IMAGE_ID="7"
SAVED_VM_ID="46"
SAVE_AS_HOT="YES"
```

где `/var/lib/one//datastores/1/f811a893808a9d8f5bf1c029b3c7e905` – адрес образа жёсткого диска ВМ.

Скопировать данный образ на хост назначения с PVE.

Примечание. В OpenNebula любой диск ВМ можно экспортировать в новый образ (если ВМ находится в состояниях RUNNING, POWEROFF или SUSPENDED):

```
$ onevm disk-saveas <vmid> <diskid> <img_name>  [--type type --snapshot snapshot]
```

где `--type <type>` – тип нового образа (по умолчанию raw); `--snapshot <snapshot_id>` – снимок диска, который будет использован в качестве источника нового образа (по умолчанию текущее состояние диска).

Экспорт диска ВМ:

```
$ onevm disk-saveas 125 0 test.qcow2
Image ID: 44
```

Инфомация об образе диска ВМ:

```
$ oneimage show 44
MAGE 44 INFORMATION
ID             : 44
NAME           : test.qcow2
USER           : oneadmin
GROUP          : oneadmin
LOCK           : None
DATASTORE      : default
TYPE           : OS
REGISTER TIME  : 07/12 21:34:42
PERSISTENT     : No
SOURCE         : /var/lib/one//datastores/1/9d6336a88d6ab62ea1dce65d81e55881
FSTYPE         : save_as
SIZE           : 12G
STATE          : rdy
RUNNING_VMS    : 0

PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---

IMAGE TEMPLATE
DEV_PREFIX="vd"
DRIVER="qcow2"
SAVED_DISK_ID="0"
SAVED_IMAGE_ID="14"
SAVED_VM_ID="125"
SAVE_AS_HOT="YES"

VIRTUAL MACHINES
```

Информация о диске:

```
$ qemu-img info /var/lib/one//datastores/1/9d6336a88d6ab62ea1dce65d81e55881
image: /var/lib/one//datastores/1/9d6336a88d6ab62ea1dce65d81e55881
file format: qcow2
virtual size: 12 GiB (12884901888 bytes)
disk size: 3.52 GiB
cluster_size: 65536
```

```
Format specific information:
    compat: 1.1
    compression type: zlib
    lazy refcounts: false
    refcount bits: 16
    corrupt: false
    extended l2: false
```

На хосте назначения подключить образ диска к ВМ (рассмотрено подключение на основе Directory Storage), выполнив следующие действия:

1) создать новую ВМ в веб-интерфейсе PVE или командой:

```
# qm create 120 --bootdisk scsi0 --net0 virtio,bridge=vmbr0 --scsihw virtio-scsi-pci
```

2) чтобы использовать в PVE образ диска в формате qcow2 (полученный из другой системы KVM, либо преобразованный из другого формата), его необходимо импортировать. Команда импорта:

```
# qm importdisk <vmid> <source> <storage> [OPTIONS]
```

Команда импорта диска f811a893808a9d8f5bf1c029b3c7e905 в хранилище local, для ВМ с ID 120 (подразумевается, что образ импортируемого диска находится в каталоге, из которого происходит выполнение команды):

```
# qm importdisk 120 f811a893808a9d8f5bf1c029b3c7e905 local --format qcow2
importing disk 'f811a893808a9d8f5bf1c029b3c7e905' to VM 120 ...
…
Successfully imported disk as 'unused0:local:120/vm-120-disk-0.qcow2'
```

3) привязать диск к ВМ:

- в веб-интерфейсе PVE: перейти на вкладку «Оборудование» созданной ВМ. В списке устройств будет показан неиспользуемый жесткий диск, выбрать его, выбрать режим «SCSI» и нажать кнопку «Добавить» (Рис. 235).
- в командной строке:

```
# qm set 120 --scsi0 local:120/vm-120-disk-0.qcow2
update VM 120: -scsi0 local:120/vm-120-disk-0.qcow2
```

4) донастроить параметры процессора, памяти, сетевых интерфейсов, порядок загрузки;

5) включить ВМ.

4.9.3.2 Миграция ВМ из VMware в PVE

Экспорт ВМ из внешнего гипервизора обычно заключается в переносе одного или нескольких образов дисков с файлом конфигурации, описывающим настройки ВМ (ОЗУ, количество ядер). Образы дисков могут быть в формате vmdk (VMware или VirtualBox), или qcow2 (KVM).

В данном разделе рассмотрена миграция ВМ из VMware в PVE на примере ВМ с ОС Windows 7.

Добавление диска к ВМ

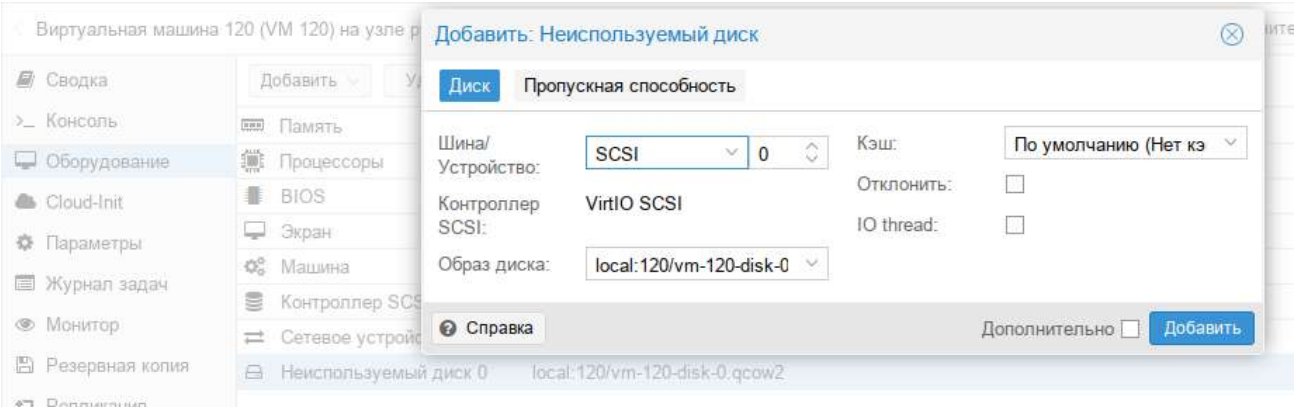


Рис. 235

Подготовить ОС Windows. ОС Windows должна загружаться с дисков в режиме IDE.

Подготовить образ диска. Необходимо преобразовать образ диска в тип `single growable virtual disk`. Сделать это можно с помощью утилиты `vmware-vdiskmanager` (поставляется в комплекте с VMWare Workstation). Для преобразования образа перейти в папку с образами дисков и выполнить команду:

```
"C:\Program Files\VMware\VMware Server\vmware-vdiskmanager"
-r win7.vmdk -t 0 win7-pve.vmdk
```

где win7.vmdk – файл с образом диска.

Подключить образ диска к ВМ одним из трёх указанных способов:

1) подключение образа диска к ВМ на основе Directory Storage:

- в веб-интерфейсе PVE создать ВМ KVM;
- скопировать преобразованный образ win7-pve.vmdk в каталог с образами ВМ /var/lib/vz/images/VMID, где VMID – VMID, созданной виртуальной машины (можно воспользоваться WinSCP);
- преобразовать файл win7-pve.vmdk в qemu формат:

```
# qemu-img convert -f vmdk win7-pve.vmdk -O qcow2 win7-pve.qcow2
```

- добавить в конфигурационный файл ВМ (`/etc/pve/nodes/pve02/qemu-server/VMID.conf`) строку:

```
unused0: local:100/win7-pve.qcow2
```

где 100 – VMID, а local – хранилище в PVE.

- перейти в веб-интерфейсе PVE на вкладку «Оборудование» созданной ВМ. В списке устройств будет показан неиспользуемый жесткий диск, выбрать его, выбрать режим IDE и нажать кнопку «Добавить» (Рис. 236).

Добавление диска к ВМ

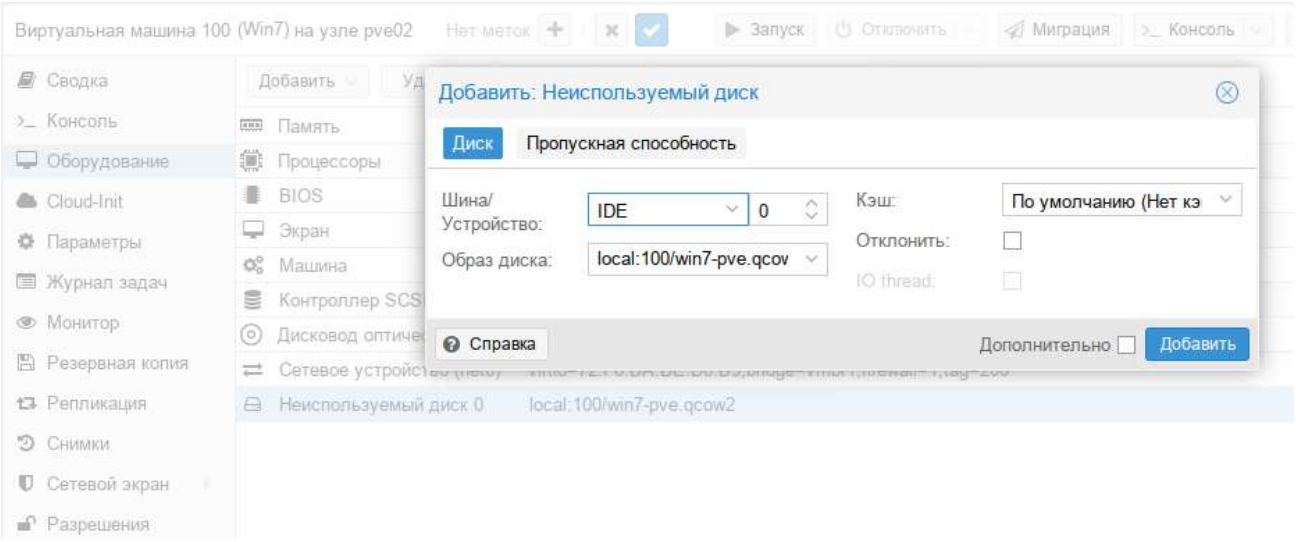


Рис. 236

2) подключение образа диска к ВМ на основе LVM Storage:

- в веб-интерфейсе PVE создать ВМ с диском большего размера, чем диск в образе vmdk. Посмотреть размер диска в образе можно командой:

```
# qemu-img info win7-pve.vmdk
image: win7-pve.vmdk
file format: vmdk
virtual size: 127G (136365211648 bytes)
disk size: 20.7 GiB
cluster_size: 65536
Format specific information:
   cid: 3274246794
   parent cid: 4294967295
   create type: streamOptimized
   extents:
       [0]:
       compressed: true
       virtual size: 136365211648
       filename: win7-pve.vmdk
       cluster size: 65536
       format:
```

В данном случае необходимо создать диск в режиме IDE размером не меньше 127GB.

- скопировать преобразованный образ win7-pve.vmdk в каталог с образами ВМ `/var/lib/vz/images/VMID`, где VMID – VMID, созданной ВМ (можно воспользоваться WinSCP);
- перейти в консоль ноды кластера и посмотреть, как называется LVM диск созданной ВМ (диск должен быть в статусе ACTIVE):

```
# lvscan
```

```
ACTIVE            '/dev/sharedsv/vm-101-disk-1' [130,00 GiB] inherit
```

- сконвертировать образ vdmk в raw формат непосредственно на LVM-устройство:

```
# qemu-img convert -f vmdk win7-pve.vmdk -O raw /dev/sharedsv/vm-101-disk-1
```

3) подключение образа диска к ВМ на основе CEPH Storage:

- в веб-интерфейсе PVE создать ВМ с диском большего размера, чем диск в образе vmdk. Посмотреть размер диска в образе можно командой:

```
# qemu-img info win7-pve.vmdk
```

- скопировать преобразованный образ win7-pve.vmdk в каталог с образами ВМ `/var/lib/vz/images/VMID`, где VMID – VMID, созданной виртуальной машины;
- перейти в консоль ноды кластера. Отобразить образ из пула CEPH в локальное блочное устройство:

```
# rbd map rbd01/vm-100-disk-1
/dev/rbd0
```

Примечание. Имя нужного пула можно посмотреть на вкладке «Центр обработки данных» → «Хранилище» → «rbd-storage».

- сконвертировать образ vdmk в raw формат непосредственно на отображенное устройство:

```
# qemu-img convert -f vmdk win7-pve.vmdk -O raw /dev/rbd0
```

Адаптация новой ВМ:

1) проверить режим работы жесткого диска: для Windows – IDE, для Linux – SCSI.

2) установить режим VIRTIO для жесткого диска (режим VIRTIO также доступен для Windows, но сразу загрузиться в этом режиме система не может):

- загрузиться в режиме IDE и выключить машину. Добавить еще один диск в режиме VIRTIO и включить машину. Windows установит нужные драйвера;
- выключить машину;
- изменить режим основного диска с IDE на VIRTIO;
- загрузить систему, которая должна применить VIRTIO драйвер и выдать сообщение, что драйвер от RedHat.

3) включить ВМ. Первое включение займет какое-то время (будут загружены необходимые драйвера).

4.9.3.3 Пример импорта Windows OVF

Скопировать файлы ovf и vmdk на хост PVE. Создать новую ВМ, используя имя ВМ, информацию о ЦП и памяти из файла конфигурации OVF, и импортировать диски в хранилище local-lvm:

```
# qm importovf 999 WinDev2212Eval.ovf local-lvm
```

Примечание. Сеть необходимо настроить вручную.

4.10 Клонирование виртуальных машин

Простой способ развернуть множество ВМ одного типа – создать клон существующей ВМ.

Существует два вида клонов:

- Полный клон – результатом такой копии является независимая ВМ. Новая ВМ не имеет общих ресурсов с оригинальной ВМ. При таком клонировании можно выбрать целевое хранилище, поэтому его можно использовать для миграции ВМ в совершенно другое хранилище. При создании клона можно также изменить формат образа диска, если хранилище поддерживает несколько форматов.
- Связанный клон – такой клон является перезаписываемой копией, исходное содержимое которой совпадает с исходными данными. Создание связанного клона происходит практически мгновенно и изначально не требует дополнительного места. Клоны называются связанными потому что новый образ диска ссылается на оригинал. Немодифицированные блоки данных считываются из исходного образа, а изменения записываются (и затем считываются) из нового местоположения (исходный образ при этом должен работать в режиме только для чтения). С помощью PVE можно преобразовать любую ВМ в шаблон (см. ниже). Такие шаблоны впоследствии могут быть использованы для эффективного создания связанных клонов. При создании связанных клонов невозможно изменить целевое хранилище.

Примечание. При создании полного клона необходимо прочитать и скопировать все данные образа ВМ. Это обычно намного медленнее, чем создание связанного клона.

Весь функционал клонирования доступен в веб-интерфейсе PVE.

Для клонирования ВМ необходимо выполнить следующие шаги:

1) создать ВМ с необходимыми настройками (все создаваемые из такой ВМ клоны будут иметь идентичные настройки) или воспользоваться уже существующей ВМ;
2) в контекстном меню ВМ выбрать пункт «Клонировать» (Рис. 237);
3) откроется диалоговое окно (Рис. 238), со следующими полями:

- «Целевой узел» – узел получатель клонируемой ВМ (для создания новой ВМ на другом узле необходимо чтобы ВМ находилась в общем хранилище и это хранилище должно быть доступно на целевом узле);
- «VM ID» – идентификатор ВМ;
- «Имя» – название новой ВМ;
- «Пул ресурсов» – пул, к которому будет относиться ВМ;
- «Режим» – метод клонирования (если клонирование происходит из шаблона ВМ). Доступны значения: «Полное клонирование» и «Связанная копия»;
- «Снимок» – снимок, из которого будет создаваться клон (если снимки существуют);

- «Целевое хранилище» – хранилище для клонируемых виртуальных дисков;
- «Формат» – формат образа виртуального диска.

4) для запуска процесса клонирования нажать кнопку «Клонировать».

Контекстное меню ВМ

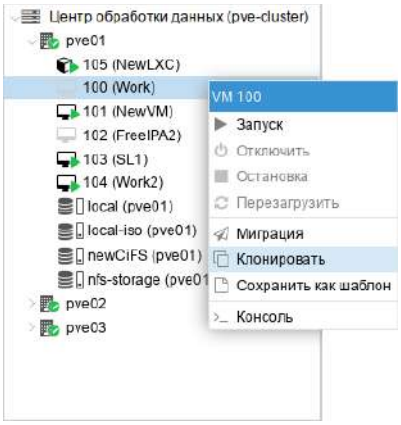


Рис. 237

Настройки клонирования

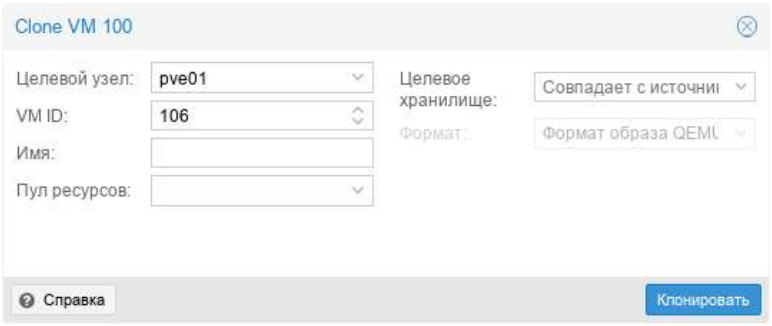


Рис. 238

Некоторые типы хранилищ позволяют копировать определенный снимок ВМ (Рис. 239), который по умолчанию соответствует текущим данным ВМ. Клон ВМ никогда не содержит дополнительных снимков оригинальной ВМ.

Выбор снимка для клонирования

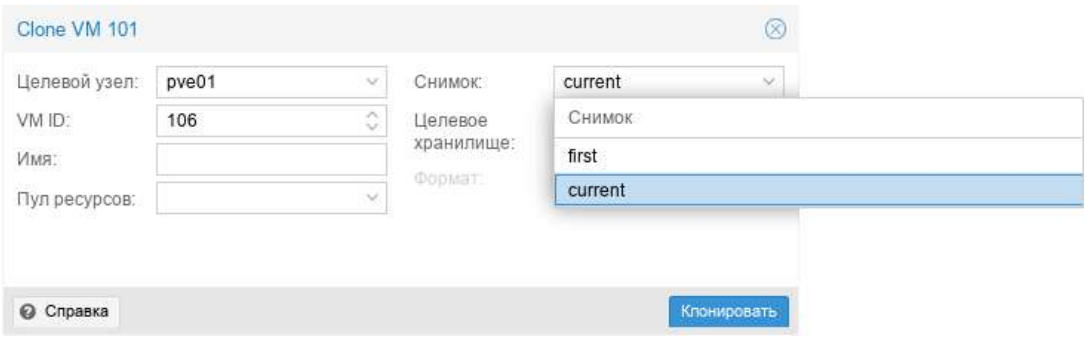


Рис. 239

Чтобы избежать конфликтов ресурсов, при клонировании MAC-адреса сетевых интерфейсов рандомизируются, и генерируется новый UUID для настройки BIOS ВМ (smbios1).

4.11 Шаблоны ВМ

ВМ можно преобразовать в шаблон. Такие шаблоны доступны только для чтения, и их можно использовать для создания связанных клонов.

Примечание. Если ВМ содержит снимки (snapshot), то преобразовать такую ВМ в шаблон нельзя. Для преобразования ВМ в шаблон необходимо предварительно удалить все снимки этой ВМ.

Для преобразования ВМ в шаблон необходимо в контекстном меню ВМ выбрать пункт «Сохранить как шаблон» (Рис. 240) и в ответ на запрос на подтверждения, нажать кнопку «Да».

Создание шаблона ВМ

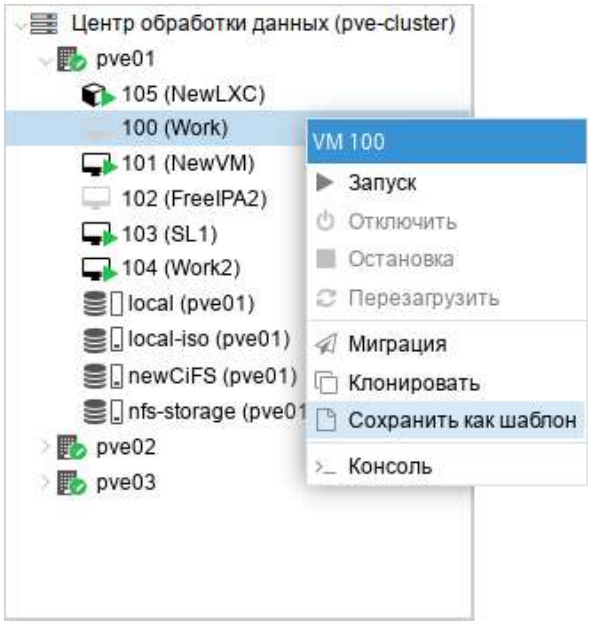


Рис. 240

Примечание. Запустить шаблоны невозможно, так как это приведет к изменению образов дисков. Если необходимо изменить шаблон, следует создать связанный клон и изменить его.

Для создания связанного клона необходимо выполнить следующие шаги:

1) в контекстном меню шаблона ВМ выбрать пункт «Клонировать» (Рис. 241);

2) в открывшемся диалоговом окне (Рис. 242) указать параметры клонирования (в поле «Режим» следует выбрать значение «Связанная копия»);

3) для запуска процесса клонирования нажать кнопку «Клонировать».

Создание связанного клона

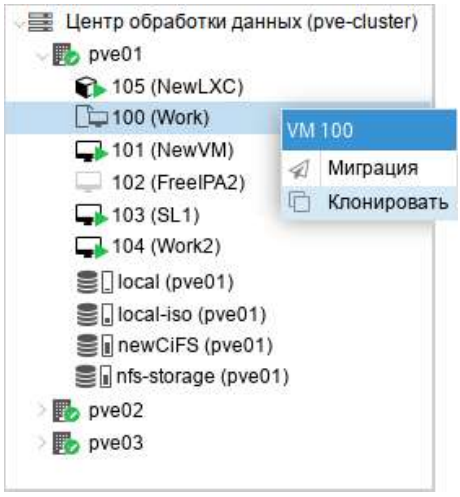


Рис. 241

Создание связанного клона

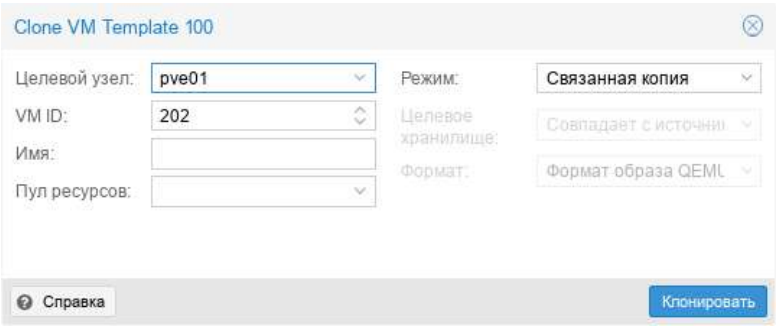


Рис. 242

4.12 Теги (метки) ВМ

В организационных целях для ВМ (KVM и LXC) можно установить теги (метки). Теги отображаются в дереве ресурсов и в строке статуса при выборе ВМ (Рис. 243). Теги позволяют фильтровать ВМ (Рис. 244).

Теги ВМ 103

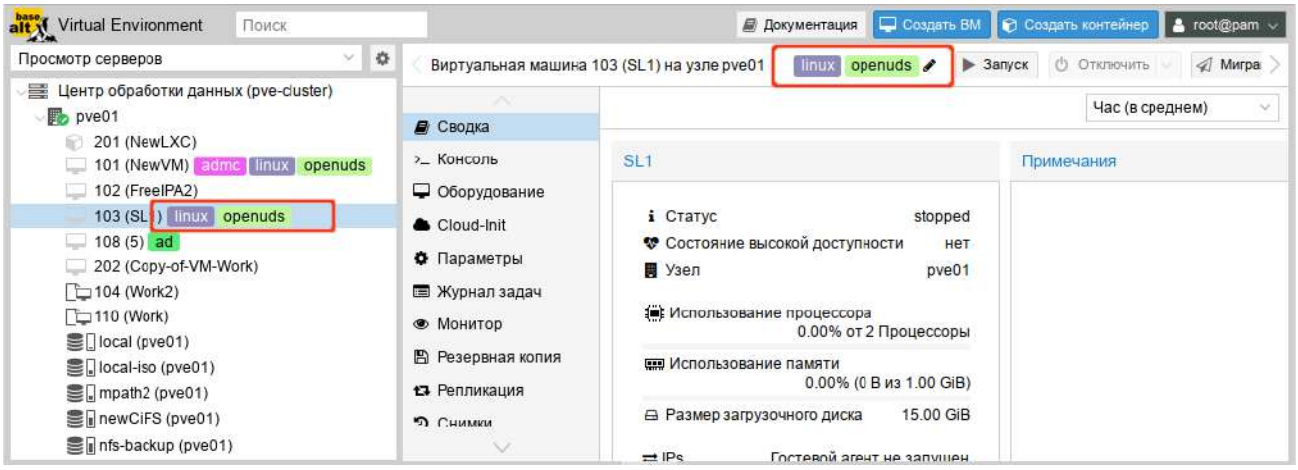


Рис. 243

Фильтрация ВМ по тегам (меткам)

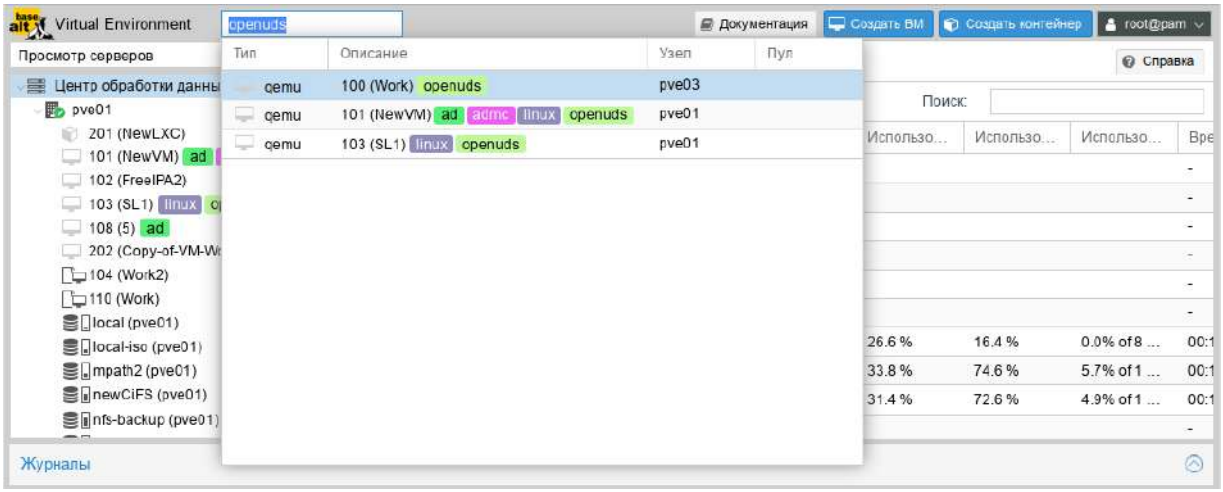


Рис. 244

4.12.1 Работа с тегами

Для добавления, редактирования, удаления тегов необходимо в строке статуса ВМ нажать на значок карандаша (Рис. 245). Можно добавить несколько тегов, нажав кнопку «+», и удалить их, нажав кнопку «-». Чтобы сохранить или отменить изменения, используются кнопки « ✓ » и «x» соответственно (Рис. 246).

Строка статуса ВМ

Рис. 245

Теги ВМ 102

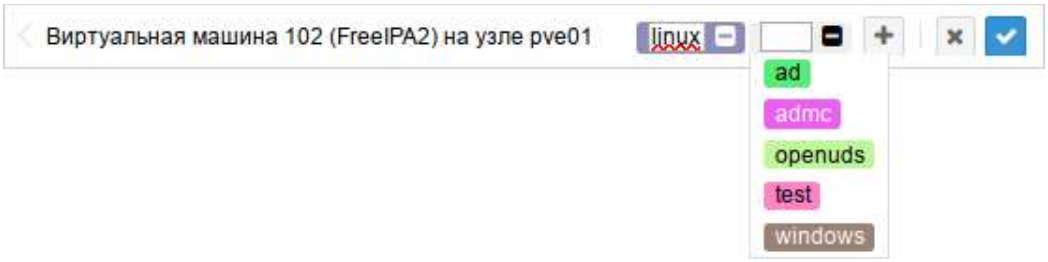


Рис. 246

Теги также можно устанавливать в командной строке (несколько тегов разделяются точкой с запятой):

```
# qm set ID --tags 'myfirsttag;mysecondtag'
```

Например:

```
# qm set 103 --tags 'linux;openuds'
```

4.12.2 Настройка тегов

В глобальных параметрах центра обработки данных PVE (раздел «Центр обработки данных» → «Параметры») есть 3 пункта меню, посвященных тегам (Рис. 247). Здесь можно, среди прочего, заранее определить теги и напрямую назначить им цвет.

Настройки тегов

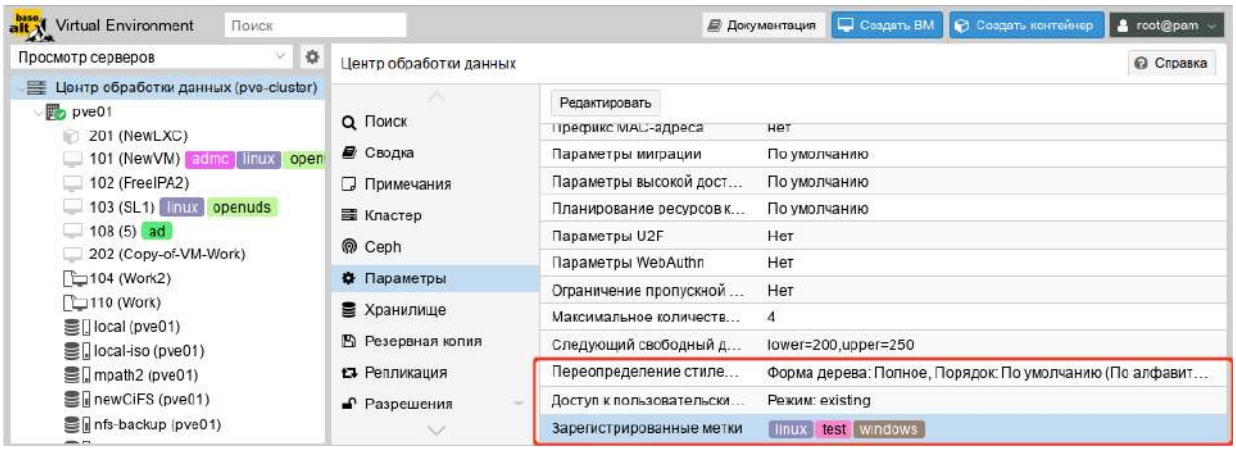


Рис. 247

4.12.2.1 Стиль тегов

Цвет, форму избражения тегов в дереве ресурсов, чувствительность к регистру, а также способ сортировки тегов можно настроить в параметре «Переопределение стилей меток» (Рис. 248):

- «Форма дерева» – позволяет указать форму отображения тегов:
  - «Полное» – отображать полнотекстовую версию;
  - «Круговое» – использовать только цветовой акцент: круг с цветом фона (по умолчанию);
  - «Плотное» – использовать только цветовой акцент: небольшой прямоугольник (полезно, когда каждой ВМ назначено много тегов);
  - «Нет» – отключить отображение тегов;
- «Порядок» – управляет сортировкой тегов в веб-интерфейсе;
- «С учётом регистра» – позволяет указать, должна ли фильтрация уникальных тегов учитывать регистр символов;
- «Переопределение цветов» – позволяет задать цвета для тегов (по умолчанию цвета тегов автоматически выбирает PVE).

Настроить стиль тегов можно также в командной строке, используя команду:

```
# pvesh set /cluster/options --tag-style [case-sensitive=<1|0>]\
[,color-map=<tag>:<hex-color> [:<hex-color-for-text>][;<tag>=...]]\
[,ordering=<config|alphabetical>][,shape=<circle|dense|full|none>]
```

Например, следущая команда установит для тега «FreeIPA» цвет фона черный (#000000), а цвет текста – белый (#FFFFFF) и форму тегов «Плотное»:

```
# pvesh set /cluster/options --tag-style color-map=FreeIPA:000000:FFFFFF,shape=dense
```

Примечание. Команда pvesh set удалит все ранее переопределённые стили тегов.

Переопределение стилей меток

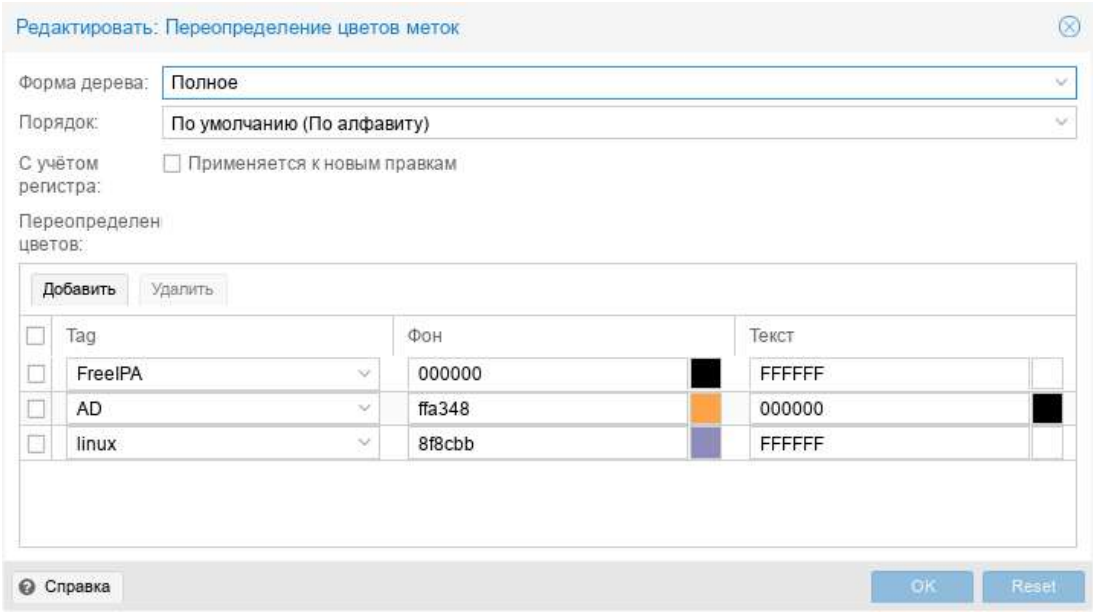


Рис. 248

4.12.2.2 Права

По умолчанию пользователи с привилегиями VM.Config.Options могут устанавливать любые теги для ВМ (/vms/ID) (см. «Управление доступом»). Если необходимо ограничить такое поведение, соответствующие разрешения можно установить в разделе «Доступ к пользовательским меткам» (Рис. 249). Доступны следующие режимы (поле «Режим»):

- «free» – пользователи не ограничены в установке тегов (по умолчанию);
- «list» – пользователи могут устанавливать теги на основе заранее определенного списка тегов;
- «existing» – аналогично режиму «list», но пользователи также могут использовать уже существующие теги;
- «none» – пользователям запрещено использовать теги.

Здесь же можно определить, какие теги разрешено использовать пользователям (поле «Предустановленные метки») если используются режимы «list» или «existing».

Назначить права можно также и в командной строке, используя команду:

```
# pvesh set /cluster/options --user-tag-access\
[user-allow=<existing|free|list|none>][,user-allow-list=<tag>[;<tag>...]]
```

Например, запретить пользователям использовать теги:

```
# pvesh set /cluster/options --user-tag-access user-allow=none
```

Доступ к пользовательским меткам

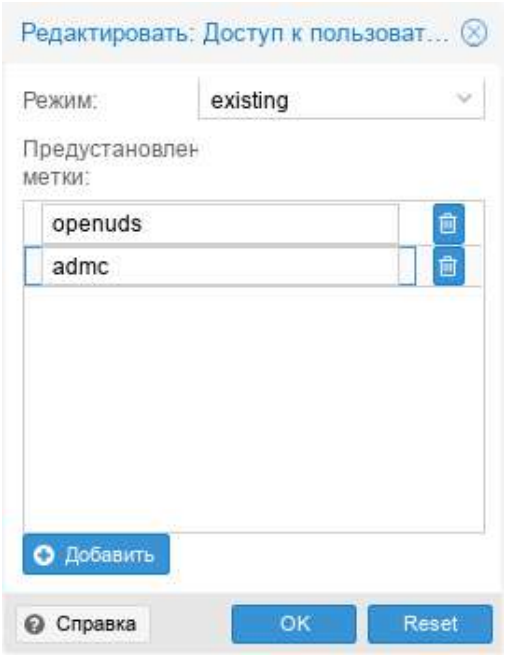


Рис. 249

Следует обратить внимание, что пользователь с привилегиями Sys.Modify на / всегда может устанавливать или удалять любые теги, независимо от настроек в разделе «Доступ к пользовательским меткам». Кроме того, существует настраиваемый список зарегистрированных тегов, которые могут добавлять и удалять только пользователи с привилегией Sys.Modify на /. Список зарегистрированных тегов можно редактировать в разделе «Зарегистрированные метки» (Рис. 250) или через интерфейс командной строки:

```
# pvesh set /cluster/options --registered-tags <tag>[;<tag>...]
```

Зарегистрированные метки

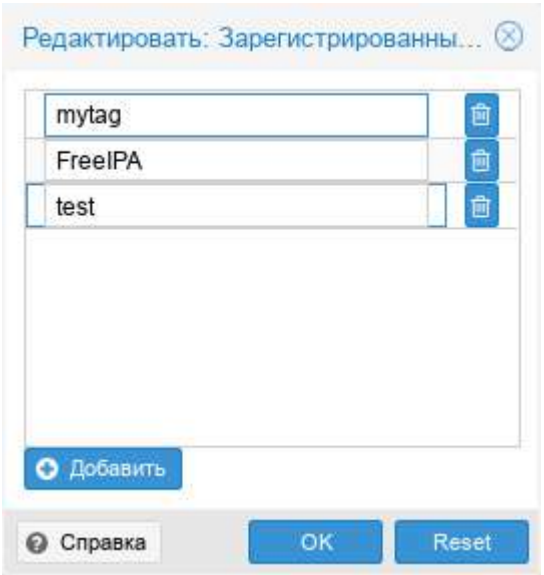


Рис. 250

4.13 Резервное копирование (backup)

PVE предоставляет полностью интегрированное решение, использующее возможности всех хранилищ и всех типов гостевых систем.

Резервные копии PVE представляют собой полные резервные копии – они содержат конфигурацию ВМ/CT и все данные. Резервное копирование может быть запущено через графический интерфейс или с помощью утилиты командной строки `vzdump`.

Задания для резервного копирования можно запланировать так, чтобы они выполнялись автоматически в определенные дни и часы для конкретных узлов и гостевых систем.

4.13.1 Режимы резервного копирования

Существует несколько способов обеспечения согласованности (параметр `mode`) в зависимости от типа гостевой системы.

Режимы резервного копирования для ВМ:

- режим остановки (Stop) – обеспечивает самую высокую надежность резервного копирования, но требует полного выключения ВМ. В этом режиме ВМ отправляется команда на штатное выключение, после остановки выполняется резервное копирование и затем отдается команда на включение ВМ. Количество ошибок при таком подходе минимально и чаще всего сводится к нулю;
- режим ожидания (Suspend) – ВМ временно «замораживает» свое состояние, до окончания процесса резервного копирования. Содержимое оперативной памяти не стирается, что позволяет продолжить работу ровно с той точки, на которой работа была приостановлена. Сервер простаивает во время копирования информации, но при этом нет необходимости выключения/включения ВМ, что достаточно критично для некоторых сервисов;
- режим снимка (Snapshot) – обеспечивает минимальное время простоя ВМ (использование этого механизма не прерывает работу ВМ), но имеет два очень серьезных недостатка – могут возникать проблемы из-за блокировок файлов операционной системой и самая низкая скорость создания. Резервные копии, созданные этим методом, надо всегда проверять в тестовой среде.

Примечание. Live резервное копирование PVE обеспечивает семантику, подобную моментальным снимкам, для любого типа хранилища (не требуется, чтобы базовое хранилище поддерживало снимки). Так как резервное копирование выполняется с помощью фонового процесса QEMU, остановленная ВМ на короткое время будет отображаться как работающая, пока QEMU читает диски ВМ. Однако сама ВМ не загружается, читаются только ее диски.

Режимы резервного копирования для контейнеров:

- режим остановки (Stop) – остановка контейнера на время резервного копирования. Это может привести к длительному простою;
- режим ожидания (Suspend) – этот режим использует `rsync` для копирования данных контейнера во временную папку (опция `--tmpdir`). Затем контейнер приостанавливается и `rsync` копирует измененные файлы. После этого контейнер возобновляет свою работу. Это приводит к минимальному времени простоя, но требует дополнительное пространство для хранения копии контейнера. Когда контейнер находится в локальной файловой системе и хранилищем резервной копии является сервер NFS, необходимо установить `--tmpdir` также и на локальную файловую систему, так как это приведет к повышению производительности. Использование локального `tmpdir` также необходимо, если требуется сделать резервную копию локального контейнера с использованием списков контроля доступа (ACL) в режиме ожидания, если хранилище резервных копий – сервер NFS;
- режим снимка (Snapshot) – этот режим использует возможности мгновенных снимков основного хранилища. Сначала контейнер будет приостановлен для обеспечения согласованности данных, будет сделан временный снимок томов контейнера, а содержимое снимка будет заархивировано в tar-файле, далее временный снимок удаляется. Для возможности использования этого режима необходимо, чтобы тома резервных копий находились в хранилищах, поддерживающих моментальные снимки. Используя опцию `backup=0` для точки монтирования, можно исключить отдельные тома из резервной копии (и, следовательно, обойти это требование).

Примечание.По умолчанию дополнительные точки монтирования, кроме точки монтирования «Корневой диск» («Root Disk»), не включаются в резервные копии. Для точек монтирования томов можно настроить опцию «Резервная копия» (Рис. 251), чтобы включить точку монтирования в резервную копию.

Настройки точки монтирования

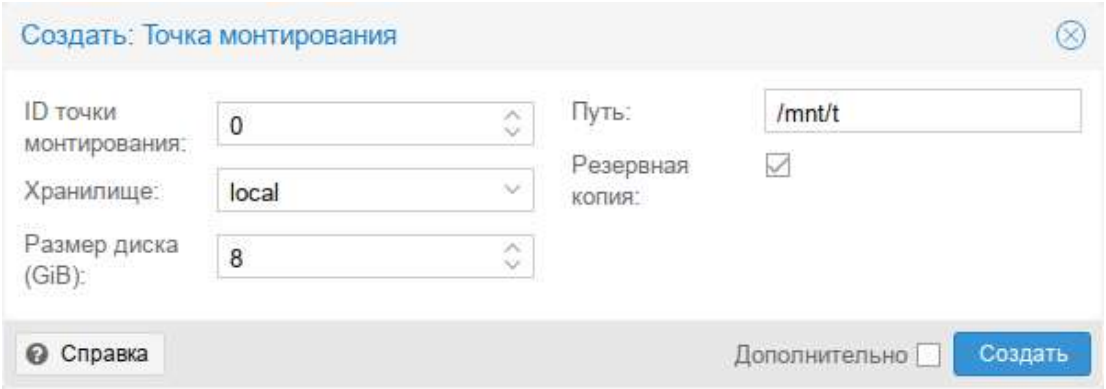


Рис. 251

4.13.2 Хранилище резервных копий

Перед тем как настроить резервное копирование, необходимо определить хранилище резервных копий. Это может быть хранилище Proxmox Backup Server, где резервные копии хранятся в виде дедуплицированных фрагментов и метаданных, или хранилище на уровне файлов, где резервные копии хранятся в виде обычных файлов. Рекомендуется использовать Proxmox Backup Server на выделенном узле из-за его расширенных функций. Использование сервера NFS является хорошей альтернативой.

Если хранилище будет использоваться только для резервных копий, следует выставить соответствующие настройки (Рис. 252).

Настройка хранилища NFS

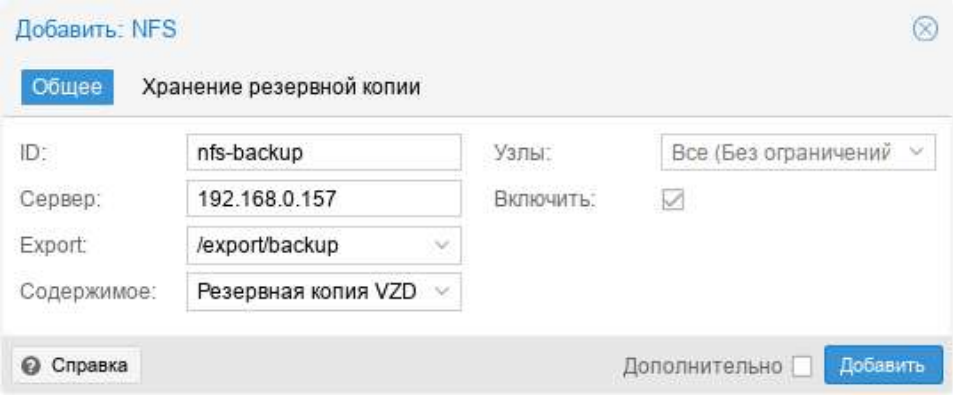


Рис. 252

На вкладке «Хранение резервной копии» можно указать параметры хранения резервных копий (Рис. 253).

Параметры хранения резервных копий в хранилище NFS

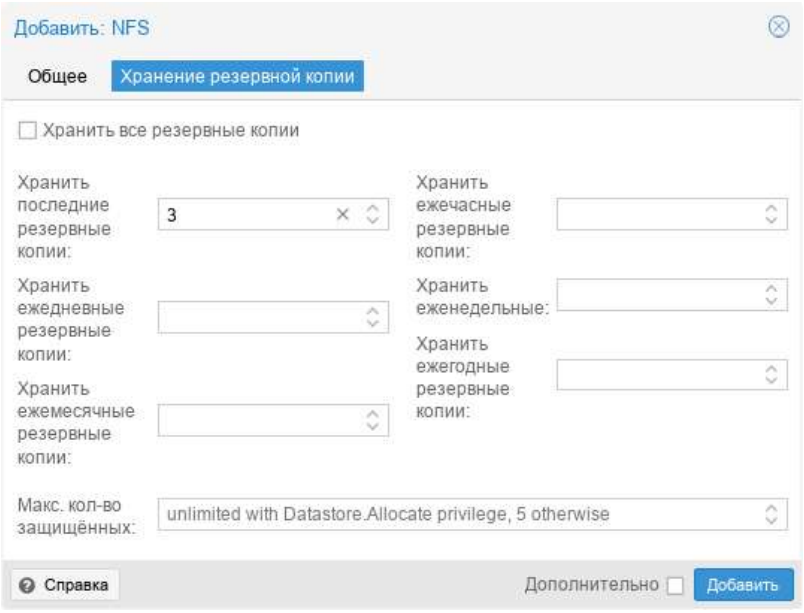


Рис. 253

Доступны следующие варианты хранения резервных копий (в скобках указаны параметры опции `prune-backups` команды `vzdump`):

- «Хранить все резервные копии» (`keep-all=<1|0>`) – хранить все резервные копии (если отмечен этот пункт, другие параметры не могут быть установлены);
- «Хранить последние резервные копии» (`keep-last=<N>`) – хранить <N> последних резервных копий;
- «Хранить ежечасные резервные копии» (`keep-hourly=<N>`) – хранить резервные копии за последние <N> часов (если за один час создается более одной резервной копии, сохраняется только последняя);
- «Хранить ежедневные резервные копии» (`keep-daily=<N>`) – хранить резервные копии за последние <N> дней (если за один день создается более одной резервной копии, сохраняется только самая последняя);
- «Хранить еженедельные» (`keep-weekly=<N>`) – хранить резервные копии за последние <N> недель (если за одну неделю создается более одной резервной копии, сохраняется только последняя);
- «Хранить ежемесячные резервные копии» (`keep-monthly=<N>`) – хранить резервные копии за последние <N> месяцев (если за один месяц создается более одной резервной копии, сохраняется только самая последняя);
- «Хранить ежегодные резервные копии» (`keep-yearly <N>`) – хранить резервные копии за последние <N> лет (если за один год создается более одной резервной копии, сохраняется только самая последняя).

«Макс. кол-во защищенных» (параметр хранилища: `max-protected-backups`) – количество защищённых резервных копий на гостевую систему, которое разрешено в хранилище. Для указания неограниченного количества используется значение -1. Значение по умолчанию: неограниченно для пользователей с привилегией `Datastore.Allocate` и 5 для других пользователей.

Варианты хранения обрабатываются в указанном выше порядке. Каждый вариант распространяется только на резервное копирование в определенный период времени.

Пример указания параметров хранения резервных копий при создании задания:

```
# vzdump 777 --prune-backups keep-last=3,keep-daily=13,keep-yearly=9
```

Несмотря на то что можно передавать параметры хранения резервных копий непосредственно при создании задания, рекомендуется настроить эти параметры на уровне хранилища.

4.13.3 Сжатие файлов резервной копии

Инструментарий для создания резервных копий PVE поддерживает следующие механизмы сжатия:

- сжатие LZO – алгоритм сжатия данных без потерь (реализуется в PVE утилитой lzop). Особенностью этого алгоритма является скоростная распаковка. Следовательно, любая резервная копия, созданная с помощью этого алгоритма, может при необходимости быть развернута за минимальное время.
- сжатие GZIP – при использовании этого алгоритма резервная копия будет «на лету» сжиматься утилитой GNU Zip, использующей мощный алгоритм Deflate. Упор делается на максимальное сжатие данных, что позволяет сократить место на диске, занимаемое резервными копиями. Главным отличием от LZO является то, что процедуры компрессии/декомпрессии занимают достаточно большое количество времени.
- сжатие Zstandard (zstd) – алгоритм сжатия данных без потерь. В настоящее время Zstandard является самым быстрым из этих трех алгоритмов. Многопоточность – еще одно преимущество zstd перед lzo и gzip.

4.13.4 Файлы резервных копий

Все создаваемые резервные копии будут сохраняться в каталоге `dump`. Имя файла резервной копии будет иметь вид:

- vzdump-qemu-номер_машины-дата-время.vma.zst, vzdump-lxc-номер_контейнера-дата-время.tar.zst в случае выбора метода сжатия ZST;
- vzdump-qemu-номер_машины-дата-время.vma.gz, vzdump- lxc-номер_контейнера -дата-время.vma.gz в случае выбора метода сжатия GZIP;
- vzdump-qemu-номер_машины-дата-время.vma.lzo, vzdump- lxc-номер_контейнера -дата-время.vma. lzo для использования метода LZO.

Если имя файла резервной копии не заканчивается одним из указанных выше расширений, то он не был сжат vzdump.

4.13.5 Шифрование резервных копий

Для хранилищ Proxmox Backup Server можно дополнительно настроить шифрование резервных копий на стороне клиента (см. «Шифрование»).

4.13.1 Выполнение резервного копирования в веб-интерфейсе

Для того чтобы разово создать резервную копию конкретной ВМ, достаточно выбрать ВМ, перейти в раздел «Резервная копия» и нажать кнопку «Создать резервную копию сейчас» (Рис. 254).

Вкладка «Резервная копия» ВМ

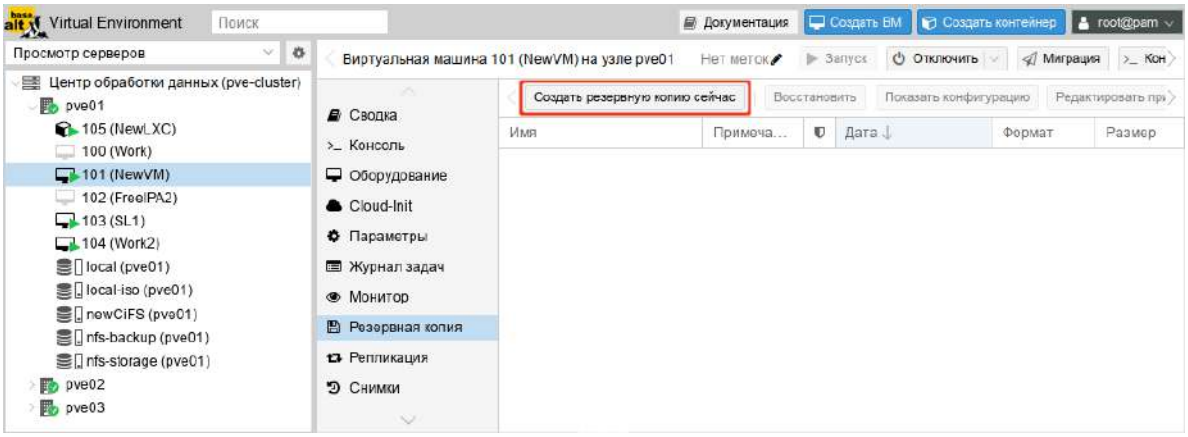


Рис. 254

Далее, в открывшемся окне (Рис. 255), следует указать параметры резервного копирования.

После создания резервной копии рекомендуется сразу убедиться, что из нее можно восстановить ВМ. Для этого необходимо открыть хранилище с резервной копией копией, выбрать резервную копию (Рис. 256) и начать процесс восстановления (Рис. 257). При восстановлении можно указать новое имя и переопределить некоторые параметры ВМ.

Выбор режима создания резервной копии

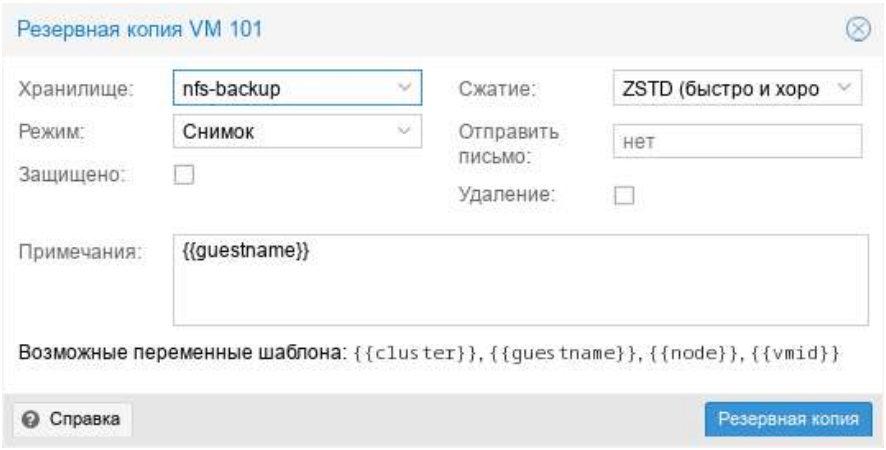


Рис. 255

Резервная копия в хранилище nfs-backup

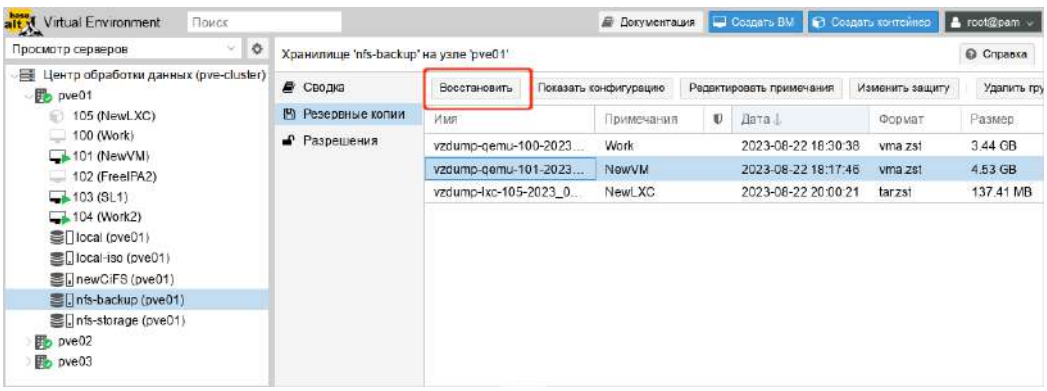


Рис. 256

Восстановить ВМ из резервной копии

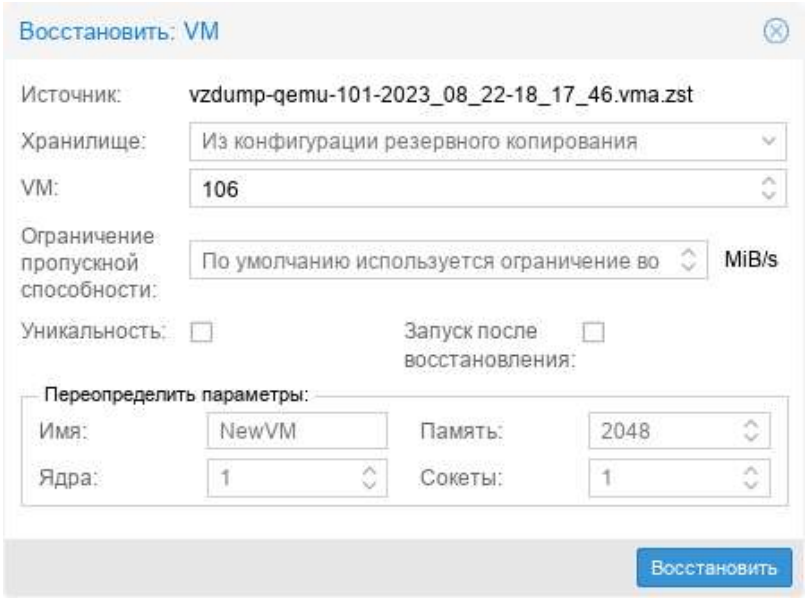


Рис. 257

Если восстанавливать из резервной копии в интерфейсе ВМ (Рис. 258), то будет предложена только замена существующей ВМ.

Восстановление из резервной копии в интерфейсе ВМ

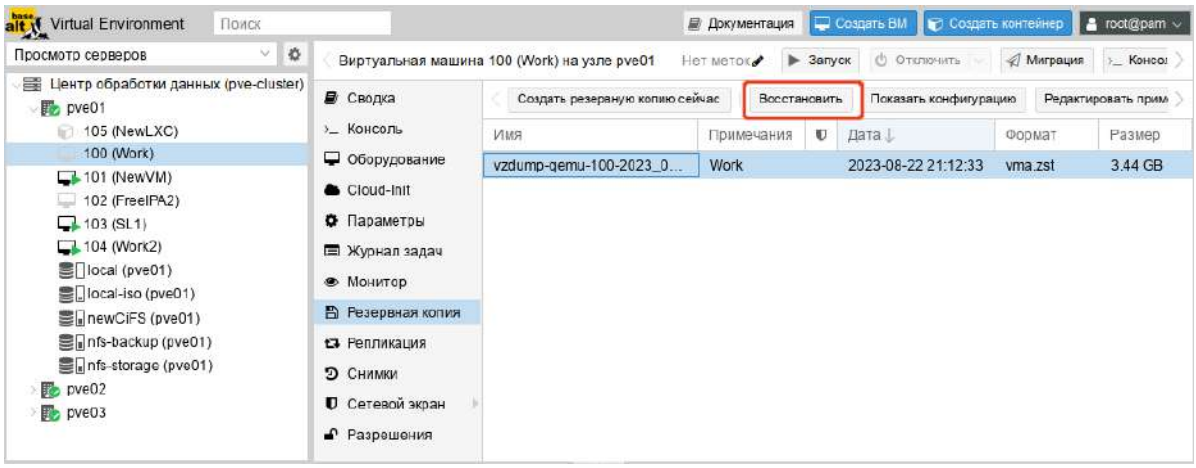


Рис. 258

Резервную копию можно пометить как защищённую (кнопка «Изменить защиту»), чтобы предотвратить ее удаление (Рис. 259).

Защищенная резервная копия

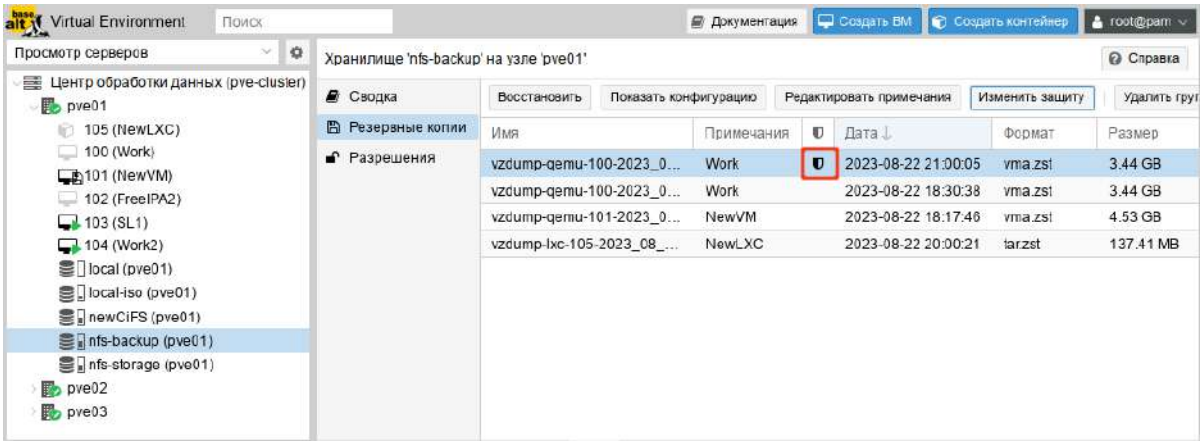


Рис. 259

Примечание. Попытка удалить защищенную резервную копию через пользовательский интерфейс, интерфейс командной строки или API PVE не удастся. Но так как это обеспечивается PVE, а не файловой системой, ручное удаление самого файла резервной копии по-прежнему возможно для любого, у кого есть доступ на запись к хранилищу резервных копий.

4.13.2 Задания резервного копирования

Помимо запуска резервного копирования вручную, можно также настроить периодические задания, которые выполняют резервное копирование всех или выбранных виртуальных гостевых систем в хранилище. Управлять заданиями можно в пользовательском интерфейсе (раздел «Центр обработки данных» → «Резервное копирование») или через конечную точку API /cluster/backup. Оба метода будут генерировать записи заданий в /etc/pve/jobs.cfg, которые анализируются и выполняются демоном `pvescheduler`.

Задание настраивается либо для всех узлов кластера, либо для определенного узла.

4.13.2.1 Формат расписания

Для настройки расписания используются события календаря системного времени (см. `man 7 systemd.time`).

Используется следующий формат:

```
[WEEKDAY] [[YEARS-]MONTHS-DAYS] [HOURS:MINUTES[:SECONDS]]
```

WEEKDAY – дни недели, указанные в трёх буквенном варианте на английском: mon,tue,wed,thu,fri,sat и sun. Можно использовать несколько дней в виде списка, разделённого запятыми. Можно задать диапазон дней, указав день начала и окончания, разделённые двумя точками («..»), например, mon..fri. Форматы можно смешивать. Если опущено, подразумевается «*».

Формат времени – время указывается в виде списка интервалов часов и минут. Часы и минуты разделяются знаком «:». И часы, и минуты могут быть списком и диапазонами значений в

том же формате, что и дни недели. Можно не указывать часы, если они не нужны. В этом случае подразумевается «*». Допустимый диапазон значений: 0–23 для часов и 0–59 для минут.

Специальные значения приведены в табл. 20. В таблице 21 приведены примеры периодов времени.

Т а б л и ц а 20 – Специальные значения

Расписание	Значение	Синтаксис
minutely	Каждую минуту	*-*-* *:*:00
hourly	Каждый час	*-*-* *:00:00
daily	Раз в день	*-*-* 00:00:00
weekly	Раз в неделю	mon *-*-* 00:00:00
monthly	Раз в месяц	*-*-01 00:00:00
yearly или annually	Раз в год	*-01-01 00:00:00
quarterly	Раз в квартал	*-01,04,07,10-01 00:00:00
semiannually или semi-annually	Раз в полгода	*-01,07-01 00:00:00

Т а б л и ц а 21 – Примеры периодов времени

Расписание	Эквивалент	Значение
mon,tue,wed,thu,fri	mon..fri	Каждый будний день в 00:00
sat,sun	sat..sun	В субботу и воскресенье в 00:00
mon,wed,fri	-	В понедельник, среду и пятницу в 00:00
12:05	12:05	Каждый день в 12:05
*/5	0/5	Каждые пять минут
mon..wed 30/10	mon,tue,wed 30/10	В понедельник, среду и пятницу в 30, 40 и 50 минут каждого часа
mon..fri 8..17,22:0/15	-	Каждые 15 минут с 8 часов до 18 и с 22 до 23 в будний день
fri 12..13:5/20	fri 12,13:5/20	В пятницу в 12:05, 12:25, 12:45, 13:05, 13:25 и 13:45
12,14,16,18,20,22:5	12/2:5	Каждые два часа каждый день с 12:05 до 22:05
*	*/1	Ежеминутно (минимальный интервал)
*-05	-	Пятого числа каждого месяца
Sat *-1..7 15:00	-	Первую субботу каждого месяца в 15:00
2023-10-22	-	22 октября 2023 года в 00:00

Проверить правильность задания расписания, можно в окне «Имитатор расписания заданий» («Центр обработки данных» →«Резервная копия» кнопка «Имитатор расписания»): указать

расписание в поле «Расписание», задать число итераций и нажать кнопку «Моделировать» (Рис. 260).

Имитатор расписания заданий

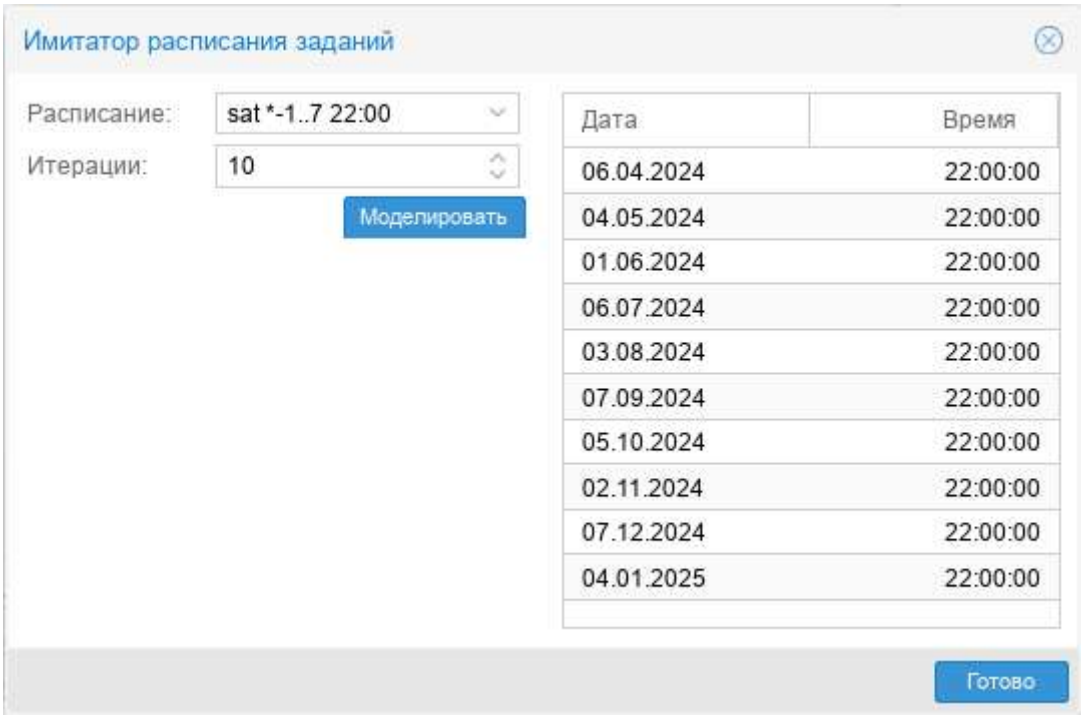


Рис. 260

4.13.2.2 Настройка заданий резервного копирования в веб-интерфейсе

Для того чтобы создать расписание резервного копирования, необходимо перейти во вкладку «Центр обработки данных» →«Резервная копия» (Рис. 261) и нажать кнопку «Добавить».

Вкладка «Резервная копия»

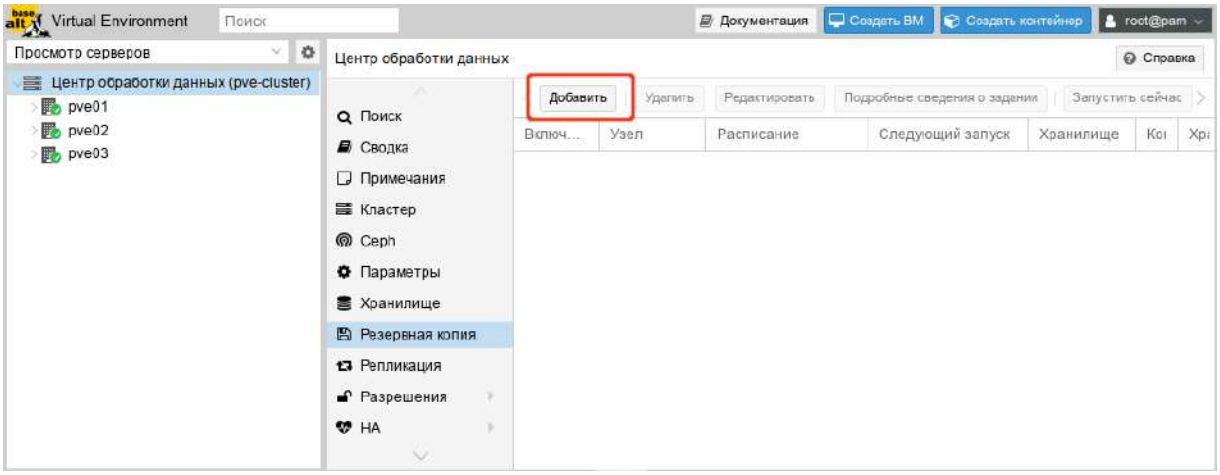


Рис. 261

При создании задания на резервирование, необходимо указать (Рис. 262):

- «Узел» – можно создавать график из одного места по разным узлам (серверам);
- «Хранилище» – точка смонтированного накопителя, куда будет проходить копирование;

- «Расписание» – здесь можно указать расписание резервного копирования. Можно выбрать период из списка (Рис. 263) или указать вручную;
- «Режим выбора» – возможные значения: «Учитывать выбранные ВМ», «Все», «Исключить выбранные ВМ», «На основе пула»;
- «Отправить письмо» – адрес, на который будут приходить отчёты о выполнении резервного копирования;
- «Адрес эл.почты» – принимает два значения: «Уведомлять всегда» – сообщение будет приходить при любом результате резервного копирования, «Только при ошибках» – сообщение будет приходить только в случае неудачной попытки резервного копирования;

  Примечание. Для возможности отправки электронной почты должен быть запущен postfix:

  ```
  # systemctl enable --now postfix
  ```
- «Сжатие» – метод сжатия, принимает четыре значения: «ZSTD (быстро и хорошо)» (по умолчанию), «LZO (быстро)», «GZIP (хорошо)» и «нет»;
- «Режим» – режим ВМ, в котором будет выполняться резервное копирование. Может принимать три значения (Рис. 264): «Снимок», «Приостановить», «Остановка».

Создание задания для резервного копирования. Вкладка «Общее»

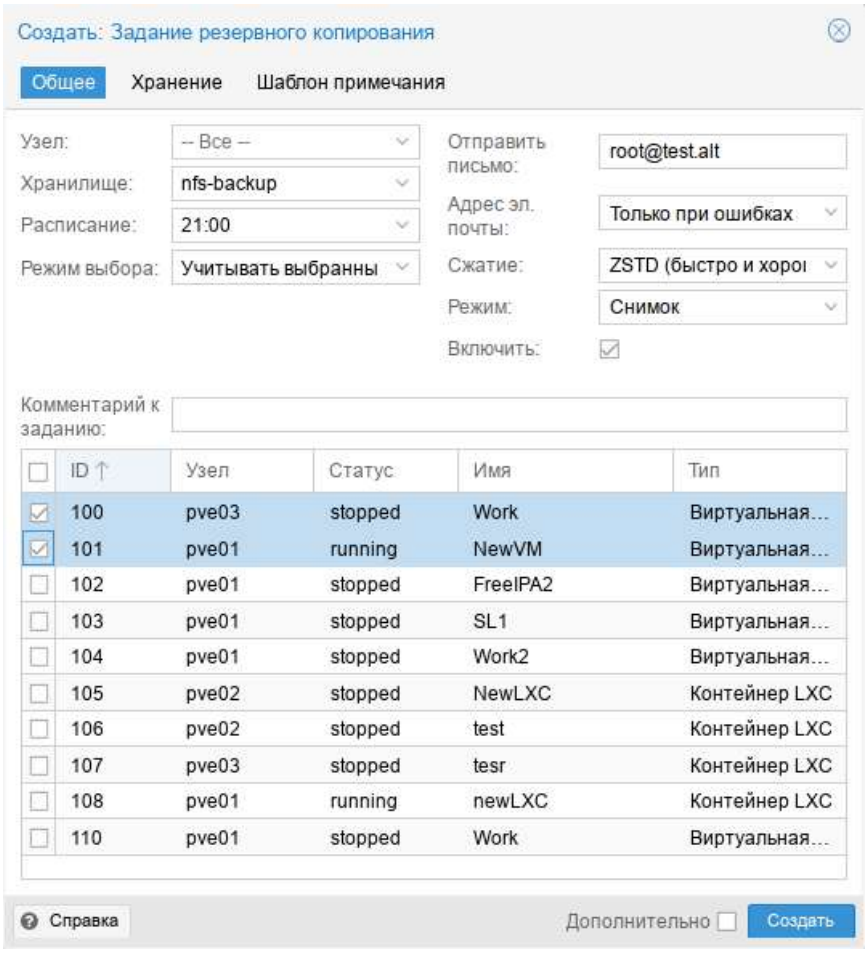


Рис. 262

Выбор расписания резервного копирования

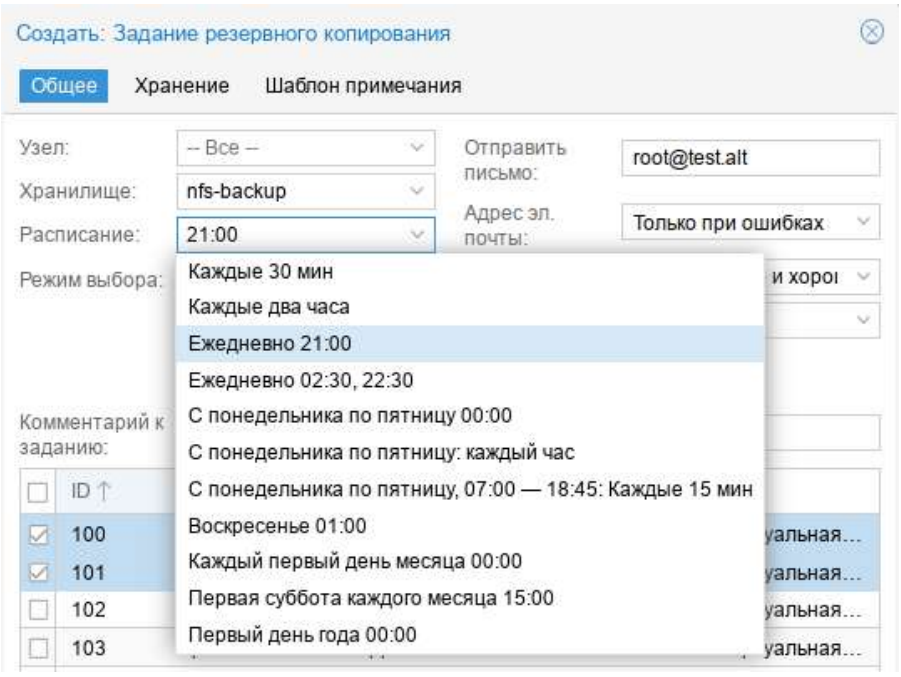


Рис. 263

Выбор режима создания резервной копии

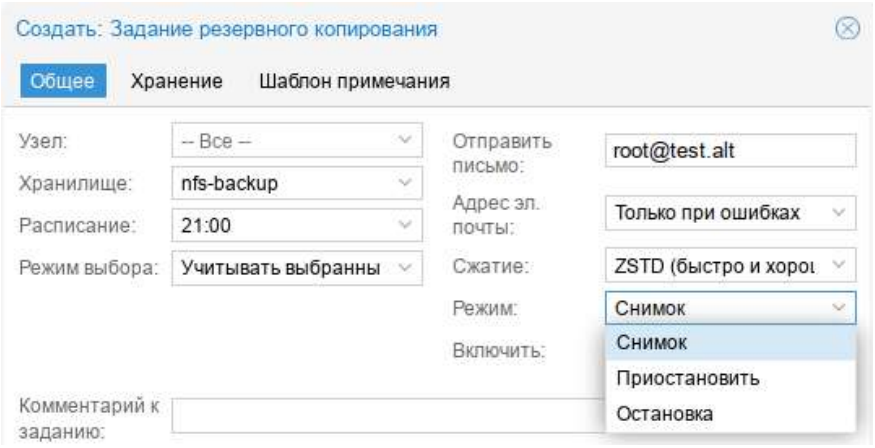


Рис. 264

Далее в списке необходимо выбрать ВМ/контейнеры, для которых создаётся задание резервного копирования. Для сокращения списка выбора можно использовать фильтры (Рис. 265). Фильтры доступны для полей «ID», «Статус», «Имя», «Тип».

Настройка фильтра

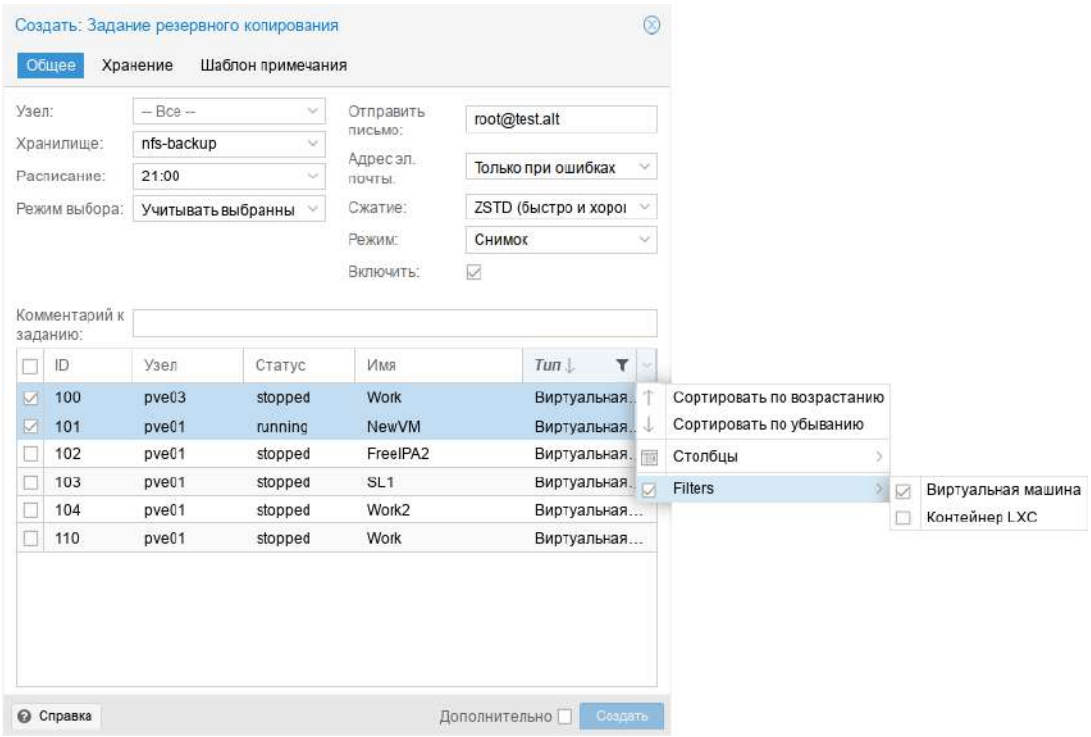


Рис. 265

Примечание. Поскольку запланированные задания не выполняются, если узел был в автономном режиме или pvescheduler был отключен в запланированное время, можно настроить поведение для наверстывания упущенного. Включив параметр «Повторять пропущенные» (доступно если установлена отметка в поле «Дополнительно») или указав параметр `repeat-missed` в конфигурации задания, можно указать планировщику, что он должен запустить пропущенные задания как можно скорее.

На вкладке «Хранение» можно настроить параметры хранения резервных копий (Рис. 266).

Создание задания для резервного копирования. Вкладка «Хранение»

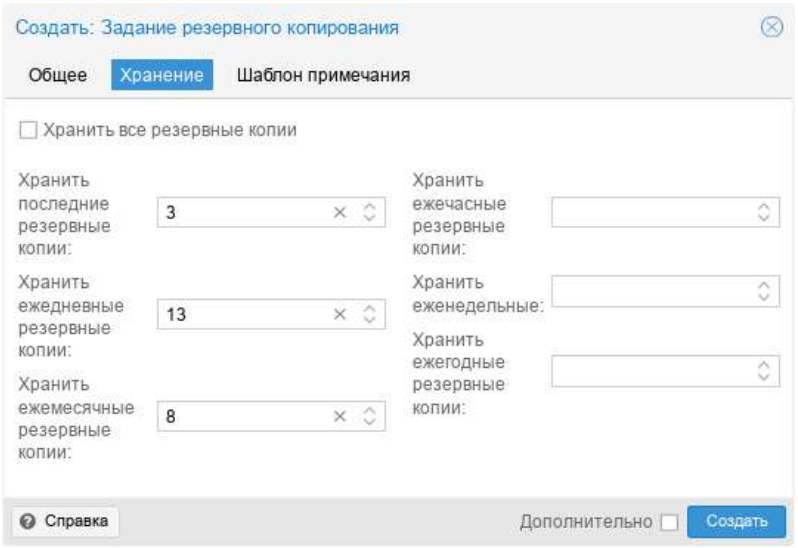


Рис. 266

На вкладке «Шаблон примечания» можно настроить примечание, которое будет добавляться к резервным копиям. Строка примечания может содержать переменные, заключенные в две фигурные скобки. Поддерживаются следующие переменные:

- {{cluster}} – имя кластера;
- {{guestname}} – имя ВМ/контейнера;
- {{node}} – имя узла, для которого создается резервная копия;
- {{vmid}} – VMID ВМ/контейнера.

Создание задания для резервного копирования. Вкладка «Шаблон примечания»

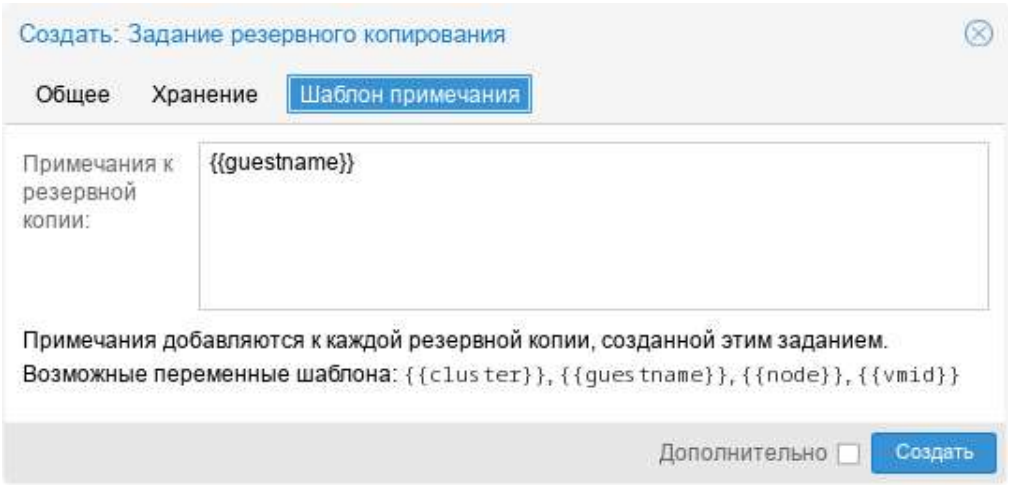


Рис. 267

После указания необходимых параметров и нажатия кнопки «Создать», задание для резервного копирования появляется в списке (Рис. 268). Запись о задании создаётся в файле `/etc/pve/jobs.cfg`.

Данное задание будет запускаться в назначенное время. Время следующего запуска задания отображается в столбце «Следующий запуск». Существует также возможность запустить задание по требованию – кнопка «Запустить сейчас».

Задание резервного копирования

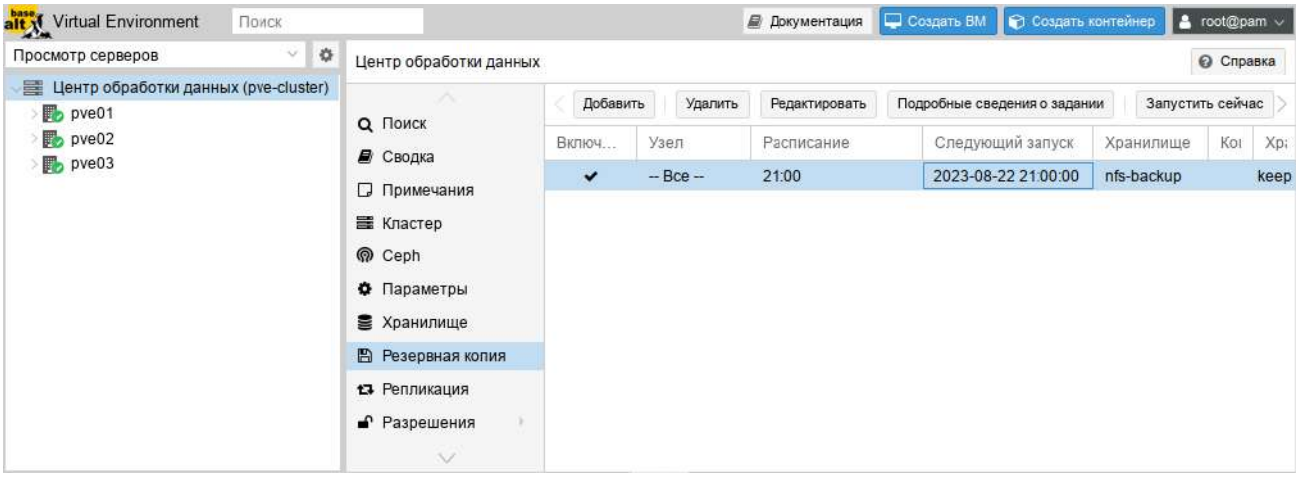


Рис. 268

4.13.3 Восстановление

Восстановить данные из резервных копий можно в веб-интерфейсе PVE или с помощью следующих утилит:

- `pct restore` – утилита восстановления контейнера;
- `qmrestore` – утилита восстановления ВМ.

4.13.4 Ограничение пропускной способности

Для восстановления одной или нескольких больших резервных копий может потребоваться много ресурсов, особенно пропускной способности хранилища как для чтения из резервного хранилища, так и для записи в целевое хранилище. Это может негативно повлиять на работу других ВМ, так как доступ к хранилищу может быть перегружен. Чтобы этого избежать, можно установить ограничение полосы пропускания для задания резервного копирования. В PVE есть два вида ограничений для восстановления и архивирования:

- per-restore limit – максимальный объем полосы пропускания для чтения из архива резервной копии;
- per-storage write limit – максимальный объем полосы пропускания, используемый для записи в конкретное хранилище.

Ограничение чтения косвенно влияет на ограничение записи. Меньшее ограничение на задание перезапишет большее ограничение на хранилище. Увеличение лимита на задание приведёт к перезаписи лимита на хранилище, только если для данного хранилища есть разрешения «Data.Allocate».

Чтобы задать ограничение пропускной способности для конкретного задания восстановления, используется параметр `bwlimit`. В качестве единицы ограничения используется Кб/с, это означает, что значение 10240 ограничит скорость чтения резервной копии до 10 Мб/с, гарантируя, что остальная часть возможной пропускной способности хранилища будет доступна для уже работающих гостевых систем, и, таким образом, резервное копирование не повлияет на их работу.

Примечание. Чтобы отключить все ограничения для конкретного задания можно использовать значение `0` для параметра `bwlimit`. Это может быть полезно, если требуется как можно быстрее восстановить ВМ.

Установить ограничение пропускной способности по умолчанию для хранилища, можно с помощью команды:

```
# pvesm set STORAGEID --bwlimit restore=KIBs
```

4.13.5 Восстановление в реальном времени (Live-Restore)

Восстановление большой резервной копии может занять много времени, в течение которого гостевая система будет недоступна. Для резервных копий ВМ, хранящихся на сервере резервного копирования Proxmox (PBS), это время ожидания можно сократить с помощью параметра live-restore.

Включение live-restore с помощью отметки в веб-интерфейсе (Рис. 269) или указания параметра `live-restore` в команде `qmrestore` приводит к запуску ВМ сразу после начала восстановления. Данные копируются в фоновом режиме, отдавая приоритет фрагментам, к которым ВМ активно обращается.

Восстановление в реальном времени

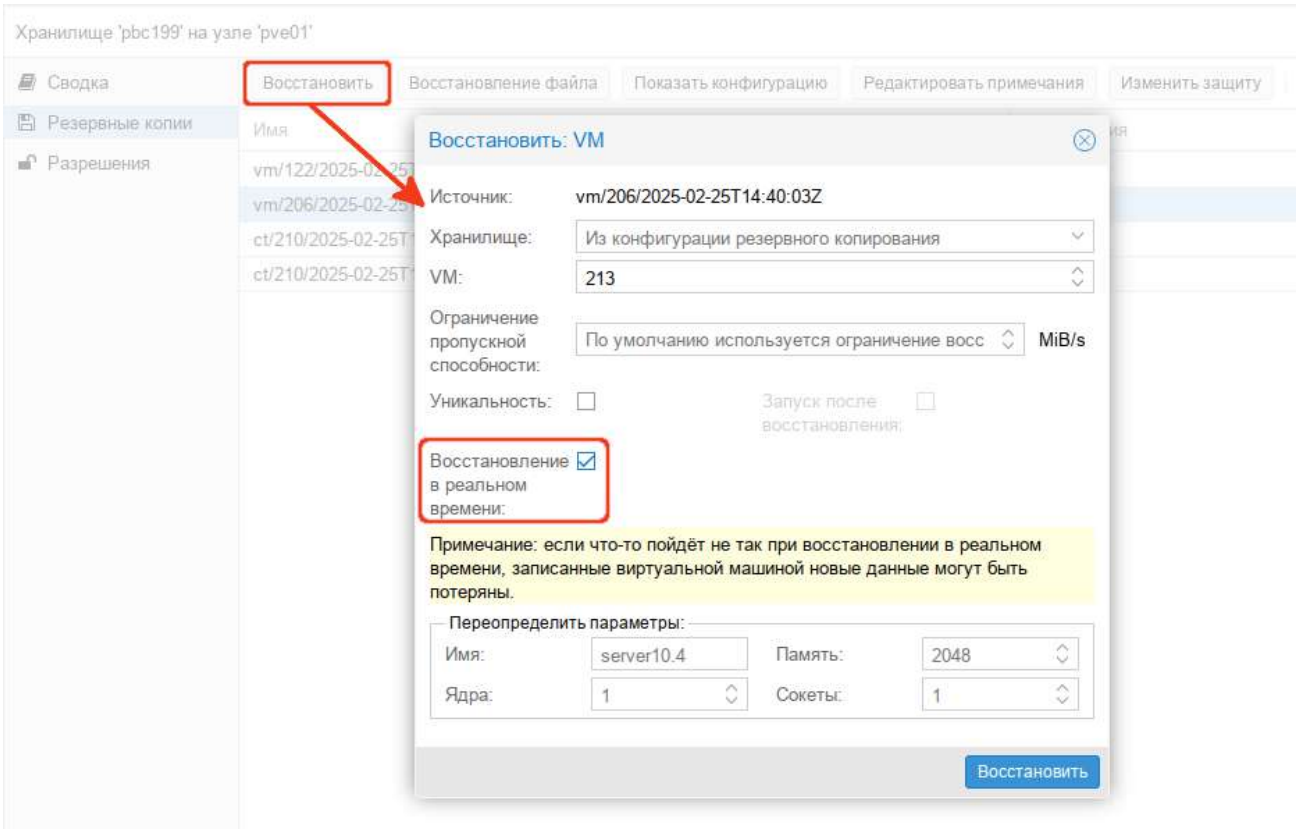


Рис. 269

При этом следует обратить внимание на следующее:

- во время live-restore ВМ будет работать с ограниченной скоростью чтения с диска, поскольку данные должны быть загружены с сервера резервного копирования (однако после загрузки они немедленно становятся доступны в целевом хранилище, поэтому двойной доступ к данным влечет за собой штраф только в первый раз). Скорость записи в основном не изменяется;
- если по какой-либо причине live-restore не удается, ВМ останется в неопределенном состоянии, т.к. не все данные могли быть скопированы из резервной копии, и, скорее всего, невозможно сохранить какие-либо данные, записанные во время неудачной операции восстановления.

Этот режим работы особенно полезен для больших ВМ, где для начальной работы требуется лишь небольшой объем данных, например, веб-серверов – после запуска ОС и необходимых служб ВМ становится работоспособной, в то время как фоновая задача продолжает копировать редко используемые данные.

4.13.6 Восстановление отдельных файлов

Кнопка «Восстановление файла» на вкладке «Резервные копии» хранилища PBS (Рис. 270) может использоваться для открытия браузера файлов непосредственно на данных, содержащихся

в резервной копии. Эта функция доступна только для резервных копий на сервере резервного копирования Proxmox (PBS).

Восстановление отдельных файлов

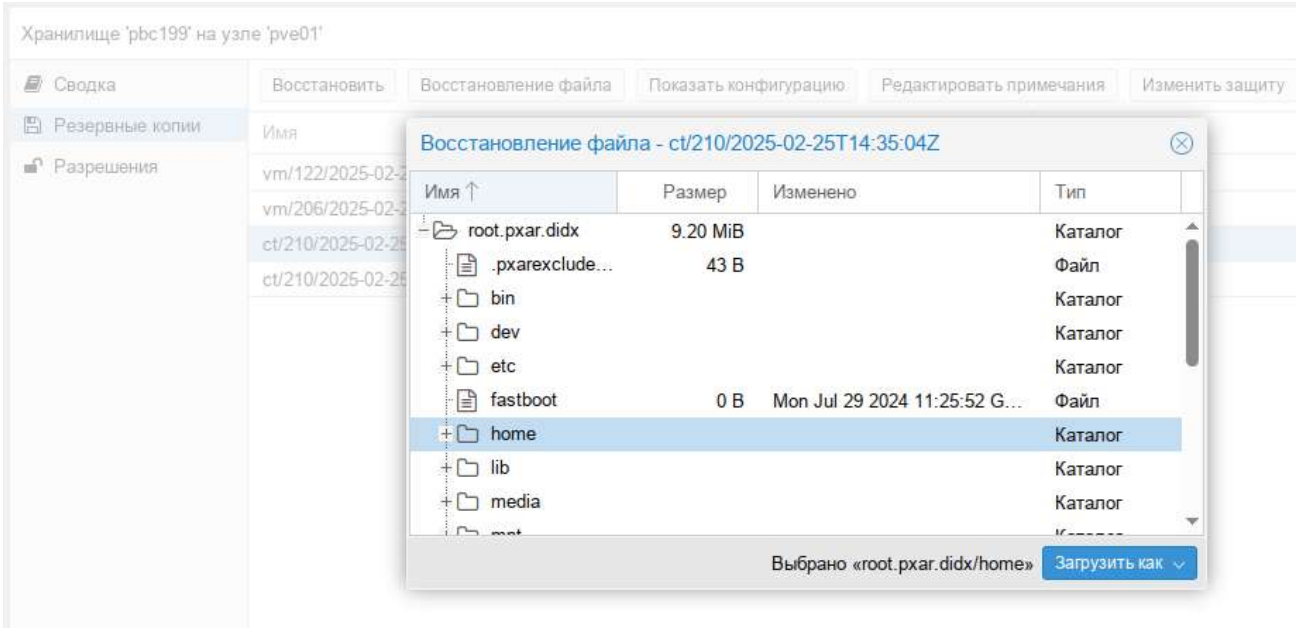


Рис. 270

Для контейнеров первый уровень дерева файлов показывает все включенные архивы pxar, которые можно открывать и просматривать. Для ВМ первый уровень показывает образы дисков. Следует обратить внимание, что для ВМ не все данные могут быть доступны (неподдерживаемые гостевые файловые системы, технологии хранения и т. д.).

Если нужно загрузить отдельный файл, необходимо его выбрать и нажать кнопку «Загрузить». Для загрузки каталога следует нажать кнопку «Загрузить как» и выбрать формат архива (zip или tar.zst).

Чтобы обеспечить безопасный доступ к образам ВМ, которые могут содержать ненадежные данные, запускается временная ВМ (не видимая как гостевая). Это не означает, что данные, загруженные из такого архива, по своей сути безопасны, но это позволяет избежать опасности для системы гипервизора. ВМ остановится сама по истечении времени ожидания. Весь этот процесс происходит прозрачно с точки зрения пользователя.

4.13.7 Файл конфигурация vzdump.conf

Глобальные настройки создания резервных копий хранятся в файле конфигурации `/etc/vzdump.conf`. Каждая строка файла имеет следующий формат (пустые строки в файле игнорируются, строки, начинающиеся с символа #, рассматриваются как комментарии и также игнорируются):

```
OPTION: value
```

Поддерживаемые опции представлены в табл. 22.

Т а б л и ц а 22 – Опции резервного копирования

Опция	Описание
bwlimit: integer (0 - N) (default=0)	Ограничение пропускной способности ввода/вывода (Кб/с)
compress: (0 | 1 | gzip | lzo | zstd) (default=0)	Сжатие файла резервной копии
dumpdir: string	Записать результирующие файлы в указанный каталог
exclude-path: string	Исключить определенные файлы/каталоги. Пути, начинающиеся с /, привязаны к корню контейнера, другие пути вычисляются относительно каждого подкаталога
ionice: integer (0 - 8) (default=7)	Установить CFQ приоритет ionice
lockwait: integer (0 - N) (default=180)	Максимальное время ожидания для глобальной блокировки (в минутах)
mailnotification: (always | failure) (default=always)	Указание, когда следует отправить отчет по электронной почте
mailto: string	Разделенный запятыми список адресов электронной почты, на которые будут приходить уведомления
maxfiles: integer (1 - N) (default=1)	Устарело: следует использовать вместо этого prune-backups. Максимальное количество файлов резервных копий ВМ
mode: (snapshot | stop | suspend) (default=snapshot)	Режим резервного копирования
notes-template: string	Строка шаблона для создания заметок для резервных копий. Может содержать переменные, которые будут заменены их значениями. В настоящее время поддерживаются следующие переменные {{cluster}}, {{guestname}}, {{node}} и {{vmid}}. Шаблон должен быть записан в одну строку, новая строка и обратная косая черта должны быть экранированы как \n и \\ соответственно
performance: [max-workers=<integer>] max-workers=<integer> (1 - 256) (default = 16)	Другие настройки, связанные с производительностью. max-workers – разрешить до этого количества рабочих процессов ввода-вывода одновременно (применяется к ВМ)
pigz: integer (default=0)	Использует pigz вместо gzip при N>0. N=1 использует половину ядер (uses half of cores), при N>1 N – количество потоков
pool: string	Резервное копирование всех известных гостевых систем, включенных в указанный пул
protected: boolean	Если true, пометить резервную копию как защищенную
prune-backups: [keep-all=<1|0>] [,keep-daily=<N>] [,keep-	Использовать эти параметры хранения вместо параметров из конфигурации хранилища (см.выше)

hourly=<N>] [,keep-last=<N>] [,keep-monthly=<N>] [,keep-weekly=<N>] [,keep-yearly=<N>]	
remove: boolean (default=1)	Удалить старые резервные копии, если их больше, чем установлено опцией prune-backups
script: string	Использовать указанный скрипт
stdexcludes: boolean (default=1)	Исключить временные файлы и файлы журналов
stopwait: integer (0 - N) (default=10)	Максимальное время ожидания пока гостевая система не остановится (минуты)
storage: string	Хранить полученный файл в этом хранилище
tmpdir: string	Хранить временные файлы в указанном каталоге
zstd: integer (default = 1)	Количество потоков zstd. N = 0 использовать половину доступных ядер, N> 0 использовать N как количество потоков

Пример vzdump.conf:

```
tmpdir: /mnt/fast_local_disk
storage: my_backup_storage
mode: snapshot
bwlimit: 10000
```

4.13.8 Скрипты-ловушки (hookscripts)

Скрипт-ловушку можно указать с помощью опции --script. Этот скрипт вызывается на различных этапах процесса резервного копирования с соответствующими параметрами (см. пример скрипта /usr/share/doc/pve-manager/examples/vzdump-hook-script.pl).

4.13.9 Файлы, не включаемые в резервную копию

Примечание. Эта опция доступна только при создании резервных копий контейнеров.

Команда vzdump по умолчанию пропускает следующие файлы (отключается с помощью опции --stdexcludes 0):

```
/tmp/?*
/var/tmp/?*
/var/run/?*pid
```

Кроме того, можно вручную указать какие файлы исключать (дополнительно), например:

```
# vzdump 777 --exclude-path /tmp/ --exclude-path '/var/foo*'
```

Если путь не начинается с символа «/», то он не будет привязан к корню контейнера и будет соответствовать любому подкаталогу. Например:

```
# vzdump 777 --exclude-path bar
```

исключает любые файлы и каталоги с именами /bar, /var/bar, /var/foo/bar и т.д.

Файлы конфигурации ВМ и контейнеров также хранятся внутри архива резервных копий (в /etc/vzdump/) и будут корректно восстановлены.

4.13.10 Примеры создания резервных копий в командной строке

Создать простую резервную копию гостевой системы 103 – без снимков, только архив гостевой части и конфигурационного файла в каталог резервного копирования по умолчанию (обычно `/var/lib/vz/dump/`):

```
# vzdump 103
```

Использовать `rsync` и режим приостановки для создания снимка (минимальное время простоя):

```
# vzdump 103 --mode suspend
```

Сделать резервную копию всей гостевой системы и отправить отчет пользователям root и admin:

```
# vzdump --all --mode suspend --mailto root --mailto admin
```

Использовать режим мгновенного снимка (снапшота) (нет времени простоя) и каталог для хранения резервных копий `/mnt/backup`:

```
# vzdump 103 --dumpdir /mnt/backup --mode snapshot
```

Резервное копирование более чем одной ВМ (выборочно):

```
# vzdump 101 102 103 --mailto root
```

Резервное копирование всех ВМ, исключая 101 и 102:

```
# vzdump --mode suspend --exclude 101,102
```

Восстановить контейнер в новый контейнер 600:

```
# pct restore 600 /mnt/backup/vzdump-lxc-777.tar
```

Восстановить QemuServer VM в ВМ 601:

```
# qmrestore /mnt/backup/vzdump-qemu-888.vma 601
```

Клонировать существующий контейнер 101 в новый контейнер 300 с 4GB корневой файловой системы:

```
# vzdump 101 --stdout | pct restore --rootfs 4 300 -
```

4.14 Снимки (snapshot)

Снимки ВМ – это файловые снимки состояния, данных диска и конфигурации ВМ в определенный момент времени. Можно создать несколько снимков ВМ даже во время ее работы. Затем можно возвратить ее в любое из предыдущих состояний, применив моментальный снимок к ВМ.

Чтобы создать снимок состояния системы, необходимо в меню ВМ выбрать пункт «Снимки» и нажать кнопку «Сделать снимок» (Рис. 271). В открывшемся окне (Рис. 272) следует ввести название снимка и нажать кнопку «Сделать снимок».

Окно управления снимками ВМ

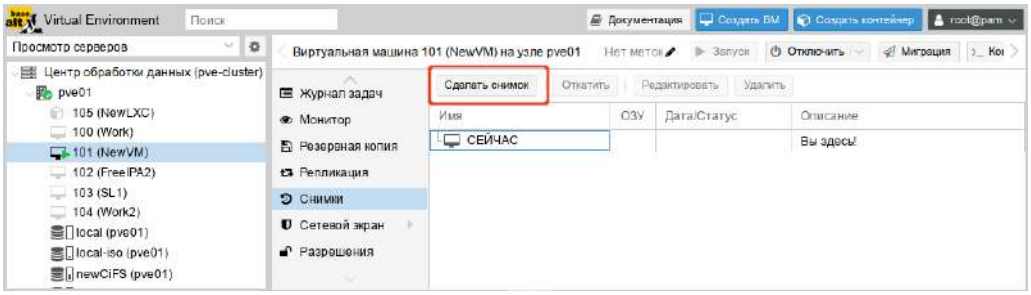


Рис. 271

Создание снимка ВМ

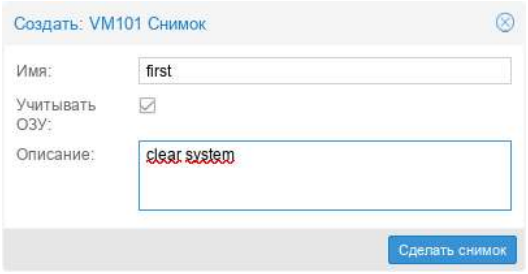


Рис. 272

Для того чтобы восстановить ВМ из снимка, необходимо в меню ВМ выбрать пункт «Снимки», выбрать снимок (Рис. 273) и нажать кнопку «Откатить».

Восстановление ОС из снимка

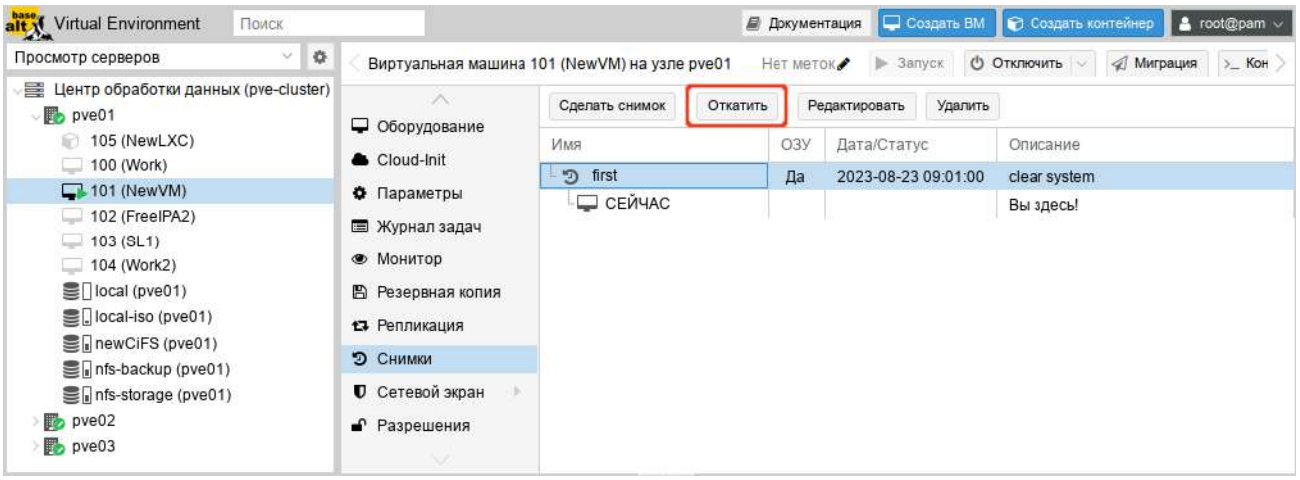


Рис. 273

При создании снимков, `qm` сохраняет конфигурацию ВМ во время снимка в отдельном разделе в файле конфигурации ВМ. Например, после создания снимка с именем `first` файл конфигурации будет выглядеть следующим образом:

```
boot: order=scsi0;sata2;net0
cores: 1
memory: 1024
meta: creation-qemu=7.1.0,ctime=1671708251
name: NewVM
```

```
net0: virtio=3E:E9:24:FF:85:D9,bridge=vmbr0,firewall=1
numa: 0
ostype: l26
parent: first
sata2: local-iso:iso/slinux-10.1-x86_64.iso,media=cdrom,size=4586146K
scsi0: local:100/vm-100-disk-0.qcow2,size=42G
scsihw: virtio-scsi-pci
smbios1: uuid=ee9db068-5427-4934-bf7a-5895c377b5af
sockets: 1
vmgenid: dfec8e3b-d391-40cb-8983-b4938461b79a

[first]
#clear system
boot: order=scsi0;sata2;net0
cores: 1
memory: 1024
meta: creation-qemu=7.1.0,ctime=1671708251
name: NewVM
net0: virtio=3E:E9:24:FF:85:D9,bridge=vmbr0,firewall=1
numa: 0
ostype: l26
runningcpu: kvm64,enforce,+kvm_pv_eoi,+kvm_pv_unhalt,+lahf_lm,+sep
runningmachine: pc-i440fx-7.1+pve0
sata2: local-iso:iso/slinux-10.1-x86_64.iso,media=cdrom,size=4586146K
scsi0: local:100/vm-100-disk-0.qcow2,size=42G
scsihw: virtio-scsi-pci
smbios1: uuid=ee9db068-5427-4934-bf7a-5895c377b5af
snaptime: 1671724448
sockets: 1
vmgenid: dfec8e3b-d391-40cb-8983-b4938461b79a
vmstate: local:100/vm-100-state-first.raw
```

Свойство `parent` используется для хранения родительских/дочерних отношений между снимками, `snaptime` – это отметка времени создания снимка (эпоха Unix).

4.15 Встроенный мониторинг PVE

Все данные о потреблении ресурсов и производительности можно найти на вкладках «Сводка» узлов PVE и ВМ. Можно просматривать данные на основе почасового, ежедневного, еженедельного или за год периодов.

На Рис. 274 показана «Сводка» узла pve01 со списком для выбора периода данных.

Просмотреть список всех узлов, ВМ и контейнеров в кластере можно, выбрав «Центр обработки данных» → «Поиск» **(Рис. 275**). В этом списке отображается потребление ресурсов только в реальном масштабе времени. К полям, отображаемым по умолчанию, можно добавить дополнительные поля (Рис. 276) и указать порядок сортировки дерева ресурсов (например, сортировать по имени ВМ).

Выбор периода данных, для отображения отчета

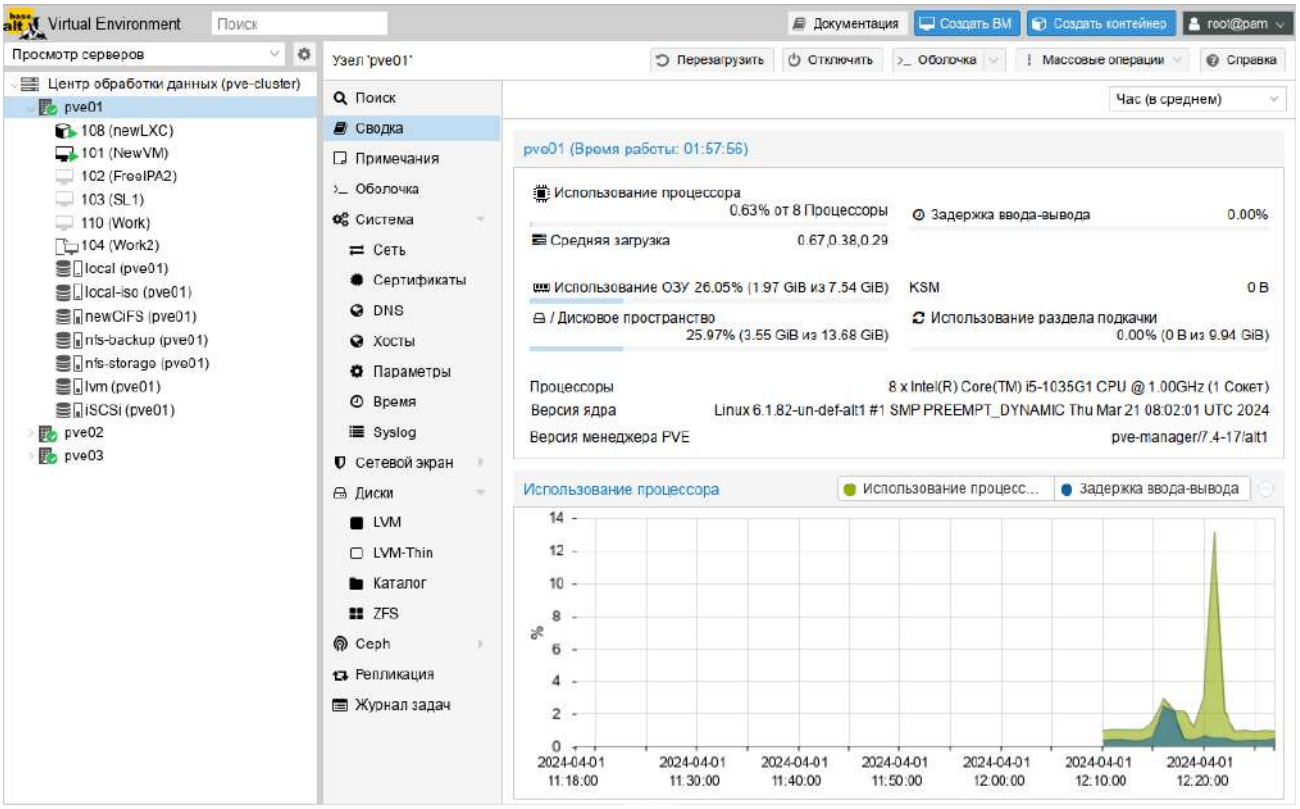


Рис. 274

Потребление ресурсов

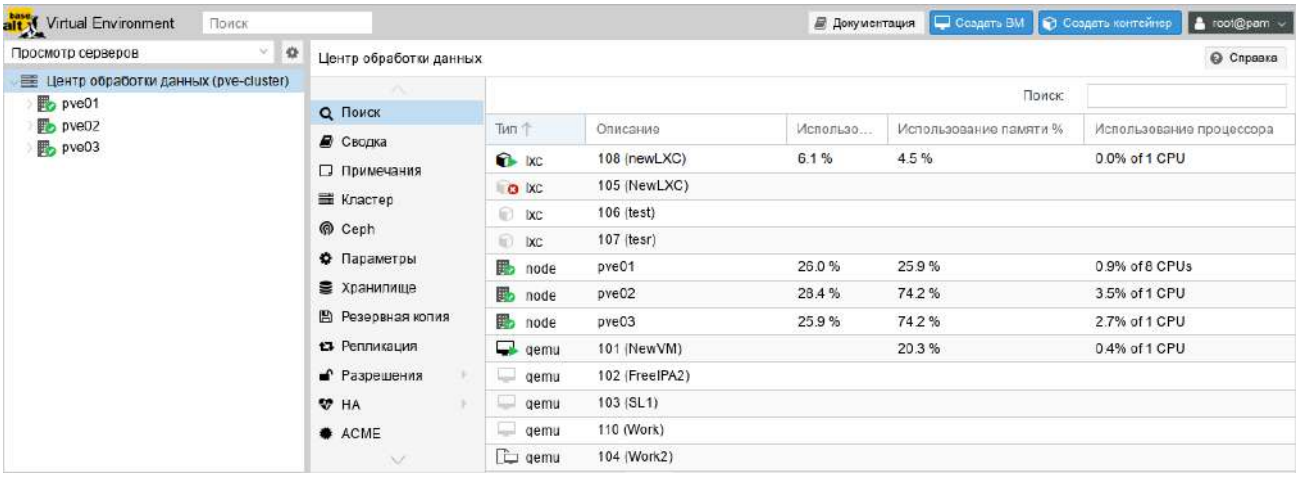


Рис. 275

Выбор отображаемых полей

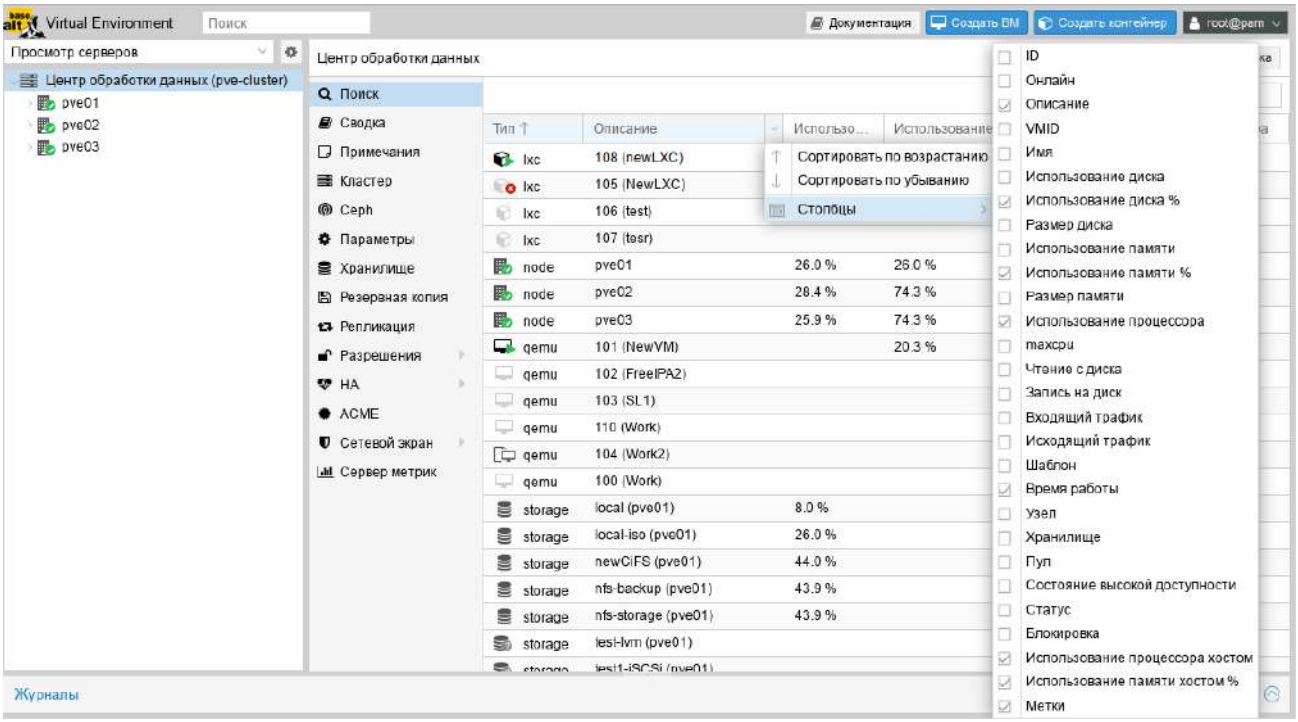


Рис. 276

Для мониторинга состояния локальных дисков используется пакет `smartmontools`. Он содержит набор инструментов для мониторинга и управления S.M.A.R.T. системой для локальных жестких дисков.

Получить статус диска можно, выполнив следующую команду:

```
# smartctl -a /dev/sdX
```

где /dev/sdX – это путь к одному из локальных дисков.

Включить поддержку SMART для диска, если она отключена:

```
# smartctl -s on /dev/sdX
```

Просмотреть S.M.A.R.T. статус диска в веб-интерфейсе можно, выбрав в разделе «Диски» нужный диск и нажав кнопку «Показать данные S.M.A.R.T.» (Рис. 277).

Кнопка «Показать данные S.M.A.R.T.»

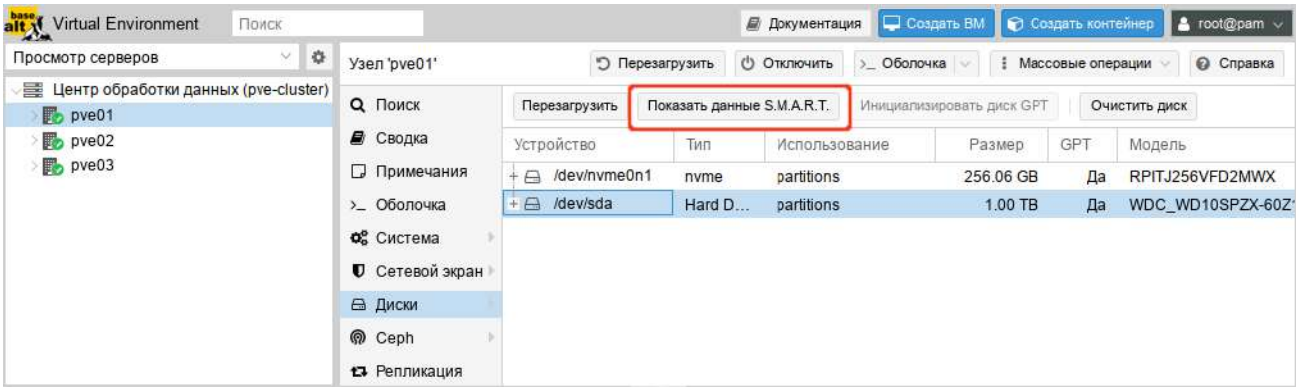


Рис. 277

По умолчанию `smartmontools daemon smartd` активен и включен, и сканирует диски в `/dev` каждые 30 минут на наличие ошибок и предупреждений, а также отправляет сообщение электронной почты пользователю root в случае обнаружения проблемы (для пользователя root в PVE должен быть введен действительный адрес электронной почты).

Электронное сообщение будет содержать имя узла, где возникла проблема, а также параметры самого устройства, такие как серийный номер и идентификатор дискового устройства. Если та же самая ошибка продолжит возникать, узел будет отсылать электронное сообщение каждые 24 часа. Основываясь на содержащейся в электронном сообщении информации можно определить отказавшее устройство и заменить его в случае такой необходимости.

4.16 Высокая доступность PVE

Высокая доступность PVE (High Availability, HA) позволяет кластеру перемещать или мигрировать ВМ с отказавшего узла на жизнеспособный узел без вмешательства пользователя.

Для функционирования HA в PVE необходимо чтобы все ВМ использовали общее хранилище. HA PVE обрабатывает только узлы PVE и ВМ в пределах кластера PVE. Такую функциональность HA не следует путать с избыточностью общих хранилищ, которую PVE может применять в своем развертывании HA. Общие хранилища сторонних производителей могут предоставлять свою собственную функциональность HA.

В вычислительном узле PVE могут существовать свои уровни избыточности, например, применение RAID, дополнительные источники питания, объединение/агрегация сетей. HA в PVE не подменяет собой ни один из этих уровней, а просто способствует использованию функций избыточности ВМ для сохранения их в рабочем состоянии при отказе какого-либо узла.

4.16.1 Как работает высокая доступность PVE

PVE предоставляет программный стек ha-manager, который может автоматически обнаруживать ошибки и выполнять автоматический переход на другой ресурс. Основной блок управления, управляемый ha-manager называется ресурсом. Ресурс (сервис) однозначно идентифицируется идентификатором сервиса (SID), который состоит из типа ресурса и идентификатора, специфичного для данного типа, например, vm: 100 (ресурс типа ВМ с идентификатором 100).

В случае, когда по какой-либо причине узел становится недоступным, HA PVE ожидает 60 секунд прежде чем выполнить ограждение (fencing) отказавшего узла. Ограждение предотвращает службы кластера от возврата в рабочее состояние в этом месте. Затем HA перемещает ВМ и контейнеры на следующий доступный узел в группе участников HA. Даже если узел с ВМ включен, но потерял связь с сетевой средой, HA PVE попытается переместить все ВМ с этого узла на другой узел.

При возврате отказавшего узла в рабочее состояние, HA не переместит ВМ на первоначальный узел. Это необходимо выполнять вручную. При этом ВМ может быть перемещена вручную только если HA запрещен для данной ВМ. Поэтому сначала следует выключить HA, а затем переместить на первоначальный узел и включить HA на данной ВМ вновь.

4.16.2 Требования для настройки высокой доступности

Среда PVE для настройки HA должна отвечать следующим требованиям:

- кластер, содержащий, как минимум, три узла (для получения надежного кворума);
- общее хранилище для ВМ и контейнеров;
- аппаратное резервирование;
- использование надежных «серверных» компонентов;
- аппаратный сторожевой таймер (если он недоступен, используется программный таймер ядра Linux);
- дополнительные устройства ограждения (fencing).

Примечание. В случае построения виртуальной инфраструктуры на серверах HP необходимо запретить загрузку модуля ядра hpwdt. Для этого необходимо создать файл `/etc/modprobe.d/nohpwdt.conf` со следующим содержимым (для применения изменений следует перезагрузить систему):

```
# Do not load the 'hpwdt' module on boot.
blacklist hpwdt
```

4.16.3 Настройка высокой доступности PVE

Все настройки HA PVE могут быть выполнены в веб-интерфейсе в разделе «Центр обработки данных» → «HA» (Рис. 278).

Меню HA. Статус настройки HA

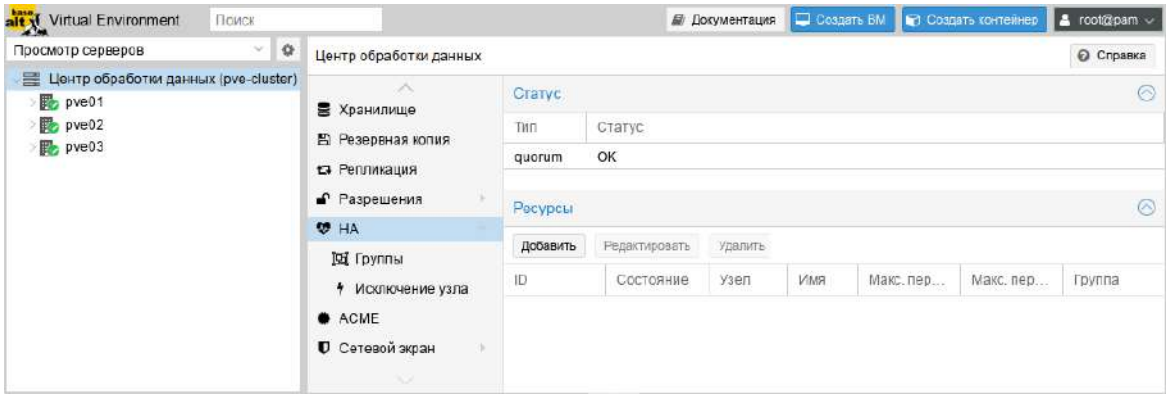


Рис. 278

4.16.3.1 Создание группы высокой доступности

Наиболее характерным примером использования групп HA являются некие программные решения или инфраструктура ВМ, которые должны работать совместно (например, контроллер

домена, файловый сервер и т.д.). Назначенные в определенную группу ВМ могут перемещаться только между узлами участниками этой группы. Например, есть шесть узлов, три из которых обладают всей полнотой ресурсов, достаточной для исполнения виртуального сервера базы данных, а другие три узла выполняют виртуальные рабочие столы или решения VDI. Можно создать две группы, для которых виртуальные серверы баз данных могут перемещаться только в пределах тех узлов, которые будут назначены для данной группы. Это гарантирует, что ВМ переместится на тот узел, который будет способен исполнять такие ВМ.

Для включения HA необходимо создать как минимум одну группу.

Для создания группы следует нажать кнопку «Создать» в подменю «Группы».

Элементы, доступные в блоке диалога «Группа высокой доступности» (Рис. 279):

- «ID» – название HA группы;
- «Узел» – назначение узлов в создаваемую группу (нужно выбрать, по крайней мере, один узел);
- «restricted» – разрешение перемещения ВМ со стороны HA PVE только в рамках узлов участников данной группы HA. Если перемещать ВМ некуда, то эти ВМ будут автоматически остановлены;
- «nofailback» – используется для предотвращения автоматического восстановления состояния ВМ/контейнера при восстановлении узла в кластере (не рекомендуется включать эту опцию).

Диалог создания группы

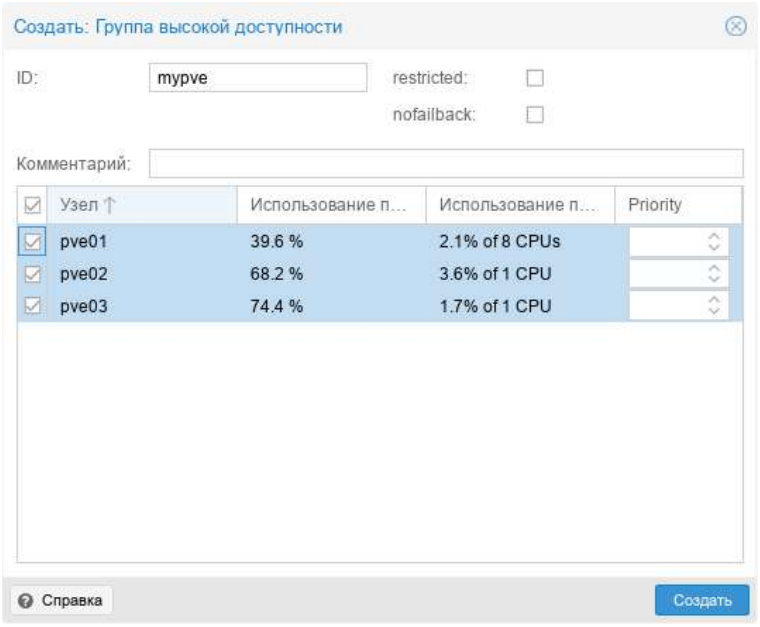


Рис. 279

На Рис. 280 представлено подменю «Группы» с созданной группой.

Подменю «Группы» с созданной группой

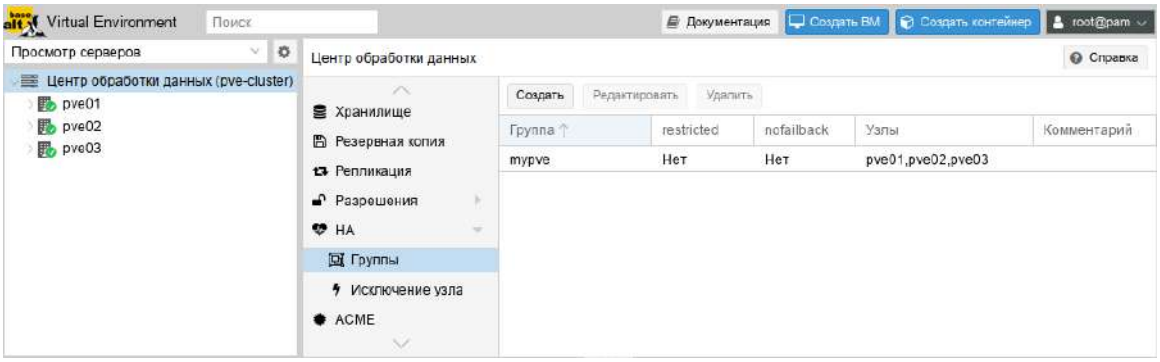


Рис. 280

4.16.3.2 Добавление ресурсов

Для включения HA для ВМ или контейнера следует нажать на кнопку «Добавить» в разделе «Ресурсы» меню «HA». В открывшемся диалоговом окне нужно выбрать ВМ/контейнер и группу HA (Рис. 281).

Добавление ресурса в группу

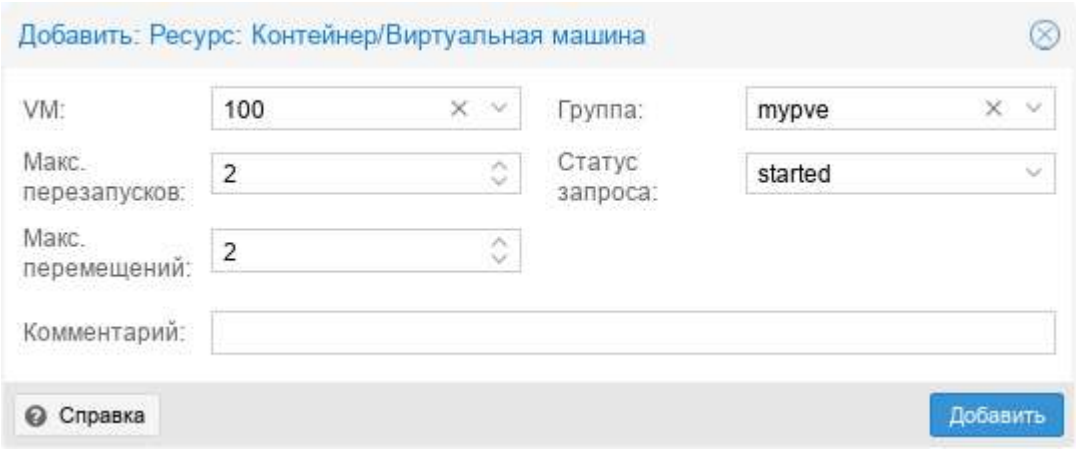


Рис. 281

В окне можно настроить следующие параметры:

- «Макс. перезапусков» – количество попыток запуска ВМ/контейнера на новом узле после перемещения;
- «Макс. перемещений» – количество попыток перемещения ВМ/контейнера на новый узел;
- «Статус запроса» – доступны варианты: «started» – кластер менеджер будет пытаться поддерживать состояние машины в запущенном состоянии; «stopped» – при отказе узла перемещать ресурс, но не пытаться запустить; «ignored» – ресурс, который не надо перемещать при отказе узла; «disabled» – в этот статус переходят ВМ, которые находятся в состоянии «error».

На Рис. 282 показана группа HA PVE и добавленные в нее ВМ и контейнеры, которыми будет управлять HA.

Раздел «Статус» отображает текущее состояние функциональности HA:

- кворум кластера установлен;
- главный узел pve01 группы HA активен и последний временной штамп жизнеспособности (heartbeat timestamp) проверен;
- все узлы, участвующие в группе HA активны и последний временной штамп жизнеспособности (heartbeat timestamp) проверен.

Список ресурсов

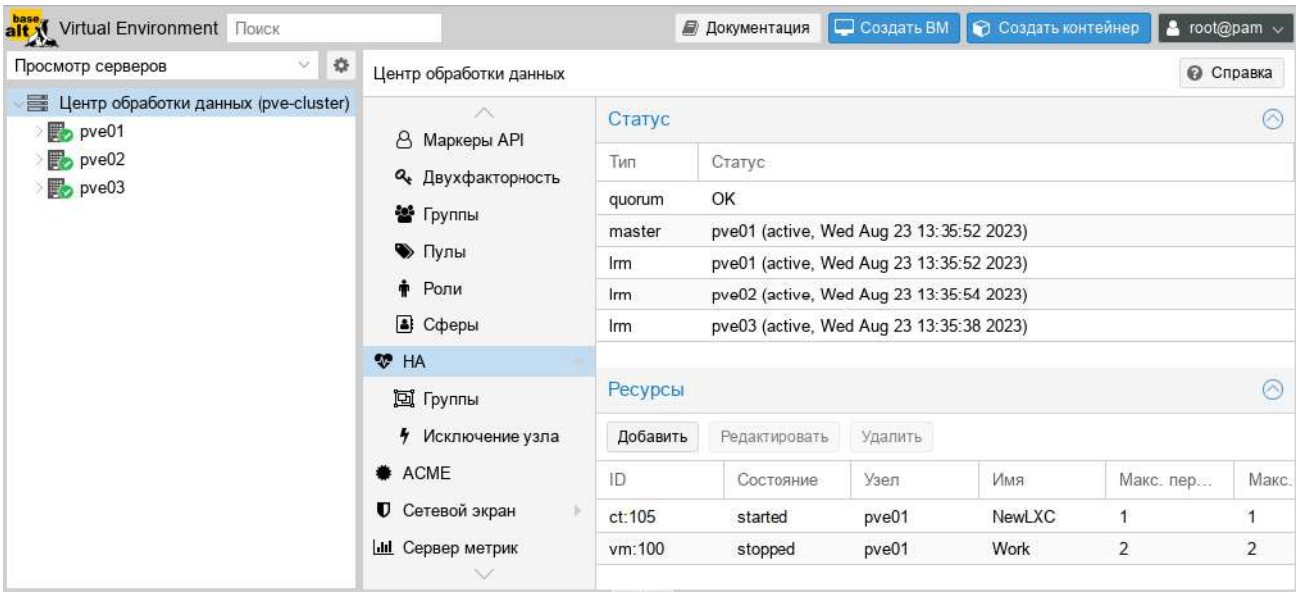


Рис. 282

Просмотреть состояние функциональности HA можно и в консоли:

```
# ha-manager status
quorum OK
master pve01 (active, Wed Aug 23 13:26:31 2023)
lrm pve01 (active, Wed Aug 23 13:26:31 2023)
lrm pve02 (active, Wed Aug 23 13:26:33 2023)
lrm pve03 (active, Wed Aug 23 13:26:26 2023)
service ct:105 (pve01, started)
service vm:100 (pve01, stopped)
```

4.16.4 Тестирование настройки высокой доступности PVE

Для того чтобы убедиться, что HA действительно работает, можно отключить сетевое соединение для pve01 и понаблюдать за окном «Статус» (Рис. 282) на предмет изменений HA.

После того как соединение с узлом pve01 будет потеряно, он будет помечен как недоступный. По истечению 60 секунд, HA PVE предоставит следующий доступный в группе HA узел в качестве главного (Рис. 283).

После того как HA PVE предоставит новый ведущий узел для группы HA, будет запущено ограждение для ресурсов ВМ/контейнера для подготовки к перемещению их на другой узел. В

процессе ограждения, все связанные с данной ВМ службы ограждаются, что означает, что даже если отказавший узел вернется в строй на этом этапе, ВМ не смогут восстановить свою нормальную работу. Затем ВМ/контейнер полностью останавливается. Так как узел сам по себе отключен, ВМ/контейнер не может выполнить миграцию в реальном режиме времени, поскольку состояние оперативной памяти исполняемой ВМ не может быть получено с отключенного узла.

Изменение главного узла на pve02

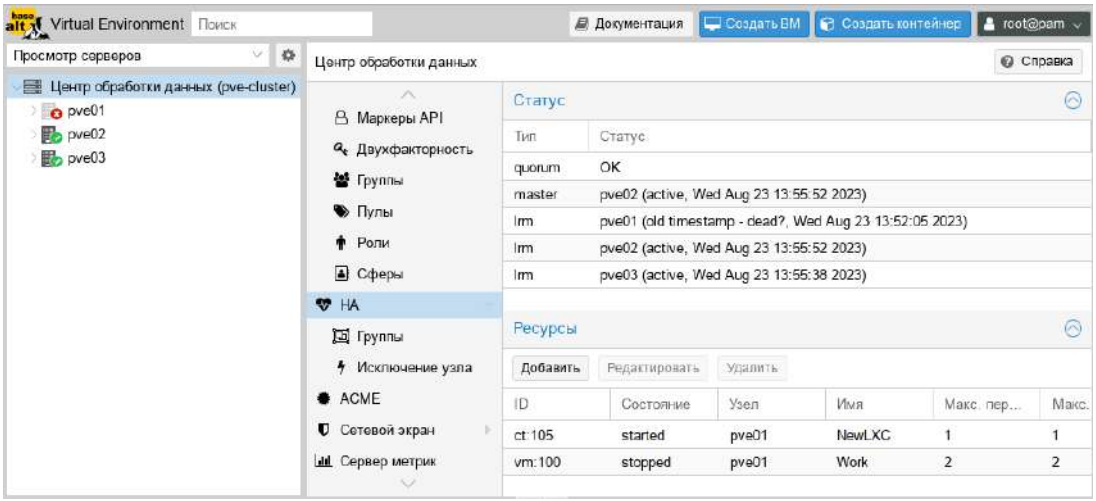


Рис. 283

После остановки, ВМ/контейнер перемещается на следующий свободный узел в группе HA и автоматически запускается. В данном примере контейнер 105 перемещен на узел pve02 и запущен (Рис. 284).

Контейнер 105 запущен на узле pve02

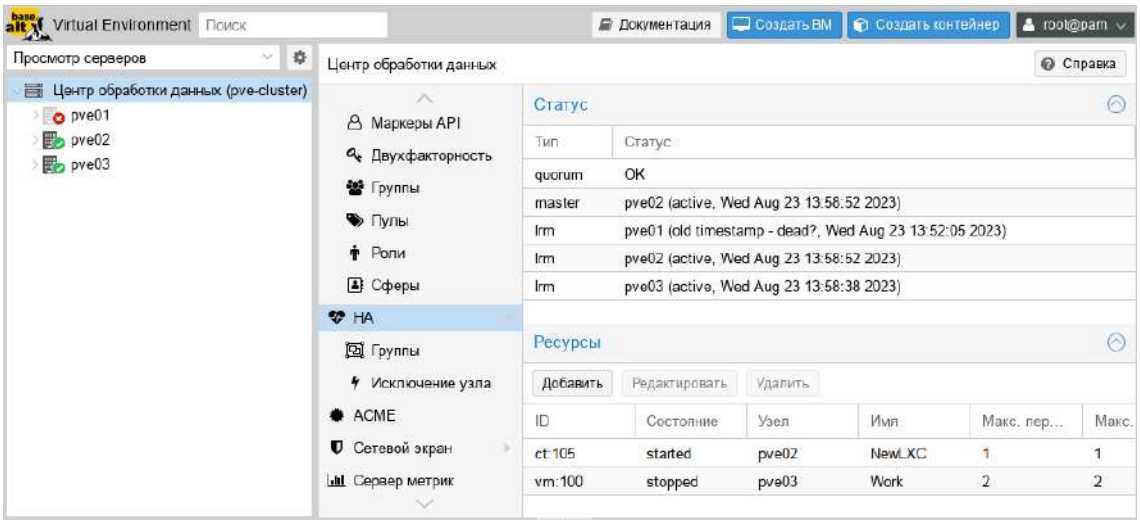


Рис. 284

В случае возникновения любой ошибки, HA PVE выполнит несколько попыток восстановления в соответствии с политиками `restart` и `relocate`. Если все попытки окажутся неудачными, HA PVE поместит ресурсы в ошибочное состояние и не будет выполнять для них никаких задач.

4.17 Межсетевой экран PVE (firewall)

Межсетевой экран PVE обеспечивает простой способ защиты ИТ-инфраструктуры. Можно настроить правила межсетевого экрана для всех узлов внутри кластера или определить правила для ВМ и контейнеров.

Хотя вся конфигурация хранится в файловой системе кластера, служба межсетевого экрана на основе iptables работает на каждом узле кластера и, таким образом, обеспечивает полную изоляцию между ВМ. Распределенная природа этой системы также обеспечивает гораздо более высокую пропускную способность, чем центральное решение межсетевого экрана.

Межсетевой экран поддерживает протоколы IPv4 и IPv6. По умолчанию фильтруется трафик для обоих протоколов, поэтому нет необходимости поддерживать другой набор правил для IPv6.

4.17.1 Зоны

Межсетевой экран PVE группирует сеть в следующие логические зоны:

- Узел – трафик из/в узел кластера;
- ВМ – трафик из/в определенную ВМ.

Для каждой зоны можно определить правила межсетевого экрана для входящего и/или исходящего трафика.

4.17.2 Файлы конфигурации

Вся конфигурация, связанная с межсетевым экраном, хранится в файловой системе кластера. Поэтому эти файлы автоматически распространяются на все узлы кластера, а служба pve-firewall при изменениях автоматически обновляет базовые правила iptables.

Управление правилами осуществляется через веб-интерфейс PVE (например, «Центр обработки данных»→«Сетевой экран» или «Узел»→«Сетевой экран») или через конфигурационные файлы (/etc/pve/firewall/).

Файлы конфигурации брандмауэра содержат разделы пар ключ-значение. Строки, начинающиеся с символа #, и пустые строки считаются комментариями. Разделы начинаются со строки заголовка, которая содержит имя раздела, заключенное в квадратные скобки.

4.17.2.1 Настройка кластера

Файл `/etc/pve/firewall/cluster.fw` используется для хранения конфигурации PVE Firewall на уровне всего кластера. Этот файл содержит глобальные правила и параметры, которые применяются ко всем узлам и ВМ в кластере. Файл автоматически синхронизируется между всеми узлами кластера через PVE Cluster File System (pmxcfs).

Файл `/etc/pve/firewall/cluster.fw` состоит из нескольких секций, каждая из которых отвечает за определённые аспекты конфигурации firewall. В файле содержатся следующие секции:

- [OPTIONS] – используется для установки параметров межсетевого экрана;
- [RULES] – правила межсетевого экрана;
- [IPSET <имя_набора>] – определения набора IP-адресов;
- [GROUP <имя_группы>] – определения групп безопасности;
- [ALIASES] – определения псевдонимов.

Опции секции [OPTIONS] файла cluster.fw приведены в табл. Таблица 23.

Т а б л и ц а 23 – Опции секции [OPTIONS] файла cluster.fw

Опция	Описание
ebtables: <1|0> (по умолчанию = 1)	Включить правила ebtables для всего кластера
enable: <1|0>	Включить или отключить межсетевой экран для всего кластера
log_ratelimit: [enable=]<1|0> [,burst=<integer>] [,rate=<rate>]	Настройки ограничения частоты записи логов (rate limiting) в PVE Firewall. - burst=<integer> (0 – N) (по умолчанию = 5) – максимальное количество логов, которые могут быть записаны за один раз (пиковое значение); - enable=<1|0> (по умолчанию = 1) – включить или отключить ограничение частоты записи логов; - rate=<rate> (по умолчанию = 1/second) – средняя скорость записи логов. Формат: <число>/<интервал> (например, 1/second, 5/minute).
policy_in: <ACCEPT | DROP | REJECT>	Политика по умолчанию для входящего трафика. Возможные значения: - ACCEPT – разрешить; - DROP – отбросить; - REJECT – отклонить с отправкой уведомления.
policy_out: <ACCEPT | DROP | REJECT>	Политика по умолчанию для исходящего трафика (аналогично policy_in)

Параметры межсетевого экрана кластера можно настроить в веб-интерфейсе «Центр обработки данных»→«Сетевой экран» → «Параметры» (Рис. 285).

Параметры межсетевого экрана кластера

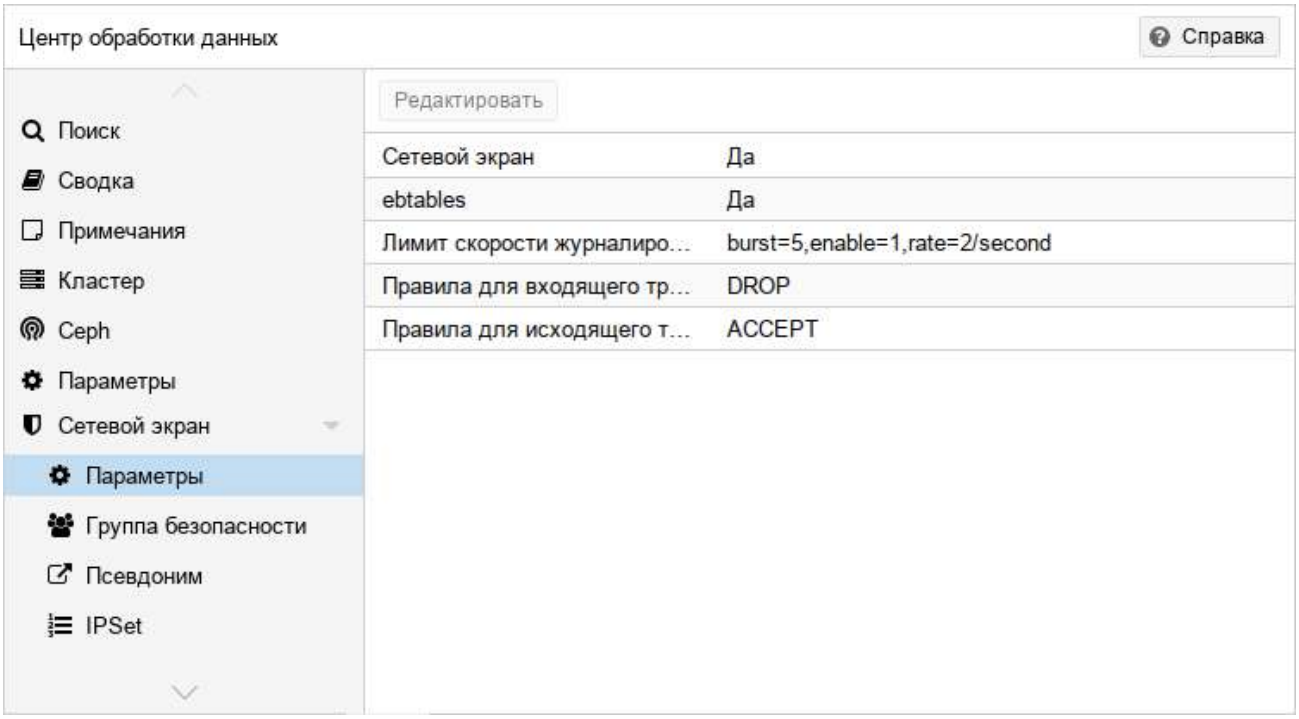


Рис. 285

По умолчанию межсетевой экран отключен. Включить межсетевой экран можно, установив опцию `enable` в файле `/etc/pve/firewall/cluster.fw`:

```
[OPTIONS]
# enable firewall (настройка для всего кластера, по умолчанию отключено)
enable: 1
```

Примечание. При включении межсетевого экрана трафик ко всем узлам будет заблокирован по умолчанию. Исключениями являются только WebGUI (порт 8006) и SSH (порт 22) из локальной сети.

Чтобы администрировать узлы PVE удаленно, нужно создать правила, разрешающие трафик с этих удаленных IP-адресов в веб-интерфейс (порт 8006). Можно также разрешить SSH (порт 22) и, возможно, SPICE (порт 3128).

Примечание. Перед включением межсетевого экрана можно создать SSH-подключение к одному из узлов PVE. В этом случает, если что-то пойдет не так, доступ к узлу сохранится.

Чтобы упростить задачу удалённого администрирования, можно создать IPSet под названием «management» и добавить туда все удаленные IP-адреса. При этом будут созданы все необходимые правила межсетевого экрана для доступа к GUI из удаленного режима.

4.17.2.2 Конфигурация узла

Конфигурация, связанная с узлом, считывается из файла `/etc/pve/nodes/<node-name>/host.fw`. Здесь можно перезаписать правила из конфигурации `cluster.fw` или увеличить уровень детализации журнала и задать параметры, связанные с `netfilter`.

В файле `host.fw` содержатся следующие секции:

- [OPTIONS] – используется для настройки параметров межсетевого экрана, связанных с узлом;
- [RULES] – правила межсетевого экрана, спецефичные для узла.

Опции секции [OPTIONS] конфигурации узла приведены в табл. Таблица 24.

Т а б л и ц а 24 – Опции секции [OPTIONS] конфигурации узла

Опция	Описание
enable: <1|0>	Включить или отключить межсетевой экран узла
log_level_in: <alert | crit | debug | emerg | err | info | nolog | notice | warning>	Уровень журнала для входящего трафика. Возможные значения: - alert – логировать важные события; - crit – логировать критические события; - debug – логировать всё (для отладки); - emerg – логировать только аварийные события; - err – логировать ошибки; - info – логировать информационные сообщения; - nolog – не логировать; - notice – логировать уведомления; - warning – логировать предупреждения.
log_level_out: <alert | crit | debug | emerg | err | info | nolog | notice | warning>	Уровень журнала для исходящего трафика (аналогично log_level_in)
log_nf_conntrack: <1|0> (по умолчанию = 0)	Включить регистрацию информации о conntrack
ndp: <1|0> (по умолчанию = 0)	Включить NDP (протокол обнаружения соседей)
nf_conntrack_allow_invalid: <1|0> (по умолчанию = 0)	Разрешить недействительные пакеты при отслеживании соединения
nf_conntrack_helpers: <string> (по умолчанию = ``)	Включить conntrack helpers для определенных протоколов. Поддерживаемые протоколы: amanda, ftp, irc, netbios-ns, pptp, sane, sip, snmp, tftp
nf_conntrack_max: <integer> (32768 – N) (по умолчанию = 262144)	Максимальное количество отслеживаемых соединений
nf_conntrack_tcp_timeout_installed: <integer> (7875 – N) (по умолчанию = 432000)	Тайм-аут, установленный для conntrack
nf_conntrack_tcp_timeout_syn_recv: <integer> (30 – 60) (по умолчанию = 60)	Тайм-аут syn recv conntrack
nosmurfs: <1|0>	Включить фильтр SMURFS
protection_synflood: <1|0> (по умолчанию = 0)	Включить защиту от synflood
protection_synflood_burst: <integer> (по умолчанию = 1000)	Уровень защиты от Synflood rate burst по IP-адресу источника
protection_synflood_rate: <integer> (по умолчанию = 200)	Скорость защиты Synflood syn/sec по IP-адресу источника
smurf_log_level: <alert | crit | debug | emerg | err | info | nolog | notification | warning>	Уровень журнала для фильтра SMURFS
tcp_flags_log_level: <alert | crit | debug |	Уровень журнала для фильтра нелегальных флагов TCP

emerg | err | info | nolog | notification | warning>	
tcpflags: <1|0> (по умолчанию = 0)	Фильтрация недопустимых комбинаций флагов TCP

Параметры межсетевого экрана узла можно настроить в веб-интерфейсе «Узел»→«Сетевой экран» → «Параметры» (Рис. 286).

Параметры межсетевого экрана узла

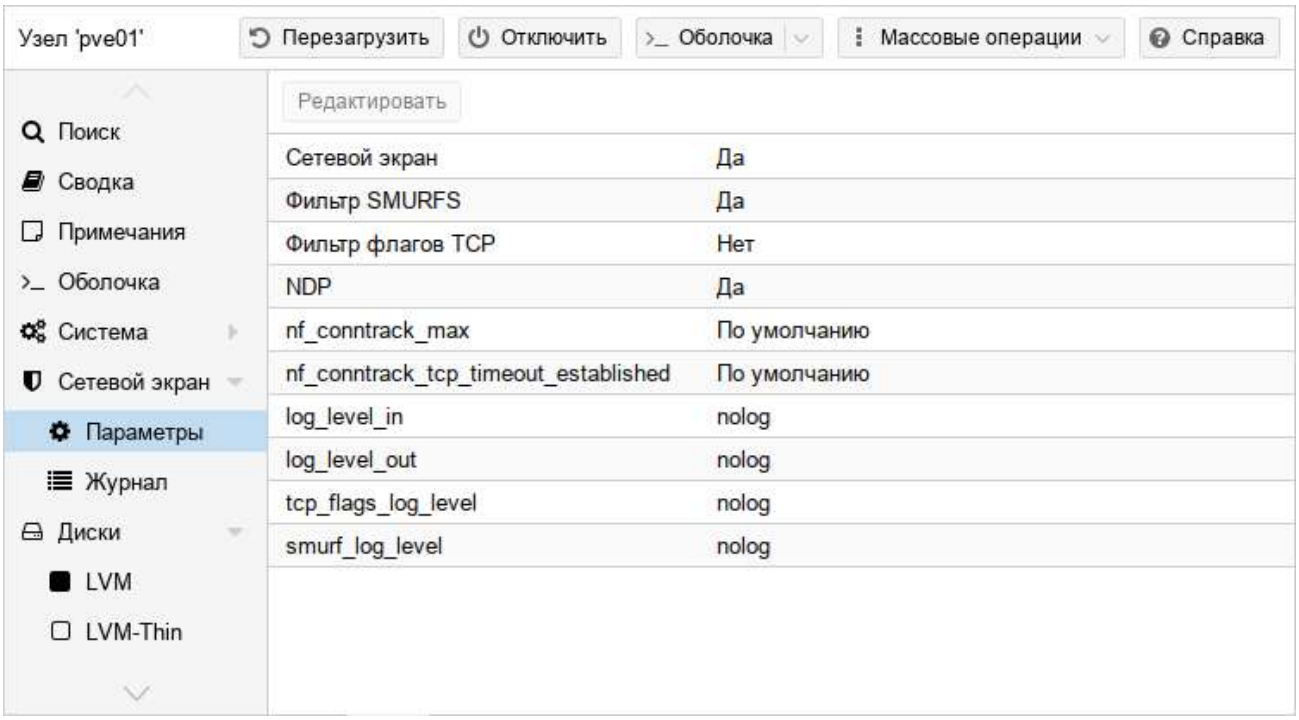


Рис. 286

4.17.2.3 Конфигурация ВМ/контейнера

Конфигурация межсетевого экрана ВМ считывается из файла /etc/pve/firewall/<VMID>.fw. Этот файл используется для установки параметров межсетевого экрана, связанных с ВМ/контейнером.

В файле содержатся следующие секции:

- [OPTIONS] – параметры межсетевого экрана ВМ/контейнера;
- [RULES] – правила межсетевого экрана;
- [IPSET <имя_набора>] – определения набора IP-адресов;
- [ALIASES] – определения псевдонимов.

Опции секции [OPTIONS] файла конфигурации ВМ/контейнера приведены в табл. Таблица 25.

Т а б л и ц а 25 – Опции секции [OPTIONS] файла конфигурации ВМ/контейнера

Опция	Описание
dhcp: <1|0> (по умолчанию = 0)	Включить DHCP
enable: <1|0>	Включить или отключить межсетевой экран
ipfilter: <1|0>	Включить фильтры IP по умолчанию. Это эквивалентно добавлению

	пустого ipfilter-net<id> ipset для каждого интерфейса. Такие ipset неявно содержат разумные ограничения по умолчанию, такие как ограничение локальных адресов ссылок IPv6 до одного, полученного из MAC-адреса интерфейса. Для контейнеров будут неявно добавлены настроенные IP-адреса
log_level_in: <alert | crit | debug | emerg | err | info | nolog | notice | warning>	Уровень журнала для входящего трафика. Возможные значения: - alert – логировать важные события; - crit – логировать критические события; - debug – логировать всё (для отладки); - emerg – логировать только аварийные события; - err – логировать ошибки; - info – логировать информационные сообщения; - nolog – не логировать; - notice – логировать уведомления; - warning – логировать предупреждения.
log_level_out: <alert | crit | debug | emerg | err | info | nolog | notice | warning>	Уровень журнала для исходящего трафика (аналогично log_level_in)
macfilter: <1|0> (по умолчанию = 1)	Включить/выключить фильтр MAC-адресов
ndp: <1|0> (по умолчанию = 0)	Включить NDP (протокол обнаружения соседей)
policy_in: <ACCEPT | DROP | REJECT>	Политика по умолчанию для входящего трафика. Возможные значения: - ACCEPT – разрешить; - DROP – отбросить; - REJECT – отклонить с отправкой уведомления.
policy_out: <ACCEPT | DROP | REJECT>	Политика по умолчанию для исходящего трафика (аналогично policy_in)
radv: <1|0>	Разрешить отправку объявлений маршрутизатора

Параметры межсетевого экрана ВМ/Контейнера можно настроить в веб-интерфейсе «ВМ»/«Контейнер»→«Сетевой экран» → «Параметры» (Рис. 287).

Параметры межсетевого экрана ВМ

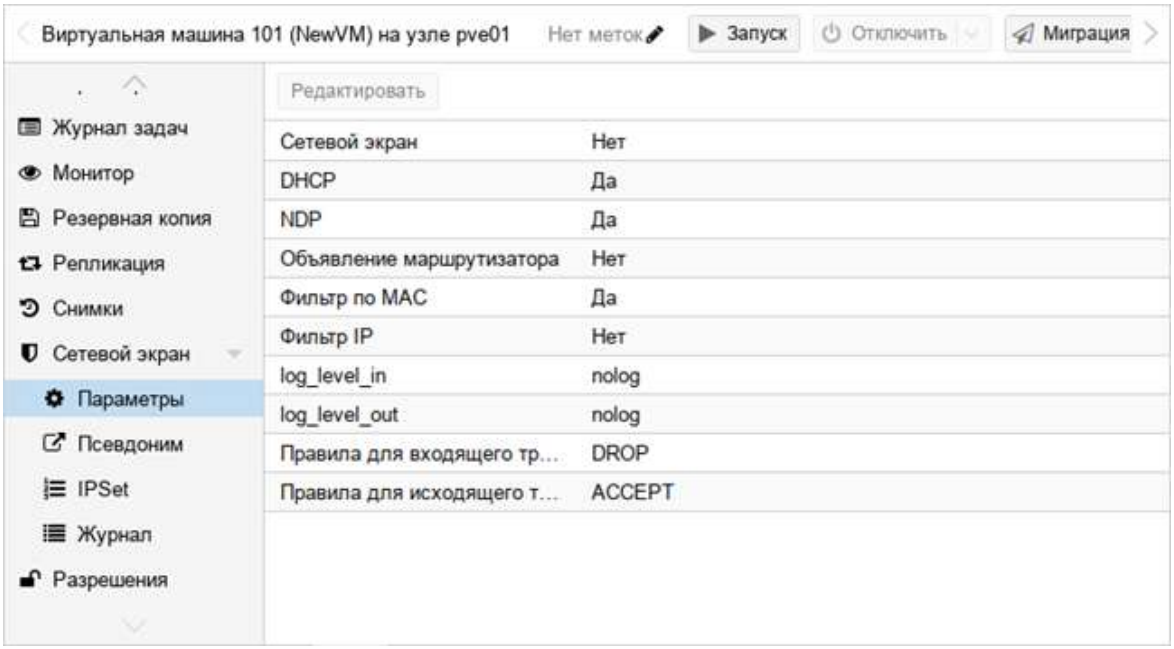


Рис. 287

Каждое виртуальное сетевое устройство, в дополнение к общей опции включения межсетевого экрана, имеет свой собственный флаг включения межсетевого экрана (Рис. 288). Таким образом, можно выборочно включить межсетевой экрана для каждого интерфейса.

Включение межсетевого экрана для сетевого устройства ВМ

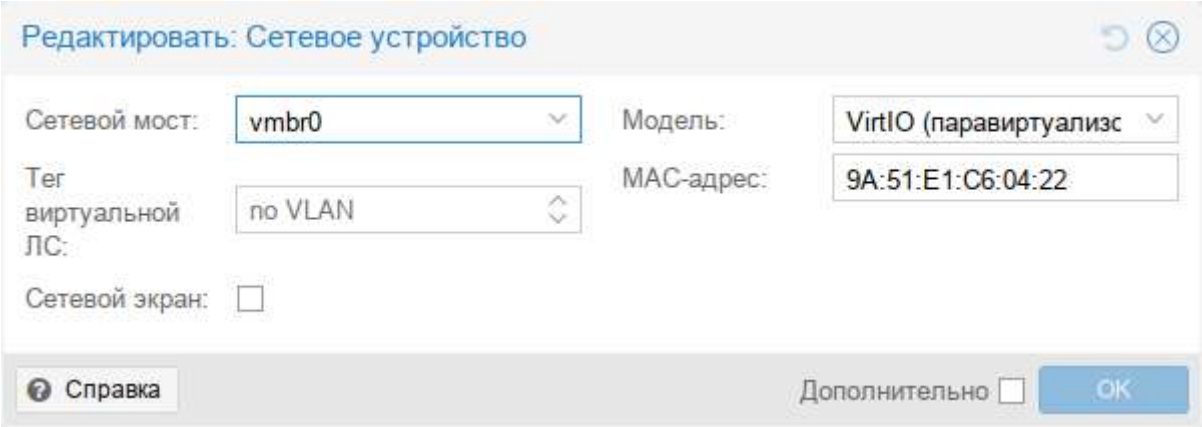


Рис. 288

4.17.3 Правила межсетевого экрана

Правила межсетевого экрана состоят из направления (IN или OUT) и действия (ACCEPT, DENY, REJECT). Можно также указать имя макроса. Макросы содержат предопределенные наборы правил и параметров. Правила можно отключить, добавив к ним префикс |.

Синтаксис правил:

```
[RULES]

DIRECTION ACTION [OPTIONS]
```

```
|DIRECTION ACTION [OPTIONS] # отключенное правило

DIRECTION MACRO(ACTION) [OPTIONS] # использовать предопределенный макрос
```

Параметры, которые можно использовать для уточнения соответствия правил приведены в табл. Таблица 26.

Т а б л и ц а 26 – Опции файла конфигурации

Опция	Описание
--dest <string>	Ограничить адрес назначения пакета. Может быть указан одиночный IP-адрес, набор IP-адресов (+ipsetname) или псевдоним IP (alias). Можно указать диапазон адресов (например, 200.34.101.207-201.3.9.99) или список IP-адресов и сетей (записи разделяются запятой). Не следует смешивать адреса IPv4 и IPv6 в таких списках
--dport <string>	Ограничить порт назначения TCP/UDP. Можно использовать имена служб или простые числа (0-65535), как определено в /etc/services. Диапазоны портов можно указать с помощью \d+:\d+, например 80:85. Для сопоставления нескольких портов или диапазонов можно использовать список, разделенный запятыми
--icmp-type <string>	Тип icmp. Действителен, только если proto равен icmp
--iface <string>	Сетевой интерфейс, к которому применяется правило. В правилах для ВМ и контейнеров необходимо указывать имена ключей конфигурации сети net\d+, например, net0. Правила, связанные с узлом, могут использовать произвольные строки
--log <alert | crit | debug | emerg | err | info | nolog | notice | warning>	Уровень журналирования для правила межсетевого экрана
--proto <string>	IP-протокол. Можно использовать названия протоколов (tcp/udp) или простые числа, как определено в /etc/protocols
--source <string>	Ограничить исходный адрес пакета. Может быть указан одиночный IP-адрес, набор IP-адресов (+ipsetname) или псевдоним IP (alias). Можно указать диапазон адресов (например, 200.34.101.207-201.3.9.99) или список IP-адресов и сетей (записи разделяются запятой). Не следует смешивать адреса IPv4 и IPv6 в таких списках
--sport <string>	Ограничить исходный порт TCP/UDP. Можно использовать имена служб или простые числа (0-65535), как определено в /etc/services. Диапазоны портов можно указать с помощью \d+:\d+, например, 80:85. Для сопоставления нескольких портов или диапазонов можно использовать список, разделенный запятыми

Примеры:

```
[RULES]
IN SSH(ACCEPT) -i net0
IN SSH(ACCEPT) -i net0 # a comment
IN SSH(ACCEPT) -i net0 -source 192.168.0.192 # разрешить SSH только из 192.168.0.192
IN SSH(ACCEPT) -i net0 -source 10.0.0.1-10.0.0.10 # разрешить SSH для диапазона IP
```

```
IN SSH(ACCEPT) -i net0 -source 10.0.0.1,10.0.0.2,10.0.0.3 # разрешить SSH для списка IP-адресов
IN SSH(ACCEPT) -i net0 -source +mynetgroup # разрешить SSH для ipset mynetgroup
IN SSH(ACCEPT) -i net0 -source myserveralias # разрешить SSH для псевдонима myserveralias

|IN SSH(ACCEPT) -i net0 # отключенное правило

IN  DROP # отбросить все входящие пакеты
OUT ACCEPT # принять все исходящие пакеты
```

Для добавления правила в веб-интерфейсе необходимо перейти в раздел «Сетевой экран» (например, «Центр обработки данных»→«Сетевой экран»), нажать кнопку «Добавить», в открывшемся окне задать параметры правила (Рис. 289) и нажать кнопку «Добавить».

Добавление правила

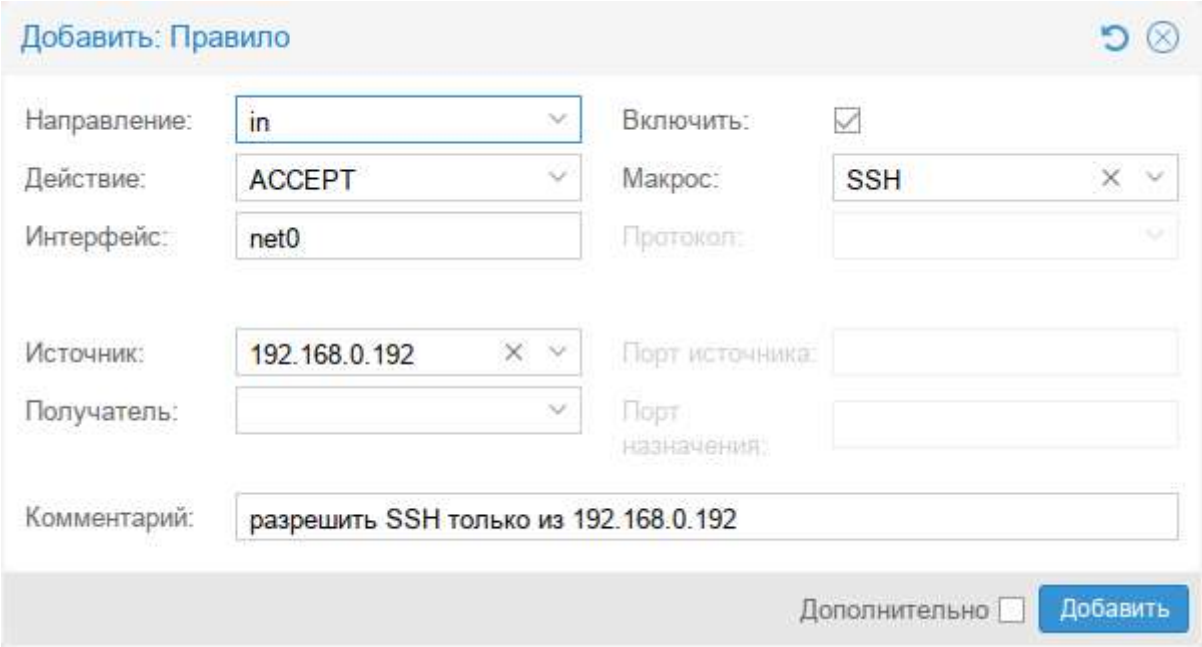


Рис. 289

4.17.4 Группы безопасности

Группа безопасности – это набор правил, определенных на уровне кластера, которые можно использовать во всех правилах ВМ. Например, можно определить группу с именем «webserver» с правилами для открытия портов http и https:

```
# /etc/pve/firewall/cluster.fw

[group webserver]
IN ACCEPT -p tcp -dport 80
IN ACCEPT -p tcp -dport 443
```

Затем можно добавить эту группу в сетевой экран ВМ:

```
# /etc/pve/firewall/<VMID>.fw

[RULES]
GROUP webserver
```

Пример работы с группой безопасности в веб-интерфейсе:

1) перейти в раздел «Центр обработки данных» →«Сетевой экран» → «Группа безопасности»;
2) в секции «Группа» нажать кнопку «Создать», в открывшемся окне ввести название группы (Рис. 290) и нажать кнопку «Создать»;
3) выделить созданную группу безопасности;
4) в секции «Правила» нажать кнопку «Добавить», в открывшемся окне установить параметры правила (Рис. 291) и нажать кнопку «Добавить»;
5) повторить п.3 нужное число раз для добавления всех правил к группе (Рис. 292);
6) перейти в раздел «ВМ»→«Сетевой экран»;
7) нажать кнопку «Вставить: Группа безопасности», в открывшемся окне в поле «Группа безопасности» выбрать группу (Рис. 293), установить отметку в поле «Включить» и нажать кнопку «Добавить».

Создание группы безопасности

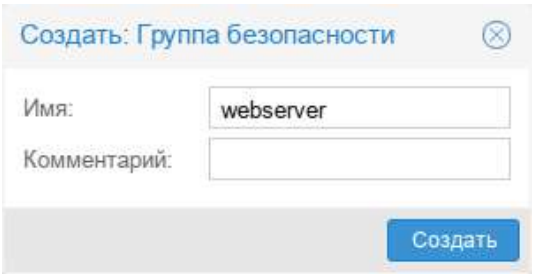


Рис. 290

Добавление правила к группе безопасности

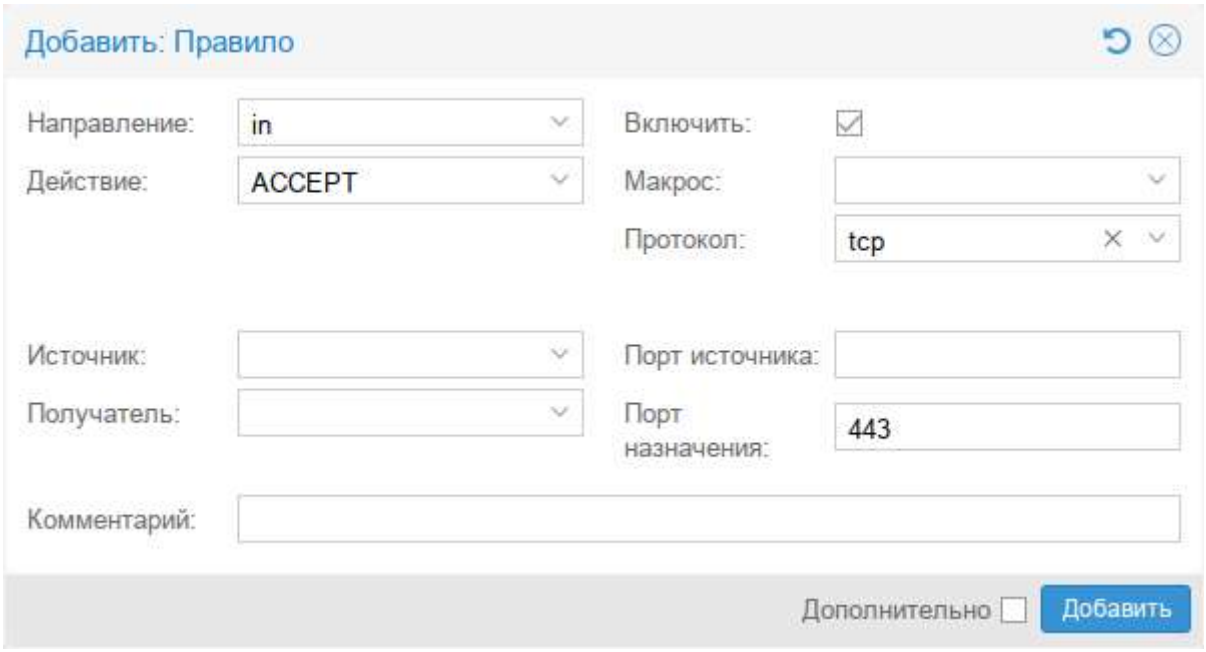


Рис. 291

Правила группы безопасности webserver

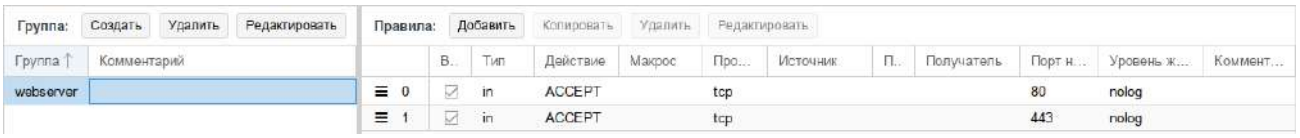


Рис. 292

Добавление правил группы безопасности к ВМ

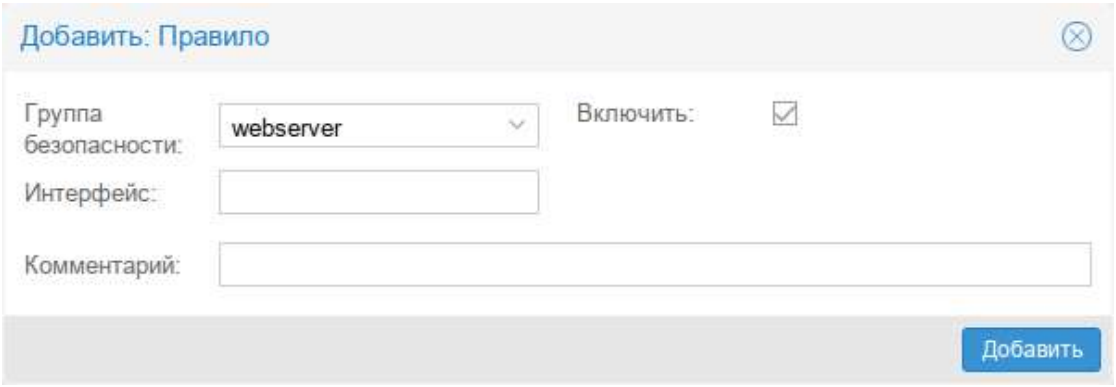


Рис. 293

4.17.5 IP-псевдонимы

IP-псевдонимы могут быть полезны для упрощения управления сетевыми правилами. Псевдонимы позволяют связывать IP-адреса сетей с именем, затем можно ссылаться на эти имена:

- внутри определений набора IP-адресов;
- в параметрах `source` и `dest` правил межсетевого экрана.

IP-псевдоним `local_network` определяется автоматически. Чтобы увидеть назначенные псевдониму значения можно выполнить команду:

```
# pve-firewall localnet
local hostname: pve01
local IP address: 192.168.0.186
network auto detect: 192.168.0.0/24
using detected local_network: 192.168.0.0/24

accepting corosync traffic from/to:
 - pve02: 192.168.0.90 (link: 0)
 - pve03: 192.168.0.70 (link: 0)
```

Межсетевой экран автоматически устанавливает правила, чтобы разрешить все необходимое для кластера (corosync, API, SSH) с помощью этого псевдонима.

Переопределить эти значения можно в разделе [ALIASES] в файле `/etc/pve/firewall/cluster.fw`. Если используется один узел в публичной сети, лучше явно назначить локальный IP-адрес:

```
# /etc/pve/firewall/cluster.fw
[ALIASES]
local_network 192.168.0.186 # использовать одиночный IP-адрес
```

Пример создания IP-псевдонима («Центр обработки данных»→«Сетевой экран» → «Псевдоним» кнопка «Добавить») показан на Рис. 294.

Создание псевдонима

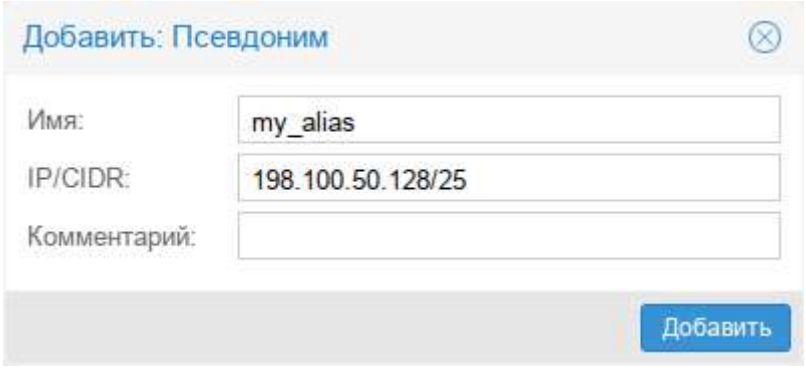


Рис. 294

4.17.6 Наборы IP-адресов

Наборы IP-адресов (IPSet) можно использовать для определения групп сетей и узлов. На них можно ссылаться в свойствах источника и назначения правил межсетевого экрана с помощью «+name».

Следующий пример разрешает HTTP-трафик из набора IP-адресов `management`:

```
IN HTTP(ACCEPT) -source +management
```

Пример работы с набором IP-адресов в веб-интерфейсе:

1) перейти в раздел «Центр обработки данных»→«Сетевой экран» → «IPSet»;

2) в секции «IPSet» нажать кнопку «Создать», в открывшемся окне ввести название набора (Рис. 295) и нажать кнопку «ОК»;
3) выделить созданный набор;
4) в секции «IP/CIDR» нажать кнопку «Добавить», в открывшемся окне указать IP/CIDR (Рис. 296) и нажать кнопку «Создать»;
5) повторить п.3 нужное число раз для добавления всех IP к набору (Рис. 297);
6) перейти в раздел «ВМ»→«Сетевой экран»;
7) нажать кнопку «Добавить», в открывшемся окне в поле «Источник» выбрать созданный набор (Рис. 298), установить другие параметры правила и нажать кнопку «Добавить».

Создание набора IP-адресов

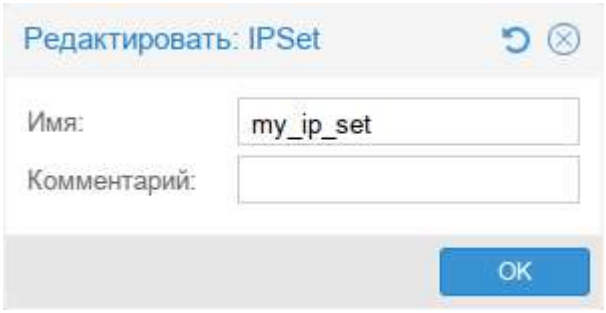


Рис. 295

Добавление сети к набору

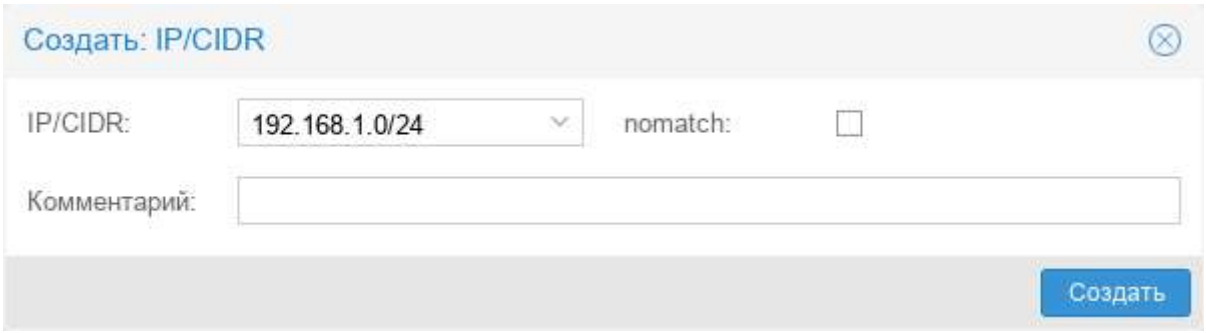


Рис. 296

IP-адреса в наборе my_ip_set

Рис. 297

Указание набора IP-адресов в правиле межсетевого экрана

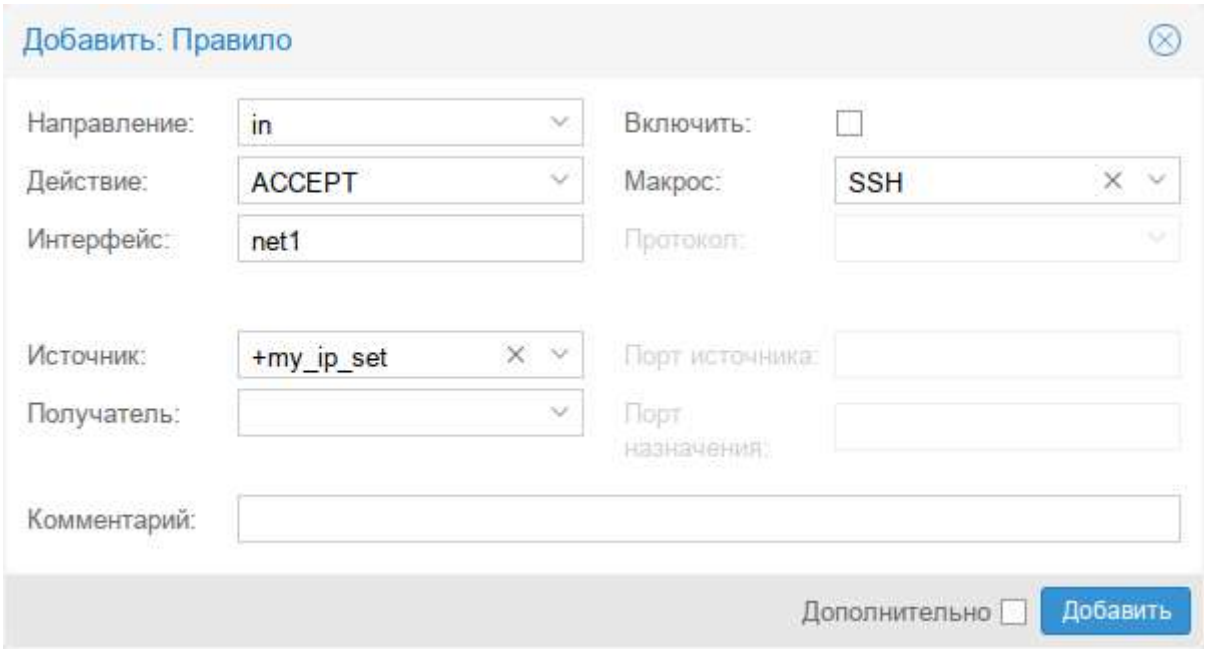


Рис. 298

4.17.6.1 Стандартный набор IP-адресов management

Стандартный набор IP-адресов `management` применяется только к межсетевым экранам узлов (не к межсетевым экранам ВМ). Этим IP-адресам разрешено выполнять обычные задачи управления (PVE GUI, VNC, SPICE, SSH).

Локальная сеть кластера автоматически добавляется в этот набор IP-адресов (псевдоним cluster_network), чтобы включить межкластерную связь узлов (multicast, ssh, …):

```
# /etc/pve/firewall/cluster.fw

[IPSET management]
192.168.0.90
192.168.0.90/24
```

4.17.6.2 Стандартный набор IP-адресов blacklist

Трафик с IP-адресов, внесенных в черный список (`blacklist`), отбрасывается межсетевым экраном каждого узла и ВМ:

```
# /etc/pve/firewall/cluster.fw

[IPSET blacklist]
77.240.159.182
213.87.123.0/24
```

*4.17.6.3 Стандартный набор IP-адресов ipfilter-net**

Фильтры ipfilter-net* относятся к сетевому интерфейсу ВМ и в основном используются для предотвращения подмены IP-адресов. Если набор IP-адресов ipfilter-net* существует для

интерфейса, то любой исходящий трафик с исходным IP-адресом, не соответствующим соответствующему набору ipfilter его интерфейса, будет отброшен.

Для контейнеров с настроенными IP-адресами эти наборы, если они существуют (или активированы с помощью параметра общего фильтра IP-адресов на вкладке «Параметры» межсетевого экрана ВМ), неявно содержат связанные IP-адреса.

Как для ВМ, так и для контейнеров они также неявно содержат стандартный локальный адрес IPv6, полученный из MAC-адреса, чтобы обеспечить работу протокола обнаружения соседей.

```
/etc/pve/firewall/<VMID>.fw

[IPSET ipfilter-net0] # only allow specified IPs on net0
192.168.0.90
```

4.17.7 Службы и команды

Межсетевой экран запускает две службы на каждом узле:

- pve-firewall – служба, отвечающая за применение и управление правилами firewall;
- pvefw-logger – служба, отвечающая за логирование событий, связанных с работой firewall.

Для запуска и остановки службы межсетевого экрана можно также использовать команду `pve-firewall`:

```
# pve-firewall start
# pve-firewall stop
```

Получение статуса службы:

```
# pve-firewall status
```

Данная команда считывает и компилирует все правила межсетевого экрана, поэтому если конфигурация межсетевого экрана содержит какие-либо ошибки, будут выведены ошибки.

Для просмотра сгенерированных правил iptables можно использовать команду:

```
# iptables-save
```

4.17.8 Правила по умолчанию

4.17.8.1 Входящий/исходящий DROP/REJECT центра обработки данных

Если политика ввода или вывода для межсетевого экрана установлена на DROP или REJECT, следующий трафик все равнобудет разрешен для всех узлов PVE в кластере:

- трафик через интерфейс обратной связи;
- уже установленные соединения;
- трафик с использованием протокола IGMP;
- TCP-трафик от узлов управления на порт 8006 для разрешения доступа к веб-интерфейсу;

- TCP-трафик от узлов управления на диапазон портов 5900-5999 для разрешения трафика для веб-консоли VNC;
- TCP-трафик от узлов управления на порт 3128 для подключений к прокси-серверу SPICE;
- TCP-трафик от узлов управления на порт 22 для разрешения доступа по SSH;
- UDP-трафик в сети кластера на порты 5405-5412 для corosync;
- UDP-многоадресный трафик в кластере сеть;
- ICMP трафик типа 3 (Destination Unreachable), 4 (Congestion control) или 11 (Time Exceeded).

Следующий трафик отбрасывается, но не регистрируется даже при включенном ведении журнала:

- TCP-соединения с недопустимым состоянием соединения;
- широковещательный, многоадресный и anycast-трафик, не связанный с corosync, т.е. не проходящий через порты 5405-5412;
- TCP-трафик на порт 43;
- UDP-трафик на порты 135 и 445;
- UDP-трафик на диапазон портов 137-139;
- UDP-трафик с исходного порта 137 на диапазон портов 1024-65535;
- UDP-трафик на порт 1900;
- TCP-трафик на порты 135, 139 и 445;
- UDP-многоадресный трафик в кластере сеть;
- UDP-трафик, исходящий из исходного порта 53.

Остальной трафик отбрасывается или отклоняется, а также регистрируется в соответствии с правилами. Это может зависеть от дополнительных параметров, таких как «NDP», «Фильтр SMURFS» и «Фильтр флагов TCP» («Сетевой экран» → «Параметры»).

Чтобы увидеть активные цепочки и правила межсетевого экрана, можно использовать команду:

```
# iptables-save
```

Этот вывод также включается в системный отчет, выводимый при выполнении команды:

```
# pvereport
```

4.17.8.2 Входящий/исходящий DROP/REJECT ВМ/Контейнера

Весь трафик к ВМ отбрасывается/отклоняется, с некоторыми исключениями для DHCP, NDP, Router Advertisement, MAC и IP-фильтрации в зависимости от установленной конфигурации. Эти же правила для отбрасывания/отклонения пакетов наследуются от центра обработки данных, в то время как исключения для принятого входящего/исходящего трафика узла не применяются.

4.17.9 Ведение журнала

По умолчанию ведение журнала трафика, отфильтрованного правилами межсетевого экрана, отключено. Чтобы включить ведение журнала, необходимо установить уровень журнала для входящего и/или исходящего трафика в разделе «Сетевой экран» → «Параметры». Это можно сделать как для узла, так и для ВМ по отдельности. При этом ведение журнала стандартных правил сетевого экрана PVE включено, а вывод можно наблюдать в разделе «Сетевой экран»→ «Журнал». Кроме того, для стандартных правил регистрируются только некоторые отброшенные или отклоненные пакеты (см. «Правила по умолчанию»).

`loglevel` не влияет на объем регистрируемого фильтрованного трафика. Он изменяет LOGID (табл. Таблица 27), добавленный в качестве префикса к выводу журнала для упрощения фильтрации и постобработки.

Таблица 27 – Флаги loglevel

loglevel	LOGID
nolog	-
emerg	0
alert	1
crit	2
err	3
warning	4
notice	5
info	6
debug	7

Типичная запись журнала межсетевого экрана выглядит следующим образом:

```
VMID LOGID CHAIN TIMESTAMP POLICY: PACKET_DETAILS
```

В случае межсетевого экрана узла VMID равен 0.

Чтобы регистрировать пакеты, отфильтрованные пользовательскими правилами, можно задать параметр уровня журнала для каждого правила индивидуально. Это позволяет вести журнал детально и независимо от уровня журнала, определенного для стандартных правил.

Хотя уровень журнала для каждого отдельного правила можно легко определить или изменить в веб-интерфейсе во время создания или изменения правила, его также можно задать с помощью соответствующих вызовов API pvesh.

Уровень журнала также можно задать с помощью файла конфигурации межсетевого экрана, добавив `-log <loglevel>` к выбранному правилу.

Например, следующие два правила идентичны:

```
IN REJECT -p icmp -log nolog
IN REJECT -p icmp
```

А правило:

```
IN REJECT -p icmp -log debug
```

создает вывод журнала, помеченный уровнем отладки.

4.17.10 Особенности IPv6

Межсетевой экран содержит несколько специфичных для IPv6 опций. Следует отметить, что IPv6 больше не использует протокол ARP, а вместо этого использует NDP (Neighbor Discovery Protocol), который работает на уровне IP и, следовательно, для успешной работы нуждается в IP-адресах. Для этой цели используются локальные адреса, полученные из MAC-адреса интерфейса. По умолчанию опция NDP включена как на уровне узла, так и на уровне ВМ, чтобы разрешить отправку и получение пакетов обнаружения соседей (NDP).

Помимо обнаружения соседей, NDP также используется для нескольких других вещей, таких как автоматическая настройка и объявление маршрутизаторов.

По умолчанию ВМ разрешено отправлять сообщения запроса маршрутизатора (для запроса маршрутизатора) и получать пакеты объявления маршрутизатора. Это позволяет им использовать автоматическую настройку без сохранения состояния. С другой стороны, ВМ не могут объявлять себя маршрутизаторами, если не установлена опция «Объявление маршрутизатора» (`radv: 1`).

Что касается локальных адресов ссылок, необходимых для NDP, также можно включить опцию «Фильтр IP» (`ipfilter: 1`), которая имеет тот же эффект, что и добавление `ipfilter-net* ipset` для каждого сетевого интерфейса ВМ, содержащего соответствующие локальные адреса ссылок.

4.17.11 Порты, используемые PVE

Таблица 28 – Используемые порты

Порт	Функция
8006 (TCP, HTTP/1.1 через TLS)	Веб-интерфейс PVE
5900-5999 (TCP, WebSocket)	Доступ к консоли VNC
3128 (TCP)	Доступ к консоли SPICE
22 (TCP)	SSH доступ
111 (UDP)	rpcbind
25 (TCP, исходящий)	sendmail
5405-5412 UDP	Трафик кластера corosync
60000-60050 (TCP)	Живая миграция (память ВМ и данные локального диска)

4.18 Пользователи и их права

PVE поддерживает несколько источников аутентификации, например, Linux PAM, интегрированный сервер аутентификации PVE (Рис. 299), LDAP, Active Directory и OpenID Connect.

Выбор типа аутентификации в веб-интерфейсе

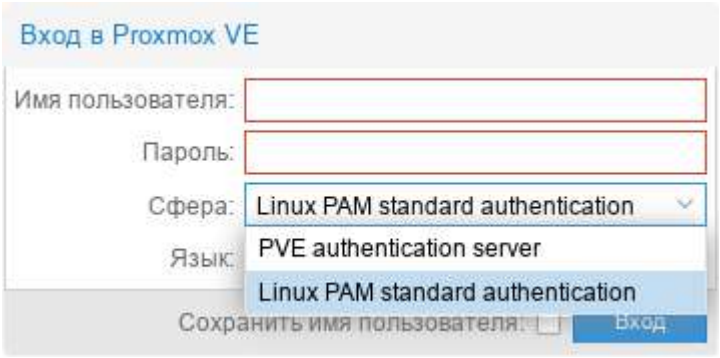


Рис. 299

Используя основанное на ролях управление пользователями и разрешениями для всех объектов (ВМ, хранилищ, узлов и т. д.), можно определить многоуровневый доступ.

PVE хранит данные пользователей в файле `/etc/pve/user.cfg`:

```
# cat /etc/pve/user.cfg
user:root@pam:1:0::::::
user:test@pve:1:0::::::
user:testuser@pve:1:0::::Just a test::
user:user@pam:1:0::::::

group:admin:user@pam::
group:testgroup:test@pve::
```

Пользователя часто внутренне идентифицируют по его имени и области аутентификации в форме <user>@<realm>.

После установки PVE существует один пользователь root@pam, который соответствует суперпользователю ОС. Этого пользователя нельзя удалить, все системные письма будут отправляться на адрес электронной почты, назначенный этому пользователю. Суперпользователь имеет неограниченные права, поэтому рекомендуется добавить других пользователей с меньшими правами.

Каждый пользователь может быть членом нескольких групп. Группы являются предпочтительным способом организации прав доступа. Всегда следует предоставлять права доступа группам, а не отдельным пользователям.

4.18.1 API-токены

API-токены позволяют получить доступ без сохранения состояния к REST API из другой системы. Токены могут быть сгенерированы для отдельных пользователей. Токенам, для ограничения объема и продолжительности доступа, могут быть предоставлены отдельные разрешения и даты истечения срока действия. Если API-токен скомпрометирован, его можно отозвать, не отключая самого пользователя.

API-токены бывают двух основных типов:

- токен с раздельными привилегиями – токену необходимо предоставить явный доступ с помощью ACL. Эффективные разрешения токена вычисляются путем пересечения разрешений пользователя и токена;
- токен с полными привилегиями – разрешения токена идентичны разрешениям связанного с ним пользователя.

API-токен состоит из двух частей:

- идентификатор (Token ID), который состоит из имени пользователя, области и имени токена (user@realm!имя токена);
- секретное значение.

Для генерации API-токена в веб-интерфейсе необходимо в окне «Центр обработки данных»→«Разрешения»→«Маркеры API» нажать кнопку «Добавить». В открывшемся окне следует выбрать пользователя и указать ID-токена (Рис. 300).

Генерация API-токена в веб-интерфейсе

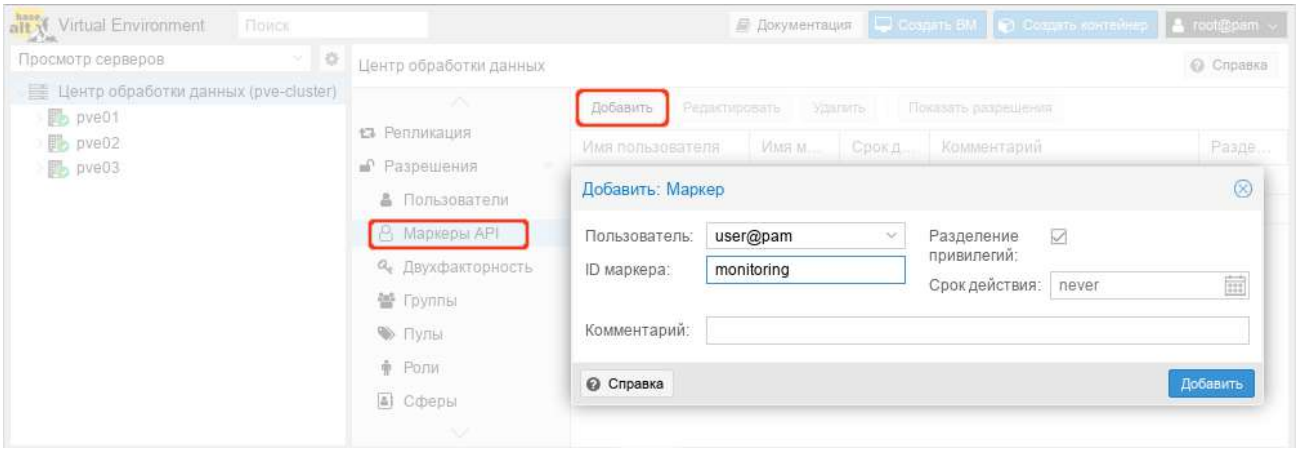


Рис. 300

Примечание. Отметку «Разделение привилегий» следует снять, в противном случае токену необходимо назначить явные права. Подробнее см. в разделе «Управление доступом».

После нажатия кнопки «Добавить» будет сгенерирован API-токен (Рис. 301).

API-токен

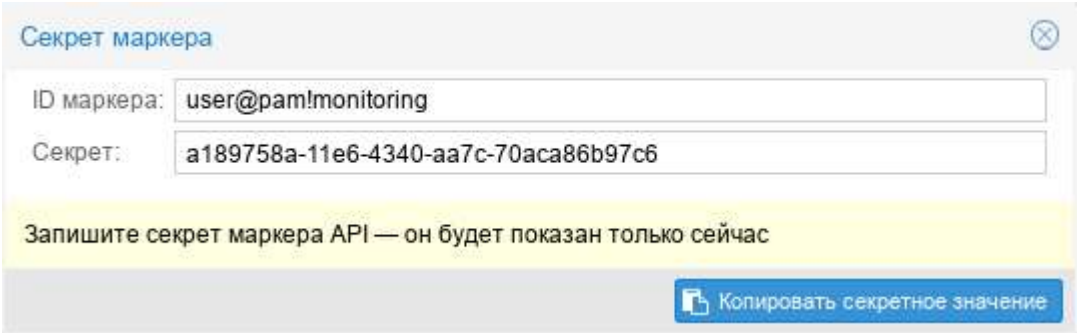


Рис. 301

Отображаемое секретное значение необходимо сохранить.

Примечание. Значение токена отображается/возвращается только один раз при создании токена. Его нельзя будет снова получить через API позже!

Если был создан токен с раздельными привилегиями, токену необходимо предоставить разрешения:

1) в окне «Центр обработки данных»→«Разрешения» нажать кнопку «Добавить»→«Разрешения маркера API» (Рис. 302);

2) в открывшемся окне выбрать путь, токен и роль и нажать кнопку «Добавить» (Рис. 303).

PVE. Добавление разрешений

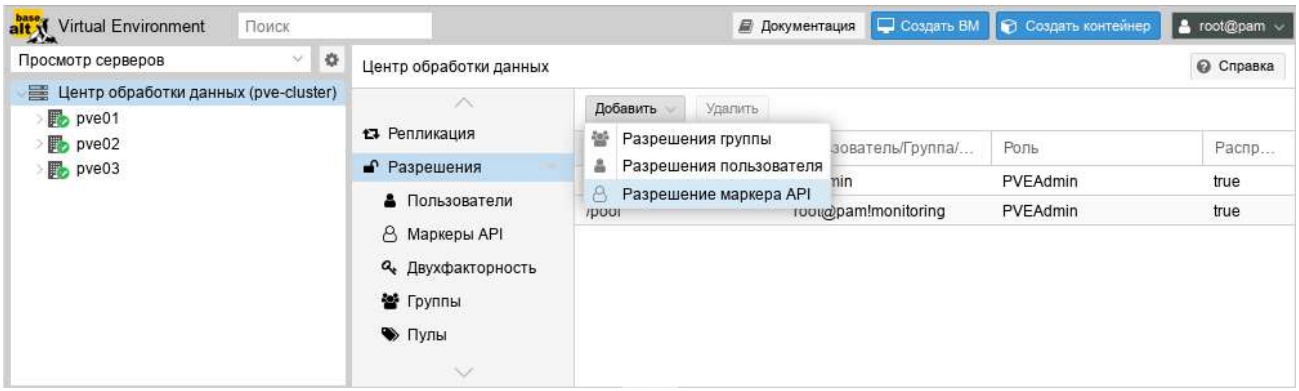


Рис. 302

Добавление разрешений для API-токена

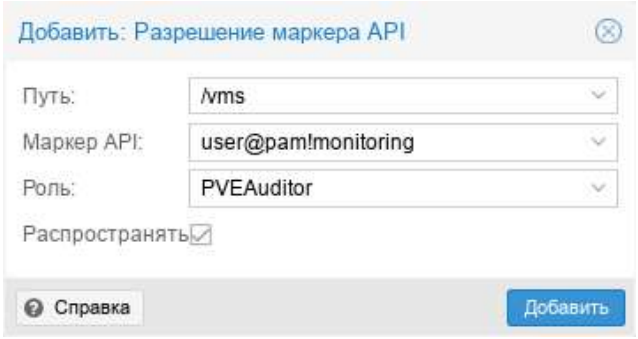


Рис. 303

Для создания API-токена в консоли используется команда:

```
# pveum user token add <userid> <tokenid> [ОПЦИИ]
```

Возможные опции:

- --comment <строка> – комментарий к токену;
- --expire <целое число> – дата истечения срока действия API-токена в секундах с начала эпохи (по умолчанию срок действия API-токена совпадает со сроком действия пользователя). Значение 0 указывает, что срок действия токена не ограничен;

- --privsep <логическое значение> – ограничить привилегии API-токена с помощью отдельных списков контроля доступа (по умолчанию) или предоставить полные привилегии соответствующего пользователя (значение 0).

Примеры команд для работы с токенами:

- создать токен t2 для пользователя user@pam с полными привилегиями:

```
# pveum user token add user@pam t2 --privsep 0
```

```
┌──────────────┬──────────────────────────────────────┐
│ key          │ value                                │
╞══════════════╪══════════════════════════════════════╡
│ full-tokenid │ user@pam!t2                          │
├──────────────┼──────────────────────────────────────┤
│ info         │ {"privsep":"0"}                      │
├──────────────┼──────────────────────────────────────┤
│ value        │ 3c749375-e189-493d-8037-a1179317c406 │
└──────────────┴──────────────────────────────────────┘
```

- вывести список токенов пользователя:

```
# pveum user token list user@pam
```

```
┌──────────────┬─────────┬─────────┬─────────┐
│ tokenid      │ comment │ expire  │ privsep │
╞══════════════╪═════════╪═════════╪═════════╡
│ monitoring   │         │       0 │ 1       │
├──────────────┼─────────┼─────────┼─────────┤
│ t2           │         │       0 │ 0       │
└──────────────┴─────────┴─────────┴─────────┘
```

- вывести эффективные разрешения для токена:

```
# pveum user token permissions user@pam t2
```

Можно использовать опцию `--path`, чтобы вывести разрешения для этого пути, а не всё дерево:

```
# pveum user token permissions user@pam t2 --path /storage
```

- добавить разрешения для токена с раздельными привилегиями:

```
# pveum acl modify /vms --tokens 'user@pam!monitoring' --roles
PVEAdmin,PVEAuditor
```

- удалить токен пользователя:

```
# pveum user token remove user@pam t2
```

Примечание. Разрешения на API-токены всегда являются подмножеством разрешений соответствующего пользователя. То есть API-токен не может использоваться для выполнения задачи, на которую у пользователя владельца токена нет разрешения.

Пример:

- предоставить пользователю test@pve роль PVEVMAdmin на всех ВМ:

```
# pveum acl modify /vms --users test@pve --roles PVEVMAdmin
```

- создать API-токен с раздельными привилегиями с правами только на просмотр информации о ВМ:

```
# pveum user token add test@pve monitoring --privsep 1
# pveum acl modify /vms --tokens 'test@pve!monitoring' --roles PVEAuditor
```

- проверить разрешения пользователя и токена:

```
# pveum user permissions test@pve
# pveum user token permissions test@pve monitoring
```

Чтобы использовать API-токен при выполнении API-запросов, следует установить заголовок HTTP Authorization в значение `PVEAPIToken=USER@REALM!TOKENID=UUID`.

4.18.2 Пулы ресурсов

Пул ресурсов – это набор ВМ, контейнеров и хранилищ. Пул ресурсов удобно использовать для обработки разрешений в случаях, когда определенные пользователи должны иметь контролируемый доступ к определенному набору ресурсов. Пулы ресурсов часто используются в тандеме с группами, чтобы члены группы имели разрешения на набор машин и хранилищ.

Пример создания пула ресурсов в веб-интерфейсе:

1) В окне «Центр обработки данных»→«Разрешения»→«Пулы» нажать кнопку «Создать». В открывшемся окне указать название пула и нажать кнопку «ОК» (Рис. 304);

2) Добавить в пул ВМ. Для этого выбрать пул («Пул»→«Члены»), нажать кнопку «Добавить»→«Виртуальная машина», выбрать ВМ и нажать кнопку «Добавить» (Рис. 305);

3) Добавить в пул хранилища. Для этого выбрать пул («Пул»→«Члены»), нажать кнопку «Добавить»→«Хранилище», выбрать хранилище и нажать кнопку «Добавить» (Рис. 306).

Создание пула ресурсов в веб-интерфейсе

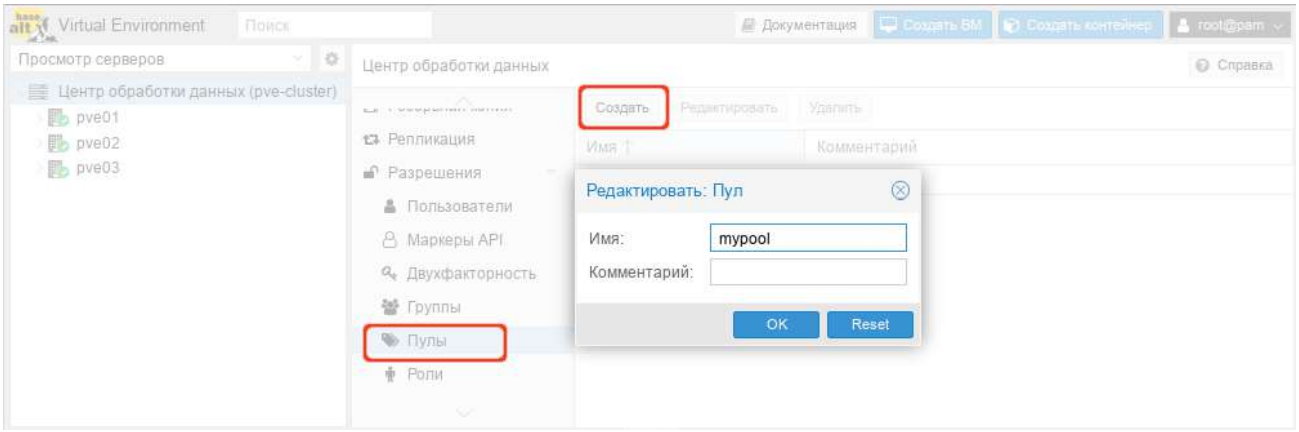


Рис. 304

Добавление ВМ в пул ресурсов

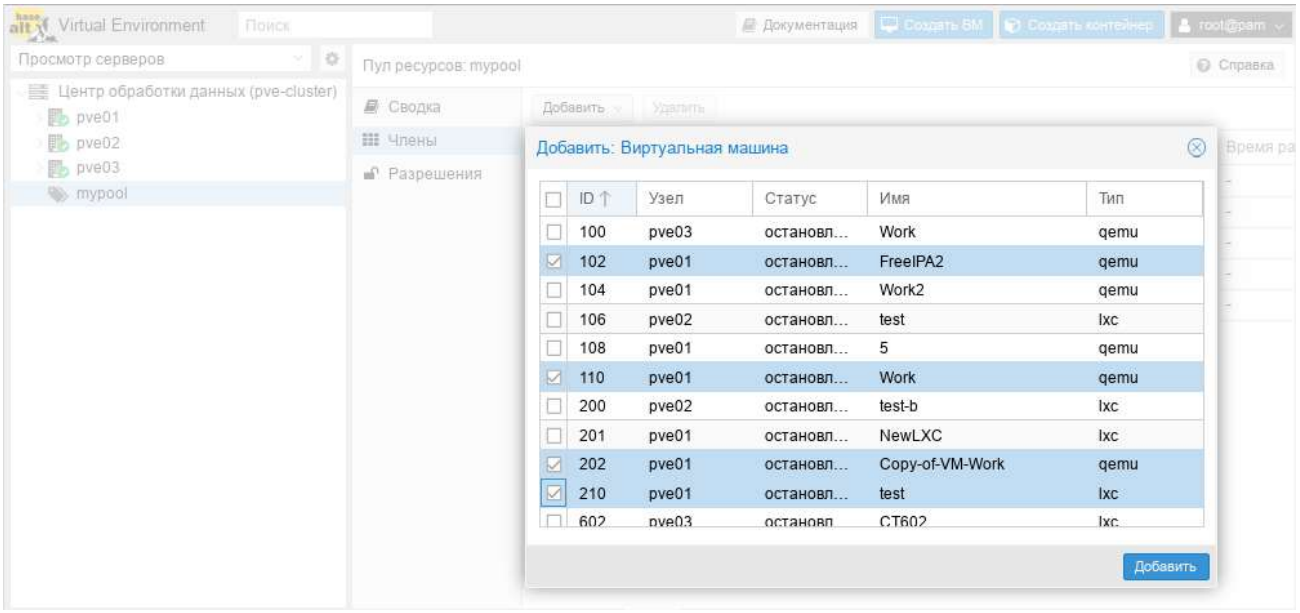


Рис. 305

Добавление хранилища в пул ресурсов

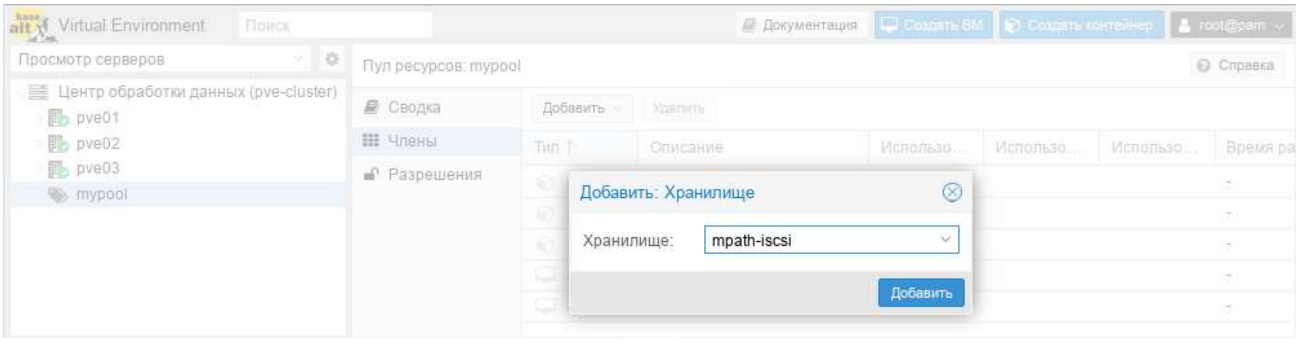


Рис. 306

Работа с пулами ресурсов в командной строке:

- создать пул:

```
# pveum pool add IT --comment 'IT development pool'
```

- вывести список пулов:

```
# pveum pool list
```

```
┌──────────┐
│ poolid   │
╞══════════╡
│ IT       │
├──────────┤
│ mypool   │
└──────────┘
```

- добавить ВМ и хранилища в пул:

```
# pveum pool modify IT --vms 201,108,202,104,208 --storage mpath2,nfs-storage
```

- удалить ВМ из пула:

```
# pveum pool modify IT --delete 1 --vms 108,104
```

- удалить пул:

```
# pveum pool delete mypool
```

Примечание. Можно удалить только пустой пул.

4.18.3 Области аутентификации

Чтобы пользователь мог выполнить какое-либо действие (например, просмотр, изменение или удаление ВМ), ему необходимо иметь соответствующие разрешения.

Доступны следующие сферы (методы) аутентификации:

- «Стандартная аутентификация Linux PAM» – общесистемная аутентификация пользователей;
- «Сервер аутентификации PVE» – пользователи полностью управляются PVE и могут менять свои пароли через графический интерфейс. Этот метод аутентификации удобен для небольших (или даже средних) установок, где пользователям не требуется доступ ни к чему, кроме PVE;
- «Сервер LDAP» – позволяет использовать внешний LDAP-сервер для аутентификации пользователей (например, OpenLDAP);
- «Сервер Active Directory» – позволяет аутентифицировать пользователей через AD. Поддерживает LDAP в качестве протокола аутентификации;
- «Сервер OpenID Connect» – уровень идентификации поверх протокола OATH 2.0. Позволяет аутентифицировать пользователей на основе аутентификации, выполняемой внешним сервером авторизации.

Настройки сферы аутентификации хранятся в файле `/etc/pve/domains.cfg`.

4.18.3.1 Стандартная аутентификация Linux PAM

При использовании «Стандартная аутентификация Linux PAM» системный пользователь должен существовать (должен быть создан, например, с помощью команды adduser) на всех узлах, на которых пользователю разрешено войти в систему. Если пользователи PAM существуют в хост-системе PVE, соответствующие записи могут быть добавлены в PVE, чтобы эти пользователи могли входить в систему, используя свое системное имя и пароль.

Область Linux PAM создается по умолчанию и не может быть удалена. Администратор может добавить требование двухфакторной аутентификации для пользователей данной области («Требовать двухфакторную проверку подлинности») и установить её в качестве области по умолчанию для входа в систему («По умолчанию») (Рис. 307).

Для добавления нового пользователя, необходимо в окне «Центр обработки данных» → «Разрешения» → «Пользователи» нажать кнопку «Добавить». На Рис. 308 показано создание нового пользователя с использованием PAM аутентификации (системный пользователь user должен существовать, в качестве пароля будет использоваться пароль для входа в систему).

Конфигурация PAM аутентификации

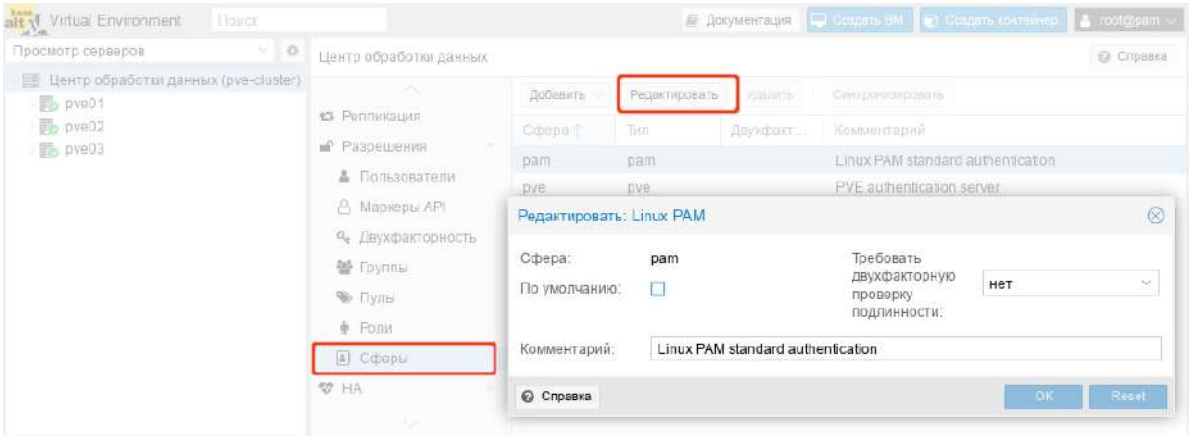


Рис. 307

Создание нового пользователя с использованием PAM аутентификации

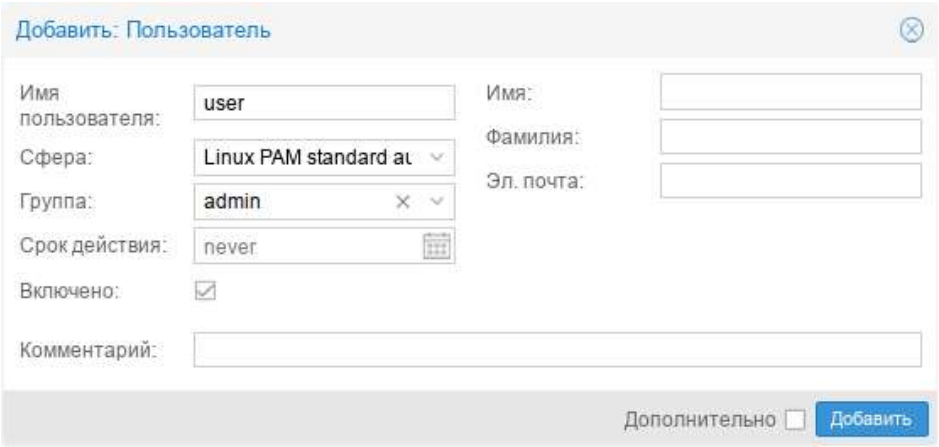


Рис. 308

4.18.3.2 Сервер аутентификации PVE

Область «Сервер аутентификации PVE» представляет собой хранилище паролей в стиле Unix (`/etc/pve/priv/shadow.cfg`). Пароль шифруется с использованием метода хеширования SHA-256.

Область создается по умолчанию, и, как и в случае с Linux PAM, для неё можно добавить требование двухфакторной аутентификации («Требовать двухфакторную проверку подлинности») и установить её в качестве области по умолчанию для входа в систему («По умолчанию») (Рис. 309).

Для добавления нового пользователя необходимо в окне «Центр обработки данных» → «Разрешения» → «Пользователи» нажать кнопку «Добавить». На Рис. 310 показано создание нового пользователя с использованием PVE аутентификации.

Конфигурация PVE аутентификации

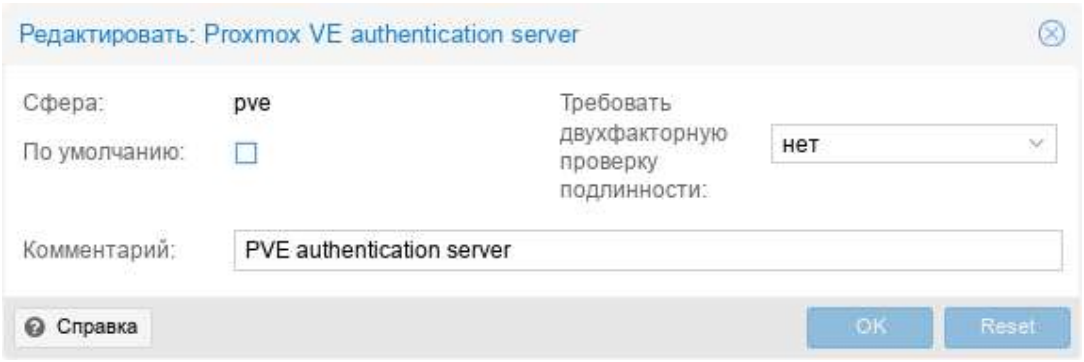


Рис. 309

Создание нового пользователя с использованием PVE аутентификации

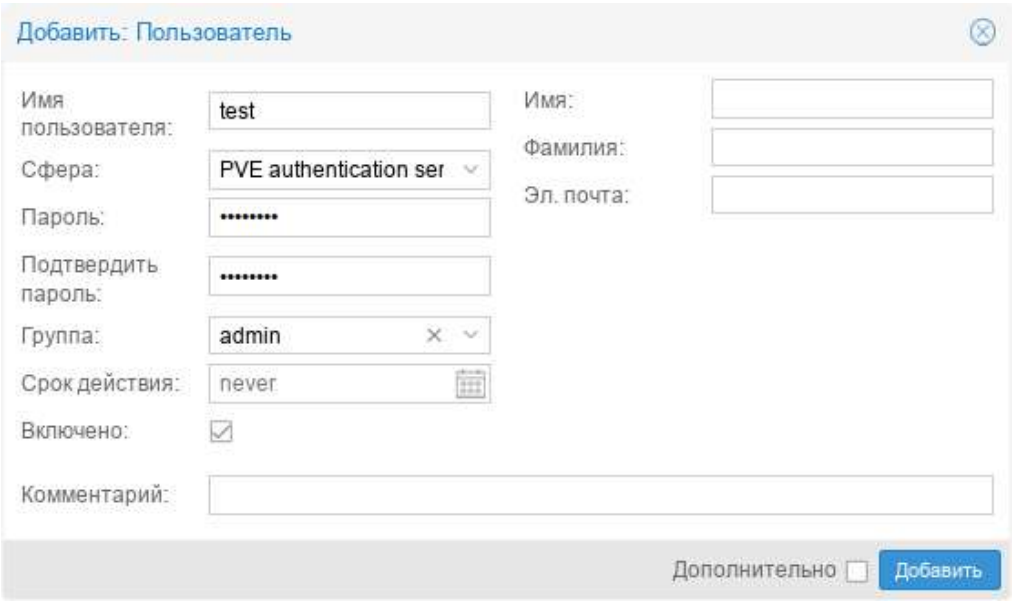


Рис. 310

Примеры использования командной строки для управления пользователями PVE:

- создать пользователя:

```
# pveum useradd testuser@pve -comment "Just a test"
```

- задать или изменить пароль:

```
# pveum passwd testuser@pve
```

- отключить пользователя:

```
# pveum usermod testuser@pve -enable 0
```

- создать новую группу:

```
# pveum groupadd testgroup
```

- создать новую роль:

```
# pveum roleadd PVE_Power-only -privs "VM.PowerMgmt VM.Console"
```

4.18.3.3 LDAP аутентификация

В данном разделе приведён пример настройки LDAP аутентификации для аутентификации на сервере FreeIPA. В примере используются следующие исходные данные:

- ipa.example.test, 192.168.0.113 – сервер FreeIPA;
- admin@example.test – учётная запись с правами чтения LDAP;
- pve – группа, пользователи которой имеют право аутентифицироваться в PVE.

Для настройки аутентификации FreeIPA необходимо выполнить следующие шаги:

1) создать область аутентификации LDAP. Для этого в разделе «Центр обработки данных» → «Разрешения» → «Сферы» нажать кнопку «Добавить» → «Сервер LDAP» (Рис. 311);

2) на вкладке «Общее» (Рис. 312) указать следующие данные:

- «Область» – идентификатор области;
- «Имя основного домена» (base_dn) – каталог, в котором выполняется поиск пользователей (dc=example,dc=test);
- «Имя пользовательского атрибута» (user_attr) – атрибут LDAP, содержащий имя пользователя, с которым пользователи будут входить в систему (uid);
- «По умолчанию» – установить область в качестве области по умолчанию для входа в систему;
- «Сервер» – IP-адрес или имя FreeIPA-сервера (ipa.example.test или 192.168.0.113);
- «Резервный сервер» (опционально) – адрес резервного сервера на случай, если основной сервер недоступен;
- «Порт» – порт, который прослушивает сервер LDAP (обычно 389 без ssl, 636 с ssl);
- «SSL» – использовать ssl;
- «Требовать двухфакторную проверку подлинности» – требовать двухфакторную аутентификацию.

Создать область аутентификации LDAP

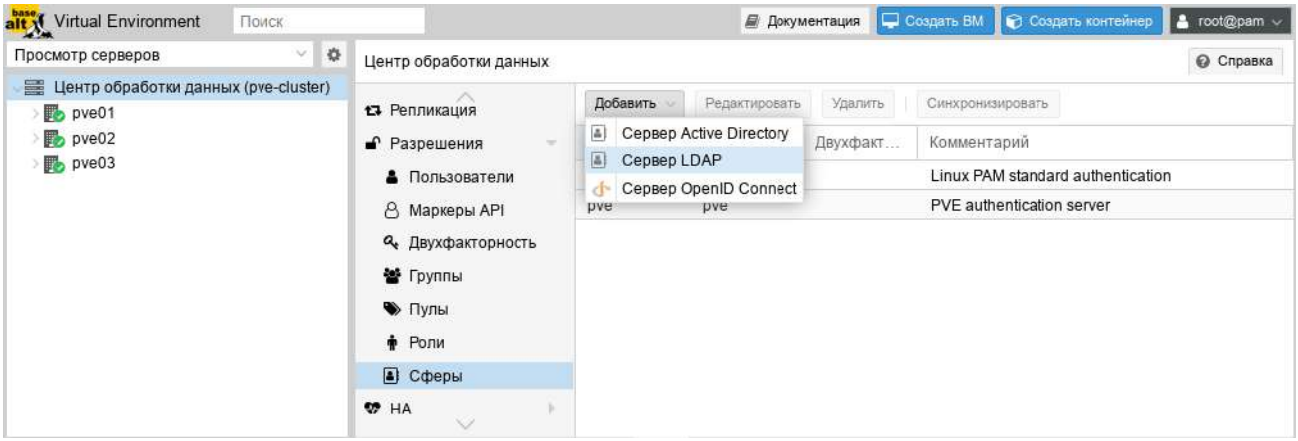


Рис. 311

Настройка LDAP аутентификации (вкладка «Общее»)

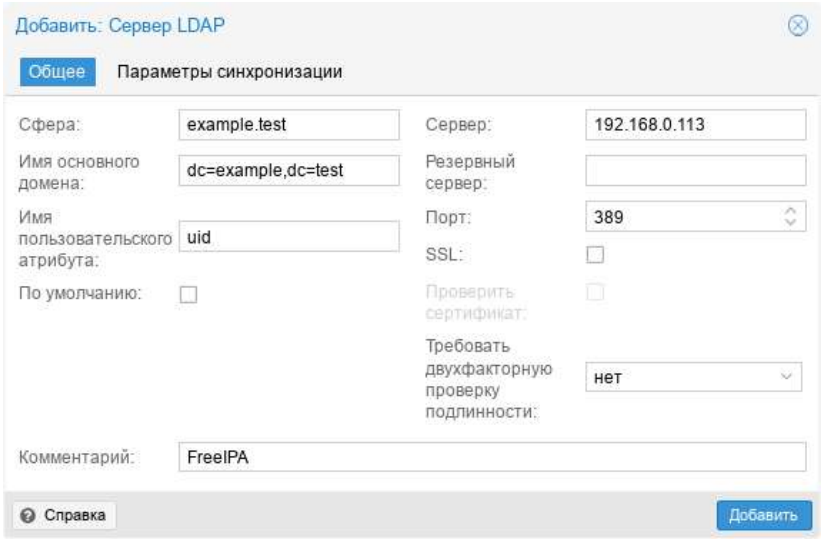


Рис. 312

3) на вкладке «Параметры синхронизации» (Рис. 313) заполнить следующие поля (в скобках указаны значения, используемые в данном примере):

- «Пользователь (bind)» – имя пользователя (*uid=admin,cn=users,cn=accounts,dc=example,dc=test*);
- «Пароль (bind)» – пароль пользователя;
- «Атрибут электронной почты» (опционально);
- «Аттр. имени группы» – атрибут имени группы (*cn*);
- «Классы пользователей» – класс пользователей LDAP (person);
- «Классы групп» – класс групп LDAP (posixGroup);
- «Фильтр пользователей» – фильтр пользователей (*memberOf=cn=pve,cn=groups,cn=accounts,dc=example,dc=test*);
- «Фильтр групп» – фильтр групп (*(|(cn=*pve*)(dc=ipa)(dc=example)(dc=test))*);

4) нажать кнопку «Добавить»;

5) выбрать добавленную область и нажать кнопку «Синхронизировать» (Рис. 314);

6) указать, если необходимо, параметры синхронизации и нажать кнопку «Синхронизировать» (Рис. 315).

В результате синхронизации пользователи и группы PVE будут синхронизированы с сервером FreeIPA LDAP. Сведения о пользователях и группах можно проверить на вкладках «Пользователи» и «Группы».

7) настроить разрешения для группы/пользователя на вкладке «Разрешения».

Настройка LDAP аутентификации (вкладка «Параметры синхронизации»)

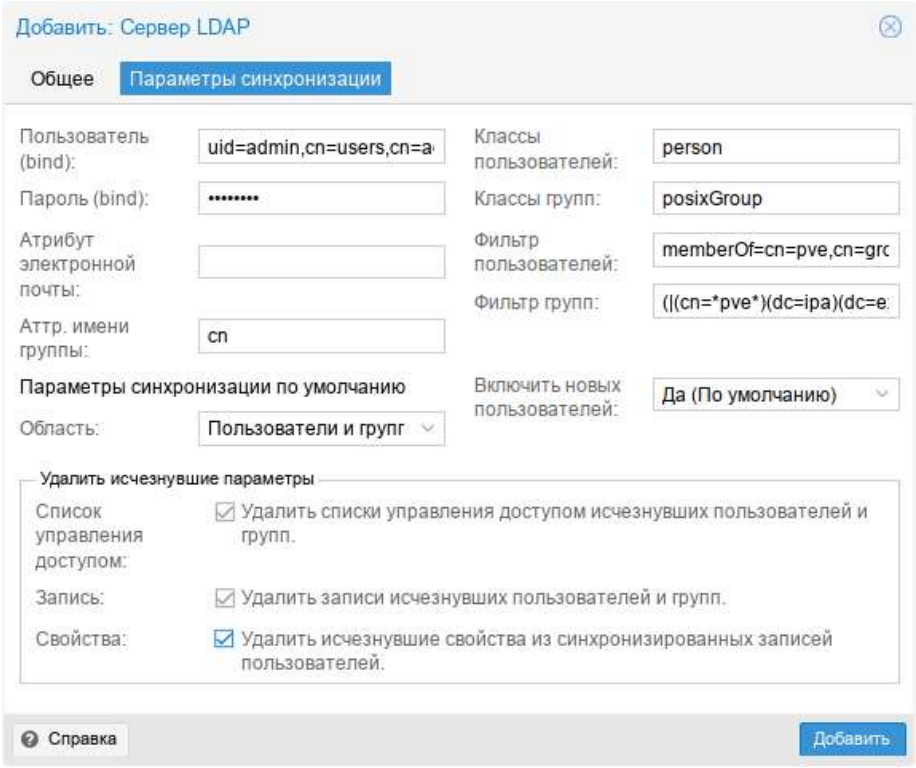


Рис. 313

Кнопка «Синхронизировать»

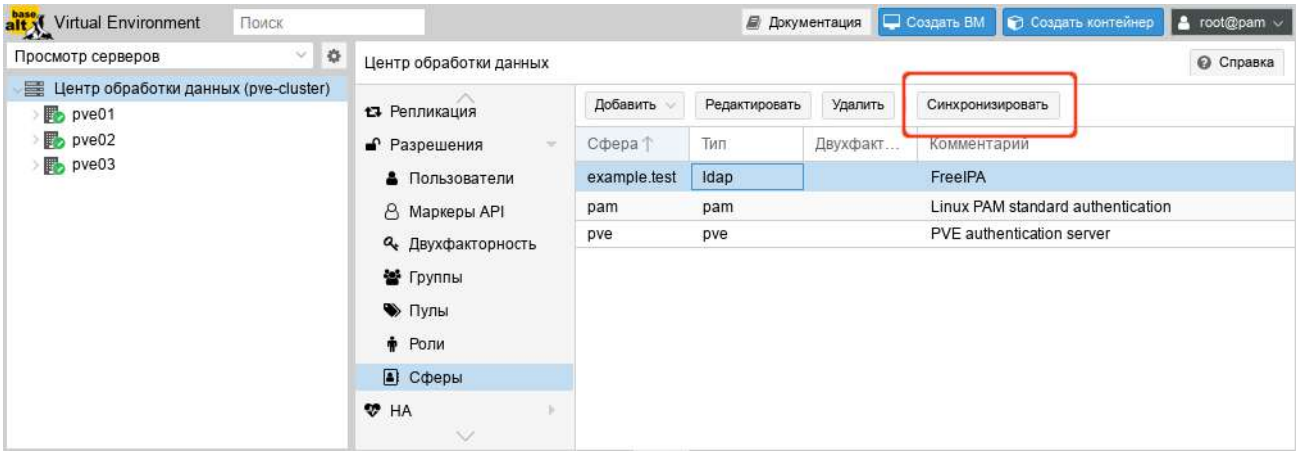


Рис. 314

Параметры синхронизации области аутентификации

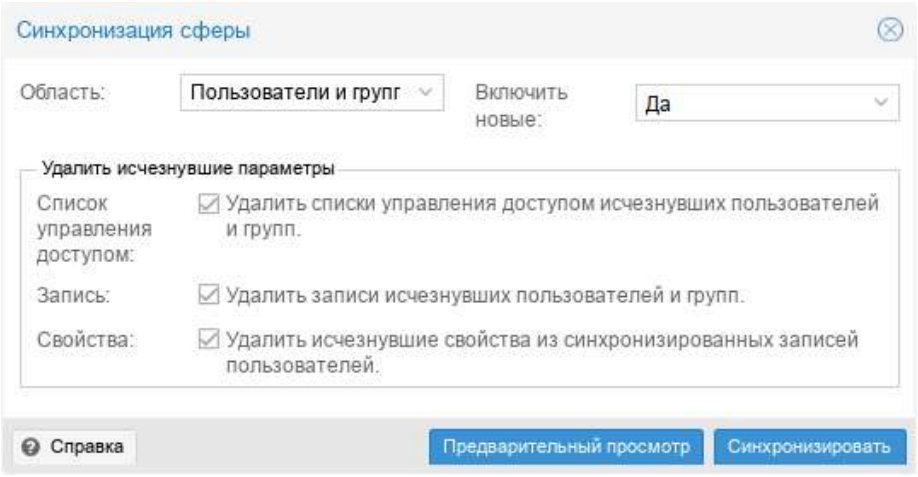


Рис. 315

Примечание. Команда синхронизации пользователей и групп:

```
# pveum realm sync example.test
```

Для автоматической синхронизации пользователей и групп можно добавить команду синхронизации в планировщик задач.

4.18.3.4 AD аутентификация

В данном разделе приведён пример настройки аутентификации на сервере AD. В примере используются следующие исходные данные:

- dc.test.alt, 192.168.0.122 – сервер AD;
- administrator@test.alt – учётная запись администратора (для большей безопасности рекомендуется создать отдельную учетную запись с доступом только для чтения к объектам домена и не использовать учётную запись администратора);
- office – группа, пользователи которой имеют право аутентифицироваться в PVE.

Для настройки AD аутентификации необходимо выполнить следующие шаги:

1) создать область аутентификации LDAP. Для этого в разделе «Центр обработки данных» → «Разрешения» → «Сферы» нажать кнопку «Добавить» → «Сервер LDAP» (Рис. 311);

2) на вкладке «Общее» (Рис. 316) указать следующие данные:

- «Сфера» – идентификатор области;
- «Домен» – домен AD (*test.alt*);
- «По умолчанию» – установить область в качестве области по умолчанию для входа в систему;
- «Сервер» – IP-адрес или имя сервера AD (dc.test.alt или 192.168.0.122);
- «Резервный сервер» (опционально) – адрес резервного сервера на случай, если основной сервер недоступен;
- «Порт» – порт, который прослушивает сервер LDAP (обычно 389 без ssl, 636 с ssl);

- «SSL» – использовать ssl;
- «Требовать двухфакторную проверку подлинности» – требовать двухфакторную аутентификацию.

Настройка AD аутентификации (вкладка «Общее»)

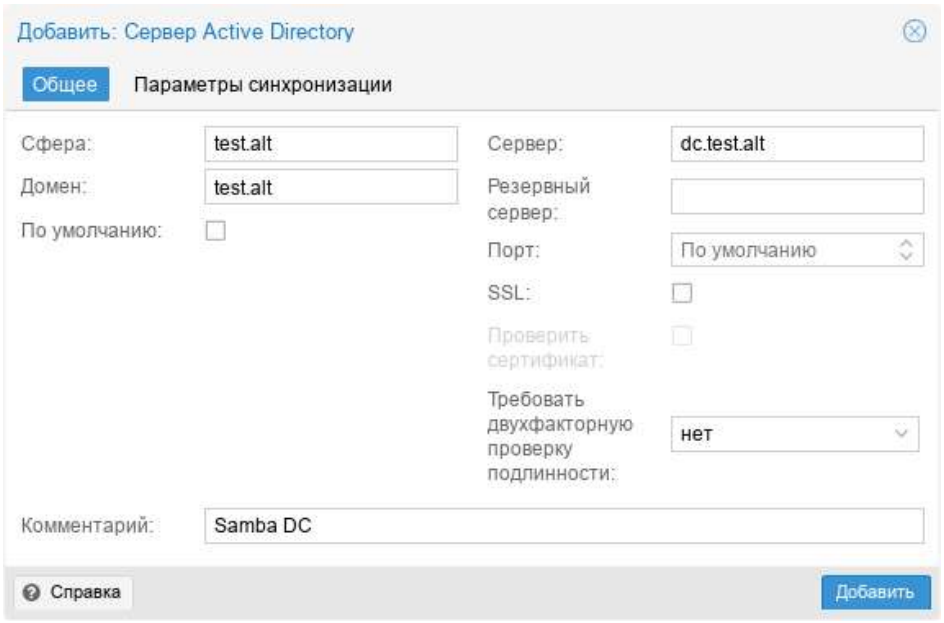


Рис. 316

3) на вкладке «Параметры синхронизации» (Рис. 317) заполнить следующие поля (в скобках указаны значения, используемые в данном примере):

- «Пользователь (bind)» – имя пользователя (*cn=Administrator,cn=Users,dc=test,dc=alt*);
- «Пароль (bind)» – пароль пользователя;
- «Атрибут электронной почты» (опционально);
- «Аттр. имени группы» – атрибут имени группы (*cn*);
- «Классы пользователей» – класс пользователей LDAP;
- «Классы групп » – класс групп LDAP;
- «Фильтр пользователей » – фильтр пользователей (*(&(objectclass=user)(samaccountname=*)(MemberOf=CN=office,ou=OU,dc=TEST,dc=ALT))*);
- « Фильтр групп» – фильтр групп (*(|(cn=*office*)(dc=dc)(dc=test)(dc=alt))*);

4) нажать кнопку «Добавить»;

5) выбрать добавленную область и нажать кнопку «Синхронизировать»;

6) указать, если необходимо, параметры синхронизации и нажать кнопку «Синхронизировать» (Рис. 315).

В результате синхронизации пользователи и группы PVE будут синхронизированы с сервером AD. Сведения о пользователях и группах можно проверить на вкладках «Пользователи» и «Группы».

7) Настроить разрешения для группы/пользователя на вкладке «Разрешения».

Настройка AD аутентификации (вкладка «Параметры синхронизации»)

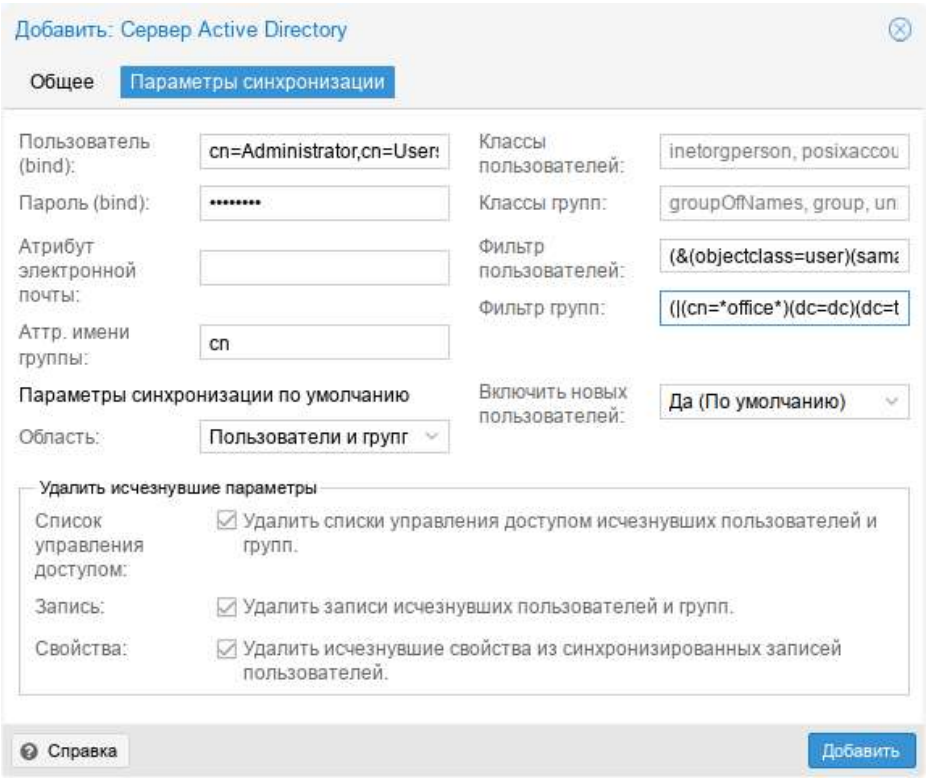


Рис. 317

Примечание. Команда синхронизации пользователей и групп:

```
# pveum realm sync test.alt
```

Для автоматической синхронизации пользователей и групп можно добавить команду синхронизации в планировщик задач.

4.18.4 Двухфакторная аутентификация

В PVE можно настроить двухфакторную аутентификацию двумя способами:

- требование двухфакторной аутентификации (TFA) можно включить при настройке области аутентификации (*Рис. 318*). Если в области аутентификации включена TFA, это становится требованием, и только пользователи с настроенным TFA смогут войти в систему. Новому пользователю необходимо сразу добавить ключи, так как возможности войти в систему без предъявления второго фактора нет;
- пользователи могут сами настроить двухфакторную аутентификацию (Рис. 319), даже если она не требуется в области аутентификации (пункт TFA в выпадающем списке пользователя см. Рис. 320).

Настройка двухфакторной аутентификации при редактировании области

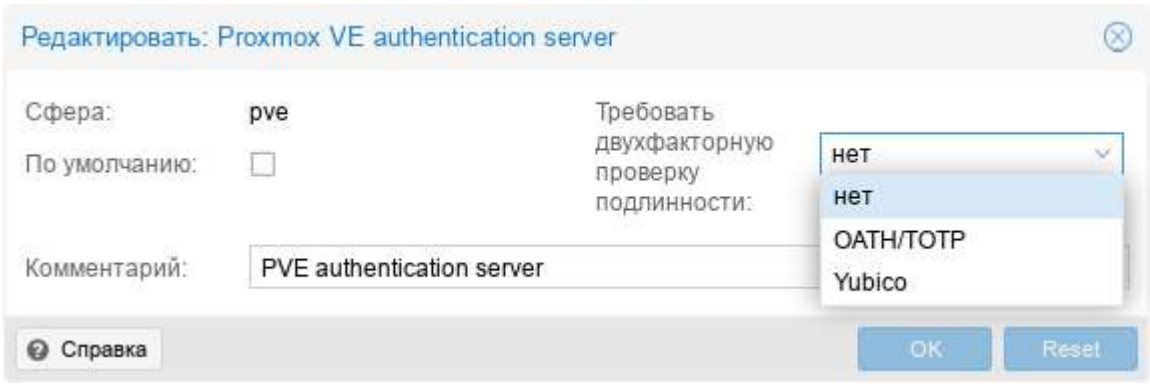


Рис. 319

Настройка двухфакторной аутентификации пользователем

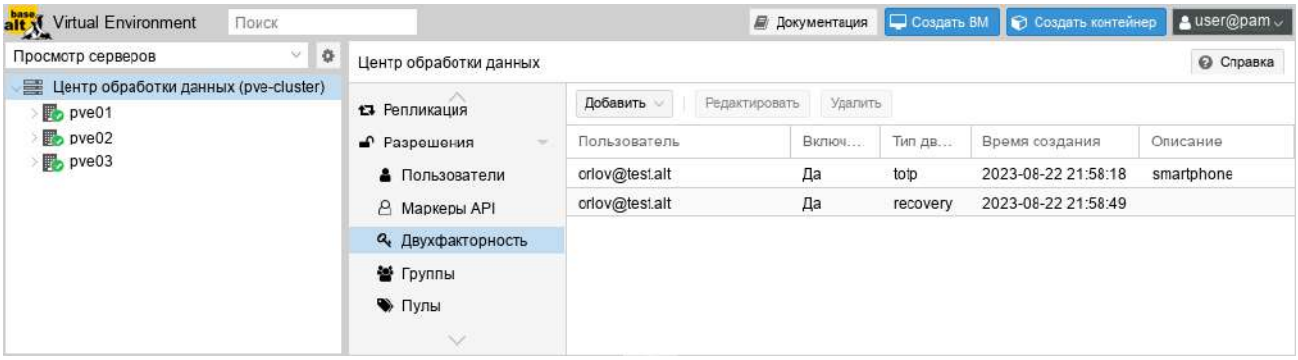


Рис. 320

Меню пользователя

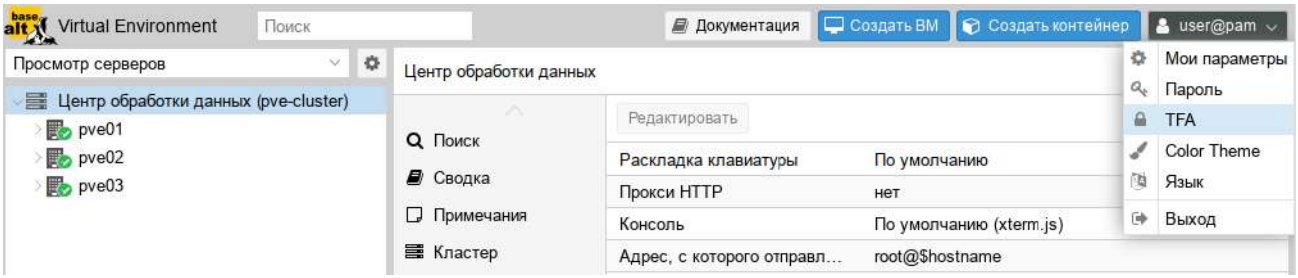


Рис. 321

При добавлении в области аутентификации доступны следующие методы двухфакторной аутентификации (Рис. 318):

- «OATH/TOTP» (основанная на времени OATH) – используется стандартный алгоритм HMAC-SHA1, в котором текущее время хэшируется с помощью настроенного пользователем ключа. Параметры временного шага и длины пароля настраиваются (Рис. 321).

  У пользователя может быть настроено несколько ключей (разделенных пробелами), и ключи могут быть указаны в Base32 (RFC3548) или в шестнадцатеричном представлении.

PVE предоставляет инструмент генерации ключей (`oathkeygen`), который печатает случайный ключ в нотации Base32. Этот ключ можно использовать непосредственно с различными инструментами OTP, такими как инструмент командной строки `oathtool`, или приложении FreeOTP и в других подобных приложениях.

- «Yubico» (YubiKey OTP) – для аутентификации с помощью YubiKey необходимо настроить идентификатор API Yubico, ключ API и URL-адрес сервера проверки, а у пользователей должен быть доступен YubiKey. Чтобы получить идентификатор ключа от YubiKey, следует активировать YubiKey после подключения его через USB и скопировать первые 12 символов введенного пароля в поле ID ключа пользователя.

Основанная на времени OATH (TOTP)

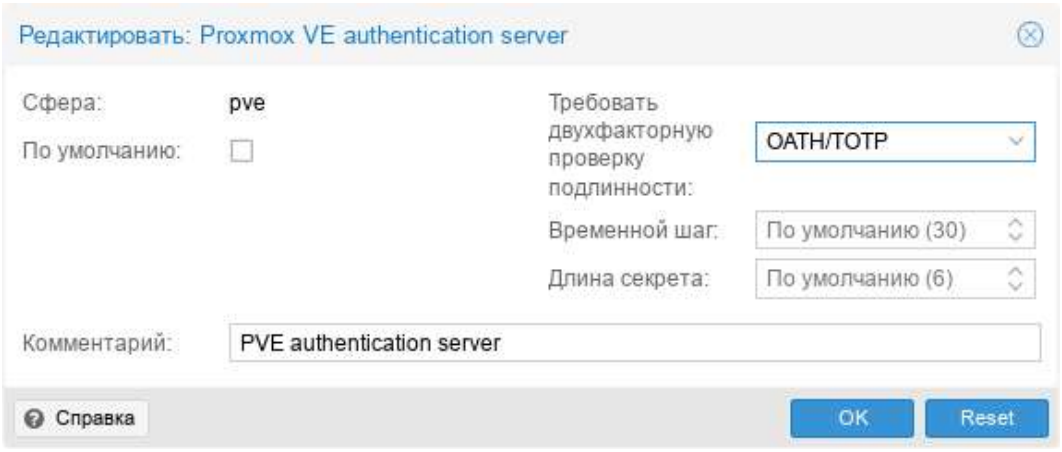


Рис. 321

В дополнение к TOTP и Yubikey OTP пользователям доступны следующие методы двухфакторной аутентификации (Рис. 319):

- «TOTP» (одноразовый пароль на основе времени) – для создания этого кода используется алгоритм одноразового пароля с учетом времени входа в систему (код меняется каждые 30 секунд);
- «WebAuthn» (веб-аутентификация) – реализуется с помощью различных устройств безопасности, таких как аппаратные ключи или доверенные платформенные модули (TPM). Для работы веб-аутентификации необходим сертификат HTTPS;
- «Ключи восстановления» (одноразовые ключи восстановления) – список ключей, каждый из которых можно использовать только один раз. В каждый момент времени у пользователя может быть только один набор одноразовых ключей. Этот метод аутентификации идеально подходит для того, чтобы гарантировать, что пользователь получит доступ, даже если все остальные вторые факторы потеряны или повреждены.

Примечание. Пользователи могут использовать TOTP или WebAuthn в качестве второго фактора при входе в систему, только если область аутентификацию не применяет YubiKey OTP.

Примечание. Чтобы избежать ситуации, когда потеря электронного ключа навсегда блокирует доступ можно настроить несколько вторых факторов для одной учетной записи (Рис. 322).

Несколько настроенных вторых факторов для учётной записи

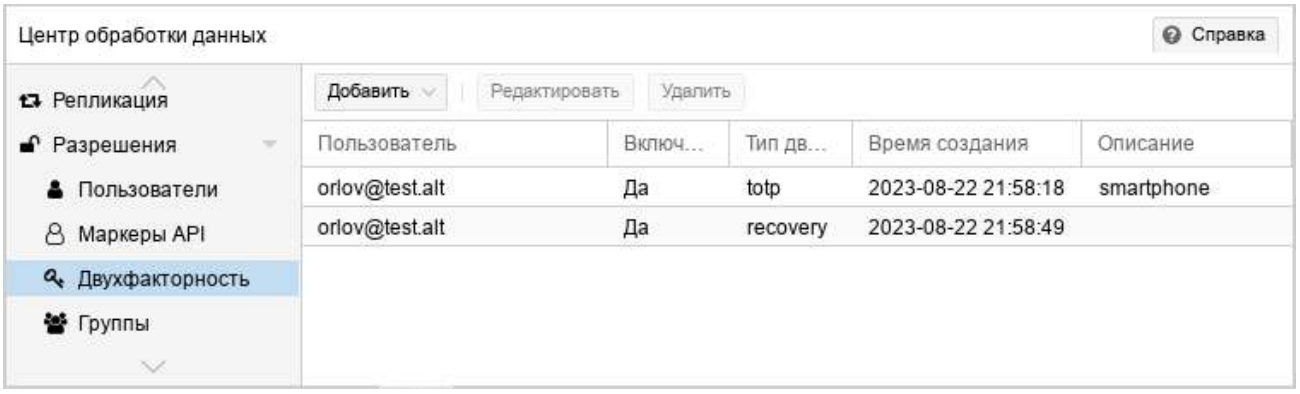


Рис. 322

Процедура добавления аутентификации «TOTP» показана на Рис. 323. При аутентификации пользователя будет запрашиваться второй фактор (Рис. 324).

PVE. Настройка аутентификации TOTP

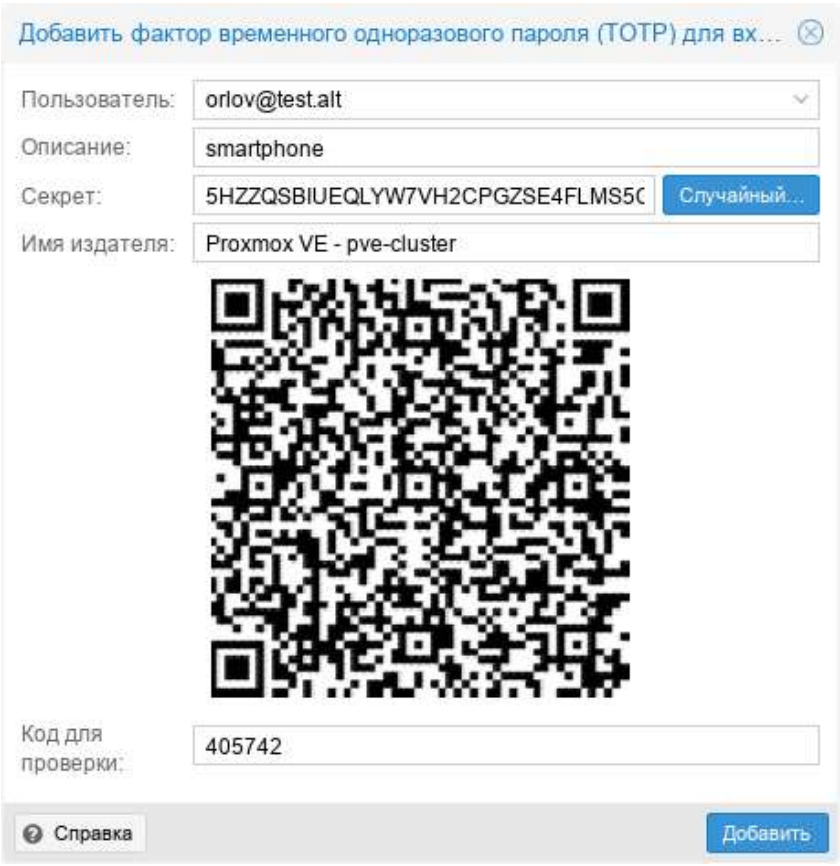


Рис. 323

Запрос второго фактора (TOTP) при аутентификации пользователя в веб-интерфейсе

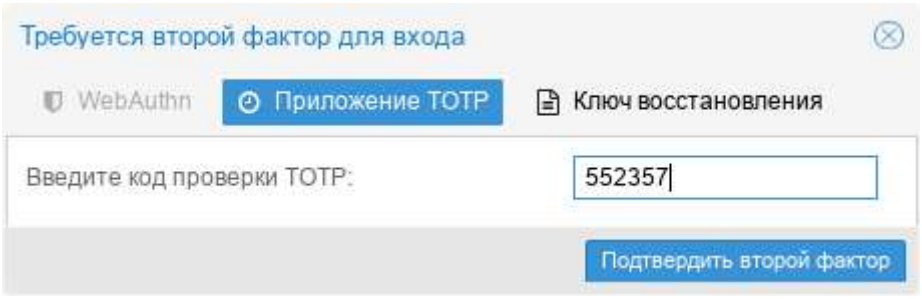


Рис. 324

При настройке аутентификации «Ключи восстановления» необходимо создать набор ключей (Рис. 325). При аутентификации пользователя будет запрашиваться второй фактор (Рис. 326).

PVE. Настройка аутентификации «Ключи восстановления»

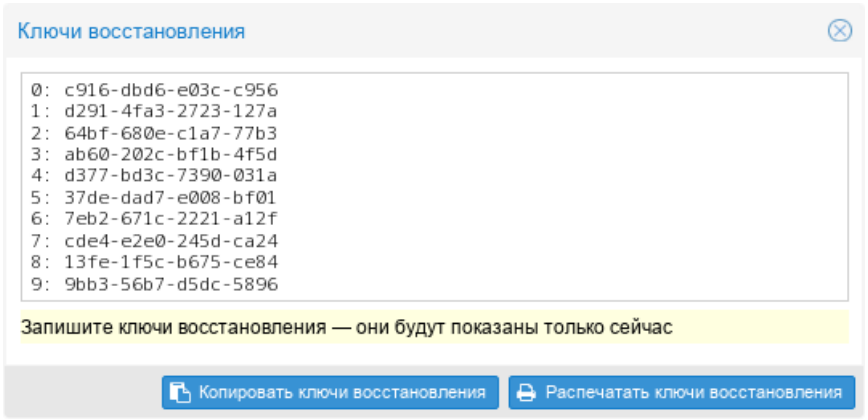


Рис. 325

Запрос второго фактора («Ключи восстановления») при аутентификации пользователя в веб-интерфейсе

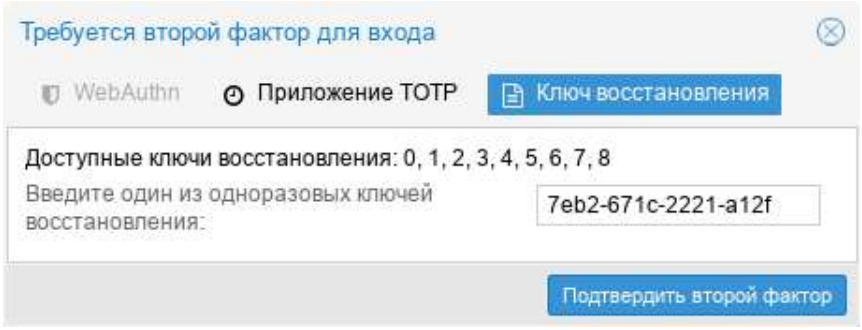


Рис. 326

4.18.5 Управление доступом

Чтобы пользователь мог выполнить какое-либо действие (например, просмотр, изменение или удаление ВМ), ему необходимо иметь соответствующие разрешения.

PVE использует систему управления разрешениями на основе ролей и путей. Запись в таблице разрешений позволяет пользователю или группе играть определенную роль при доступе к

объекту или пути. Это означает, что такое правило доступа может быть представлено как тройка (путь, пользователь, роль) или (путь, группа, роль), причем роль содержит набор разрешенных действий, а путь представляет цель этих действий.

Роль – это список привилегий. В PVE предопределён ряд ролей:

- Administrator – имеет все привилегии;
- NoAccess – нет привилегий (используется для запрета доступа);
- PVEAdmin – все привилегии, кроме прав на изменение настроек системы (Sys.PowerMgmt, Sys.Modify, Realm.Allocate);
- PVEAuditor – доступ только для чтения;
- PVEDatastoreAdmin – создание и выделение места для резервного копирования и шаблонов;
- PVEDatastoreUser – выделение места для резервной копии и просмотр хранилища;
- PVEPoolAdmin – выделение пулов, просмотр пулов;
- PVEPoolUser – просмотр пулов;
- PVESDNAdmin – выделение и просмотр SDN;
- PVESysAdmin – ACL пользователя, аудит, системная консоль и системные журналы;
- PVETemplateUser – просмотр и клонирование шаблонов;
- PVEUserAdmin – администрирование пользователей;
- PVEVMAdmin – управление ВМ;
- PVEVMUser – просмотр, резервное копирование, настройка CDROM, консоль ВМ, управление питанием ВМ.

Просмотреть список предопределенных ролей в веб-интерфейсе можно, выбрав «Центр обработки данных» → «Разрешения»→«Роли» (Рис. 327).

Список предопределенных ролей

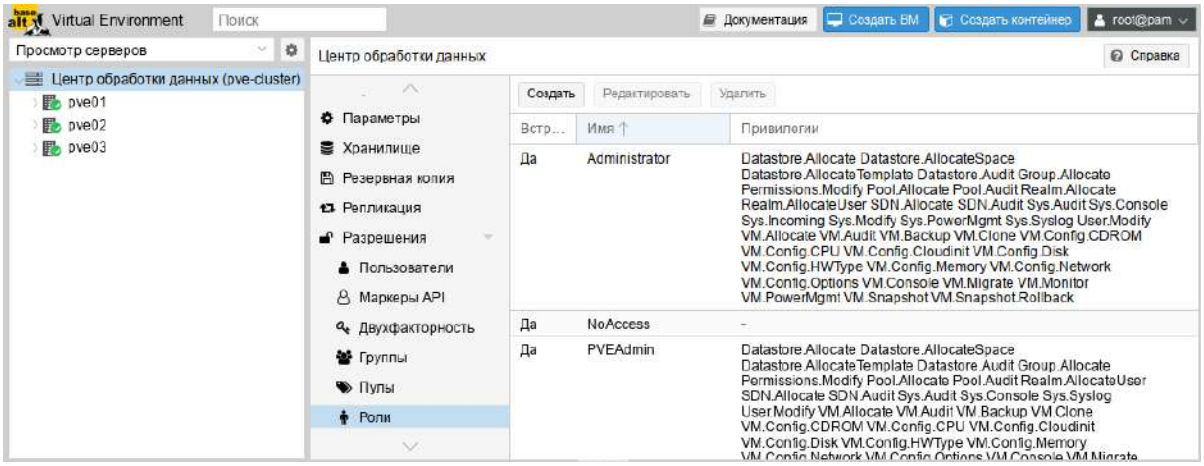


Рис. 327

Добавить новую роль можно как в веб-интерфейсе, так и в командной строке.

Пример добавления роли в командной строке:

```
# pveum role add VM_Power-only --privs "VM.PowerMgmt VM.Console"
```

Привилегия – это право на выполнение определенного действия. Для упрощения управления списки привилегий сгруппированы в роли, которые затем можно использовать в таблице разрешений. Привилегии не могут быть напрямую назначены пользователям, не будучи частью роли. Список используемых привилегий приведен в табл. 29.

Т а б л и ц а 29 – Привилегии используемые в PVE

Привилегия	Описание
Привилегии узла/системы	
Permissions.Modify	Изменение прав доступа
Sys.PowerMgmt	Управление питанием узла (запуск, остановка, сброс, выключение)
Sys.Console	Консольный доступ к узлу
Sys.Syslog	Просмотр Syslog
Sys.Audit	Просмотр состояния/конфигурации узла, конфигурации кластера Corosync и конфигурации HA
Sys.Modify	Создание/удаление/изменение параметров сети узла
Sys.Incoming	Разрешить входящие потоки данных из других кластеров (экспериментально)
Group.Allocate	Создание/удаление/изменение групп
Pool.Allocate	Создание/удаление/изменение пулов
Pool.Audit	Просмотр пула
Realm.Allocate	Создание/удаление/изменение областей аутентификации
Realm.AllocateUser	Назначение пользователю области аутентификации
SDN.Allocate	Управление конфигурацией SDN
SDN.Audit	Просмотр конфигурации SDN
User.Modify	Создание/удаление/изменение пользователя
Права, связанные с ВМ	
VM.Allocate	Создание/удаление ВМ
VM.Migrate	Миграция ВМ на альтернативный сервер в кластере
VM.PowerMgmt	Управление питанием (запуск, остановка, сброс, выключение)
VM.Console	Консольный доступ к ВМ
VM.Monitor	Доступ к монитору виртуальной машины (kvm)
VM.Backup	Резервное копирование/восстановление ВМ
VM.Audit	Просмотр конфигурации ВМ
VM.Clone	Клонирование ВМ
VM.Config.Disk	Добавление/изменение/удаление дисков ВМ
VM.Config.CDROM	Извлечь/изменить CDROM
VM.Config.CPU	Изменение настроек процессора

Привилегия	Описание
VM.Config.Memory	Изменение настроек памяти
VM.Config.Network	Добавление/изменение/удаление сетевых устройств
VM.Config.HWType	Изменение типа эмуляции
VM.Config.Options	Изменение любой другой конфигурации ВМ
VM.Config.Cloudinit	Изменение параметров Cloud-init
VM.Snapshot	Создание/удаление снимков ВМ
VM.Snapshot.Rollback	Откат ВМ к одному из её снимков
Права, связанные с хранилищем	
Datastore.Allocate	Создание/удаление/изменение хранилища данных
Datastore.AllocateSpace	Выделить место в хранилище
Datastore.AllocateTemplate	Размещение/загрузка шаблонов контейнеров и ISO-образов
Datastore.Audit	Просмотр хранилища данных

Права доступа назначаются объектам, таким как ВМ, хранилища или пулы ресурсов. PVE использует файловую систему как путь к этим объектам. Эти пути образуют естественное дерево, и права доступа более высоких уровней (более короткий путь) могут распространяться вниз по этой иерархии.

Путь может представлять шаблон. Когда API-вызов требует разрешений на шаблонный путь, путь может содержать ссылки на параметры вызова API. Эти ссылки указываются в фигурных скобках. Некоторые параметры неявно берутся из URI вызова API. Например, путь /nodes/{node} при вызове /nodes/pve01/status требует разрешений на /nodes/pve01, в то время как путь {path} в запросе PUT к /access/acl ссылается на параметр метода path.

Примеры:

- /nodes/{node} – доступ к узлам PVE;
- /vms – распространяется на все ВМ;
- /vms/{vmid} – доступ к определенным ВМ;
- /storage/{storeid} – доступ к определенным хранилищам;
- /pool/{poolid} – доступ к ресурсам из определенного пула ресурсов;
- /access/groups – администрирование групп;
- /access/realms/{realmid} – административный доступ к области аутентификации.

Используются следующие правила наследования:

- разрешения для отдельных пользователей всегда заменяют разрешения для групп;
- разрешения для групп применяются, если пользователь является членом этой группы;
- разрешения на более глубоких уровнях перекрывают разрешения, унаследованные от верхнего уровня.

Кроме того, токены с разделением привилегий (см. «API-токены») не могут обладать разрешениями на пути, которых нет у связанного с ними пользователя.

Для назначения разрешений необходимо в окне «Центр обработки данных» → «Разрешения» нажать кнопку «Добавить» (Рис. 328), в выпадающем меню выбрать «Разрешения группы», если разрешения назначаются группе пользователей, или «Разрешения пользователя», если разрешения назначаются пользователю. Далее в открывшемся окне (Рис. 329) выбрать путь, группу и роль и нажать кнопку «Добавить».

Добавление разрешений группе

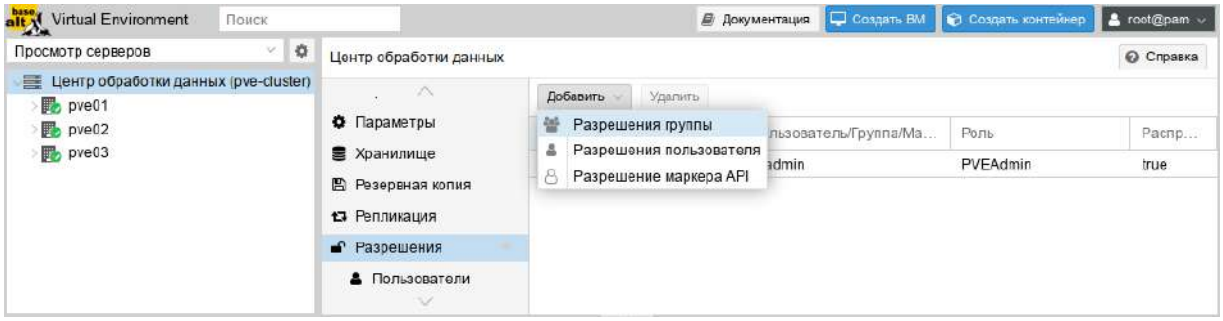


Рис. 328

Добавление разрешений группе

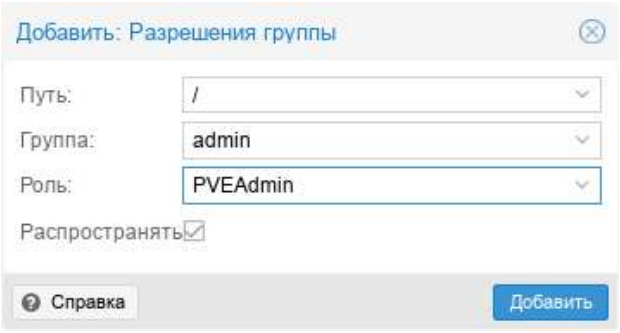


Рис. 329

Примеры работы с разрешениями в командной строке:

- предоставить группе admin полные права администратора:

```
# pveum acl modify / --groups admin --roles Administrator
```

- предоставить пользователю test@pve доступ к ВМ только для чтения:

```
# pveum acl modify /vms --users test@pve --roles PVEAuditor
```

- делегировать управление пользователями пользователю test@pve:

```
# pveum acl modify /access --users test@pve --roles PVEUserAdmin
```

- разрешить пользователю orlov@test.alt изменять пользователей в области test.alt, если они являются членами группы office-test.alt:

```
# pveum acl modify /access/realm/test.alt --users orlov@test.alt --roles
PVEUserAdmin
```

```
# pveum acl modify /access/groups/office-test.alt --users orlov@test.alt --
roles PVEUserAdmin
```

- разрешить пользователям группы developers администрировать ресурсы, назначенные пулу IT:

```
# pveum acl modify /pool/IT/ --groups developers --roles PVEAdmin
```

- удалить у пользователя test@pve право на просмотр ВМ:

```
# pveum acl delete /vms --users test@pve --roles PVEAuditor
```

Примечание. Назначение привилегий на токены см. в разделе «API-токены».

4.19 Просмотр событий PVE

При устранении неполадок сервера, например, неудачных заданий резервного копирования, полезно иметь журнал ранее выполненных задач.

Действия, такие как, например, создание ВМ, выполняются в фоновом режиме. Такое фоновое задание называется задачей. Вывод каждой задачи сохраняется в отдельный файл журнала. Получить доступ к истории задач узлов можно с помощью команды pvenode task, а также в веб-интерфейсе PVE.

4.19.1 Просмотр событий с помощью pvenode task

Команды `pvenode task` приведены в табл. 30.

Таблица 30 – Команды `pvenode task`

Команда	Описание
`pvenode task list [Параметры]`	Вывести список выполненных задач для данного узла. - --errors <логическое значение> – вывести только те задачи, которые завершились ошибкой (по умолчанию 0); - --limit <целое число> – количество задач, которые должны быть выведены (по умолчанию 50); - --since <целое число> – отметка времени (эпоха Unix), начиная с которой будут показаны задачи; - --source <active | all | archive> – вывести список активных, всех или завершенных (по умолчанию) задач; - --start <целое число> – смещение, начиная с которого будут выведены задачи (по умолчанию 0); - --statusfilter <строка> – статус задач, которые должны быть показаны; - --typefilter <строка>– вывести задачи указанного типа (например, vzstart, vzdump); - --until <целое число> – отметка времени (эпоха Unix), до которой будут показаны задачи; - --userfilter <строка> – пользователь, чьи задачи будут показаны; - --vmid <целое число> – идентификатор ВМ, задачи которой будут показаны.
`pvenode task log <upid> [Параметры]`	Вывести журнал задачи. - <upid>: <строка> – идентификатор задачи;

Команда	Описание
	- --start <целое число> – при чтении журнала задачи начать с этой строки (по умолчанию 0).
`pvenode task status <upid>`	Вывести статус задачи. - vmid – идентификатор задачи.

Примечание. Формат идентификатора задачи (UPID):

`UPID:$node:$pid:$pstart:$starttime:$dtype:$id:$user`

pid, pstart и starttime имеют шестнадцатеричную кодировку.

Примеры использования команды `pvenode task`:

- получить список завершённых задач, связанных с ВМ 105, которые завершились с ошибкой:

```
# pvenode task list --errors --vmid 105
```

Список задач будет представлен в виде таблицы см. Рис. 330.

Список задач, связанных с ВМ 105

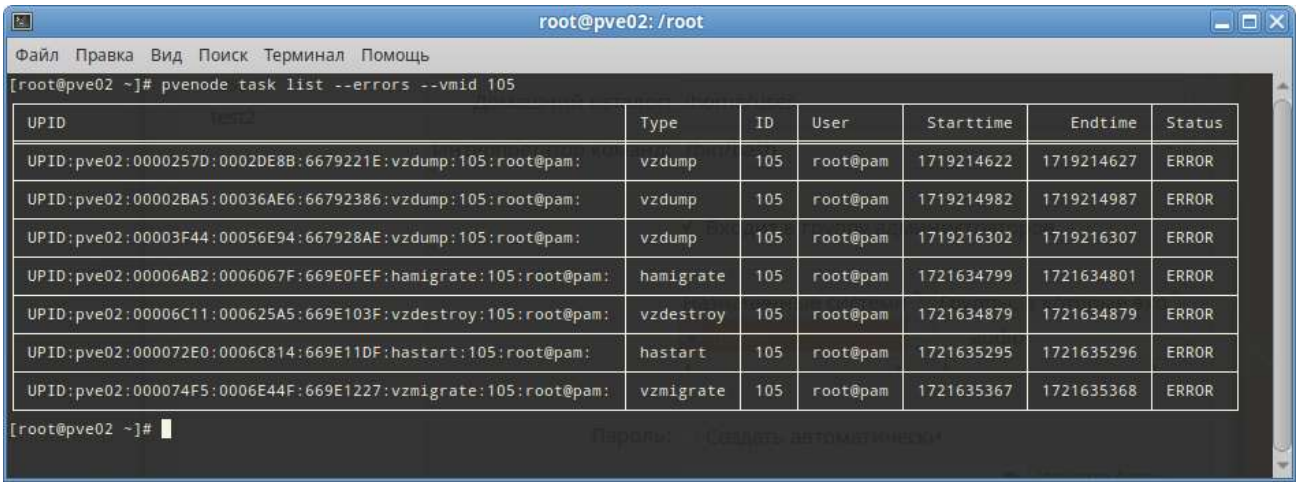


Рис. 330

- получить список задач пользователя user:

```
# pvenode task list --userfilter user
```

- вывести журнал задачи, используя её UPID:

```
# pvenode task log UPID:pve02:0000257D:0002DE8B:6679221E:vzdump:105:root@pam:
INFO: starting new backup job: vzdump 105 --node pve02 --compress zstd --
mailnotification always --notes-template '{{guestname}}' --storage nfs-backup --quiet
1 --mailto test@basealt.ru --mode snapshot
INFO: Starting Backup of VM 105 (lxc)
INFO: Backup started at 2024-06-24 09:37:03
INFO: status = stopped
INFO: backup mode: stop
INFO: ionice priority: 7
```

```
INFO: CT Name: NewLXC
INFO: including mount point rootfs ('/') in backup
INFO: creating vzdump archive '/mnt/pve/nfs-backup/dump/vzdump-lxc-105-2024_06_24-09_37_03.tar.zst'
ERROR: Backup of VM 105 failed - volume 'local:105/vm-105-disk-0.raw' does not exist
INFO: Failed at 2024-06-24 09:37:04
INFO: Backup job finished with errors
postdrop: warning: unable to look up public/pickup: No such file or directory
TASK ERROR: job errors
```

- вывести статус задачи, используя её UPID:

```
# pvenode task status UPID:pve02:0000257D:0002DE8B:6679221E:vzdump:105:root@pam:
```

key	value
exitstatus	job errors
id	105
node	pve02
pid	9597
starttime	1719214622
status	stopped
type	vzdump
upid	UPID:pve02:0000257D:0002DE8B:6679221E:vzdump:105:root@pam:
user	root@pam

4.19.2 Просмотр событий в веб-интерфейсе PVE

4.19.2.1 Панель журнала

Основная цель панели журнала – показать, что в данный момент происходит в кластере. Панель журнала раположена в нижней части интерфейса PVE (Рис. 331).

Панель журнала

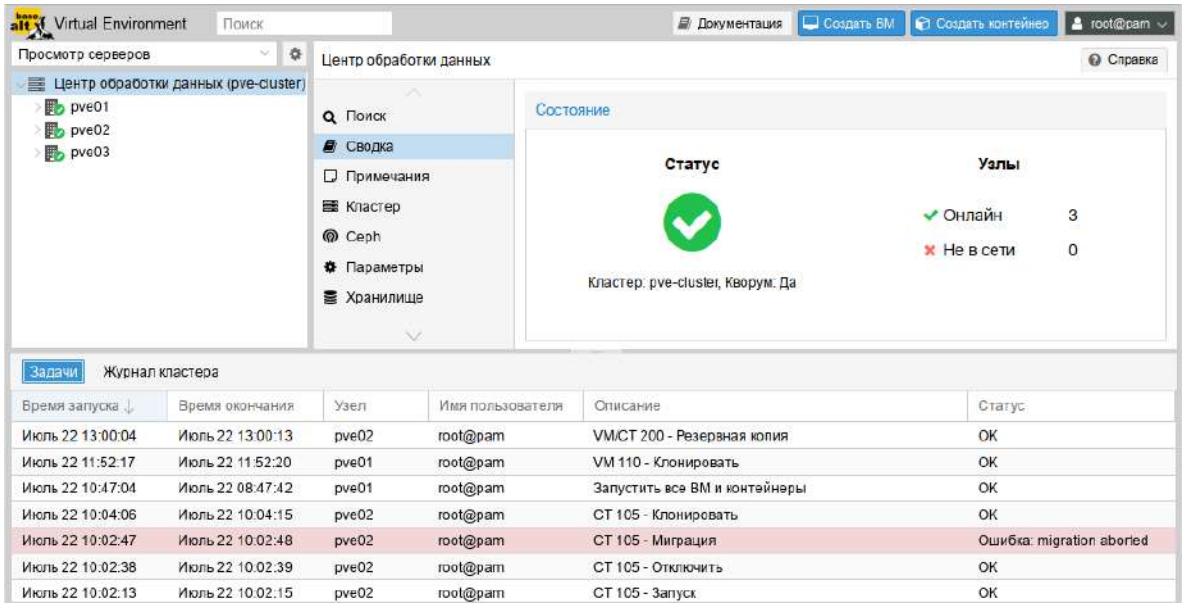


Рис. 331

На панели журнала (вкладка «Задачи») отображаются последние задачи со всех узлов кластера. Таким образом, здесь можно в режиме реального времени видеть, что кто-то еще работает на другом узле кластера.

Для того чтобы получить подробную информацию о задаче или прервать выполнение выполняемой задачи, следует дважды щелкнуть мышью по записи журнала. Откроется окно (Рис. 332) с журналом задачи (вкладка «Выход») и её статусом (вкладка «Статус»). Нажав кнопку «Остановка» можно остановить выполняемую задачу. Кнопка «Загрузка» позволяет сохранить журнал задачи в файл.

Информация о задаче

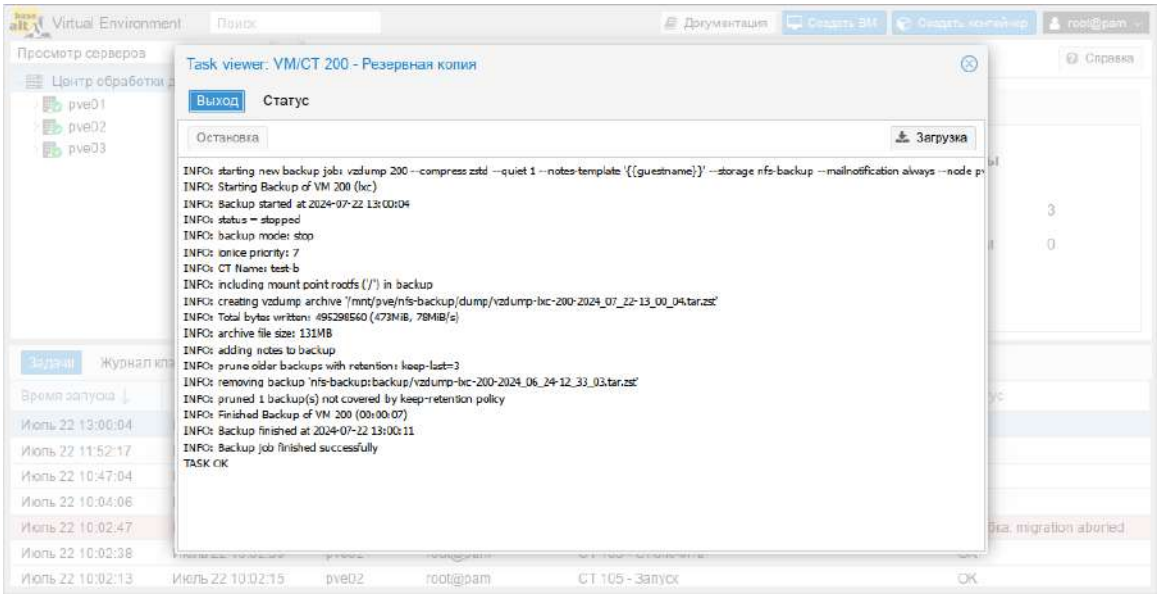


Рис. 332

Примечание. Кнопка «Остановка» доступна только если задача еще выполняется.

Некоторые кратковременные действия просто отправляют логи всем членам кластера. Эти сообщения можно увидеть на панели журнала на вкладке «Журнал кластера».

Примечание. Панель журнала можно полностью скрыть, если нужно больше места для отображения другого контента.

На вкладке «Задачи» панели журнала отображаются записи журнала только для недавних задач. Найти все задачи можно в журнале задач узла PVE.

4.19.2.2 Журнал задач узла PVE

Просмотреть список всех задач узла PVE можно, выбрав «Узел» → «Журнал задач» (Рис. 333). Записи журнала можно отфильтровать. Для этого следует нажать кнопку «Фильтр» и задать нужные значения фильтра (Рис. 334). Просмотреть журнал задачи можно, дважды щелкнув по записи или нажав кнопку «Просмотр».

Журнал задач узла pve01

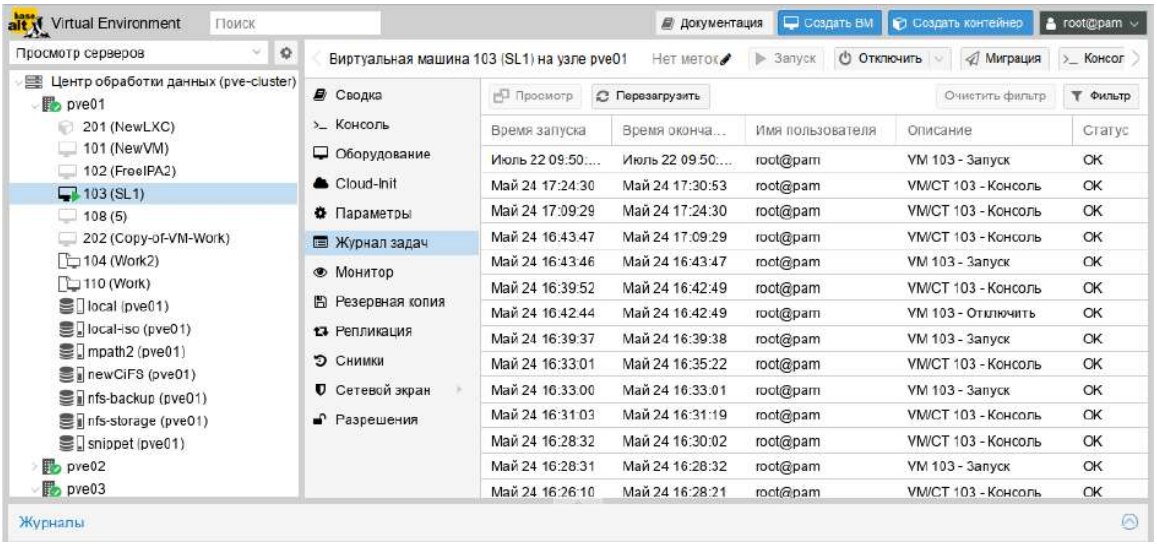


Рис. 333

Отфильтрованные задачи узла pve01

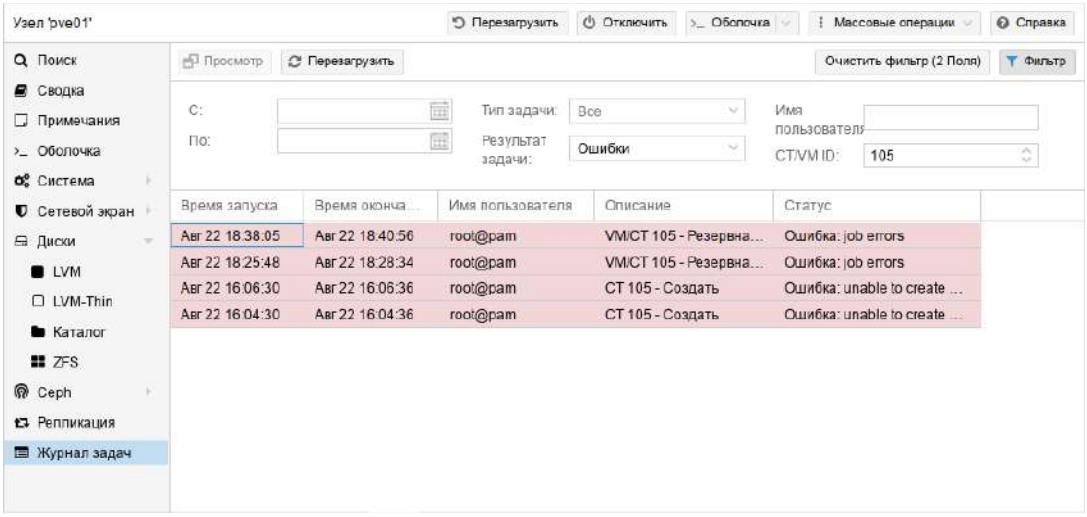


Рис. 334

4.19.2.3 Журнал задач ВМ

Для просмотра задач ВМ необходимо выбрать «Узел» → «ВМ» → «Журнал задач» (Рис. 335). Записи журнала можно отфильтровать. Для этого следует нажать кнопку «Фильтр» и задать нужные значения фильтра (Рис. 336). Просмотреть журнал задачи можно, дважды щелкнув по записи или нажав кнопку «Просмотр».

Журнал задач ВМ 103

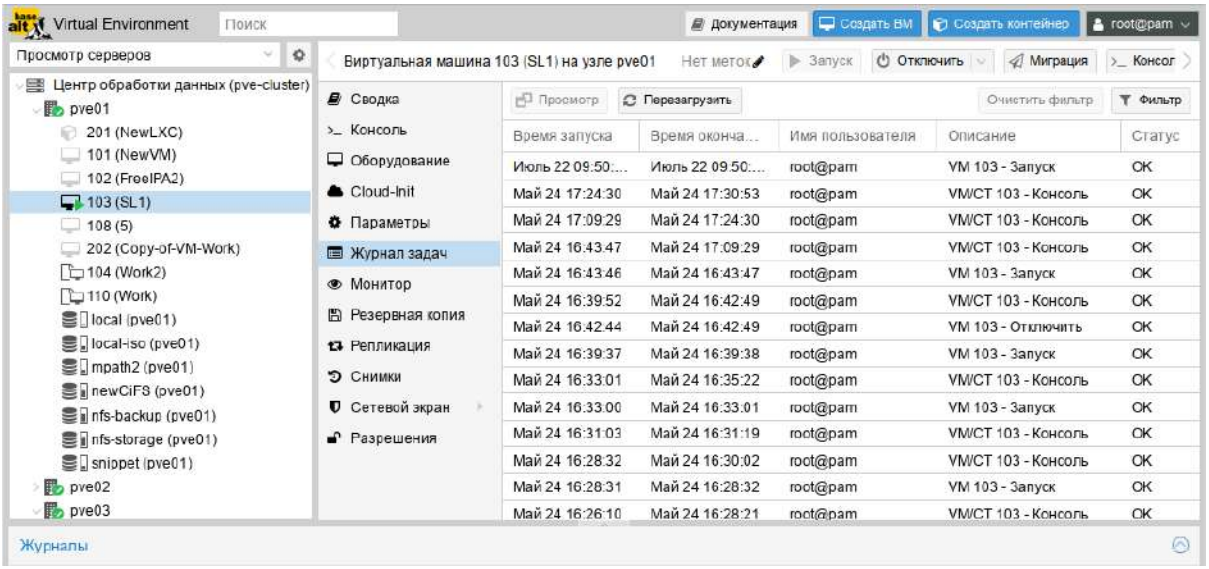


Рис. 335

Задачи ВМ 103 типа qmsnapshot

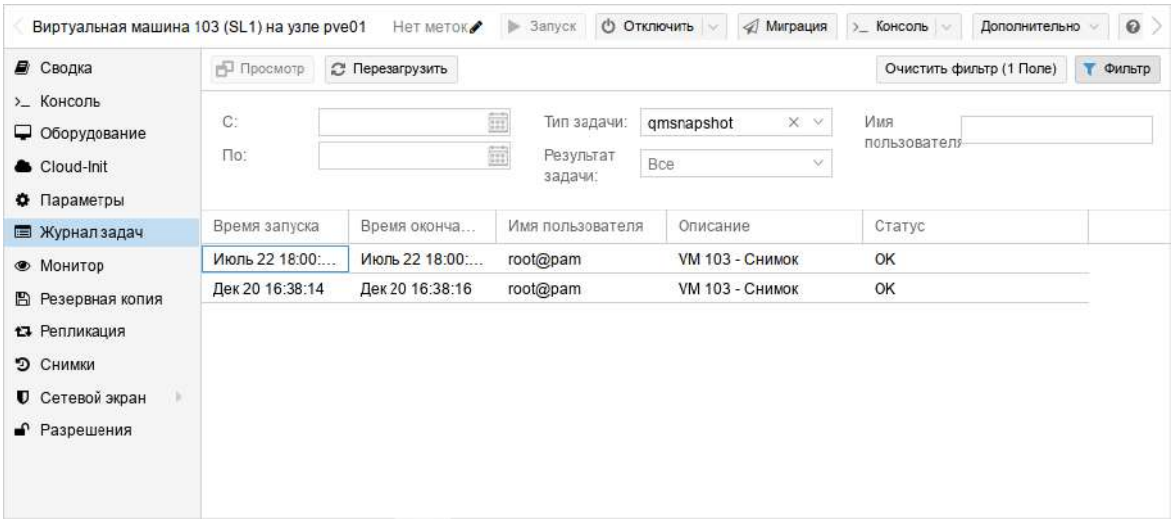


Рис. 336

4.1 PVE API

PVE использует RESTful API. В качестве основного формата данных используется JSON, и весь API формально определен с использованием JSON Schema.

Документация API доступна по адресу: https://docs.altlinux.org/pve-api/v7/index.html

Каждая команда, доступная команде pvesh (см.ниже), доступна в веб-API, поскольку они используют одну и ту же конечную точку.

Запрос (URL, к которому происходит обращение) содержит четыре компонента:

- конечная точка, являющаяся URL-адресом, по которому отправляется запрос;
- метод с типом (GET, POST, PUT, PATCH, DELETE);
- заголовки, выполняющие функции аутентификации, предоставление информации о содержимом тела (допустимо использовать параметр `-H` или `--header` для отправки заголовков HTTP) и т. д.;
- данные (или тело) – то, что отправляется на сервер с помощью опции `-d` или `--data` при запросах POST, PUT, PATCH или DELETE.

Примечание. При передаче не буквенно-цифровых параметров нужно кодировать тело HTTP-запроса. Для этого можно использовать опцию `--data-urlencode`.

HTTP-запросы разрешают работать с базой данных, например:

- GET-запрос на чтение или получение ресурса с сервера;
- POST-запрос для создания записей;
- PUT-запрос для изменения записей;
- DELETE-запрос для удаления записей;
- PATCH-запрос для обновления записей.

Для передачи команд через REST API можно использовать утилиту `curl`.

Примечание. По мере роста числа пользователей и ВМ, API PVE может начать реагировать на изменения с задержкой. Для решения этой проблемы нужно очистить `/var/lib/rrdcached/`, например, выполнив команду:

```
# find /var/lib/rrdcached -type f -mtime +5 -delete
```

Или добавив соответствующее задание в crontab.

4.1.1 URL API

API PVE использует протокол HTTPS, а сервер прослушивает порт 8006. Таким образом, базовый URL для API – https://<имя-компьютера>:8006/api2/json/

Параметры можно передавать с помощью стандартных методов HTTP:

- через URL;
- используя x-www-form-urlencoded content-type для запросов PUT и POST.

В URL можно указать формат возвращаемых данных:

- json – формат JSON;
- extjs – формат JSON, но результат вложен в объект, с объектом данных, вариант, совместимый с формами ExtJS;

- html – текст в формате HTML (иногда полезно для отладки);
- text – формат простой текст (иногда полезно для отладки);

В приведенном выше примере используется JSON.

4.1.2 Аутентификация

Есть два способа доступа к API PVE:

- использование временно сгенерированного токена (билета);
- использование API-токена.

Все API-запросы должны включать в себя билет в заголовке Cookie или отправлять API-токен через заголовок Authorization.

4.1.2.1 Билет Cookie

Билет – это подписанное случайное текстовое значение с указанием пользователя и времени создания. Билеты подписываются общекластерным ключом аутентификации, который обновляется один раз в день.

Кроме того, любой запрос на запись (POST/PUT/DELETE) должен содержать CSRF-токен для предотвращения CSRF-атак (cross-site request forgery).

Пример получения нового билета и CSRF-токена:

```
$ curl -d 'username=root@pam' --data-urlencode 'password=xxxxxxxxx' \
https://192.168.0.186:8006/api2/json/access/ticket
```

Примечание. Если запрос завершается ошибкой вида:

```
curl: (60) SSL certificate problem: unable to get local issuer certificate
```

можно дополнить запрос опцией `--insecure` (`-k`), для отключения проверки валидности сертификатов:

```
$ curl -k -d 'username=root@pam' --data-urlencode 'password=xxxxxxxxx' \
https://192.168.0.186:8006/api2/json/access/ticket
```

Примечание. Параметры командной строки видны всей системе, поэтому следует избегать запуска команды с указанием пароля на ненадежных узлах.

Пример получения нового билета и CSRF-токена с паролем, записанным в файл, доступный для чтения только пользователю:

```
$ curl -k -d 'username=root@pam' \
--data-urlencode "password@$HOME/.pve-pass-file" \
https://192.168.0.186:8006/api2/json/access/ticket
```

Примечание. Для форматированного вывода можно использовать команду `jq` (должен быть установлен пакет `jq`):

```
$ curl -k -d 'username=root@pam' \
--data-urlencode "password@$HOME/.pve-pass-file" \
https://192.168.0.186:8006/api2/json/access/ticket | jq
```

Пример ответа:

```
{
  "data": {
    "ticket":"PVE:root@pam:66AA52D6::d85E+IIFAuG731…",
    "CSRFPreventionToken":"66AA52D6:Y2zvIXjRVpxx4ZG74F14Ab0EHn8NRoso/WmVqZEnAuM",
    "username":"root@pam"
  }
}
```

Примечание. Билет действителен в течение двух часов и должен быть повторно запрошен по истечении срока его действия. Но можно получить новый билет, передав старый билет в качестве пароля методу /access/ticket до истечения срока его действия.

Полученный билет необходимо передавать с Cookie при любом запросе, например:

```
$ curl -k -b "PVEAuthCookie=PVE:root@pam:66AA52D6::d85E+IIFAuG731…" \
https://192.168.0.186:8006/api2/json/
```

Ответ:

```
{
  "data": [
    { "subdir": "version" },
    { "subdir": "cluster" },
    { "subdir": "nodes" },
    { "subdir": "storage" },
    { "subdir": "access" },
    { "subdir": "pools" }
  ]
}
```

Примечание. Для передачи данных в заголовке Cookie используется параметр `--cookie` (`-b`).

Любой запрос на запись (POST, PUT, DELETE) кроме билета должен включать заголовок CSRFPreventionToken, например:

```
$ curl -k -XDELETE \
'https://pve01:8006/api2/json/access/users/testuser@pve' \
-b "PVEAuthCookie=PVE:root@pam:66AA52D6::d85E+IIFAuG731…" \
-H "CSRFPreventionToken: 66AA52D6:Y2zvIXjRVpxx4ZG74F14Ab0EHn8NRoso/WmVqZEnAuM"
```

4.1.2.2 API-токены

API-токены позволяют другой системе, программному обеспечению или API-клиенту получать доступ без сохранения состояния к большинству частей REST API. Токены могут быть сгенерированы для отдельных пользователей и им могут быть предоставлены отдельные разрешения и даты истечения срока действия для ограничения объема и продолжительности

доступа (подробнее см. раздел «API-токены»). Если API-токен будет скомпрометирован, его можно отозвать, не отключая самого пользователя.

Примеры запросов с использованием API-токена:

- получить список пользователей:

```
$ curl \
-H 'Authorization: PVEAPIToken=root@pam!test=373007e1-4ecb-4e56-b843-d0fbed543375' \
https://192.168.0.186:8006/api2/json/access/users
```

- добавить пользователя testuser@pve:

```
$ curl -k -X 'POST' \
'https://pve01:8006/api2/json/access/users' \
--data-urlencode 'userid=testuser@pve' \
-H 'Authorization: PVEAPIToken=root@pam!test=373007e1-4ecb-4e56-b843-d0fbed543375'
```

- удалить пользователя testuser@pve:

```
$ curl -k -X 'DELETE' \
'https://pve01:8006/api2/json/access/users/testuser@pve' \
-H 'Authorization: PVEAPIToken=root@pam!test=373007e1-4ecb-4e56-b843-d0fbed543375'
```

Примечание. API-токены не нуждаются в значениях CSRF для POST, PUT или DELETE запросов. Обычно токены не используются в контексте браузера, поэтому основной вектор атаки CSRF изначально неприменим.

4.1.3 Пример создания контейнера с использованием API

Исходные данные:

- APINODE – узел, на котором производится аутентификация;
- TARGETNODE – узел, на котором будет создан контейнер;
- cookie – файл, в который будет помещен cookie;
- csrftoken – файл, в который будет помещен CSRF-токен.

Пример создания контейнера с использованием API:

1) для удобства установить переменные окружения:

```
$ export APINODE=pve01
$ export TARGETNODE=pve03
```

2) сохранить авторизационный cookie в файл cookie:

```
$ curl --silent --insecure --data "username=root@pam&password=yourpassword" \
 https://$APINODE:8006/api2/json/access/ticket \
| jq --raw-output '.data.ticket' | sed 's/^/PVEAuthCookie=/' > cookie
```

3) сохранить CSRF-токен в файл csrftoken:

```
$ curl --silent --insecure --data "username=root@pam&password=yourpassword" \
 https://$APINODE:8006/api2/json/access/ticket \
| jq --raw-output '.data.CSRFPreventionToken' | sed 's/^/CSRFPreventionToken:/' >
csrftoken
```

4) отобразить статус целевого узла, чтобы проверить, что создание cookie-билета сработало:

```
$ curl --insecure --cookie "$(<cookie)" \
https://$APINODE:8006/api2/json/nodes/$TARGETNODE/status | jq '.'
```

5) создать LXC-контейнер:

```
$ curl --silent -k --cookie "$(<cookie)" --header "$(<csrftoken)" -X POST\
 --data-urlencode net0="name=myct0,bridge=vmbr0" \
 --data-urlencode ostemplate="local:vztmpl/alt-p10-rootfs-systemd-x86_64.tar.xz" \
 --data vmid=601 \
 https://$APINODE:8006/api2/json/nodes/$TARGETNODE/lxc

{"data":"UPID:pve03:00005470:00083F6D:66A76C80:vzcreate:601:root@pam:"}
```

Команда должна вернуть структуру JSON, содержащую идентификатор задачи (UPID).

Примечание. При передаче не буквенно-цифровых параметров нужно кодировать тело HTTP POST.

Примечание. При создании контейнера должен использоваться доступный vmid.

4.1.4 Утилита pvesh

Инструмент управления PVE (`pvesh`) позволяет напрямую вызывать функции API, без использования сервера REST/HTTPS.

```
# pvesh ls /
Dr---        access
Dr---        cluster
Dr---        nodes
Dr-c-        pools
Dr-c-        storage
-r---        version
```

Примечание. `pvesh` может использовать только пользователь root.

Инструмент автоматически проксирует вызовы другим членам кластера с помощью ssh.

Примеры:

- вывести текущую версию:

```
# pvesh get /version
```

- получить список узлов в кластере:

```
# pvesh get /nodes
```

- получить список доступных опций для центра обработки данных:

```
# pvesh usage cluster/options -v
```

- создать нового пользователя:

```
# pvesh create /access/users --userid testuser@pve
```

- удалить пользователя:

```
# pvesh delete /access/users/testuser@pve
```

- установить консоль HTML5 NoVNC в качестве консоли по умолчанию:

```
# pvesh set cluster/options -console html5
```

- создать и запустить новый контейнер на узле pve03:

```
# pvesh create nodes/pve03/lxc -vmid 210 -hostname test --storage local \
 --password "supersecret" \
 --ostemplate nfs-storage:vztmpl/alt-p10-rootfs-systemd-x86_64.tar.xz \
 --memory 512 --swap 512

UPID:pve03:0000286E:0003553C:66A75FE7:vzcreate:210:root@pam:

# pvesh create /nodes/pve03/lxc/210/status/start

UPID:pve03:0000294B:00036B33:66A7601F:vzstart:210:root@pam
```

4.2 Службы PVE

Команды служб PVE на примере `pvedaemon`:

- вывести справку:

```
# pvedaemon help
```

- перезапустить службу (или запустить, если она не запущена):

```
# pvedaemon restart
```

- запустить службу:

```
# pvedaemon start
```

- запустить службу в режиме отладки:

```
# pvedaemon start --debug 1
```

- вывести статус службы:

```
# pvedaemon status
```

- остановить службу:

```
# pvedaemon stop
```

4.2.1 pvedaemon – служба PVE API

Служба `pvedaemon` предоставляет весь API PVE на 127.0.0.1:85. Она работает от имени пользователя root и имеет разрешение на выполнение всех привилегированных операций.

Примечание. Служба слушает только локальный адрес, поэтому к ней нельзя получить доступ извне. Доступ к API извне предоставляет служба pveproxy.

4.2.2 pveproxy – служба PVE API Proxy

Служба `pveproxy` предоставляет весь PVE API на TCP-порту 8006 с использованием HTTPS. Она работает от имени пользователя www-data и имеет минимальные разрешения. Операции, требующие дополнительных разрешений, перенаправляются локальному pvedaemon.

Запросы, предназначенные для других узлов, автоматически перенаправляются на них. Поэтому можно управлять всем кластером, подключившись к одному узлу PVE.

4.2.2.1 Управление доступом на основе хоста

Можно настраивать apache2-подобные списки контроля доступа. Значения считываются из файла `/etc/default/pveproxy`. Например:

```
ALLOW_FROM="10.0.0.1-10.0.0.5,192.168.0.0/22"
DENY_FROM="all"
POLICY="allow"
```

IP-адреса можно указывать с использованием любого синтаксиса, понятного Net::IP. Ключевое слово all является псевдонимом для 0/0 и ::/0 (все адреса IPv4 и IPv6).

Политика по умолчанию – allow.

Правила обработки запросов приведены в табл. 31.

Таблица 31. Правила обработки запросов

Соответствие	**POLICY=deny**	**POLICY=allow**
Соответствует только Allow	Запрос разрешён	Запрос разрешён
Соответствует только Deny	Запрос отклонён	Запрос отклонён
Нет соответствий	Запрос отклонён	Запрос разрешён
Соответствует и Allow и Deny	Запрос отклонён	Запрос разрешён

4.2.2.2 Прослушиваемый IP-адрес

По умолчанию службы `pveproxy` и `spiceproxy` прослушивают подстановочный адрес и принимают соединения от клиентов как IPv4, так и IPv6.

Установив опцию LISTEN_IP в `/etc/default/pveproxy`, можно контролировать, к какому IP-адресу будут привязываться службы `pveproxy` и `spiceproxy`. IP-адрес должен быть настроен в системе.

Установка `sysctl net.ipv6.bindv6only` в значение 1 приведет к тому, что службы будут принимать соединения только от клиентов IPv6, что может вызвать множество проблем. Если устанавливается эта конфигурация, рекомендуется либо удалить настройку `sysctl`, либо установить LISTEN_IP в значение 0.0.0.0 (что позволит использовать только клиентов IPv4).

LISTEN_IP можно использовать только для ограничения сокета внутренним интерфейсом, например:

```
LISTEN_IP="192.168.0.186"
```

Аналогично можно задать IPv6-адрес:

```
LISTEN_IP="2001:db8:85a3::1"
```

Если указывается локальный IPv6-адрес, необходимо указать имя интерфейса, например:

```
LISTEN_IP="fe80::c463:8cff:feb9:6a4e%vmbr0"
```

Примечание. Не рекомендуется устанавливать параметр LISTEN_IP в кластерных системах.

Для применения изменений нужно перезагрузить узел или полностью перезапустить `pve-proxy` и `spiceproxy`:

```
# systemctl restart pveproxy.service spiceproxy.service
```

Примечание. Перезапуск службы `pveproxy`, в отличие от перезагрузки конфигурации (reload), может прервать некоторые рабочие процессы, например, запущенную консоль или оболочку ВМ. Поэтому следует дождаться остановки системы на обслуживания, чтобы это изменение вступило в силу.

4.2.2.3 Набор SSL-шифров

Список шифров можно определить в `/etc/default/pveproxy` с помощью ключей CIPHERS (TLS = 1.2) и CIPHERSUITES (TLS >= 1.3), например:

```
CIPHERS="ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-SHA384:ECDHE-
ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-GCM-
SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-RSA-
AES256-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256"
CIPHERSUITES="TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256:TLS_AES_128
_GCM_SHA256"
```

Кроме того, можно настроить клиент на выбор шифра, используемого в `/etc/default/pveproxy` (по умолчанию используется первый шифр в списке, доступном как клиенту, так и `pveproxy`):

```
HONOR_CIPHER_ORDER=0
```

4.2.2.4 Поддерживаемые версии TLS

Для отключения TLS версий 1.2 или 1.3, необходимо установить следующий параметр в `/etc/default/pveproxy`:

```
DISABLE_TLS_1_2=1
```

или, соответственно:

```
DISABLE_TLS_1_3=1
```

Примечание. Если нет особой причины, не рекомендуется вручную настраивать поддерживаемые версии TLS.

4.2.2.5 Параметры Диффи-Хеллмана

Определить используемые параметры Диффи-Хеллмана можно в `/etc/default/pveproxy`, указав в параметре `DHPARAMS` путь к файлу, содержащему параметры DH в формате PEM, например:

```
DHPARAMS="/path/to/dhparams.pem"
```

Примечание. Параметры DH используются только в том случае, если согласован набор шифров, использующий алгоритм обмена ключами DH.

4.2.2.6 Альтернативный сертификат HTTPS

`pveproxy` использует `/etc/pve/local/pveproxy-ssl.pem` и `/etc/pve/local/pveproxy-ssl.key`, если они есть, или `/etc/pve/local/pve-ssl.pem` и `/etc/pve/local/pve-ssl.key` в противном случае. Закрытый ключ не может использовать парольную фразу.

Можно переопределить местоположение закрытого ключа сертификата `/etc/pve/local/pveproxy-ssl.key`, установив параметр TLS_KEY_FILE в `/etc/default/pveproxy`, например:

```
TLS_KEY_FILE="/secrets/pveproxy.key"
```

4.2.2.7 Сжатие ответа

По умолчанию `pveproxy` использует сжатие gzip HTTP-уровня для сжимаемого контента, если клиент его поддерживает. Это поведение можно отключить в `/etc/default/pveproxy`:

```
COMPRESSION=0
```

4.2.3 pvestatd – служба PVE Status

Служба `pveproxy` запрашивает статус ВМ, хранилищ и контейнеров через регулярные интервалы. Результат отправляется на все узлы кластера.

4.2.4 spiceproxy – служба SPICE Proxy

Служба `spiceproxy` прослушивает TCP-порт 3128 и реализует HTTP-прокси для пересылки запроса CONNECT от SPICE-клиента к ВМ PVE. Она работает от имени пользователя www-data и имеет минимальные разрешения.

Можно настраивать apache2-подобные списки контроля доступа. Значения считываются из файла `/etc/default/pveproxy`. Подробнее см. «Управление доступом на основе хоста».

4.2.5 pvescheduler – служба PVE Scheduler

Служба `pvescheduler` отвечает за запуск заданий по расписанию, например, заданий репликации и vzdump.

Для заданий vzdump служба получает свою конфигурацию из файла `/etc/pve/jobs.cfg`.

5 УПРАВЛЕНИЕ ВИРТУАЛИЗАЦИЕЙ НА ОСНОВЕ LIBVIRT

5.1 Установка и настройка libvirt

libvirt – это набор инструментов, предоставляющий единый API к множеству различных технологий виртуализации.

Кроме управления виртуальными машинами/контейнерами libvirt поддерживает управление виртуальными сетями и управление хранением образов.

Для управления из консоли разработан набор утилит virt-install, virt-clone, virsh и других. Для управления из графической оболочки можно воспользоваться virt-manager.

Любой виртуальный ресурс, необходимый для создания ВМ (compute, network, storage) представлен в виде объекта в libvirt. За процесс описания и создания этих объектов отвечает набор различных XML-файлов. Сама ВМ в терминологии libvirt называется доменом (domain). Это тоже объект внутри libvirt, который описывается отдельным XML-файлом.

При первоначальной установке и запуске libvirt по умолчанию создает мост (bridge) virbr0 и его минимальную конфигурацию. Этот мост не будет подключен ни к одному физическому интерфейсу, однако, может быть использован для связи ВМ внутри одного гипервизора.

Примечание. Компоненты libvirt будут установлены в систему, если при установке дистрибутива выбрать профиль «Базовая виртуализация».

Примечание. На этапе «Подготовка диска» рекомендуется выбрать «Generic Server KVM/Docker/LXD/Podman/CRI-O/PVE (large /var)».

Если же развертывание libvirt происходит в уже установленной системе на базе Десятой платформы, достаточно любым штатным способом (apt-get, aptitude, synaptic) установить пакет libvirt-kvm:

```
# apt-get install libvirt-kvm
```

Добавить службу в автозапуск и запустить ее:

```
# systemctl enable --now libvirtd
```

Для непривилегированного доступа (не root) к управлению libvirt, нужно добавить пользователя в группу vmusers:

```
# gpasswd -a user vmusers
```

Сервер виртуализации использует следующие каталоги хостовой файловой системы:

- `/etc/libvirt/` – каталог с файлами конфигурации libvirt;
- `/var/lib/libvirt/` – рабочий каталог сервера виртуализации libvirt;
- `/var/log/libvirt` – файлы журналов libvirt.

5.2 Утилиты управления

Основные утилиты командной строки для управления ВМ:

- qemu-img – управление образами дисков ВМ. Позволяет выполнять операции по созданию образов различных форматов, конвертировать файлы-образы между этими форматами, получать информацию об образах и объединять снимки ВМ для тех форматов, которые это поддерживают;
- virsh – консольный интерфейс управления ВМ, виртуальными дисками и виртуальными сетями;
- virt-clone –клонирование ВМ;
- virt-convert –конвертирования ВМ между различными форматами и программно-аппаратными платформами;
- virt-image – создание ВМ по их XML описанию;
- virt-install – создание ВМ с помощью опций командной строки;
- virt-xml – редактирование XML-файлов описаний ВМ.

5.2.1 Утилита Virsh

virsh – утилита для командной строки, предназначенная для управления ВМ и гипервизорами KVM.

virsh использует libvirt API и служит альтернативой графическому менеджеру виртуальных машин (virt-manager).

С помощью virsh можно сохранять состояние ВМ, переносить ВМ между гипервизорами и управлять виртуальными сетями.

В табл. 32 и табл. 33 приведены основные параметры утилиты командной строки virsh. Получить список доступных команд или параметров, можно используя команду: $ `virsh help`.

Т а б л и ц а 32 – Команды управления виртуальными машинами

Команда	Описание
`help`	Краткая справка
`list`	Просмотр всех ВМ
`dumpxml`	Вывести файл конфигурации XML для заданной ВМ
`create`	Создать ВМ из файла конфигурации XML и ее запуск
`start`	Запустить неактивную ВМ
`destroy`	Принудительно остановить работу ВМ
`define`	Определяет файл конфигурации XML для заданной ВМ
`domid`	Просмотр идентификатора ВМ
`domuuid`	Просмотр UUID ВМ
`dominfo`	Просмотр сведений о ВМ

Команда	Описание
domname	Просмотр имени ВМ
domstate	Просмотр состояния ВМ
quit	Закрыть интерактивный терминал
reboot	Перезагрузить ВМ
restore	Восстановить сохраненную в файле ВМ
resume	Возобновить работу приостановленной ВМ
save	Сохранить состояние ВМ в файл
shutdown	Корректно завершить работу ВМ
suspend	Приостановить работу ВМ
undefine	Удалить все файлы ВМ
migrate	Перенести ВМ на другой узел

Т а б л и ц а 33 – Параметры управления ресурсами ВМ и гипервизора

Команда	Описание
setmem	Определяет размер выделенной ВМ памяти
setmaxmem	Ограничивает максимально доступный гипервизору объем памяти
setvcpus	Изменяет число предоставленных ВМ виртуальных процессоров
vcpuinfo	Просмотр информации о виртуальных процессорах
vcpupin	Настройка соответствий виртуальных процессоров
domblkstat	Просмотр статистики блочных устройств для работающей ВМ
domifstat	Просмотр статистики сетевых интерфейсов для работающей ВМ
attach-device	Подключить определенное в XML-файле устройство к ВМ
attach-disk	Подключить новое дисковое устройство к ВМ
attach-interface	Подключить новый сетевой интерфейс к ВМ
detach-device	Отключить устройство от ВМ (принимает те же определения XML, что и attach-device)
detach-disk	Отключить дисковое устройство от ВМ
detach-interface	Отключить сетевой интерфейс от ВМ

5.2.2 Утилита virt-install

virt-install – это инструмент для создания ВМ, основанный на командной строке.

Должен быть установлен пакет virt-install (из репозитория p10):

```
# apt-get install virt-install
```

Далее подробно рассматриваются возможности создания ВМ при помощи утилиты командной строки virt-install. В табл. 34 приведено описание только наиболее часто используемых опций команды virt-install. Описание всех доступных опций можно получить, выполнив команду:

```
$ man virt-install
```

Утилита virt-install поддерживает как графическую установку операционных систем при помощи VNC и Spice, так и текстовую установку через последовательный порт. Гостевая система может быть настроена на использование нескольких дисков, сетевых интерфейсов, аудиоустройств и физических USB и PCI-устройств.

Установочный носитель может располагаться как локально, так и удаленно, например, на NFS, HTTP или FTP-серверах. В последнем случае virt-install получает минимальный набор файлов для запуска установки и позволяет установщику получить отдельные файлы. Поддерживается загрузка по сети (PXE) и создание виртуальной машины/контейнера без установки операционной системы.

Утилита virt-install поддерживает большое число опции, позволяющих создать полностью независимую ВМ, готовую к работе, что хорошо подходит для автоматизации установки ВМ.

Т а б л и ц а 34 – Параметры команды virt-install

Команда	Описание
`-n NAME, --name=NAME`	Имя новой ВМ. Это имя должно быть уникально внутри одного гипервизора

`--memory MEMORY`	Определяет размер выделенной ВМ памяти, например:
	`--memory 1024 (в МБ)`
	`--memory memory=1024,currentMemory=512`

`--vcpus VCPUS`	Определяет количество виртуальных ЦПУ, например:
	`--vcpus 5`
	`--vcpus 5,maxvcpus=10,cpuset=1-4,6,8`
	`--vcpus sockets=2,cores=4,threads=2`

`--cpu CPU`	Модель ЦП и его характеристики, например:
	`--cpu coreduo,+x2apic`
	`--cpu host-passthrough`
	`--cpu host`

`--metadata METADATA`	Метаданные ВМ
Метод установки	
`--cdrom CDROM`	Установочный CD-ROM. Может указывать на файл ISO-образа или на устройство чтения CD/DVD-дисков
`-l LOCATION, --location LOCATION`	Источник установки, например, https://host/path
`--pxe`	Выполнить загрузку из сети, используя протокол PXE
`--import`	Пропустить установку ОС, и создать ВМ на основе существую-

Команда	Описание
	щего образа диска

`--boot BOOT`	Параметры загрузки ВМ. Например:
	`--boot hd,cdrom,menu=on`
	`--boot init=/sbin/init (для контейнеров)`

`--os-variant=DISTRO_VARIANT`	ОС, которая устанавливается в гостевой системе. Используется для выбора оптимальных значений по умолчанию, в том числе VirtIO. Примеры значений: alt.p10, alt10.1, win10

`--disk DISK`	Настройка пространства хранения данных, например:
	`--disk size=10 (новый образ на 10 ГБ в выбранном по умолчанию месте)`
	`--disk /my/existing/disk,cache=none`
	`--disk device=cdrom,bus=scsi`
	`--disk=?`

`-w NETWORK, --network NETWORK`	Конфигурация сетевого интерфейса ВМ, например:
	`--network bridge=mybr0`
	`--network network=my_libvirt_virtual_net`
	`--network network=mynet,model=virtio,mac=00:11...`
	`--network none`

`--graphics GRAPHICS`	Настройки параметров экрана ВМ, например:
	`--graphics spice`
	`--graphics vnc,port=5901,listen=0.0.0.0`
	`--graphics none`

`--input INPUT`	Конфигурация устройства ввода, например:
	`--input tablet`
	`--input keyboard,bus=usb`

`--hostdev HOSTDEV`	Конфигурация физических USB/PCI и других устройств хоста для совместного использования ВМ
`-filesystem FILESYSTEM`	Передача каталога хоста гостевой системе, например: `--filesystem /my/source/dir,/dir/in/guest`
Параметры платформы виртуализации	
`-v, --hvm`	Эта ВМ должна быть полностью виртуализированной
`-p, --paravirt`	Эта ВМ должна быть паравиртуализированной
`--container`	Тип ВМ – контейнер
`--virt-type VIRT_TYPE`	Тип гипервизора (kvm, qemu и т.п.)
`--arch ARCH`	Имитируемая архитектура процессора
`--machine MACHINE`	Имитируемый тип компьютера
Прочие параметры	
`--autostart`	Запускать домен автоматически при запуске хоста

Команда	Описание
`--transient`	Создать временный домен
`--noautoconsole`	Не подключаться к гостевой консоли автоматически
`-q, --quiet`	Подавлять вывод (за исключением ошибок)
`-d, --debug`	Вывести отладочные данные

5.2.3 Утилита qemu-img

qemu-img – инструмент для манипулирования с образами дисков машин QEMU.

Использование:

```
qemu-img command [command options]
```

Для манипуляции с образами используются следующие команды:

- create – создание нового образа диска;
- check – проверка образа диска на ошибки;
- convert – конвертация существующего образа диска в другой формат;
- info – получение информации о существующем образе диска;
- snapshot – управляет снимками состояний (snapshot) существующих образов дисков;
- commit – записывает произведенные изменения на существующий образ диска;
- rebase – создает новый базовый образ на основании существующего.

qemu-img работает со следующими форматами:

- raw – простой формат для дисковых образов, обладающий отличной переносимостью на большинство технологий виртуализации и эмуляции. Только непосредственно записанные секторы будут занимать место на диске. Действительный объем пространства, занимаемый образом, можно определить с помощью команд qemu-img info или ls -ls;
- qcow2 – формат QEMU. Этот формат рекомендуется использовать для небольших образов (в частности, если файловая система не поддерживает фрагментацию), дополнительного шифрования AES, сжатия zlib и поддержки множества снимков ВМ;
- qcow – старый формат QEMU. Используется только в целях обеспечения совместимости со старыми версиями;
- cow – формат COW (Copy On Write). Используется только в целях обеспечения совместимости со старыми версиями;
- vmdk – формат образов, совместимый с VMware 3 и 4;
- cloop – формат CLOOP (Compressed Loop). Его единственное применение состоит в обеспечении повторного использования сжатых напрямую образов CD-ROM, например, Knoppix CD-ROM.

Команда получения сведений о дисковом образе:

```
# qemu-img info /var/lib/libvirt/images/alt-server.qcow2
image: /var/lib/libvirt/images/alt-server.qcow2
file format: qcow2
virtual size: 20 GiB (21474836480 bytes)
disk size: 3.32 MiB
cluster_size: 65536
Format specific information:
    compat: 1.1
    compression type: zlib
    lazy refcounts: true
    refcount bits: 16
    corrupt: false
    extended l2: false
```

В результате будут показаны сведения о запрошенном образе, в том числе зарезервированный объем на диске, а также информация о снимках ВМ.

Команда создания образа для жесткого диска (динамически расширяемый):

```
# qemu-img create -f qcow2 /var/lib/libvirt/images/hdd.qcow2 20G
```

Команда конвертирования образа диска из формата raw в qcow2:

```
# qemu-img convert -f raw -O qcow2 disk_hd.img disk_hd.qcow2
```

5.2.4 Менеджер виртуальных машин virt-manager

Менеджер виртуальных машин virt-manager предоставляет графический интерфейс для доступа к гипервизорам и ВМ в локальной и удаленных системах. С помощью virt-manager можно создавать ВМ. Кроме того, virt-manager выполняет управляющие функции:

- выделение памяти;
- выделение виртуальных процессоров;
- мониторинг производительности;
- сохранение и восстановление, приостановка и возобновление работы, запуск и завершение работы виртуальных машин;
- доступ к текстовой и графической консоли;
- автономная и живая миграция.

Для запуска менеджера виртуальных машин, в меню приложений необходимо выбрать «Система»→ «Менеджер виртуальных машин» («Manage virtual machines»).

Примечание. На управляющей машине должен быть установлен пакет virt-manager.

В главном окне менеджера (Рис. 337), при наличии подключения к гипервизору, будут показаны все запущенные ВМ. Двойной щелчок на имени ВМ открывает ее консоль.

Главное окно менеджера виртуальных машин

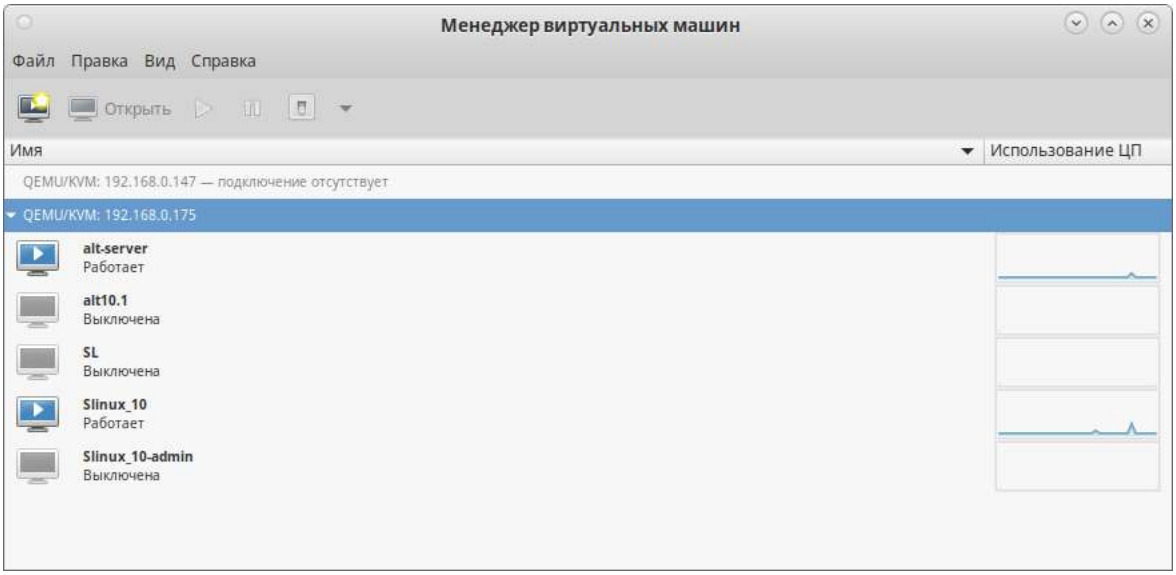


Рис. 337

5.3 Подключение к гипервизору

5.3.1 Управление доступом к libvirt через SSH

В дополнение к аутентификации SSH также необходимо определить управление доступом для службы Libvirt в хост-системе (Рис. 338).

Доступ к libvirt с удаленного узла

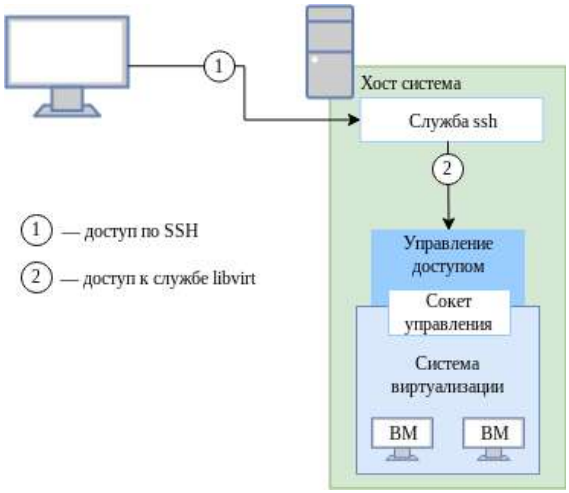


Рис. 338

Для настройки подключения к удаленному серверу виртуализации на узле, с которого будет производиться подключение, необходимо сгенерировать SSH-ключ и скопировать его публичную часть на сервер. Для этого с правами пользователя, от имени которого будет создаваться подключение, требуется выполнить в консоли следующие команды:

```
$ ssh-keygen -t ed25519
$ ssh-copy-id user@192.168.0.175
```

где 192.168.0.175 – IP-адрес сервера с libvirt.

В результате получаем возможность работы с домашними каталогами пользователя user на сервере с libvirt.

Для доступа к libvirt достаточно добавить пользователя user в группу vmusers на сервере, либо скопировать публичный ключ пользователю root и подключаться к серверу по ssh от имени root – root@server

5.3.2 Подключение к сессии гипервизора с помощью virsh

Команда подключения к гипервизору:

```
virsh -c URI
```

Если параметр URI не задан, то libvirt попытается определить наиболее подходящий гипервизор.

Параметр URI может принимать следующие значения:

- qemu:///system – подключиться к службе, которая управляет KVM/QEMU-доменами и запущена под root. Этот вариант используется по умолчанию для пользователей virt-manager;
- qemu:///session – подключиться к службе, которая управляет KVM/QEMU-доменами и запущена от имени непривилегированного пользователя;
- lxc:/// – подключиться к гипервизору для создания LXC контейнеров (должен быть установлен пакет libvirt-lxc).

Чтобы установить соединение только для чтения, к приведенной выше команде следует добавить опцию `--readonly`.

Пример создания локального подключения:

```
$ virsh -c qemu:///system list --all
ID   Имя          Состояние
------------------------------
 -    alt-server   выключен
```

Примечание. Чтобы постоянно не вводить `-c qemu:///system` можно добавить:

```
export LIBVIRT_DEFAULT_URI=qemu:///system
```

Подключение к удаленному гипервизору QEMU через протокол SSH:

```
$ virsh -c qemu+ssh://user@192.168.0.175/system
Добро пожаловать в virsh — интерактивный терминал виртуализации.

Введите «help» для получения справки по командам
         «quit», чтобы завершить работу и выйти.

virsh #
```

где user – имя пользователя на удаленном хосте, который входит в группу vmusers. 192.168.0.175 – IP-адрес или имя хоста виртуальных машин.

5.3.3 Настройка соединения с удаленным гипервизором в virt-manager

На управляющей системе можно запустить virt-manager, выполнив следующую команду:

```
virt-manager -c qemu+ssh://user@192.168.0.175/system
```

где user – имя пользователя на удаленном хосте, который входит в группу vmusers. 192.168.0.175 – IP-адрес или имя хоста виртуальных машин.

virt-manager позволяет управлять несколькими удаленными хостами ВМ.

Подключение virt-manager к удаленным хостам также можно настроить и в графическом интерфейсе менеджера виртуальных машин. Для создания нового подключения необходимо в меню менеджера выбрать «Файл» → «Добавить соединение…».

В открывшемся окне необходимо выбрать сессию гипервизора, отметить пункт «Подключиться к удаленному хосту с помощью SSH», ввести имя пользователя и адрес сервера и нажать кнопку «Подключиться» (Рис. 339).

Окно соединений менеджера виртуальных машин

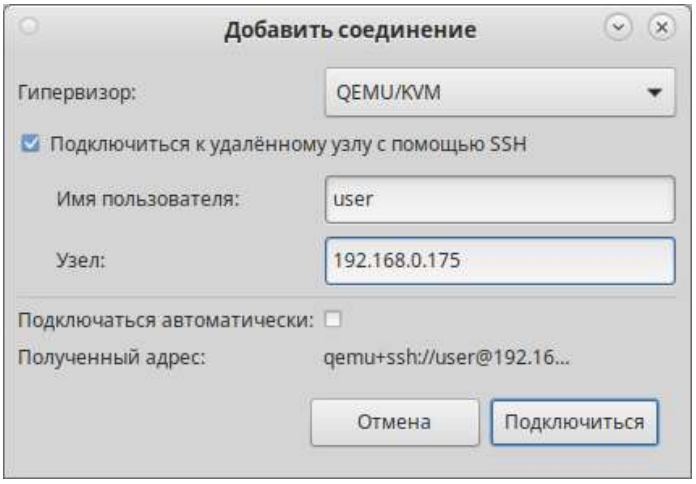


Рис. 339

5.4 Создание виртуальных машин

Наиболее важным этапом в процессе использования виртуализации является создание ВМ. Именно при создании ВМ задается используемый тип виртуализации, способы доступа к ВМ, подключение к локальной сети и другие характеристики виртуального оборудования.

Установка ВМ может быть запущена из командной строки с помощью программ virsh и virt-install или из пользовательского интерфейса программы virt-manager.

5.4.1 Создание виртуальной машины на основе файла конфигурации (утилита virsh)

ВМ могут быть созданы из файлов конфигурации. Для этого конфигурация ВМ должна быть описана в XML формате.

Команда создания ВМ из XML файла:

```
$ virsh -c qemu:///system create guest.xml
Domain 'altK' created from guest.xml
```

Для получения файла конфигурации можно сделать копию существующего XML-файла ранее созданной ВМ, или использовать опцию `dumpxml`:

```
virsh dumpxml <domain>
```

Эта команда выводит XML-файл конфигурации ВМ в стандартный вывод (stdout). Можно сохранить эти данные, отправив вывод в файл.

Пример передачи вывода в файл `guest.xml`:

```
$ virsh -c qemu:///system dumpxml alt-server > guest.xml
```

Можно отредактировать этот файл конфигурации, чтобы настроить дополнительные устройства или развернуть дополнительные ВМ.

5.4.2 Создание ВМ с помощью virt-install

Минимальные требуемые опции для создания ВМ: `--name`, `--memory`, хранилище (`--disk`, `--filesystem` или `--nodisks`) и опции установки.

Чтобы использовать команду virt-install, необходимо сначала загрузить ISO-образ той ОС, которая будет устанавливаться.

Команда создания ВМ:

```
$ virt-install --connect qemu:///system \
--name alt-server \
--os-variant=alt10.1 \
--cdrom /var/lib/libvirt/images/alt-server-10.2-x86_64.iso \
--graphics spice,listen=0.0.0.0 \
--video qxl \
--disk pool=default,size=20,bus=virtio,format=qcow2 \
--memory 2048 \
--vcpus=2 \
--network network=default \
--hvm \
--virt-type=kvm
```

где:

- --name alt-server – название ВМ;
- --os-variant=alt10.1 – версия ОС;
- --cdrom /var/lib/libvirt/images/alt-server-10.2-x86_64.iso – путь к ISO-образу установочного диска ОС;
- --graphics spice – графическая консоль;
- --disk pool=default,size=20,bus=virtio,format=qcow2 – ВМ будет создана в пространстве хранения объемом 20 ГБ, которое автоматически выделяется из пула хранилищ default. Образ диска для этой виртуальной машины будет создан в формате qcow2;
- --memory 2048 – объем оперативной памяти;

- --vcpus=2 – количество процессоров;
- --network network=default – виртуальная сеть default;
- --hvm – полностью виртуализированная система;
- --virt-type=kvm – использовать модуль ядра KVM, который задействует аппаратные возможности виртуализации процессора.

Последние две опции команды virt-install оптимизируют ВМ для использования в качестве полностью виртуализированной системы (--hvm) и указывают, что KVM является базовым гипервизором (--virt-type) для поддержки новой ВМ. Обе этих опции обеспечивают определенную оптимизацию в процессе создания и установки операционной системы; если эти опции не заданы в явном виде, то вышеуказанные значения применяются по умолчанию.

Для создания виртуальной машины с UEFI, нужно указать параметры, которые включают UEFI в качестве загрузчика, например:

```
# virt-install --connect qemu:///system \
--name alt-server-test \
--os-variant=alt10.0 \
--cdrom /var/lib/libvirt/images/alt-server-10.2-x86_64.iso \
--graphics spice,listen=0.0.0.0 \
--video qxl \
--disk pool=default,size=20,bus=virtio,format=qcow2 \
--memory 2048 \
--vcpus=2 \
--network network=default \
--hvm \
--virt-type=kvm \
--boot loader=/usr/share/OVMF/OVMF_CODE.fd
```

где /usr/share/OVMF/OVMF_CODE.fd – путь к UEFI загрузчику.

Список доступных вариантов ОС можно получить, выполнив команду:

```
$ virt-install --osinfo list
```

Запуск Live CD в ВМ без дисков:

```
# virt-install \
 --hvm \
 --name demo \
 --memory 500 \
 --nodisks \
 --livecd \
 --graphics vnc \
 --cdrom /var/lib/libvirt/images/altlive.iso
```

Запуск /bin/bash в контейнере (LXC), с ограничением памяти в 512 МБ и одним ядром хост-системы:

```
# virt-install \
 --connect lxc:/// \
 --name bash_guest \
 --memory 512 \
 --vcpus 1 \
 --init /bin/bash
```

Создать ВМ, используя существующий том хранилища:

```
# virt-install \
 --name demo \
 --memory 512 \
 --disk /home/user/VMs/mydisk.img \
 --import
```

5.4.3 Создание виртуальных машин с помощью virt-manager

Новую ВМ можно создать, нажав кнопку «Создать виртуальную машину» в главном окне virt-manager, либо выбрав в меню «Файл»→ «Создать виртуальную машину».

На первом шаге создания ВМ необходимо выбрать метод установки ОС (Рис. 340) и нажать кнопку «Вперед».

Создание ВМ. Выбор метода установки

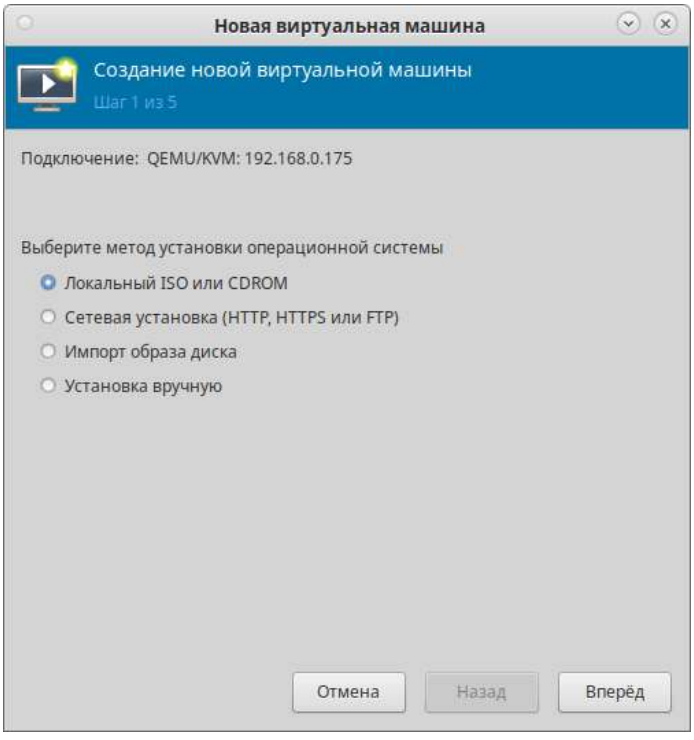


Рис. 340

В следующем окне для установки гостевой ОС требуется указать ISO-образ установочного диска ОС или CD/DVD-диск с дистрибутивом (Рис. 341). Данное окно будет выглядеть по-разному, в зависимости от выбора, сделанного на предыдущем этапе. Здесь также можно указать версию устанавливаемой ОС.

Создание ВМ. Выбор ISO-образа

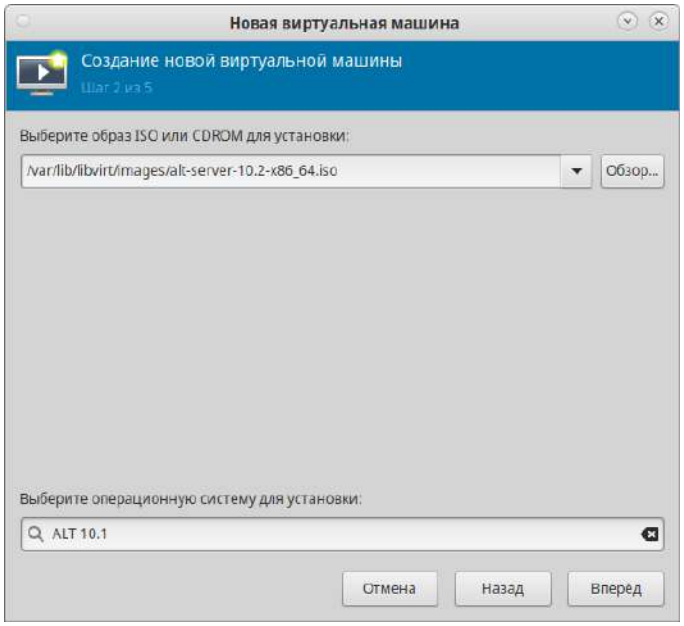


Рис. 341

На третьем шаге необходимо указать размер памяти и количество процессоров для ВМ (Рис. 342). Эти значения влияют на производительность хоста и ВМ.

Создание ВМ. Настройка ОЗУ и ЦПУ для ВМ

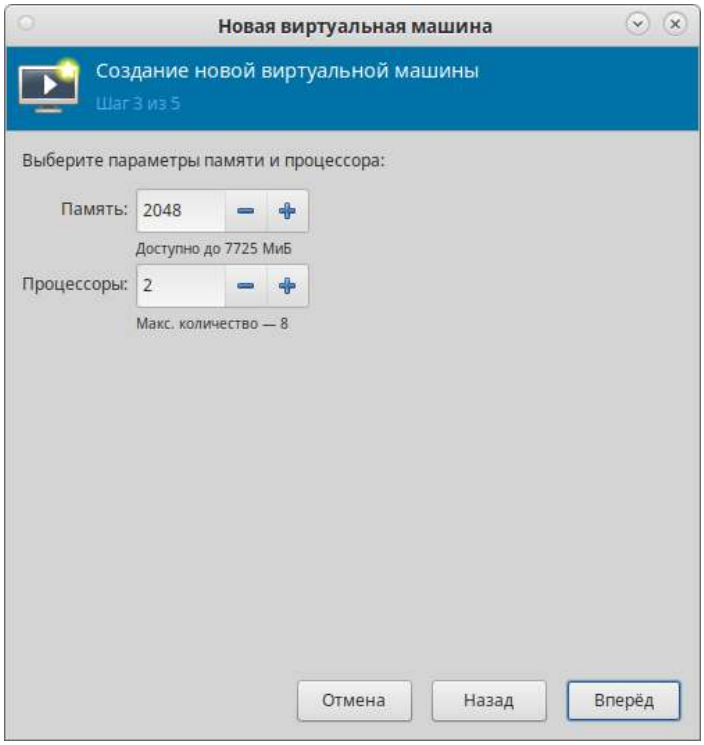


Рис. 342

На следующем этапе настраивается пространство хранения данных (Рис. 343).

Создание ВМ. Настройка пространства хранения данных

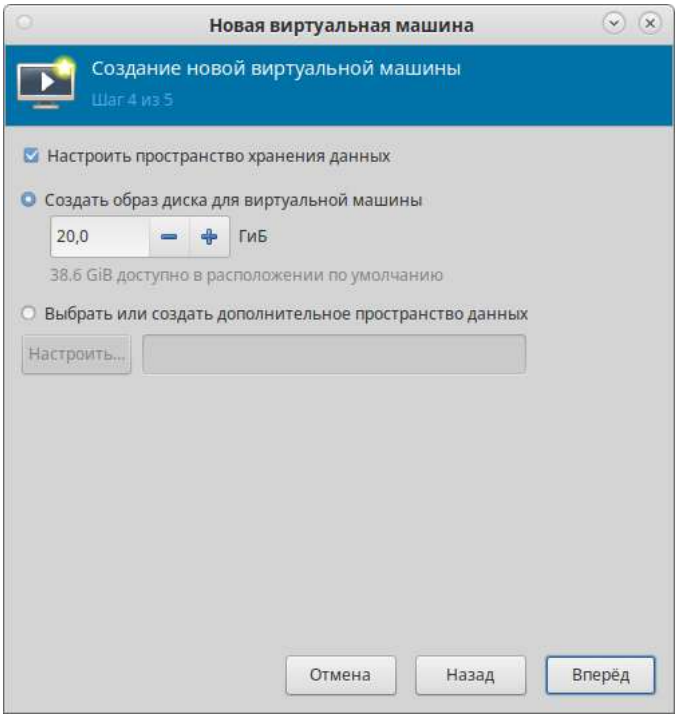


Рис. 343

На последнем этапе (Рис. 344) можно задать название ВМ, выбрать сеть и нажать кнопку «Готово».

Создание ВМ. Выбор сети

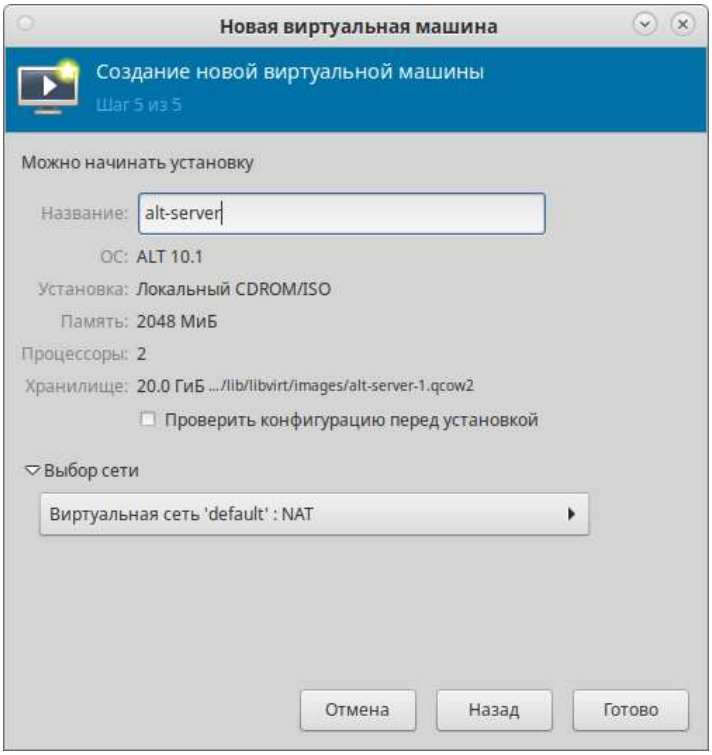


Рис. 344

В результате созданная ВМ будет запущена и после завершения исходной загрузки начнется стандартный процесс установки ОС (Рис. 345).

Установка ОС

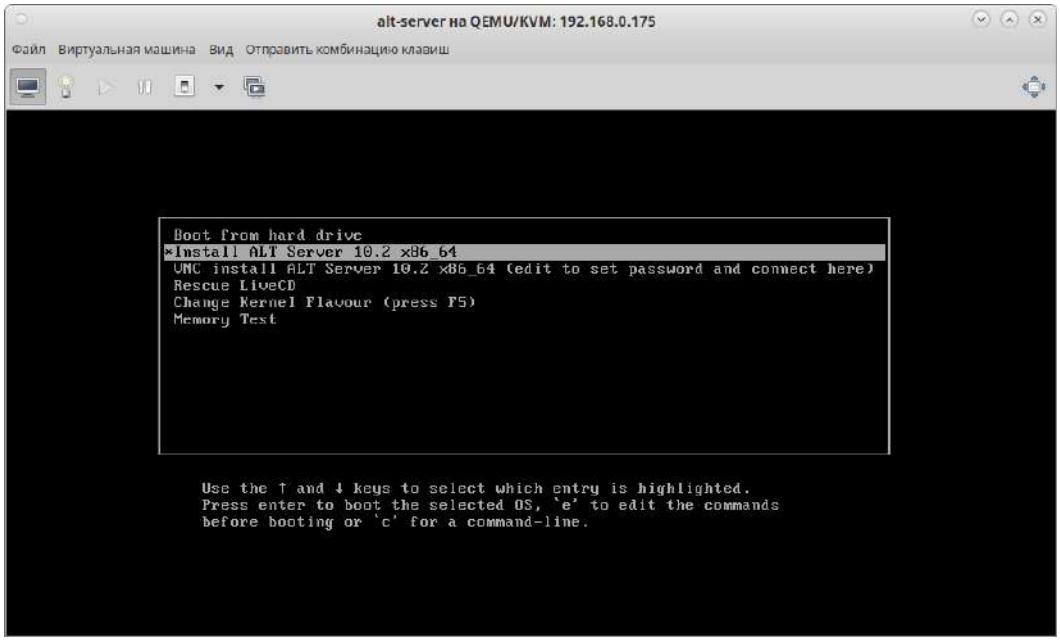


Рис. 345

Окружение локального рабочего стола способно перехватывать комбинации клавиш (например, <Ctrl>+<Alt>+<F11>) для предотвращения их отправки гостевой машине. Чтобы отправить такие последовательности, используется свойство «западания» клавиш virt-manager. Для перевода клавиши в нажатое состояние необходимо нажать клавишу модификатора (<Ctrl> или <Alt>) 3 раза. Клавиша будет считаться нажатой до тех пор, пока не будет нажата любая клавиша, отличная от модификатора. Таким образом, чтобы передать гостевой системе комбинацию <Ctrl>+<Alt>+<F11>, необходимо последовательно нажать <Ctrl>+<Ctrl>+<Ctrl>+<Alt>+<F11> или воспользоваться меню «Отправить комбинацию клавиш».

Для создания виртуальной машины с UEFI необходимо:

- на последнем этапе создания ВМ, до нажатия кнопки «Готово», установить отметку в поле «Проверить конфигурацию перед установкой» (Рис. 346);
- в открывшемся окне в разделе «Обзор» в раскрывающемся списке «Микропрограмма» выбрать опцию «UEFI» (Рис. 347);
- нажать кнопку «Применить» для сохранения изменений;
- нажать кнопку «Начать установку».

Проверить конфигурацию перед установкой

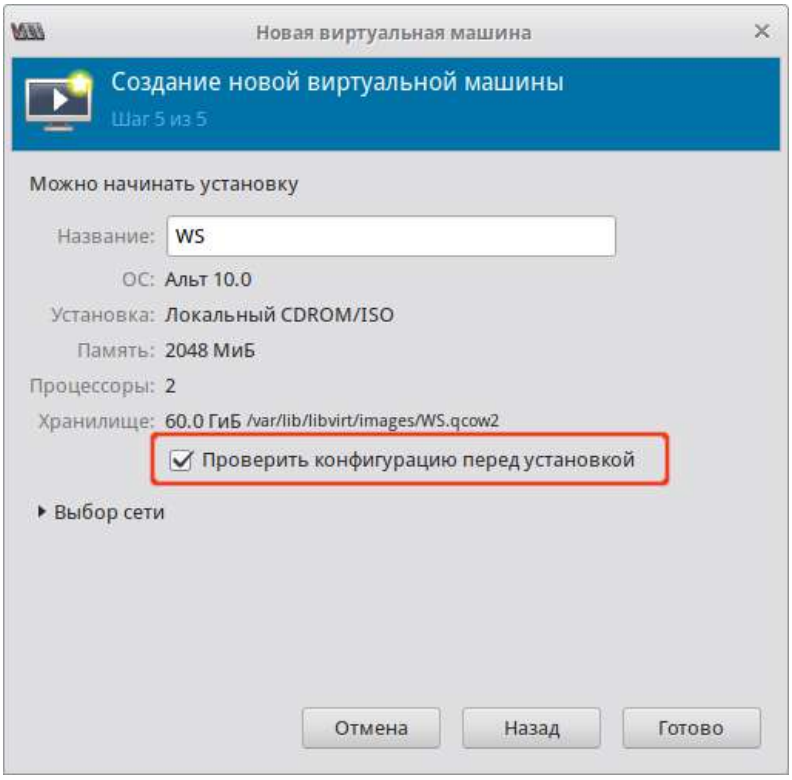


Рис. 346

Выбор типа загрузки UEFI

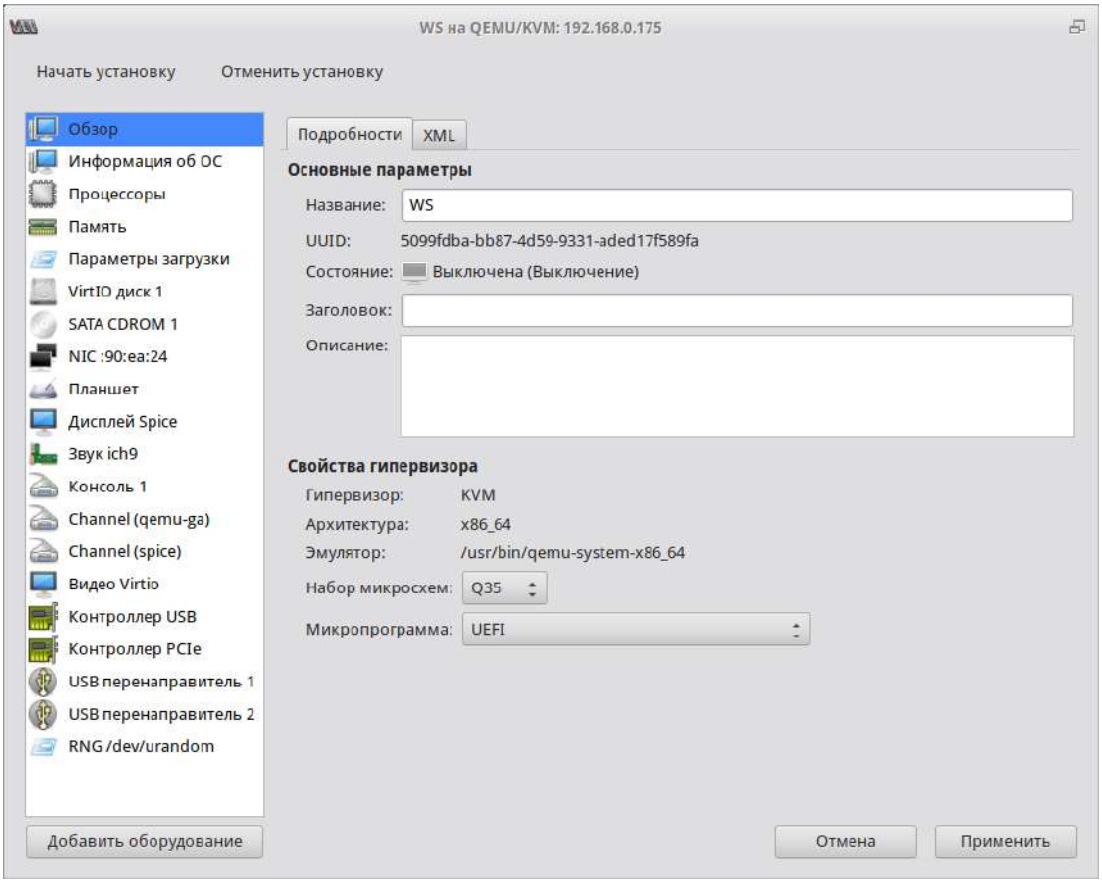


Рис. 347

5.5 Запуск и управление функционированием ВМ

5.5.1 Управление состоянием ВМ в командной строке

Команды управления состоянием ВМ:

- start – запуск ВМ;
- shutdown – завершение работы. Поведение выключаемой ВМ можно контролировать с помощью параметра on_shutdown (в файле конфигурации);
- destroy – принудительная остановка. Использование `virsh destroy` может повредить гостевые файловые системы. Рекомендуется использовать опцию `shutdown`;
- reboot – перезагрузка ВМ. Поведение перезагружаемой ВМ можно контролировать с помощью параметра on_reboot (в файле конфигурации);
- suspend – приостановить ВМ. Когда ВМ находится в приостановленном состоянии, она потребляет системную оперативную память, но не ресурсы процессора;
- resume – возобновить работу приостановленной ВМ;
- save – сохранение текущего состояния ВМ. Эта команда останавливает ВМ, сохраняет данные в файл, что может занять некоторое время (зависит от объема ОЗУ ВМ);
- restore – восстановление ВМ, ранее сохраненной с помощью команды `virsh save`. Сохраненная машина будет восстановлена из файла и перезапущена (это может занять некоторое время). Имя и идентификатор UUID ВМ останутся неизменными, но будет предоставлен новый идентификатор домена;
- undefine – удалить ВМ (конфигурационный файл тоже удаляется);
- autostart – добавить ВМ в автозагрузку;
- autostart --disable – удалить из автозагрузки.

В результате выполнения следующих команд, ВМ alt-server будет остановлена и затем удалена:

```
# virsh destroy alt-server
# virsh undefine alt-server
```

5.5.2 Управление состоянием ВМ в менеджере виртуальных машин

Для запуска ВМ в менеджере виртуальных машин virt-manager, необходимо выбрать ВМ из списка и нажать на кнопку «Включить виртуальную машину» (Рис. 348).

Для управления запущенной ВМ используются соответствующие кнопки панели инструментов virt-manager (Рис. 349).

Управлять состоянием ВМ можно, выбрав соответствующий пункт в контекстном меню ВМ (Рис. 350).

Включение ВМ

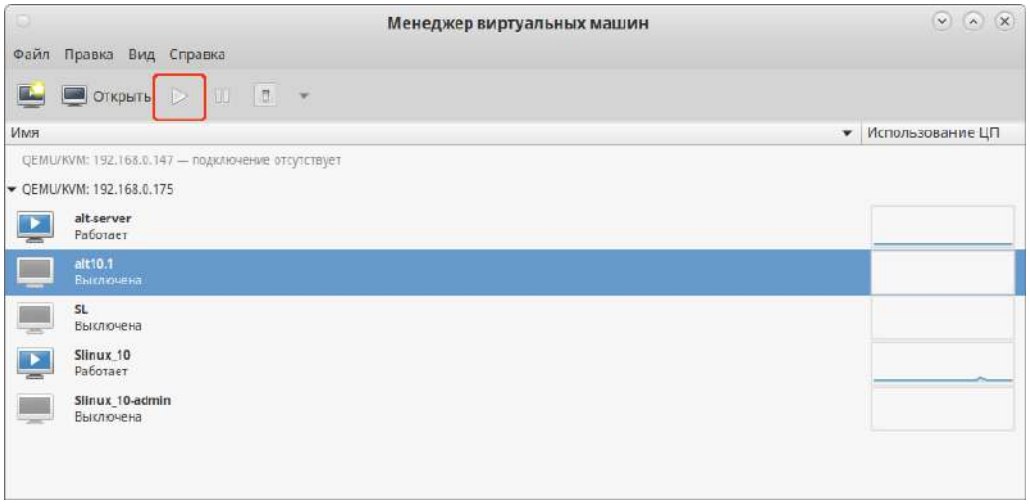


Рис. 348

Кнопки управления состоянием ВМ

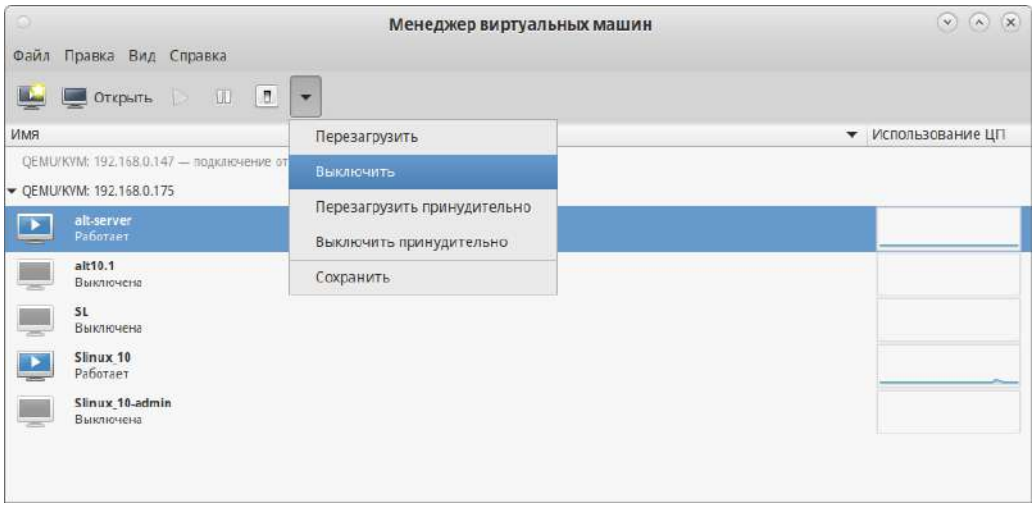


Рис. 349

Контекстное меню ВМ

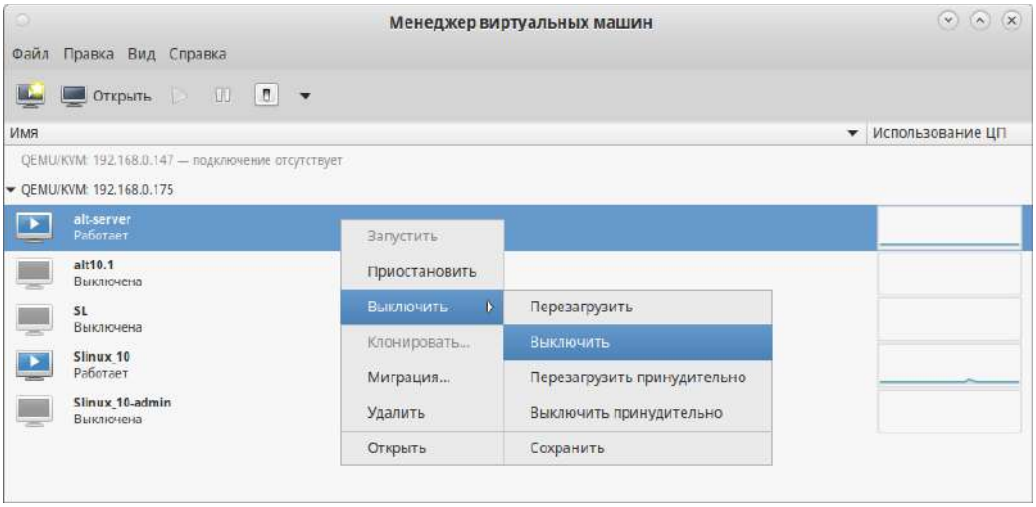


Рис. 350

5.6 Подключение к виртуальному монитору ВМ

Доступ к рабочему столу ВМ может быть организован по протоколам VNC и SPICE.

К каждой из ВМ можно подключиться, используя один IP-адрес и разные порты. Порт доступа к ВМ может быть назначен вручную или автоматически. Удаленный доступ к ВМ можно защитить паролем.

5.6.1 Использование протокола SPICE

Чтобы добавить поддержку SPICE в существующую ВМ, необходимо отредактировать её конфигурацию:

```
# virsh edit alt-server
```

Добавить графический элемент SPICE, например:

```
<graphics type='spice' port='5900' autoport='yes' listen='127.0.0.1'>
    <listen type='address' address='127.0.0.1'/>
</graphics>
```

Добавить видеоустройство QXL:

```
<video>
    <model type='qxl'/>
</video>
```

После остановки и перезапуска ВМ она должна быть доступна через SPICE.

Проверка параметров подключения к ВМ:

```
# virsh domdisplay alt-server
spice://127.0.0.1:5900
```

В данном примере доступ к ВМ будет возможен только с локального адреса (127.0.0.1). Для удаленного подключения к ВМ SPICE-сервер должен обслуживать запросы с общедоступных сетевых интерфейсов. Для возможности подключения с других машин в конфигурации ВМ необходимо указать адрес 0.0.0.0:

```
<graphics type='spice' port='5900' autoport='yes' listen='0.0.0.0' passwd='mypasswd'>
    <listen type='address' address='0.0.0.0'/>
</graphics>
```

Пример настроек доступа к рабочему столу по протоколу SPICE в менеджере ВМ показан на Рис. 351.

Для подключения к SPICE-серверу может использоваться встроенный в virt-manager просмотрщик или любой SPICE-клиент. Примеры подключений (на хосте, с которого происходит подключение, должен быть установлен пакет virt-viewer):

```
$ virt-viewer -c qemu+ssh://user@192.168.0.175/system -d alt-server

$ remote-viewer "spice://192.168.0.175:5900"
```

Менеджер ВМ. Вкладка «Дисплей Spice»

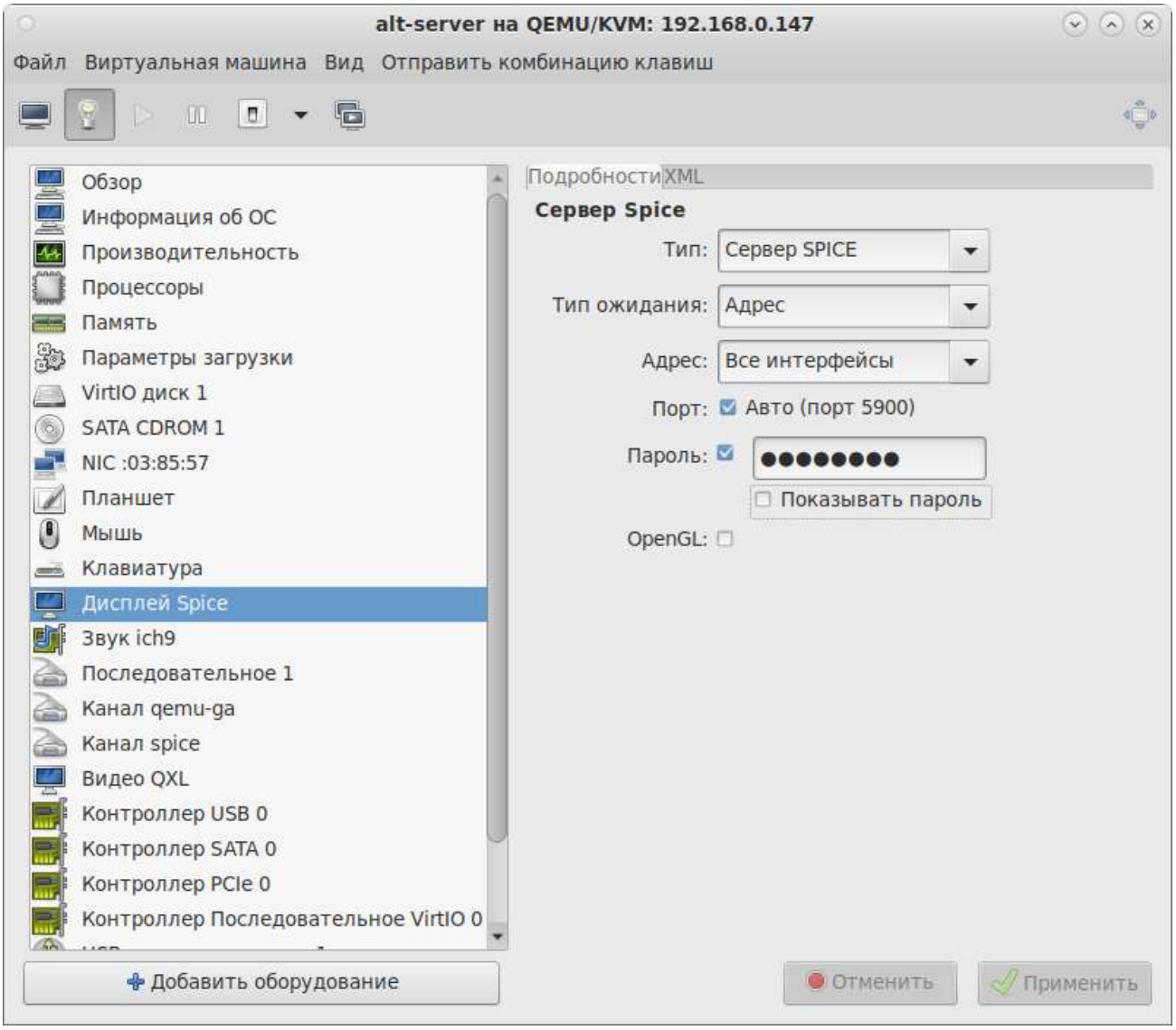


Рис. 351

Примечание. При использовании любого SPICE-клиента подключение происходит к порту и адресу хоста KVM, а не к фактическому имени/адресу ВМ.

5.6.2 Использование протокола VNC

Пример настройки доступа к рабочему столу ВМ по протоколу VNC, в файле конфигурации ВМ:

```
<graphics type='vnc' port='5900' autoport='no' listen='0.0.0.0' passwd='mypasswd'>
    <listen type='address' address='0.0.0.0'/>
</graphics>
```

Пример настроек доступа к рабочему столу по протоколу VNC в менеджере ВМ показан на Рис. 352.

Менеджер ВМ. Вкладка «Дисплей VNC»

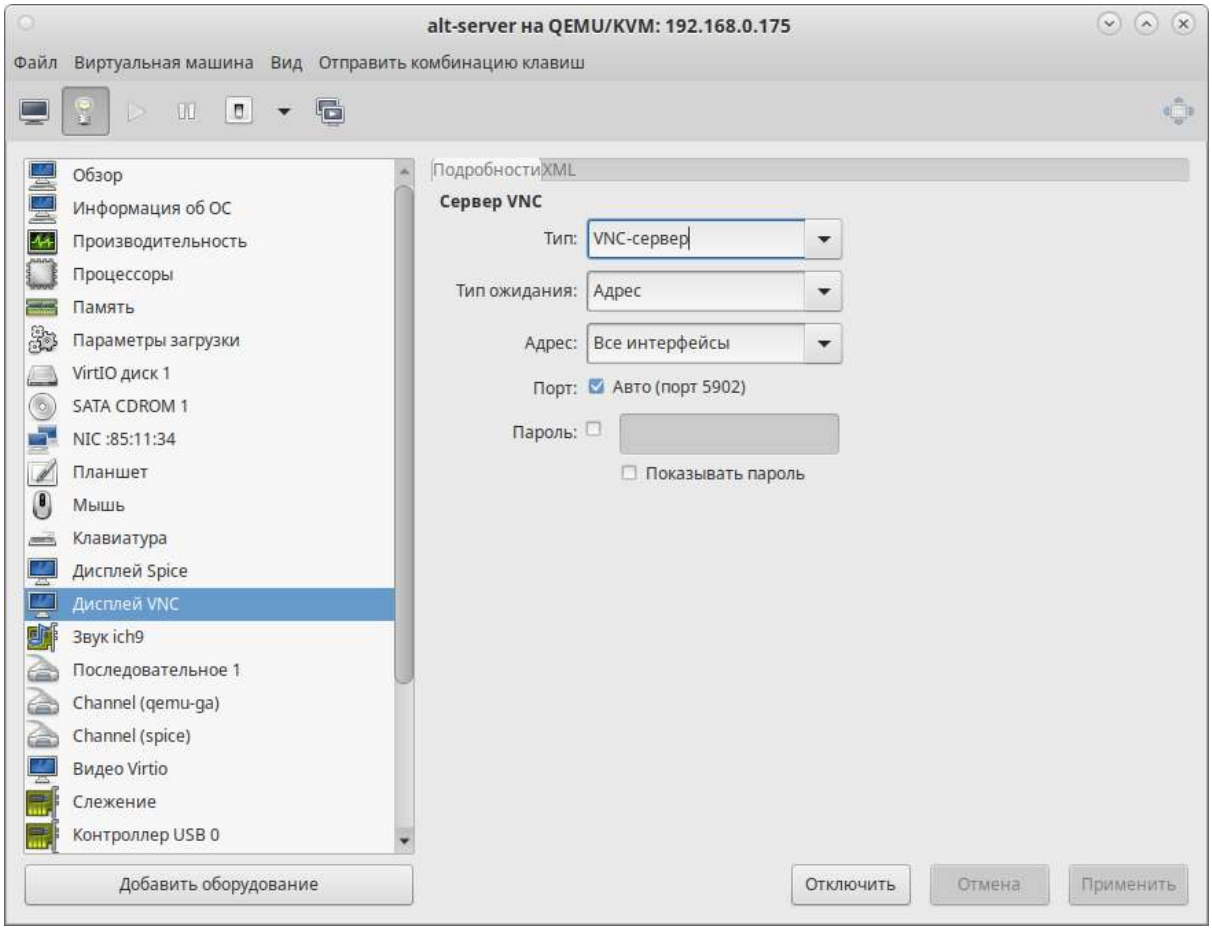


Рис. 352

Проверка параметров подключения к ВМ:

```
# virsh domdisplay alt-server
vnc://localhost:2
```

Для подключения к VNC-серверу может использоваться встроенный в virt-manager просмотрщик или любой VNC-клиент. Примеры подключений (на хосте, с которого происходит подключение, должны быть соответственно установлены пакеты virt-viewer или tigervnc):

```
$ virt-viewer -c qemu+ssh://user@192.168.0.175/system -d alt-server
$ vncviewer 192.168.0.175:5902
```

5.7 Управление ВМ

5.7.1.1 Редактирование файла конфигурации ВМ

ВМ могут редактироваться либо во время работы, либо в автономном режиме. Эту функциональность предоставляет команда `virsh edit`. Например, команда редактирования ВМ с именем alt-server:

```
# virsh edit alt-server
```

В результате выполнения этой команды откроется окно текстового редактора, заданного переменной оболочки $EDITOR.

5.7.1.2 Получение информации о ВМ

Команда для получения информации о ВМ:

```
virsh dominfo <domain>
```

где [--domain] <строка> – имя, ID или UUID домена

Пример вывода `virsh dominfo`:

```
$ virsh dominfo alt-server
ID:             3
Имя:         alt-server
UUID:           ccb6bf9e-1f8d-448e-b5f7-fa274703500b
Тип ОС:    hvm
Состояние: работает
CPU:            2
Время CPU: 90,9s
Макс.память: 2097152 KiB
Занято памяти: 2097152 KiB
Постоянство: yes
Автозапуск: выкл.
Управляемое сохранение: no
Модель безопасности: none
DOI безопасности: 0
```

Получение информации об узле:

```
$ virsh nodeinfo
Модель процессора: x86_64
CPU:                8
Частота процессора: 2827 MHz
Сокеты:        1
Ядер на сокет: 4
Потоков на ядро: 2
Ячейки NUMA:   1
Объём памяти: 8007952 KiB
```

Просмотр списка ВМ:

```
virsh list
```

Опции команды `virsh list`:

- `--inactive` – показать список неактивных доменов;
- `--all` – показать все ВМ независимо от их состояния.

Пример вывода `virsh list`:

```
$ virsh list --all
 ID   Имя          Состояние
--------------------------
 3    alt-server   работает
```

Столбец «Статус» может содержать следующие значения:

- работает (running) – работающие ВМ, то есть те машины, которые используют ресурсы процессора в момент выполнения команды;
- blocked – заблокированные, неработающие машины. Такой статус может быть вызван ожиданием ввода/вывода или пребыванием машины в спящем режиме;
- приостановлен (paused) – приостановленные домены. В это состояние они переходят, если администратор нажал кнопку паузы в окне менеджера ВМ или выполнил команду virsh suspend. В приостановленном состоянии ВМ продолжает потреблять ресурсы, но не может занимать больше процессорных ресурсов;
- выключен (shutdown) – ВМ, завершающие свою работу. При получении ВМ сигнала завершения работы, она начнет завершать все процессы (некоторые операционные системы не отвечают на такие сигналы);
- dying – сбойные домены и домены, которые не смогли корректно завершить свою работу;
- crashed – сбойные домены, работа которых была прервана. В этом состоянии домены находятся, если не была настроена их перезагрузка в случае сбоя.

Команда получения информации о виртуальных процессорах:

```
virsh vcpuinfo <domain>
```

Пример вывода:

```
# virsh vcpuinfo alt-server
Виртуальный процессор:: 0
CPU:            6
Состояние: работает
Время CPU: 67,6s
Соответствие ЦП: yyyyyyyy

Виртуальный процессор:: 1
CPU:            7
Состояние: работает
Время CPU: 7,1s
Соответствие ЦП: yyyyyyyy
```

Команда сопоставления виртуальных процессоров физическим:

```
virsh vcpupin <domain> [--vcpu <число>] [--cpulist <строка>] [--config] [--live] [--current]
```

Здесь:

[--domain] <строка> – имя, ID или UUID домена;

--vcpu <число> – номер виртуального процессора;

--cpulist <строка> – номера физических процессоров. Если номера не указаны, команда вернет текущий список процессоров;

--config – с сохранением после перезагрузки;

--live – применить к работающему домену;

--current – применить к текущему домену.

Пример вывода:

```
 # virsh vcpupin alt-server
 Виртуальный процессор:   Соответствие ЦП
------------------------------------------
 0                        0-7
 1                        0-7
```

Команда изменения числа процессоров для домена (заданное число не может превышать значение, определенное при создании ВМ):

```
virsh setvcpus <domain> <count> [--maximum] [--config] [--live] [--current] [--guest] [--hotpluggable]
```

где

[--domain] <строка> – имя, ID или UUID домена;

[--count] <число> – число виртуальных процессоров;

--maximum – установить максимальное ограничение на количество виртуальных процессоров, которые могут быть подключены после следующей перезагрузки домена;

--config – с сохранением после перезагрузки;

--live – применить к работающему домену;

--current – применить к текущему домену;

--guest – состояние процессоров ограничивается гостевым доменом.

Команда изменения выделенного ВМ объема памяти:

```
virsh setmem <domain> <size> [--config] [--live] [--current]
```

где

[--domain] <строка> – имя, ID или UUID домена;

[--size] <число> – целое значение нового размера памяти (по умолчанию в КБ);

--config – с сохранением после перезагрузки;

--live – применить к работающему домену;

--current – применить к текущему домену.

Объем памяти, определяемый заданным числом, должен быть указан в килобайтах. Объем не может превышать значение, определенное при создании ВМ, но в то же время не должен быть меньше 64 мегабайт. Изменение максимального объема памяти может оказать влияние на

функциональность ВМ только в том случае, если указанный размер меньше исходного. В таком случае использование памяти будет ограничено.

Команда изменения максимально допустимого размера выделяемой памяти:

```
virsh setmaxmem <domain> <size> [--config] [--live] [--current]
```

где

[--domain] <строка> – имя, ID или UUID домена;

[--size] <число> – целое значение максимально допустимого размера памяти (по умолчанию в КБ);

--config – с сохранением после перезагрузки;

--live – применить к работающему домену;

--current – применить к текущему домену.

Примеры изменения размера оперативной памяти и количества виртуальных процессоров соответственно:

```
# virsh setmaxmem --size 624000 alt-server
# virsh setmem --size 52240 alt-server
# virsh setvcpus --config alt-server 3 --maximum
```

Команда для получения информации о блочных устройствах работающей ВМ:

```
virsh domblkstat <domain> [--device <строка>] [--human]
```

где:

[--domain] <строка> – имя, ID или UUID домена;

--device <строка> – блочное устройство;

--human – форматировать вывод.

Команда для получения информации о сетевых интерфейсах работающей ВМ:

```
virsh domifstat <domain> <interface>
```

где:

[--domain] <строка> – имя, ID или UUID домена;

[--interface] <строка> – устройство интерфейса, указанное по имени или MAC-адресу.

5.7.1.3 Конфигурирование ВМ в менеджере виртуальных машин

С помощью менеджера виртуальных машин можно получить доступ к подробной информации о всех ВМ, для этого следует:

1) в главном окне менеджера выбрать ВМ;
2) нажать кнопку «Открыть» (Рис. 353);
3) в открывшемся окне нажать кнопку «Показать виртуальное оборудование» (Рис. 354);
4) появится окно просмотра сведений ВМ.

Окно менеджера виртуальных машин

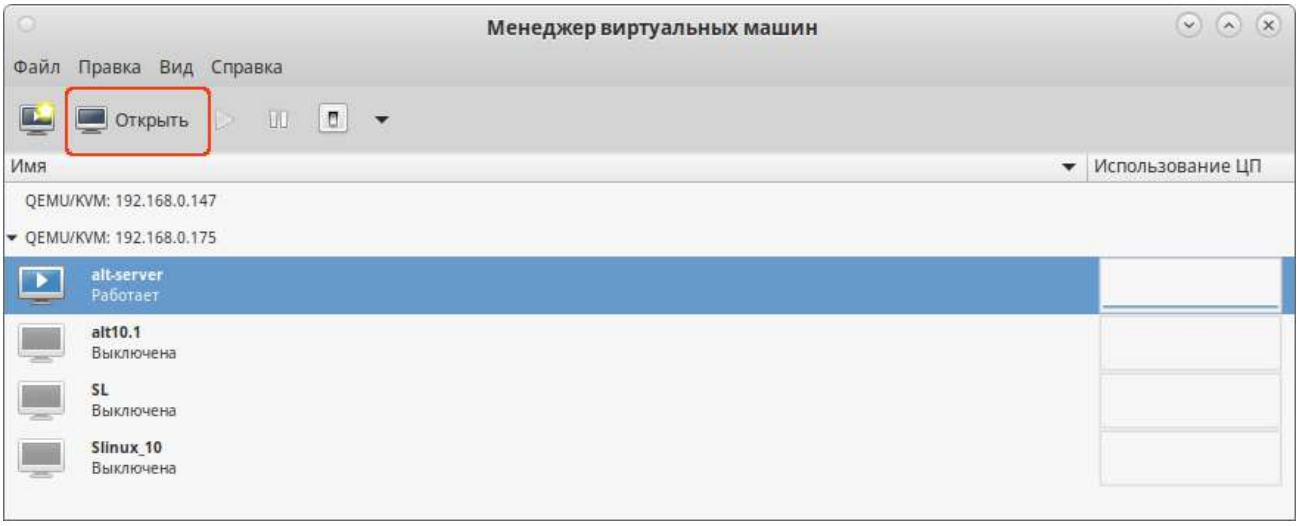


Рис. 353

Окно параметров ВМ

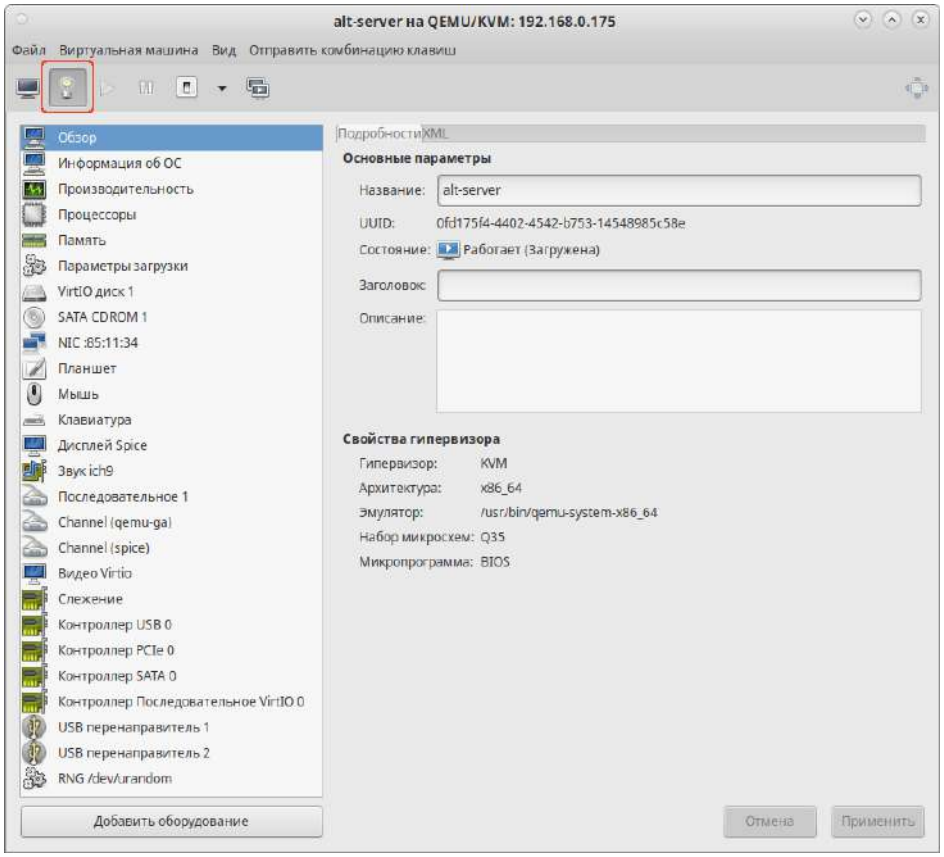


Рис. 354

Для изменения требуемого параметра необходимо перейти на нужную вкладку, внести изменения и подтвердить операцию, нажав кнопку «Применить» (Рис. 355 – Рис. 356).

Вкладка «Память»

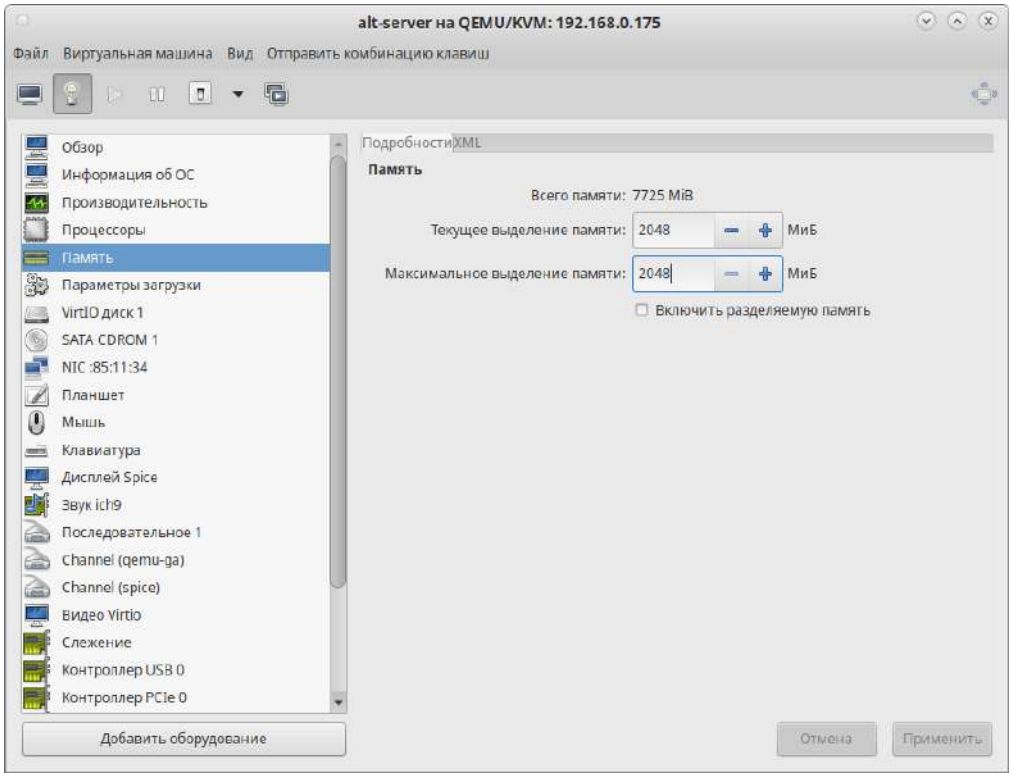


Рис. 355

Вкладка «Процессоры»

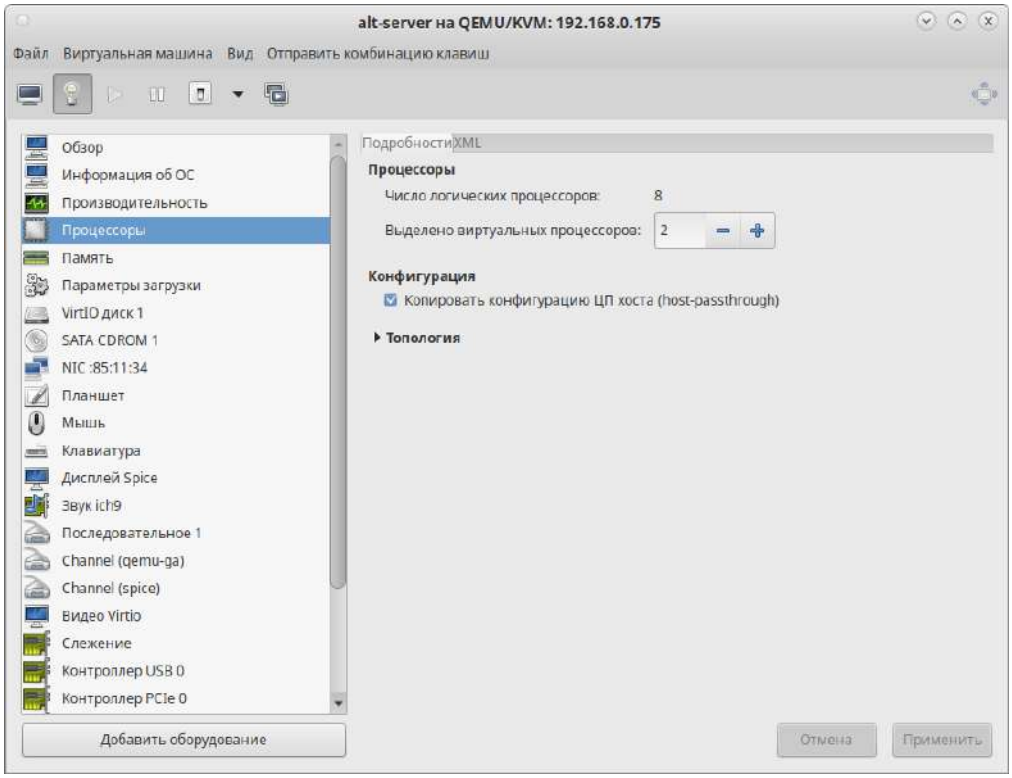


Рис. 356

5.7.1.4 Мониторинг состояния

С помощью менеджера виртуальных машин можно изменить настройки контроля состояния ВМ.

Для этого в меню «Правка» следует выбрать пункт «Настройки», в открывшемся окне на вкладке «Статистика» можно задать время обновления состояния ВМ в секундах (Рис. 357).

Вкладка «Статистика»

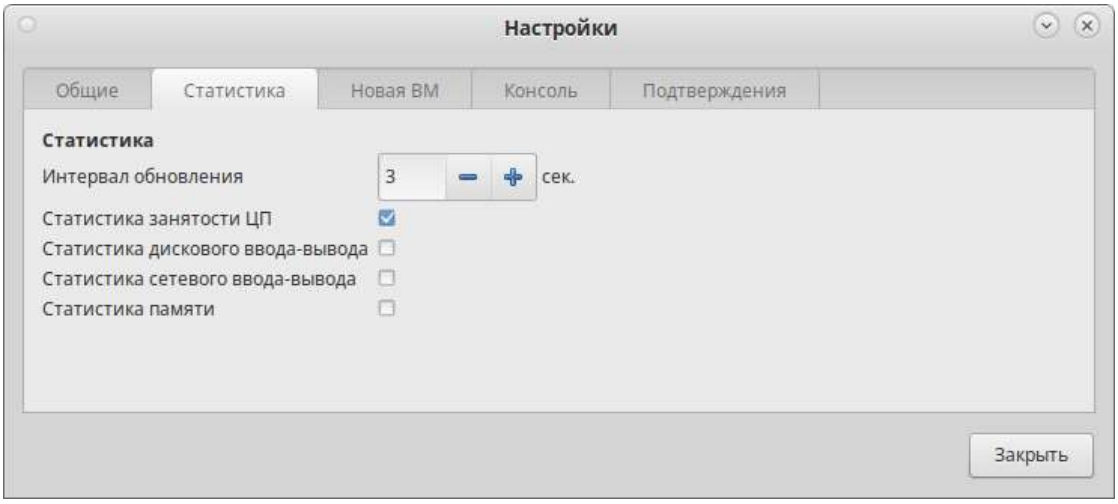


Рис. 357

Во вкладке «Консоль» (Рис. 358) можно выбрать, как открывать консоль, и указать устройство ввода.

Вкладка «Консоль»

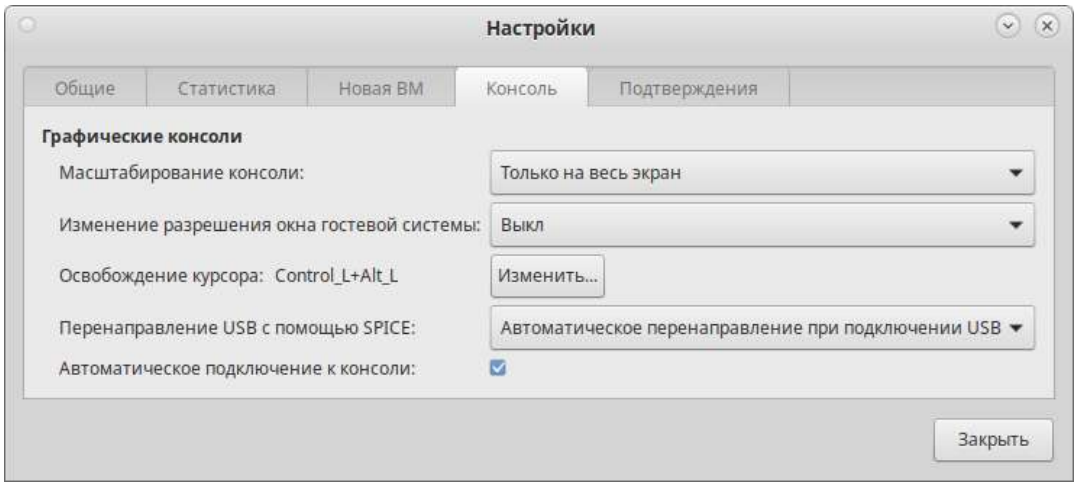


Рис. 358

5.8 Управление виртуальными сетевыми интерфейсами и сетями

Виртуальная сеть Libvirt использует концепцию виртуального сетевого коммутатора. Коммутатор виртуальной сети — это программная конструкция, которая работает на сервере физической машины. К коммутатору виртуальной сети подключаются ВМ (Рис. 359). Сетевой трафик для ВМ направляется через этот коммутатор.

Коммутатор виртуальной сети

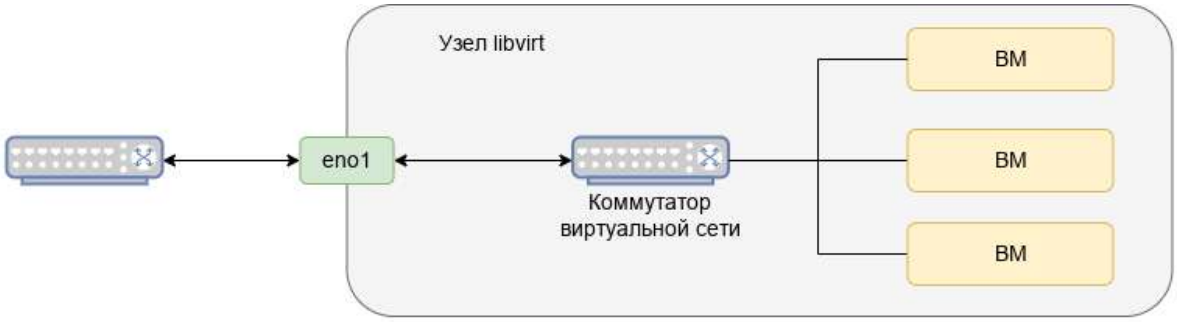


Рис. 359

Сразу после запуска службы libvirtd сетевым интерфейсом по умолчанию, представляющим виртуальный сетевой коммутатор, является virbr0:

```
$ ip addr show virbr0
9: virbr0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 52:54:00:6e:93:97 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
       valid_lft forever preferred_lft forever
```

В конфигурации по умолчанию (виртуальная сеть default на основе NAT) гостевая ОС будет иметь доступ к сетевым службам, но не будет видна другим машинам в сети.

По умолчанию гостевая ОС получит IP-адрес в адресном пространстве 192.168.122.0/24, а хостовая ОС будет доступна по адресу 192.168.122.1. Из гостевой ОС можно подключиться по SSH к хостовой ОС (по адресу 192.168.122.1) и использовать `scp` для копирования файлов.

Возможные варианты настройки сети:

- NAT –вариант по умолчанию. Внутренняя сеть, предоставляющая доступ к внешней сети с автоматическим применением NAT;
- Маршрутизация (Routed) – аналогично режиму NAT внутренняя сеть, предоставляющая доступ к внешней сети, но без NAT. Предполагает дополнительные настройки таблиц маршрутизации во внешней сети;
- Изолированный (Isolated) – в этом режиме ВМ, подключенные к виртуальному коммутатору, могут общаться между собой и с хостом. При этом их трафик не будет выходить за пределы хоста;
- Сеть на основе моста (Bridge) – подключение типа мост. Позволяет реализовать множество различных конфигураций, в том числе и назначение IP из реальной сети;
- SR-IOV pool (Single-root IOV) – перенаправление одной из PCI сетевых карт хост-машины на ВМ. Технология SR-IOV повышает производительность сетевой виртуализации, избавляя гипервизор от обязанности организовывать совместное использование физического

адаптера и перекладывая задачу реализации мультиплексирования на сам адаптер. В этом случае обеспечивается прямая пересылка ввода/вывода с ВМ непосредственно на адаптер.

Подробнее о настройках сети в разных режимах см. раздел «Режимы работы виртуальной сети».

5.8.1 Управление виртуальными сетями в командной строке

Команды управления виртуальными сетями:

- virsh net-autostart имя_сети – автоматический запуск заданной сети;
- virsh net-autostart имя_сети --disable – откоючить автозапуск заданной сети;
- virsh net-create файл_XML – создание и запуск новой сети на основе существующего XML-файла;
- virsh net-define файл_XML – создание нового сетевого устройства на основе существующего XML-файла (устройство не будет запущено);
- virsh net-destroy имя_сети – удаление заданной сети;
- virsh net-dumpxml имя_сети – просмотр информации о заданной виртуальной сети;
- virsh net-info имя_сети – просмотр основной информации о заданной виртуальной сети;
- virsh net-list – просмотр списка виртуальных сетей;
- virsh net-name UUID_сети – преобразование заданного идентификатора в имя сети;
- virsh net-start имя_неактивной_сети – запуск неактивной сети;
- virsh net-uuid имя_сети – преобразование заданного имени в идентификатор UUID;
- virsh net-update имя_сети – обновить существующую конфигурацию сети;
- virsh net-undefine имя_неактивной_сети — удаление определения неактивной сети.

Примеры:

```
# virsh net-list --all
Имя       Состояние    Автозапуск   Постоянный
-----------------------------------------------
 default   не активен   no           yes

# virsh net-start default
Сеть default запущен

# virsh net-autostart default
Добавлена метка автоматического запуска сети default

# virsh net-list
 Имя       Состояние   Автозапуск   Постоянный
-----------------------------------------------
 default   активен     yes          yes
```

```
# virsh net-dumpxml default
<network>
  <name>default</name>
  <uuid>0b37eff3-2234-4929-8a42-04a9cf35d3aa</uuid>
  <forward mode='nat'/>
  <bridge name='virbr0' stp='on' delay='0'/>
  <mac address='52:54:00:d2:30:b6'/>
  <ip address='192.168.122.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.122.2' end='192.168.122.254'/>
    </dhcp>
  </ip>
</network>
```

Иногда бывает полезно выдавать клиенту один и тот же IP-адрес независимо от момента обращения. Пример добавления статического сопоставления MAC- и IP-адреса ВМ:

1) получить MAC-адрес ВМ (alt-server – имя ВМ):

```
# virsh dumpxml alt-server | grep 'mac address'
   <mac address='52:54:00:ba:f2:76'/>
```

2) отредактировать XML-конфигурацию сети (default – имя сети):

```
# virsh net-edit default
```

после строки:

```
<range start='192.168.122.2' end='192.168.122.254'/>
```

вставить строки с MAC-адресами виртуальных адаптеров:

```
<host mac='52:54:00:ba:f2:76' name='alt-server' ip='192.168.122.50'/>
```

3) сохранить изменения и перезапустить виртуальную сеть:

```
# virsh net-destroy default
# virsh net-start default
```

Изменения, внесённые с помощью команды `virsh net-edit`, не вступят в силу в силу до тех пор, пока сеть не будет перезапущена, что приведет к потере всеми ВМ сетевого подключения к хосту до тех пор, пока их сетевые интерфейсы повторно не подключаться.

Изменения в конфигурацию сети можно внести с помощью команды `virsh net-update`, которая требует немедленного применения изменений. Например, чтобы добавить запись статического хоста, можно использовать команду:

```
# virsh net-update default add ip-dhcp-host \
   "<host mac='52:54:00:ba:f2:76' name='alt-server' ip='192.168.122.50' />" \
   --live --config
```

5.8.2 Управление виртуальными сетями в менеджере виртуальных машин

В менеджере виртуальных машин virt-manager существует возможность настройки виртуальных сетей для обеспечения сетевого взаимодействия ВМ как между собой, так и с хостовой ОС.

Для настройки виртуальной сети с помощью virt-manager необходимо:

1) в меню «Правка» выбрать «Свойства подключения» (Рис. 360);
2) в открывшемся окне перейти на вкладку «Виртуальные сети» (Рис. 361);
3) доступные виртуальные сети будут перечислены в левой части окна. Для доступа к настройкам сети необходимо выбрать сеть.

Меню «Правка»

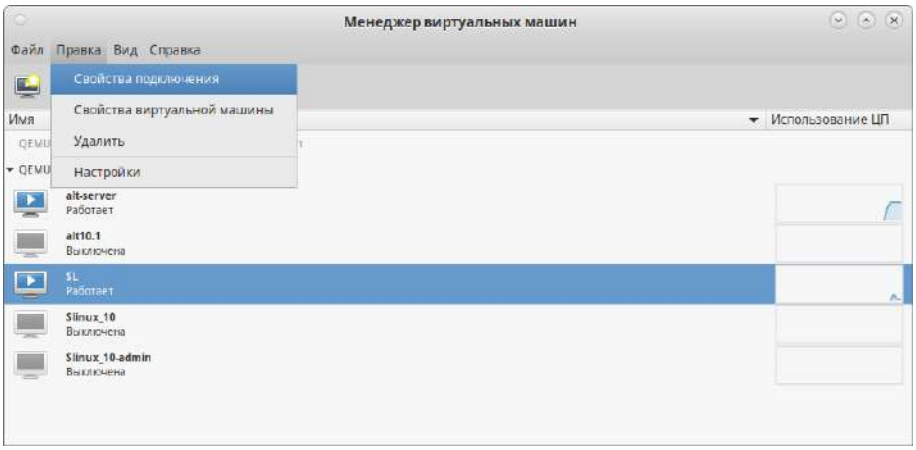


Рис. 360

Окно параметров виртуальной сети

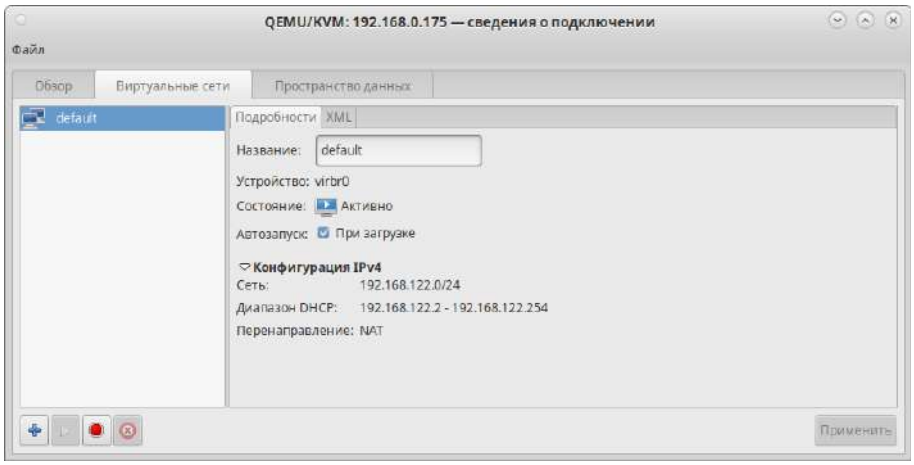


Рис. 361

Для добавления новой виртуальной сети следует нажать кнопку «Добавить сеть» («+»), расположенную в нижнем левом углу диалогового окна « Сведения о подключении» (Рис. 361). В открывшемся окне (Рис. 362) следует ввести имя для новой сети и задать необходимые настройки: выбрать способ подключения виртуальной сети к физической, ввести пространство адресов IPv4

для виртуальной сети, указать диапазон DHCP, задав начальный и конечный адрес и нажать кнопку «Готово».

Создание новой виртуальной сети

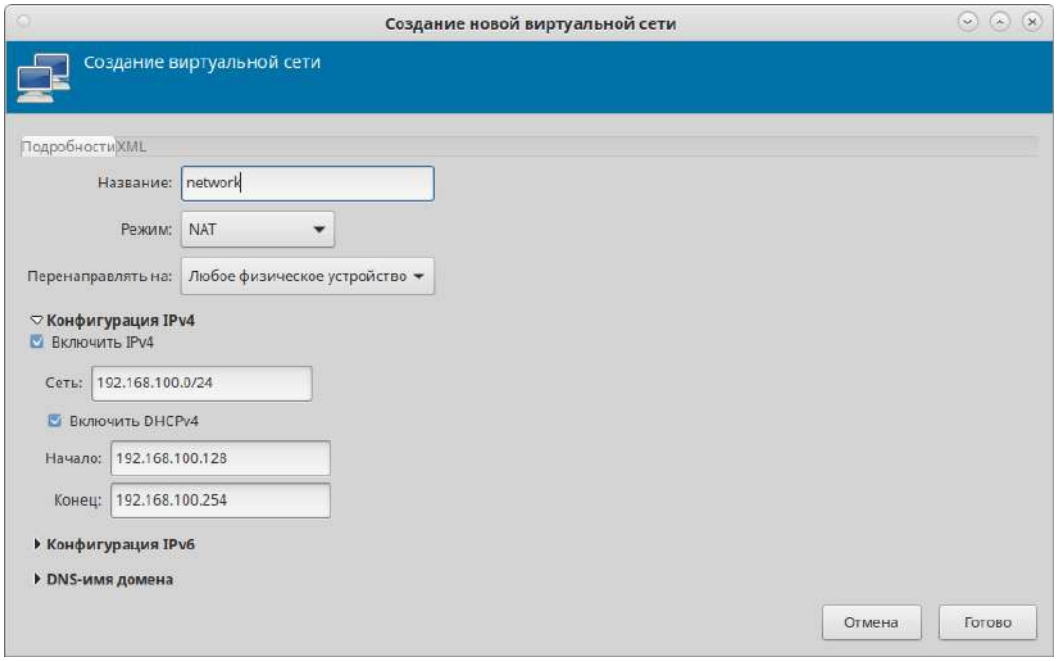


Рис. 362

5.8.3 Режимы работы виртуальной сети

5.8.3.1 Сеть на основе моста

Сеть на основе моста (Рис. 363) позволяет виртуальным интерфейсам подключаться к внешней сети через физический интерфейс, поэтому виртуальные интерфейсы выглядят как обычные хосты для остальной части сети.

Сеть на основе моста

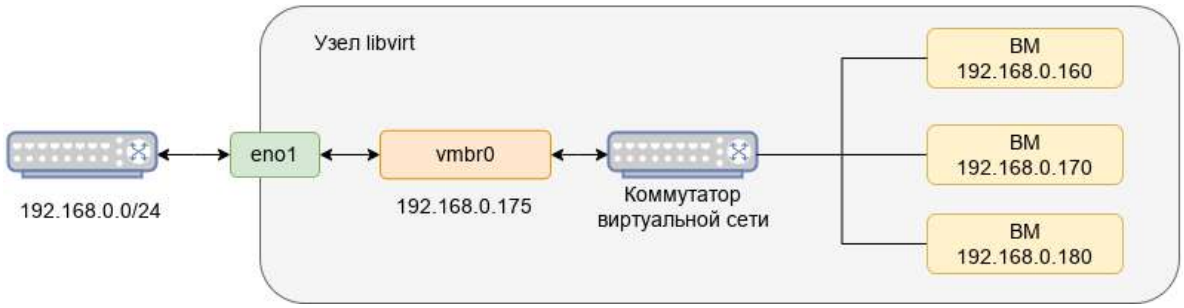


Рис. 363

Примечание. Сервер libvirt должен быть подключен к локальной сети через Ethernet. Если подключение осуществляется по беспроводной сети, следует использовать сеть с маршрутизацией или сеть на основе NAT.

Мост возможен только в том случае, если имеется достаточно IP-адресов, чтобы выделить один для каждой ВМ.

На сервере libvirt необходимо настроить Eternet-мост. Сделать это можно, например, воспользовавшись модулем ЦУС «Сетевые мосты» (см. раздел «Сетевые мосты»).

Созданный Eternet-мост можно указать при создании ВМ, например:

```
# virt-install --network bridge=vmbr0 ...
```

Для уже существующей ВМ можно указать Eternet-мост, отредактировав конфигурацию XML для ВМ. Для этого необходимо:

1) открыть конфигурацию XML ВМ в текстовом редакторе:

```
# virsh edit alt-server
```

2) найти раздел <interface>:

```
<interface type='network'>
    <mac address='52:54:00:85:11:34'/>
    <source network='default'/>
    <model type='virtio'/>
    <address type='pci' domain='0x0000' bus='0x01' slot='0x00' function='0x0'/>
</interface>
```

3) если необходимо изменить существующий интерфейс, заменить type='network' на type='bridge' и <source network='default'/> на <source bridge='vmbr0'/> (vmbr0 – интерфейс моста):

```
<interface type='bridge'>
    <mac address='52:54:00:85:11:34'/>
    <source bridge="vmbr0"/>
    <model type='virtio'/>
    <address type='pci' domain='0x0000' bus='0x01' slot='0x00' function='0x0'/>
</interface>
```

4) если необходимо добавить дополнительный интерфейс Ethernet, добавить новый раздел <interface> (libvirt сгенерирует случайный MAC-адрес для нового интерфейса, если <mac> опущен):

```
<interface type='bridge'>
    <source bridge="vmbr0"/>
</interface>
```

Чтобы указать Eternet-мост в менеджере виртуальных машин virt-manager, необходимо в окне настройки сетевого интерфейса ВМ в выпадающем списке «Создать на базе» выбрать пункт «Устройство моста...» и в поле «Название устройства» указать интерфейс моста (Рис. 364).

Виртуальный сетевой интерфейс на базе моста

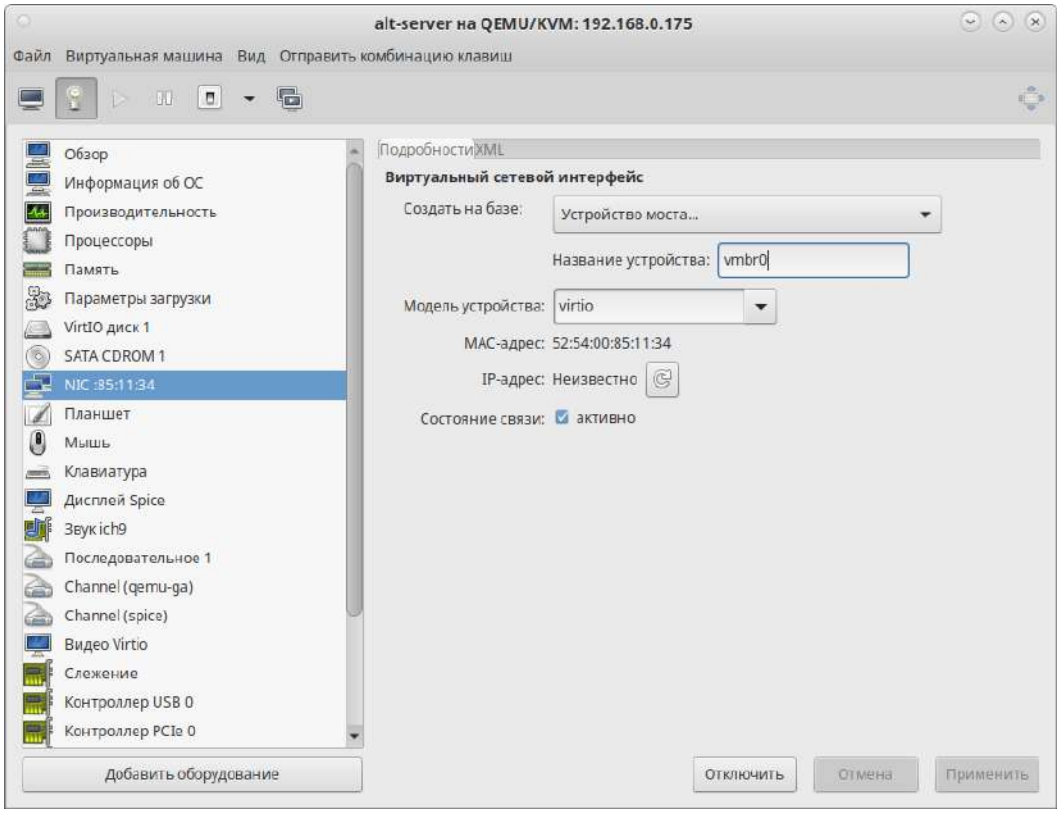


Рис. 364

5.8.3.2 Маршрутизируемая сеть

Маршрутизируемую сеть следует использовать только тогда, когда использовать сеть на базе моста невозможно (либо из-за ограничений хостинг-провайдера, либо из-за того, что сервер libvirt подключен к локальной сети по беспроводной сети.) При настройке маршрутизируемой сети все ВМ находятся в одной подсети (Рис. 365), маршрутизируемой через виртуальный коммутатор. Пакеты, предназначенные для этих адресов, статически маршрутизируются на сервер libvirt и пересылаются на ВМ (без использования NAT).

Коммутатор виртуальной сети в режиме маршрутизатора

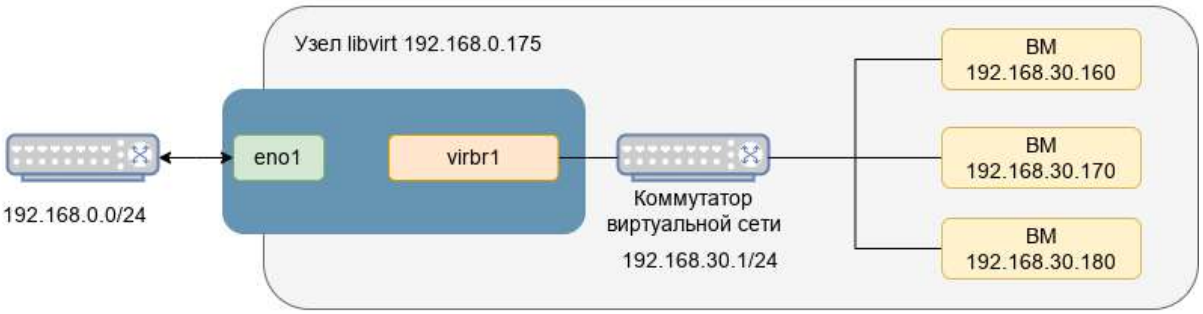


Рис. 365

Маршрутизируемая сеть возможна только в том случае, если имеется достаточно IP-адресов, чтобы выделить один для каждой ВМ.

В первую очередь необходимо выбрать, какие IP-адреса сделать доступными для ВМ (в примере 192.168.30.0/24). Так как маршрутизатор локальной сети не знает, что выбранная подсеть расположена на сервере libvirt, необходимо настроить статический маршрут на маршрутизаторе локальной сети, например:

```
# ip -4 route add 192.168.30.0/24 via 192.168.0.175
```

Далее необходимо создать виртуальную сеть.

Создание маршрутизируемой виртуальной сети в консоли:

1) создать файл `/tmp/routed_network.xml` со следующим содержимым:

```
<network>
<name>routed_network</name>
<forward mode="route"/>
<ip address="192.168.30.0" netmask="255.255.255.0">
<dhcp>
<range start="192.168.30.128" end="192.168.30.254"/>
</dhcp>
</ip>
</network>
```

2) определить новую сеть, используя файл `/tmp/routed_network.xml`:

```
# virsh net-define /tmp/routed_network.xml
Сеть routed_network определена на основе /tmp/routed_network.xml
# virsh net-autostart routed_network
Добавлена метка автоматического запуска сети routed_network
# virsh net-start routed_network
Сеть routed_network запущена
```

Создание виртуальной сети в режиме маршрутизации в virt-manager показано на **Рис. 366.**

Создание виртуальной сети в режиме маршрутизации

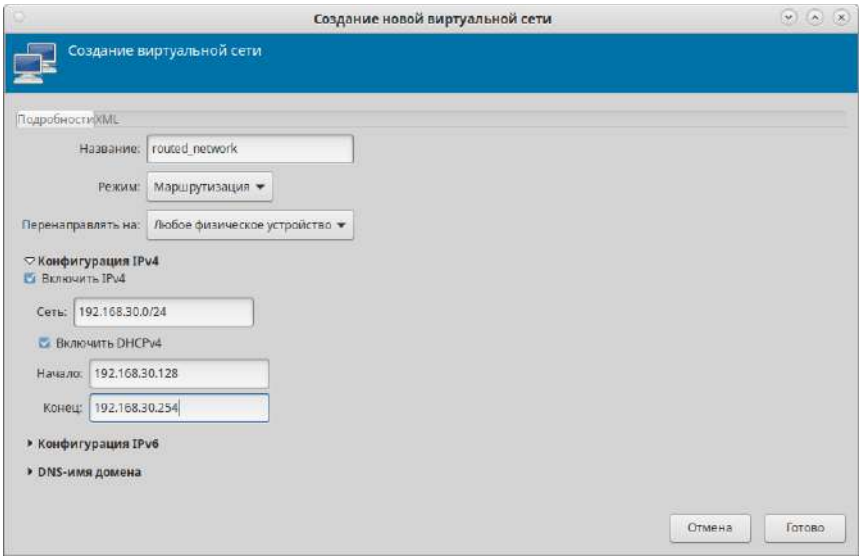


Рис. 366

Созданную виртальную сеть можно назначить ВМ, например, при создании ВМ:

```
# virt-install --network network=routed_network ...
```

Для уже существующей ВМ отредактировать конфигурацию XML для ВМ. Для этого необходимо:

1) открыть конфигурацию XML ВМ в текстовом редакторе:

```
# virsh edit alt-server
```

2) найти раздел <interface>:

```
<interface type='network'>
<mac address='52:54:00:85:11:34'/>
<source network='default'/>
<model type='virtio'/>
<address type='pci' domain='0x0000' bus='0x01' slot='0x00' function='0x0'/>
</interface>
```

3) если необходимо изменить существующий интерфейс, заменить название сети на routed_network:

```
<interface type='network'>
<mac address='52:54:00:85:11:34'/>
<source network='routed_network'/>
<model type='virtio'/>
<address type='pci' domain='0x0000' bus='0x01' slot='0x00' function='0x0'/>
</interface>
```

4) если необходимо добавить дополнительный интерфейс Ethernet, добавить новый раздел <interface>, (libvirt сгенерирует случайный MAC-адрес для нового интерфейса, если <mac> опущен):

```
<interface type='network'>
<source bridge="routed_network"/>
</interface>
```

Назначение маршрутизируемой сети в virt-manager показано на Рис. 367.

Назначение маршрутизируемой сети ВМ

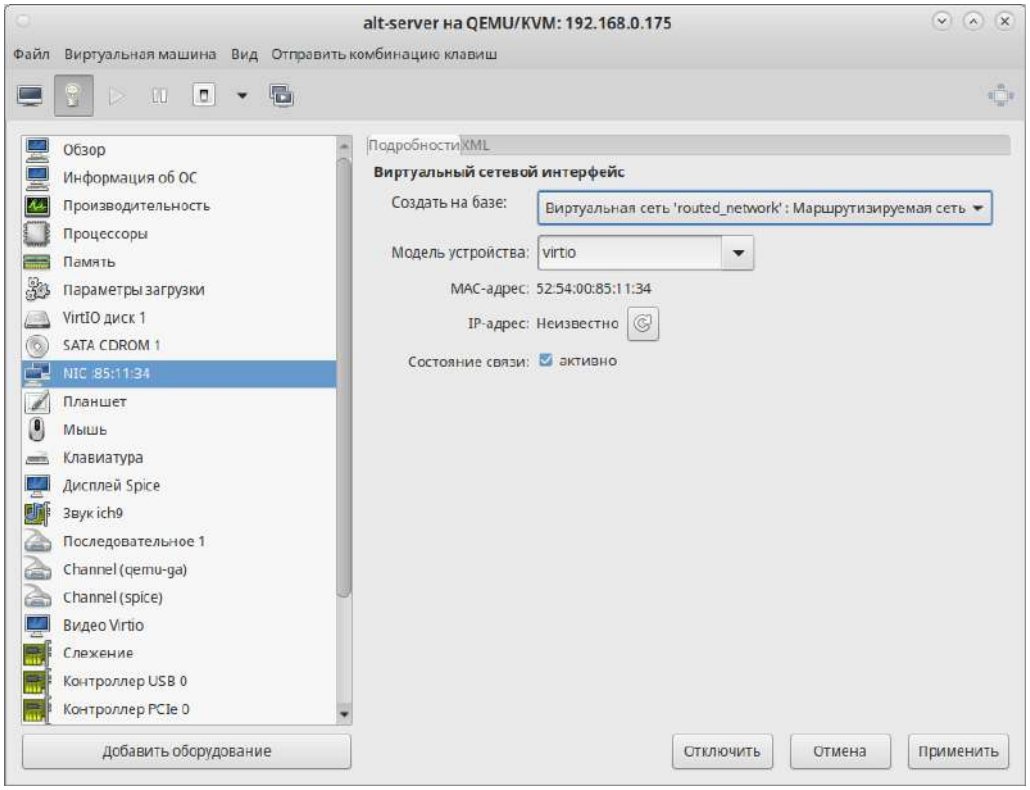


Рис. 367

5.8.3.3 Сеть на основе NAT

Сеть на основе NAT (Рис. 368) идеальна, когда требуется только доступ из ВМ к внешней сети. При этом сервер libvirt действует как маршрутизатор, и трафик ВМ исходит с IP-адреса сервера.

Коммутатор виртуальной сети в режиме NAT

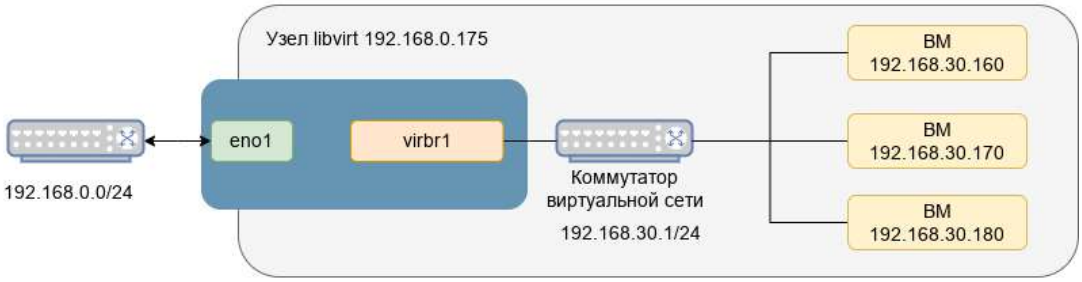


Рис. 368

Виртуальная сеть default (доступна после установки libvirt) основана на NAT. Можно также создать собственную сеть на основе NAT.

Создание виртуальной сети в режиме NAT в консоли:

1) создать файл `/tmp/nat_network.xml` со следующим содержимым:

```
<network>
<name>nat_network</name>
<forward mode="nat"/>
```

```
<ip address="192.168.20.1" netmask="255.255.255.0">
<dhcp>>
    <range start="192.168.20.128" end="192.168.20.254"/>
</dhcp>
</ip>
</network>
```

2) определить новую сеть, используя файл /tmp/nat_network.xml:

```
# virsh net-define /tmp/nat_network.xml
# virsh net-autostart nat_network
# virsh net-start nat_network
```

Создание виртуальной сети в режиме NAT в менеджере виртуальных машин показано на **Рис. 369**.

Создание виртуальной сети в режиме NAT

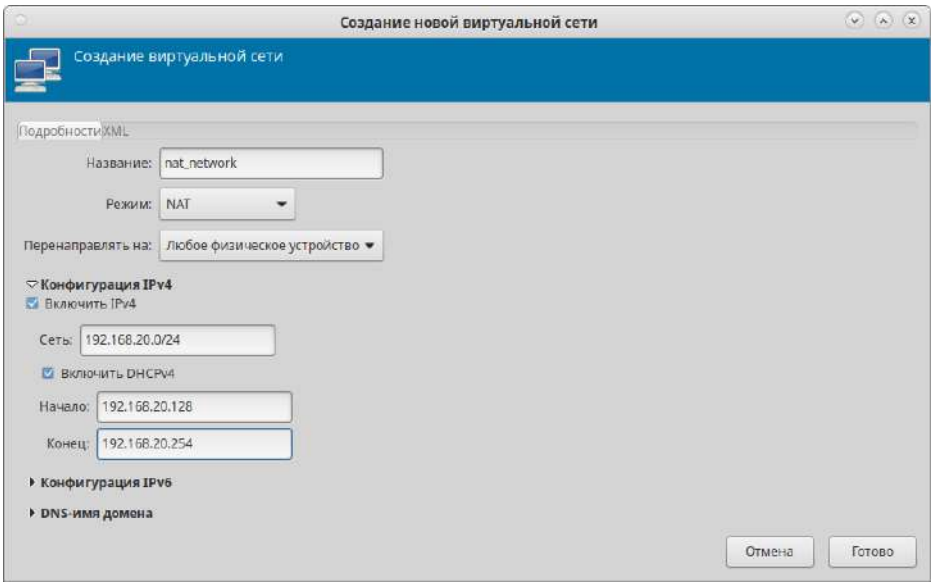


Рис. 369

Созданную виртальную сеть можно назначить ВМ, например, при создании ВМ:

```
# virt-install --network network=nat_network ...
```

Для уже существующей ВМ отредактировать конфигурацию XML для ВМ. Для этого необходимо:

1) открыть конфигурацию XML ВМ в текстовом редакторе:

```
# virsh edit alt-server
```

2) найти раздел <interface>:

```
<interface type='network'>
    <mac address='52:54:00:85:11:34'/>
    <source network='default'/>
    <model type='virtio'/>
    <address type='pci' domain='0x0000' bus='0x01' slot='0x00' function='0x0'/>
</interface>
```

3) если необходимо изменить существующий интерфейс, заменить название сети на nat_network:

```
<interface type='network'>
    <mac address='52:54:00:85:11:34'/>
    <source network='nat_network'/>
    <model type='virtio'/>
    <address type='pci' domain='0x0000' bus='0x01' slot='0x00' function='0x0'/>
</interface>
```

4) если необходимо добавить дополнительный интерфейс Ethernet, добавить новый раздел <interface> (libvirt сгенерирует случайный MAC-адрес для нового интерфейса, если <mac> опущен):

```
<interface type='network'>
    <source bridge="nat_network"/>
</interface>
```

Назначение виртуальной сети в режиме NAT в virt-manager показано на Рис. 370.

Назначение сети в режиме NAT ВМ

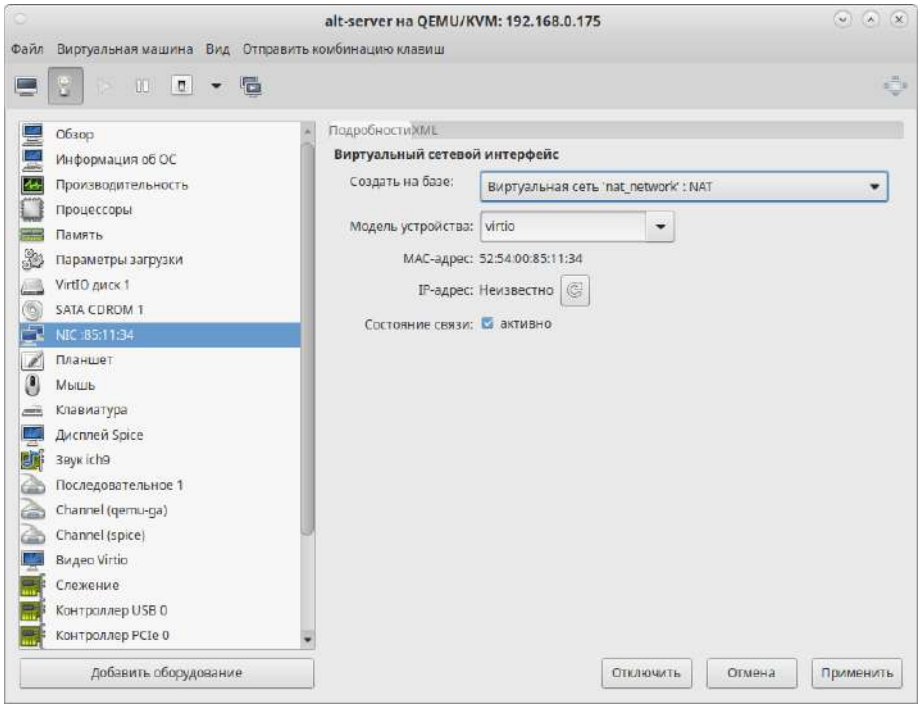


Рис. 370

5.8.3.4 Изолированная сеть

При использовании изолированного режима ВМ, подключенные к виртуальному коммутатору, могут взаимодействовать друг с другом и с физической машиной хоста, но их трафик не будет проходить за пределы физической машины хоста, и они не могут получать трафик извне физической машины хоста (Рис. 371).

Коммутатор виртуальной сети в режиме изоляции

Рис. 371

Создание изолированной виртуальной сети в консоли:

1) создать файл `/tmp/isolated_network.xml` со следующим содержимым:

```
<network>
<name>isolated_network</name>
<ip address="192.168.100.1" netmask="255.255.255.0">
<dhcp>
    <range start="192.168.100.128" end="192.168.     00.254"/>
</dhcp>
</ip>
</network>
```

2) определить новую сеть, используя файл `/tmp/isolated_network.xml`:

```
# virsh net-define /tmp/isolated_network.xml
# virsh net-autostart isolated_network
# virsh net-start isolated_network
```

Создание изолированной виртуальной сети в менеджере виртуальных машин показано на Рис. 372**.**

Созданную виртальную сеть можно назначить ВМ, например, при создании ВМ:

```
# virt-install --network network=isolated_network ...
```

Для уже существующей ВМ отредактировать конфигурацию XML для ВМ. Для этого необходимо:

1) открыть конфигурацию XML ВМ в текстовом редакторе:

```
# virsh edit alt-server
```

2) найти раздел <interface>:

```
<interface type='network'>
    <mac address='52:54:00:85:11:34'/>
    <source network='default'/>
    <model type='virtio'/>
    <address type='pci' domain='0x0000' bus='0x01' slot='0x00' function='0x0'/>
</interface>
```

3) если необходимо изменить существующий интерфейс, заменить название сети на isolated_network:

```
<interface type='network'>
    <mac address='52:54:00:85:11:34'/>
    <source network=isolated_network '/>
    <model type='virtio'/>
    <address type='pci' domain='0x0000' bus='0x01' slot='0x00' function='0x0'/>
</interface>
```

4) если необходимо добавить дополнительный интерфейс Ethernet, добавить новый раздел <interface> (libvirt сгенерирует случайный MAC-адрес для нового интерфейса, если <mac> опущен):

```
<interface type='network'>
    <source bridge="isolated_network "/>
</interface>
```

Создание виртуальной сети в режиме изоляции

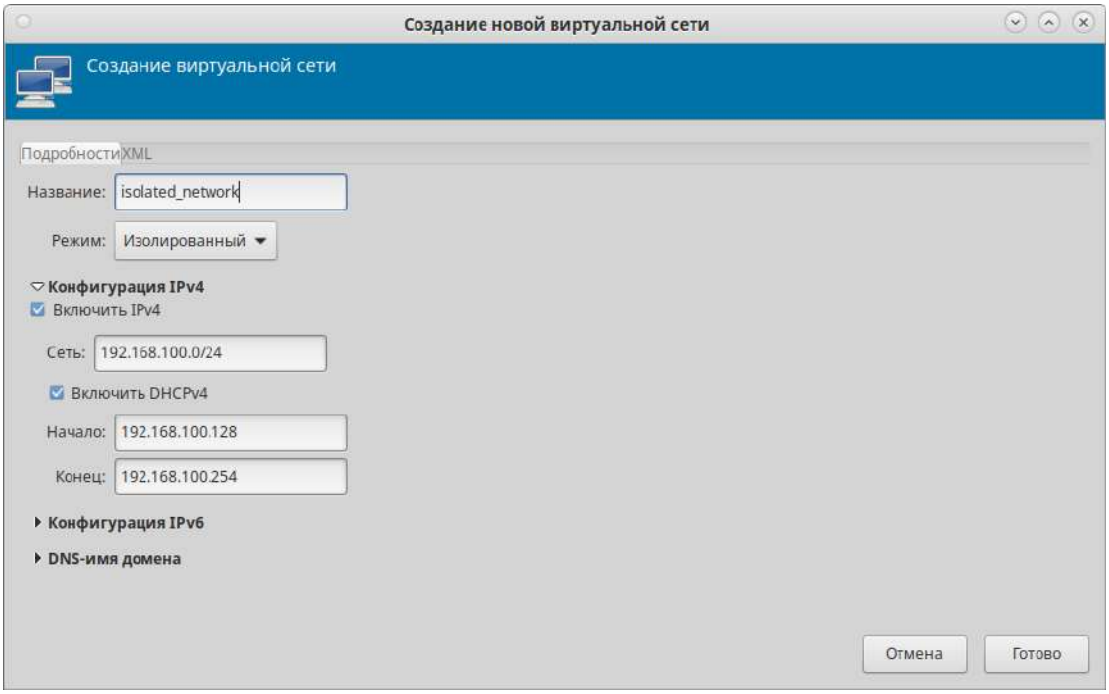


Рис. 372

Назначение виртуальной сети в режиме изоляции в virt-manager показано на Рис. 373.

Назначение сети в режиме изоляции ВМ

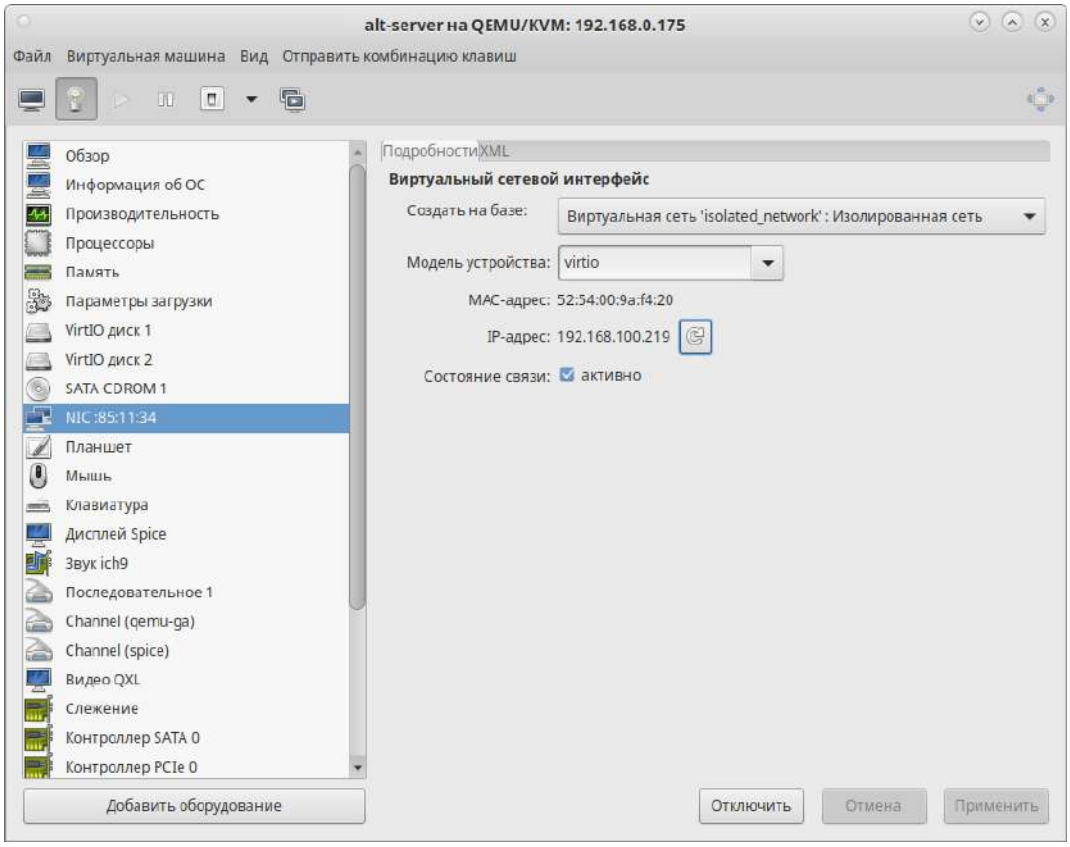


Рис. 373

5.9 Управление хранилищами

API-интерфейс libvirt обеспечивает удобную абстракцию для размещения образов ВМ и файловых систем, которая носит название storage pools (пул хранилищ). Пул хранилищ – это локальный каталог, локальное устройство хранения данных (физический диск, логический том или хранилище на основе хост-адаптера шины SCSI [SCSI HBA]), файловая система NFS (network file system), либо сетевое хранилище блочного уровня, управляемое посредством libvirt и позволяющее создать и хранить один или более образов виртуальных машин.

По умолчанию команды на базе libvirt используют в качестве исходного пула хранилищ для каталога файловой системы каталог `/var/lib/libvirt/images` на хосте виртуализации.

Образ диска – это снимок данных диска виртуальной машины, сохраненный в том или ином формате. Libvirt понимает несколько форматов образов. Возможна также работа с образами CD/DVD дисков. Каждый образ хранится в том или ином хранилище.

Типы хранилищ, с которыми работает libvirt:

- dir – каталог в файловой системе;
- disk – физический диск;
- fs – отформатированное блочное устройство;
- gluster – файловая система Gluster;

- isci – хранилище iSCSI;
- logical – группа томов LVM;
- mpath – регистратор многопутевых устройств;
- netfs – экспорт каталога из сети;
- rbd – блочное устройство RADOS/Ceph;
- scsi – хост-адаптер SCSI;
- sheepdog – файловая система Sheepdog;
- zfs – пул ZFS.

5.9.1.1 Управление хранилищами в командной строке

Команды управления хранилищами:

- pool-define – определить неактивный постоянный пул носителей на основе файла XML;
- pool-create – оздать пул из файла XML;
- pool-define-as – определить пул на основе набора аргументов;
- pool-create-as – создать пул на основе набора аргументов;
- pool-dumpxml – вывести файл конфигурации XML для заданного пула;
- pool-list – вывести список пулов;
- pool-build – собрать пул;
- pool-start – запустить ранее определённый неактивный пул;
- pool-autostart – автозапуск пула;
- pool-destroy – разрушить (остановить) пул;
- pool-delete – удалить пул;
- pool-edit – редактировать XML-конфигурацию пула носителей;
- pool-info – просмотр информации о пуле носителей;
- pool-refresh – обновить пул;
- pool-undefine – удалить определение неактивного пула.

Команда `virsh pool-define-as` создаст файл конфигурации для постоянного пула хранения. Позже этот пул можно запустить командой `virsh pool-start`, настроить его на автоматический запуск при загрузке хоста, остановить командой `virsh pool-destroy`.

Команда `virsh pool-create-as` создаст временный пул хранения (файл конфигурации не будет создан), который будет сразу запущен. Этот пул хранения будет удалён командой `virsh pool-destory`. Временный пул хранения нельзя запустить автоматически при загрузке. Преобразовать существующий временный пул в постоянный, можно создав файл XML-описания:

```
virsh pool-dumpxml имя_пула > имя_пула.xml && virsh pool-define имя_пула.xml
```

Пример создания пула хранения на основе NFS (netfs):

```
# virsh pool-create-as NFS-POOL netfs \
```

```
--source-host 192.168.0.105 \
--source-path /export/storage \
--target /var/lib/libvirt/images/NFS-POOL
Пул NFS-POOL создан
```

Первый аргумент (NFS-POOL) идентифицирует имя нового пула, второй аргумент идентифицирует тип создаваемого пула. Аргумент опции `--source-host` идентифицирует хост, который экспортирует каталог пула хранилищ посредством NFS. Аргумент опции `--source-path` определяет имя экспортируемого каталога на этом хосте. Аргумент опции `--target` идентифицирует локальную точку монтирования, которая будет использоваться для обращения к пулу хранилищ (этот каталог должен существовать).

Примечание. Для возможности монтирования NFS хранилища должен быть запущен nfs-client:

```
# systemctl enable --now nfs-client.target
```

После создания нового пула хранилищ он будет указан в выходной информации команды virsh pool-list:

```
virsh pool-list --all --details
 Имя        Состояние   Автозапуск   Постоянный   Размер      Распределение   Доступно
-------------------------------------------------------------------------------------
 default    работает    yes          yes          69,12 GiB   485,35 MiB      68,65 GiB
 NFS-POOL   работает    no           no           29,40 GiB   7,26 GiB        22,14 GiB
```

В выводе команды видно, что опция «Автозапуск» («Autostart») для пула NFS-POOL имеет значение no (нет), т. е. после перезапуска системы этот пул не будет автоматически доступен для использования, и что опция «Постоянный» («Persistent») также имеет значение «no», т.е. после перезапуска системы этот пул вообще не будет определен. Пул хранилищ является постоянным только в том случае, если он сопровождается XML-описанием пула хранилищ, которое находится в каталоге `/etc/libvirt/storage`. XML-файл описания пула хранилищ имеет такое же имя, как у пула хранилищ, с которым он ассоциирован.

Чтобы создать файл XML-описания для сформированного в ручном режиме пула, следует воспользоваться командой `virsh pool-dumpxml`, указав в качестве ее заключительного аргумента имя пула, для которого нужно получить XML-описание. Эта команда осуществляет запись в стандартный поток вывода, поэтому необходимо перенаправить выводимую ей информацию в соответствующий файл.

Например, следующая команда создаст файл XML-описания для созданного ранее пула NFS-POOL и определит постоянный пул на основе этого файла:

```
# virsh pool-dumpxml NFS-POOL > NFS-POOL.xml && virsh pool-define NFS-POOL.xml
Пул NFS-POOL определён на основе NFS-POOL.xml
```

Чтобы задать для пула хранилищ опцию «Автозапуск» («Autostart»), можно воспользоваться командой `virsh pool-autostart`:

```
# virsh pool-autostart NFS-POOL
Добавлена метка автоматического запуска пула NFS-POOL
```

Маркировка пула хранилищ как автозапускаемого говорит о том, что этот пул хранилищ будет доступен после любого перезапуска хоста виртуализации (каталог `/etc/libvirt/storage/autostart` будет содержать символьную ссылку на XML-описание этого пула хранилищ).

Пример создания постоянного локального пула:

```
# virsh pool-define-as boot --type dir --target /var/lib/libvirt/boot
Пул boot определён

# virsh pool-list --all
 Имя         Состояние    Автозапуск
-------------------------------------
 boot        не активен   no
 default     активен      yes
 NFS-POOL    активен      yes

# virsh pool-build boot
Пул boot собран

# virsh pool-start boot
Пул boot запущен

# virsh pool-autostart boot
Добавлена метка автоматического запуска пула newpool

# virsh pool-list --all
 Имя         Состояние    Автозапуск
-------------------------------------
 boot        активен     yes
 default     активен     yes
 NFS-POOL    активен     yes
```

5.9.1.2 Настройка хранилищ в менеджере виртуальных машин

Для настройки хранилищ с помощью virt-manager необходимо:

1) в меню «Правка» выбрать «Свойства подключения» (Рис. 360);
2) в открывшемся окне перейти на вкладку «Пространство данных» (Рис. 374).

Вкладка «Пространство данных»

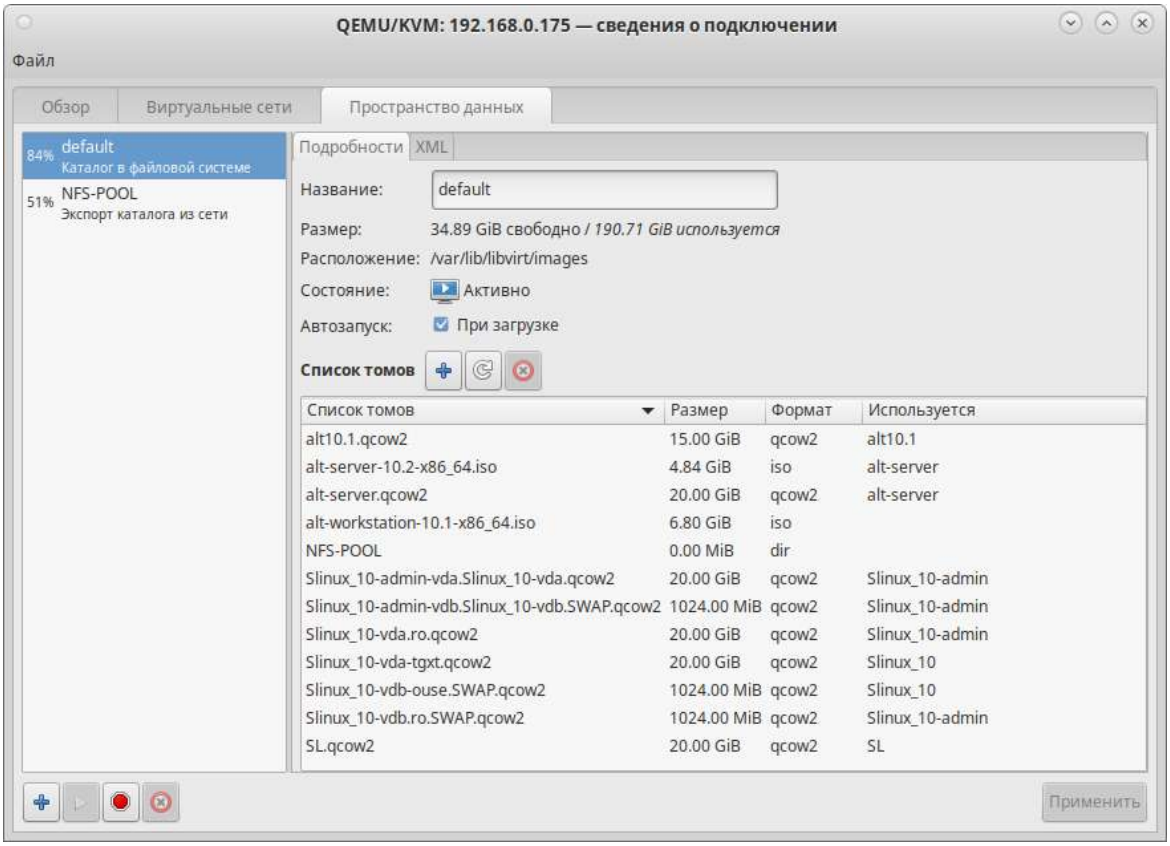


Рис. 374

Для добавления пула следует нажать кнопку «Добавить пул» («+»), расположенную в нижнем левом углу диалогового окна «Сведения о подключении» (Рис. 374). В открывшемся окне (Рис. 375) следует выбрать тип пула, далее необходимо задать параметры пула (Рис. 376).

Создание пула хранения. Выбор типа пула

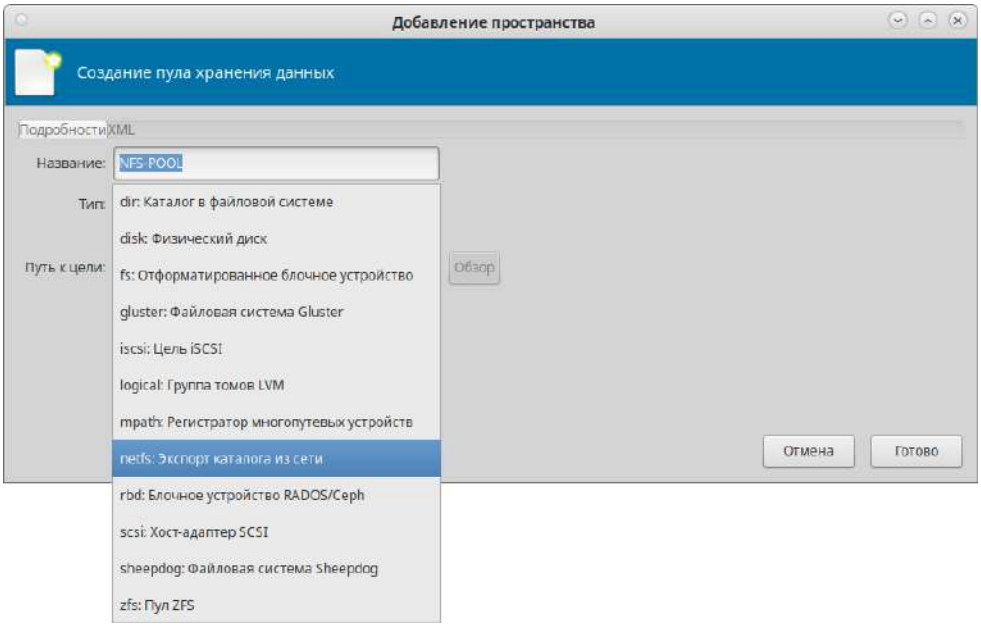


Рис. 375

Создание пула хранения. Ввод параметров

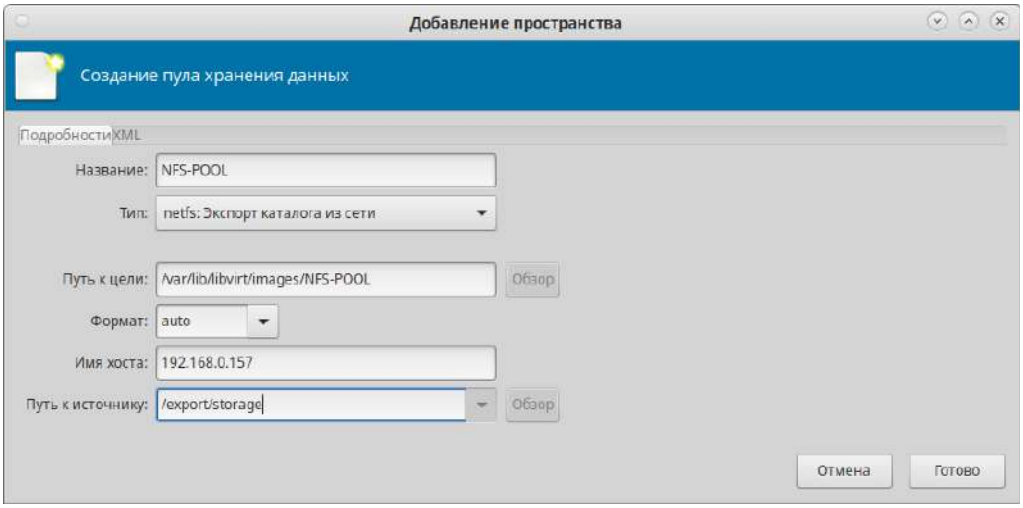


Рис. 376

5.10 Миграция ВМ

Под миграцией понимается процесс переноса ВМ с одного узла на другой.

Живая миграция позволяет перенести работу ВМ с одного физического хоста на другой без остановки ее работы.

Для возможности миграции, ВМ должна быть создана с использованием общего пула хранилищ (NFS, ISCSI, GlusterFS, CEPH).

Примечание. Живая миграция возможна даже без общего хранилища данных (с опцией `--copy-storage-all`). Но это приведет к большому трафику при копировании образа ВМ между серверами виртуализации и к заметному простою сервиса. Что бы миграция была по-настоящему «живой» с незаметным простоем необходимо использовать общее хранилище.

5.10.1 Миграция с помощью virsh

ВМ можно перенести на другой узел с помощью команды virsh. Для выполнения живой миграции нужно указать параметр `--live`. Команда переноса:

```
# virsh migrate --live VMName DestinationURL
```

где VMName – имя перемещаемой ВМ;

DestinationURL – URL или имя хоста узла назначения. Узел назначения должен использовать тот же гипервизор и служба libvirt на нем должна быть запущена.

После ввода команды будет запрошен пароль администратора узла назначения.

Для выполнения живой миграции ВМ alt-server на узел 192.168.0.195 с помощью virsh, необходимо выполнить следующие действия:

1) убедиться, что ВМ запущена:

```
# virsh list
ID   Имя          Состояние
```

```
------------------------------
 7    alt-server    работает
```

2) выполнить следующую команду, чтобы начать перенос ВМ на узел 192.168.0.195 (после ввода команды будет запрошен пароль пользователя root системы назначения):

```
# virsh migrate --live alt-server qemu+ssh://192.168.0.195/system
```

3) процесс миграции может занять некоторое время в зависимости от нагрузки и размера ВМ. virsh будет сообщать только об ошибках. ВМ будет продолжать работу на исходном узле до завершения переноса;

4) проверить результат переноса, выполнив на узле назначения команду:

```
# virsh list
```

5.10.2 Миграция с помощью virt-manager

Менеджер виртуальных машин virt-manager поддерживает возможность миграции ВМ между серверами виртуализации.

Для выполнения миграции, в virt-manager необходимо выполнить следующие действия:

1) подключить второй сервер виртуализации («Файл»→ «Добавить соединение…»);
2) в контекстном меню ВМ (она должна быть запущена) (Рис. 377) выбрать пункт «Миграция…»;
3) в открывшемся окне (Рис. 378) выбрать конечный узел и нажать кнопку «Миграция».

Пункт «Миграция…» в контекстном меню ВМ

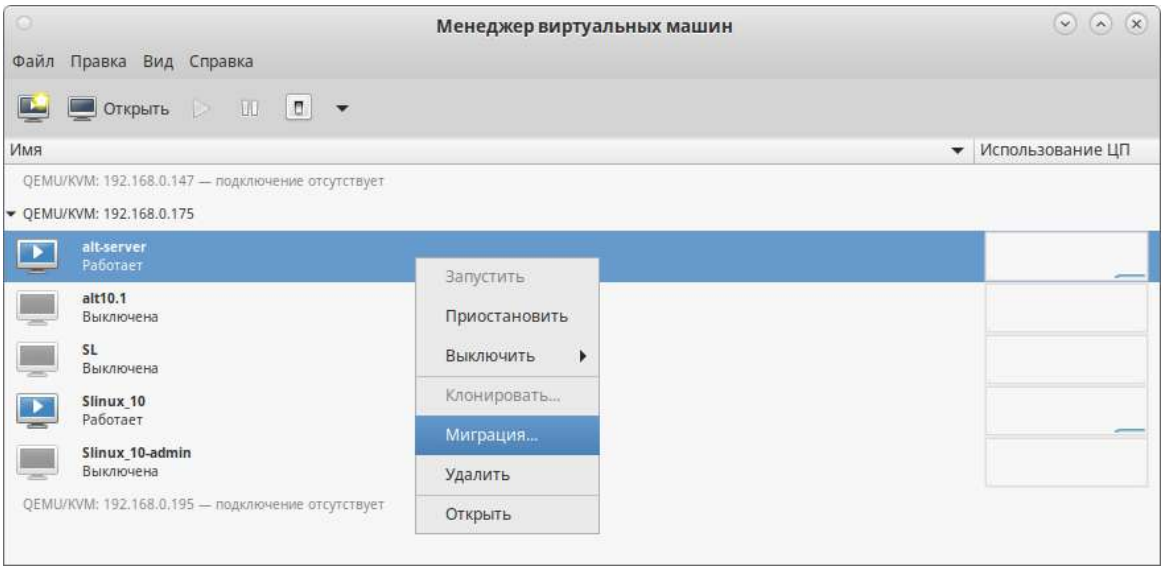


Рис. 377

Миграция ВМ

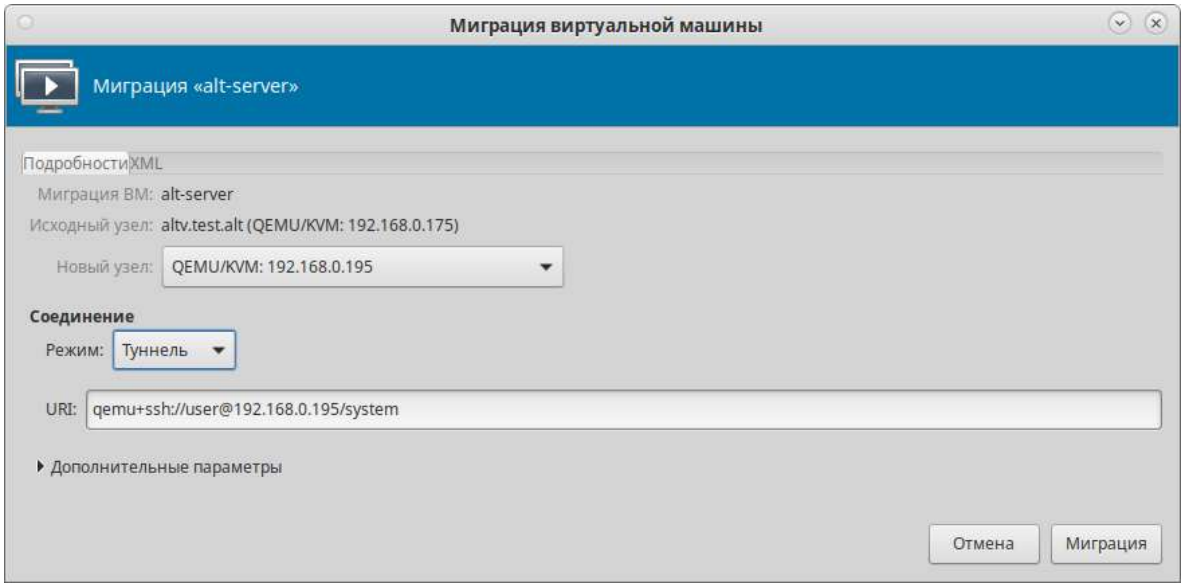


Рис. 378

При этом конфигурационный файл перемещаемой машины не перемещается на новый узел, поэтому при выключении ВМ она вновь появится на старом хосте. В связи с этим, для совершения полной живой миграции, при которой конфигурация ВМ будет перемещена на новый узел, необходимо воспользоваться утилитой командной строки virsh:

```
# virsh migrate --live --persistent --undefinesource \
alt-server qemu+ssh://192.168.0.195/system
```

5.11 Снимки машины

Примечание. Снимок (snapshot) текущего состояния машины можно создать только если виртуальный жесткий диск в формате *.qcow2.

5.11.1 Управления снимками ВМ в консоли

Команда создания снимка (ОЗУ и диск) из файла XML:

```
# virsh snapshot-create <domain> [--xmlfile <строка>] [--disk-only] [--live]...
```

Команда создания снимка (ОЗУ и диск) напрямую из набора параметров:

```
# virsh snapshot-create-as <domain> [--name <строка>] [--disk-only] [--live]...
```

Пример создания снимка ВМ:

```
# virsh snapshot-create-as --domain alt-server --name alt-server-17mar2024
Снимок домена alt-server-17mar2024 создан
```

где

alt-server – имя ВМ;

alt-server-17mar2024 – название снимка.

После того как снимок ВМ будет сделан, резервные копии файлов конфигураций будут находиться в каталоге `/var/lib/libvirt/qemu/snapshot/`.

Пример создания снимка диска ВМ:

```
# virsh snapshot-create-as --domain alt-server --name 03apr2024 \
--diskspec vda,file=/var/lib/libvirt/images/sn1.qcow2 --disk-only --atomic
Снимок домена 03apr2024 создан
```

Просмотр существующих снимков для домена alt-server:

```
# virsh snapshot-list --domain alt-server
Имя                    Время создания              Состояние
------------------------------------------------------------------
 03apr2024              2024-04-03 18:14:39 +0200   disk-snapshot
 alt-server-17mar2024   2024-03-17 18:06:29 +0200   running
```

Восстановить ВМ из снимка:

```
# virsh snapshot-revert --domain alt-server --snapshotname 03apr2024 --running
```

Удалить снимок:

```
# virsh snapshot-delete --domain alt-server --snapshotname 03apr2024
```

5.11.2 Управления снимками ВМ virt-manager

Для управления снимками ВМ в менеджере виртуальных машин virt-manager, необходимо:

1) в главном окне менеджера выбрать ВМ;
2) нажать кнопку «Открыть»;
3) в открывшемся окне нажать кнопку «Управление снимками» (Рис. 379). Появится окно управления снимками ВМ.

Управление снимками ВМ

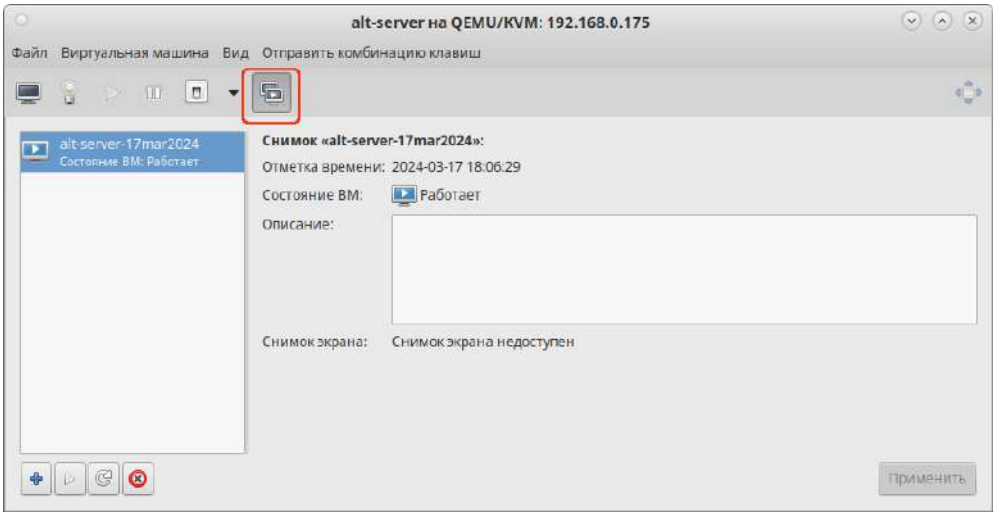


Рис. 379

Для создания нового снимка следует нажать кнопку «Создать новый снимок», расположенную в нижнем левом углу окна управления снимками ВМ. В открывшемся окне (Рис. 380) следует указать название снимка и нажать кнопку «Готово».

Создание снимка

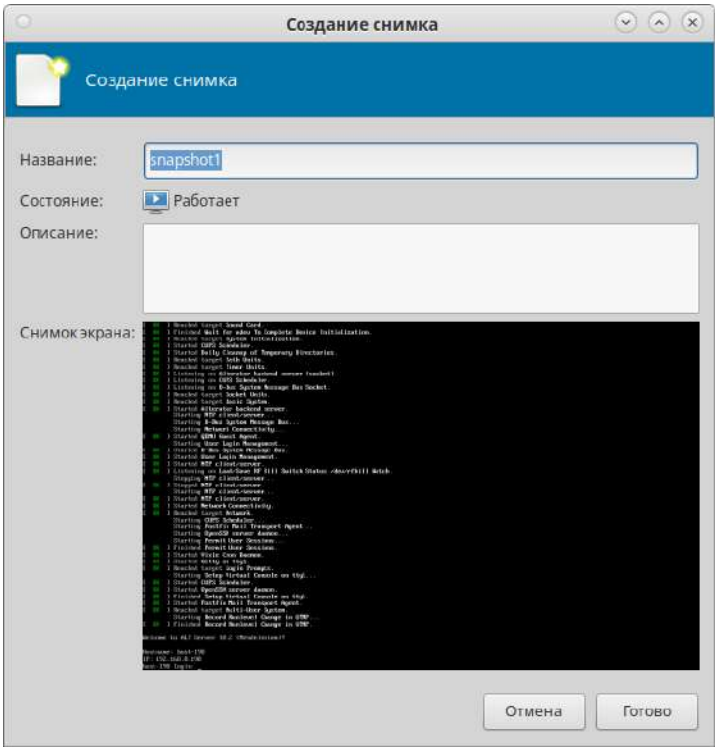


Рис. 380

Для того чтобы восстановить ВМ из снимка или удалить снимок, следует воспользоваться контекстным меню снимка (Рис. 381).

Контекстное меню снимка

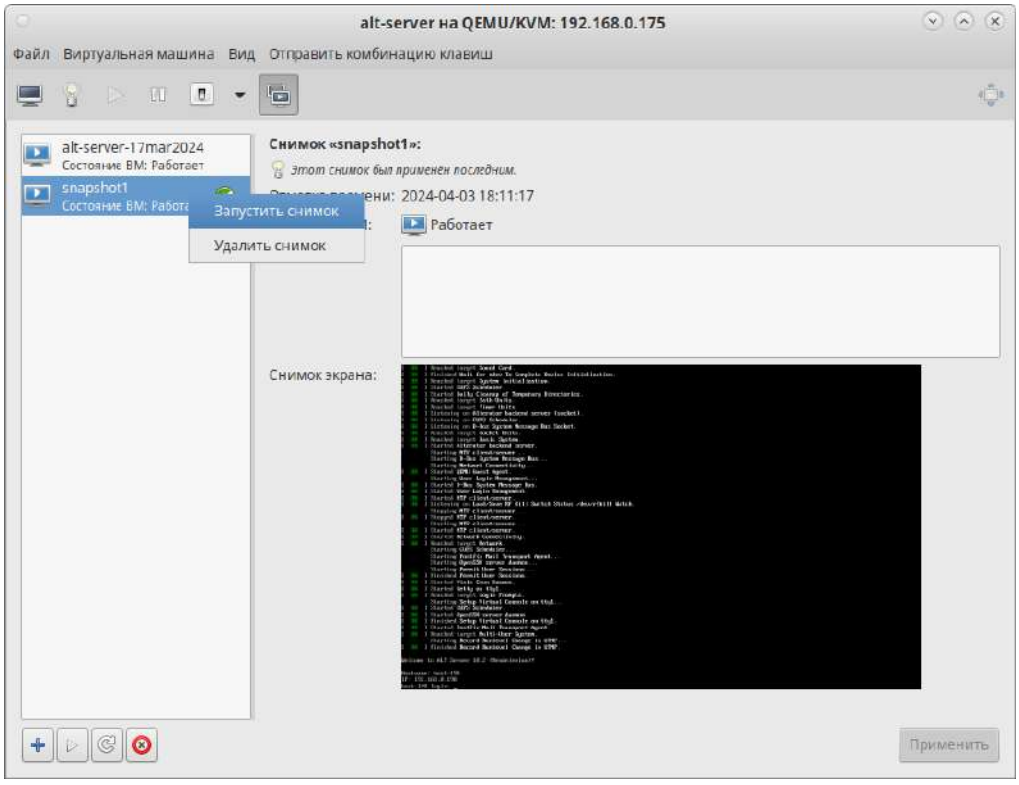


Рис. 381

5.12 Регистрация событий libvirt

Настройка регистрации событий в libvirt осуществляется в файле `/etc/libvirt/libvirtd.conf`. Логи сохраняются в каталоге `/var/log/libvirt`.

Функция журналирования в libvirt основана на трех ключевых понятиях:

- сообщения журнала;
- фильтры;
- формат ввода.

Сообщения журнала – это информация, полученная во время работы libvirt. Каждое сообщение включает в себя уровень приоритета (отладочное сообщение – 1, информационное – 2, предупреждение – 3, ошибка – 4). По умолчанию log_level=1, т. е. журналируются все сообщения.

Фильтры – это набор шаблонов и приоритетов для принятия или отклонения сообщений журнала. Если категория сообщения совпадает с фильтром, приоритет сообщения сравнивается с приоритетом фильтра, если он ниже, сообщение отбрасывается, иначе сообщение записывается в журнал. Если сообщение не соответствует ни одному фильтру, то применяется общий уровень приоритета. Это позволяет, например, захватить все отладочные сообщения для QEMU, а для остальных, только сообщения об ошибках.

Формат для фильтра:

```
x:name  (log message only)
x:+name (log message + stack trace)
```

где:

- name – строка, которая сравнивается с заданной категорией, например, remote, qemu, или util.json;
- + – записывать каждое сообщение с данным именем;
- x – минимальный уровень ошибки (1, 2, 3, 4).

Пример фильтра:

```
log_filtrers="3:remote 4:event"
```

Как только сообщение прошло через фильтрацию набора выходных данных, формат вывода определяет, куда отправить сообщение. Формат вывода также может фильтровать на основе приоритета, например, он может быть полезен для вывода всех сообщений в файл отладки.

Формат вывода может быть:

- x:stderr – вывод в STDERR;
- x:syslog:name – использовать системный журнал для вывода и использовать данное имя в качестве идентификатора;
- x:file:file_path – вывод в файл, с соответствующим filepath;
- x:journal – вывод в systemd журнал.

Пример:

```
log_outputs="3:syslog:libvirtd 1:file:/tmp/libvirt.log"
```

Журналы работы ВМ хранятся в каталоге `/var/log/libvirt/qemu/`. Например, для машины alt-server журнал будет находиться по адресу: `/var/log/libvirt/qemu/alt-server.log`.

5.13 Управление доступом в виртуальной инфраструктуре

Права пользователя могут управляться с помощью правил polkit.

В каталоге `/usr/share/polkit-1/actions/` имеются два файла с описанием возможных действий для работы с ВМ, предоставленные разработчиками libvirt:

- файл org.libvirt.unix.policy описывает мониторинг ВМ и управление ими;
- в файле org.libvirt.api.policy перечислены конкретные действия (остановка, перезапуск и т. д.), которые возможны, если предыдущая проверка пройдена.

Перечисление конкретных свойств с комментариями доступно в файле `/usr/share/polkit-1/actions/org.libvirt.api.policy`.

В libvirt названия объектов и разрешений отображаются в имена polkit действий по схеме:

```
org.libvirt.api.$объект.$разрешение
```

Например, разрешение search-storage-vols на объекте storage_pool отображено к действию polkit:

```
org.libvirt.api.storage-pool.search-storage-vols
```

Чтобы определить правила авторизации, polkit должен однозначно определить объект. Libvirt предоставляет ряд атрибутов для определения объектов при выполнении проверки прав доступа. Набор атрибутов изменяется в зависимости от типа объекта.

Например, необходимо разрешить пользователю test (должен быть в группе vmusers) действия только с доменом alt-server. Для этого необходимо выполнить следующие действия:

1) раскомментировать в файле `/etc/libvirt/libvirtd.conf` строку:

```
access_drivers = [ "polkit" ]
```

2) перезапустить libvirt:

```
# systemctl restart libvirtd
```

3) создать файл `/etc/polkit-1/rules.d/100-libvirt-acl.rules` (имя произвольно) следующего вида:

```
polkit.addRule(function(action, subject) {
   if (action.id == "org.libvirt.unix.manage" &&
      subject.user == "test") {
      return polkit.Result.YES;
   }
```

```
});
 polkit.addRule(function(action, subject) {
   // разрешить пользователю test действия с доменом "alt-server"
   if (action.id.indexOf("org.libvirt.api.domain.") == 0  &&
       subject.user == "test") {
         if (action.lookup("domain_name") == 'alt-server') {
           return polkit.Result.YES;
         }
         else { return polkit.Result.NO; }
   }
   else {
   // разрешить пользователю test действия с
   //подключениями, хранилищем и прочим
if (action.id.indexOf("org.libvirt.api.") == 0 &&
       subject.user == "test") {
       polkit.log("org.libvirt.api.Yes");
           return polkit.Result.YES;
         }
         else { return polkit.Result.NO; }
   }})
================================
```

4) перелогиниться;
5) убедиться, что пользователю test доступна только машина alt-server, выполнив команду (от пользователя test):

```
$ virsh --connect qemu:///system list --all
 ID   Имя          Состояние
------------------------------
 4    alt-server   работает
```

Права можно настраивать более тонко, например, разрешив пользователю test запускать ВМ, но запретить ему все остальные действия с ней, для этого надо разрешить действие `org.libvirt.api.domain.start`:

```
polkit.addRule(function(action, subject) {
   // разрешить пользователю test только запускать ВМ в
   // домене "alt-server"
   if (action.id. == "org.libvirt.api.domain.start") &&
       subject.user == "test") {
         if (action.lookup("domain_name") == 'alt-server') {
           return polkit.Result.YES;
         }
         else { return polkit.Result.NO; }
}
```

```
   });
```

Предоставить право запускать ВМ только пользователям группы wheel:

```
if (action.id == "org.libvirt.api.domain.start") {
        if (subject.isInGroup("wheel")) {
            return polkit.Result.YES;
        } else {
            return polkit.Result.NO;
        }
    };
```

Предоставить право останавливать ВМ только пользователям группы wheel:

```
if (action.id == "org.libvirt.api.domain.stop") {
        if (subject.isInGroup("wheel")) {
            return polkit.Result.YES;
        } else {
            return polkit.Result.NO;
        }
    };
```

Можно также вести файл журнала, используя правила polkit. Например, делать запись в журнал при старте ВМ:

```
if (action.id.match("org.libvirt.api.domain.start") ) {
           polkit.log("action=" + action);
           polkit.log("subject=" + subject);
           return polkit.Result.YES;
         }
```

Запись в журнал при останове ВМ:

```
if (action.id.match("org.libvirt.api.domain.stop") ) {
           polkit.log("action=" + action);
           polkit.log("subject=" + subject);
           return polkit.Result.YES;
         }
```

6 KUBERNETES

6.1 Краткое описание возможностей

Kubernetes – это система для автоматизации развёртывания, масштабирования и управления контейнеризированными приложениями. Поддерживает основные технологии контейнеризации (Docker, Rocket) и аппаратную виртуализацию.

Основные задачи Kubernetes:

- развертывание контейнеров и все операции для запуска необходимой конфигурации (перезапуск остановившихся контейнеров, перемещение контейнеров для выделения ресурсов на новые контейнеры и т.д.);
- масштабирование и запуск нескольких контейнеров одновременно на большом количестве хостов;
- балансировка множества контейнеров в процессе запуска. Для этого Kubernetes использует API, задача которого заключается в логическом группировании контейнеров.

Утилиты для создания и управления кластером Kubernetes:

- `kubectl` – создание и настройка объектов в кластере;
- `kubelet` – запуск контейнеров на узлах;
- `kubeadm` – настройка компонентов, составляющих кластер.

6.2 Установка и настройка Kubernetes

Для создания управляющего или вычислительного узла, при установке дистрибутива в группе «Контейнеры» следует соответственно отметить пункт «Сервисы Kubernetes для управляющего хоста» или «Сервисы Kubernetes для вычислительного хоста» (Рис. 382).

Примечание. На этапе «Подготовка диска» рекомендуется выбрать «Server KVM/Docker/LXD/Podman/CRI-O (large /var/lib/)» и не создавать раздел Swap.

Примечание. В данном руководстве рассмотрен процесс разворачивания кластера с использованием CRI-O.

Установка Kubernetes при установке системы

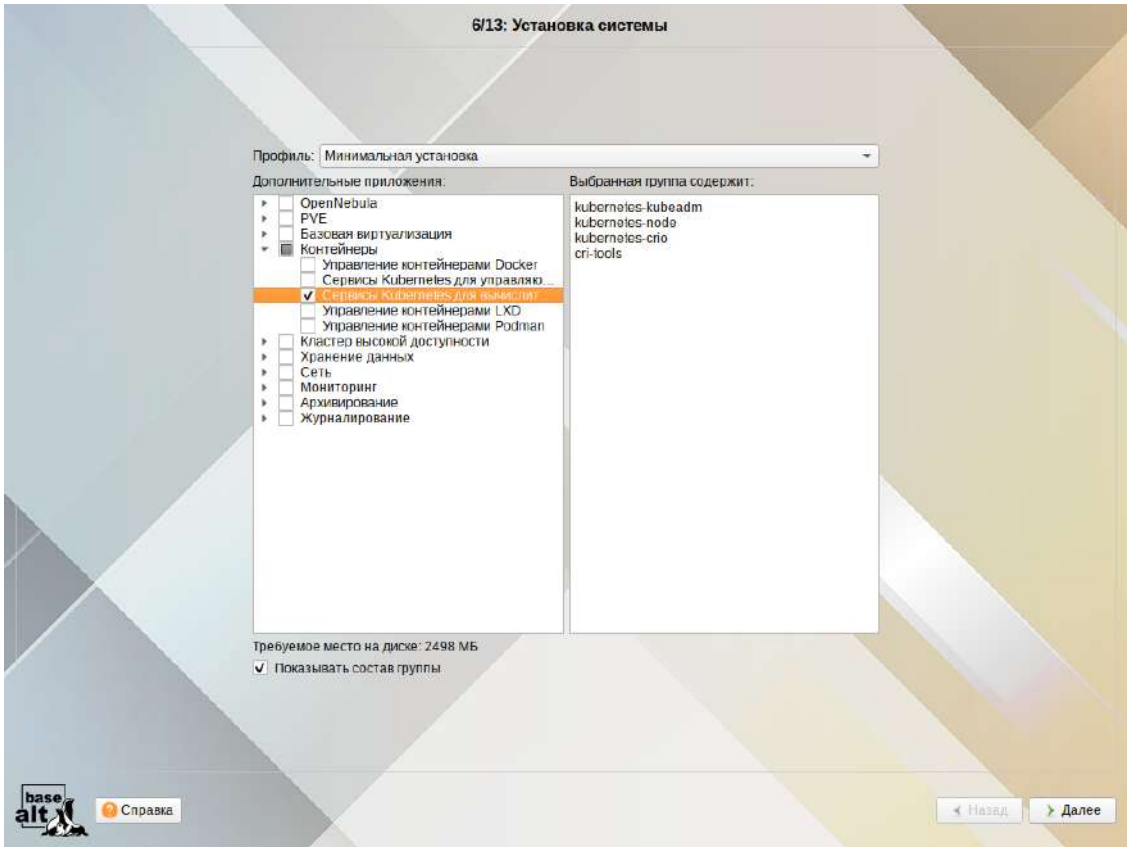


Рис. 382

6.2.1 Создание кластера Kubernetes

Для создания кластера необходимо несколько машин (nodes), одна из которых будет мастером. Системные требования:

- 2 ГБ или больше ОЗУ на машину;
- 2 ядра процессора или больше;
- все машины должны быть доступны по сети друг для друга;
- все машины должны успешно разрешать имена hostname друг друга (через DNS или hosts);
- Swap должен быть выключен.

Примечание. Для отключения Swap нужно выполнить команду:

```
# swapoff -a
```

и удалить соответствующую строку в `/etc/fstab`.

6.2.1.1 Инициализация кластера

Для инициализации кластера запустить одну из двух следующих команд (на мастере):

- для настройки сети с использованием Flannel:

```
# kubeadm init --pod-network-cidr=10.244.0.0/16
```

- для настройки сети с использованием Calico:

```
# kubeadm init --pod-network-cidr=10.168.0.0/16
```

где:

- `--pod-network-cidr=10.244.0.0/16` – адрес внутренней (разворачиваемой Kubernetes) сети, рекомендуется оставить данное значение для правильной работы Flannel;
- `--pod-network-cidr=192.168.0.0/16` – адрес внутренней (разворачиваемой Kubernetes) сети, рекомендуется оставить данное значение для правильной работы Calico.

Если все сделано правильно, на экране отобразится команда, позволяющая присоединить остальные ноды кластера к мастеру:

```
…
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as
root:

kubeadm join 192.168.0.103:6443 --token vr1hyp.anh6jecr9m1tskja \
      --discovery-token-ca-cert-hash  \
      sha256:8914081137bae4e13c741066a6b4394b68f62ab915735c4c4c92fc14b02fa5a3
```

Настроить kubernetes для работы от пользователя (на мастер-ноде):

1) создать каталог `~/.kube` (с правами администратора):

```
$ mkdir ~/.kube
```

2) скопировать конфигурацию (с правами администратора):

```
# cp /etc/kubernetes/admin.conf /home/<пользователь>/.kube/config
```

3) изменить владельца конфигурационного файла (с правами администратора):

```
# chown <пользователь>: /home/<пользователь>/.kube/config
```

6.2.1.2 Настройка сети

Развернуть сеть (Container Network Interface), запустив один из двух наборов команд (на мастер-ноде):

- для Flannel:

```
$ kubectl apply -f
https://gitea.basealt.ru/alt/flannel-manifests/raw/branch/main/p10/latest/kube-flannel.yml
```

- для Calico:
  - перейти в каталог /etc/cni/net.d/:

```
# cd /etc/cni/net.d/
```

  - создать файл 100-crio-bridge.conflist:

```
# cp 100-crio-bridge.conflist.sample 100-crio-bridge.conflist
```

  - запустить POD'ы из calico-манифестов:

```
$ kubectl create -f
https://raw.githubusercontent.com/projectcalico/calico/refs/tags/v3.25.0/manifests/tigera-operator.yaml

$ kubectl apply -f
https://raw.githubusercontent.com/projectcalico/calico/refs/tags/v3.25.0/manifests/custom-resources.yaml
```

В выводе будут отображены имена всех созданных ресурсов.

Проверить, что всё работает:

```
$ kubectl get pods --namespace kube-system

NAME                              READY   STATUS    RESTARTS   AGE
coredns-5dd5756b68-4t4tb          1/1     Running   0          10m
coredns-5dd5756b68-5cqjz          1/1     Running   0          10m
etcd-kube01                       1/1     Running   0          10m
kube-apiserver-kube01             1/1     Running   0          10m
kube-controller-manager-kube01    1/1     Running   0          10m
kube-proxy-2ncd6                  1/1     Running   0          10m
kube-scheduler-kube01             1/1     Running   0          10m
```

coredns должны находиться в состоянии Running. Количество kube-flannel и kube-proxy зависит от общего числа нод.

6.2.1.3 Добавление узлов (нод) в кластер

Подключить остальные узлы (ноды) в кластер. Для этого на узле выполнить команду:

```
# kubeadm join <ip адрес>:<порт> --token <токен> \
    --discovery-token-ca-cert-hash sha256:<хеш> \
    --ignore-preflight-errors=SystemVerification
```

Данная команда была выведена при выполнении команды `kubeadm init` на мастер-ноде. В данном случае:

```
# kubeadm join 192.168.0.103:6443 --token vr1hyp.anh6jecr9m1tskja \
    --discovery-token-ca-cert-hash \
    sha256:8914081137bae4e13c741066a6b4394b68f62ab915735c4c4c92fc14b02fa5a3

[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm
kubeadm-config -o yaml'
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file
"/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

Примечание. Получить токен, если его нет, можно выполнив команду (на мастер-ноде):

```
$ kubeadm token list

TOKEN                     TTL      EXPIRES                USAGES
vr1hyp.anh6jecr9m1tskja   23h      2024-10-29T07:45:18Z   authentication,signing
```

По умолчанию срок действия токена – 24 часа. Если требуется добавить новый узел в кластер по окончанию этого периода, можно создать новый токен:

```
$ kubeadm token create
```

Если значение параметра `--discovery-token-ca-cert-hash` неизвестно, его можно получить, выполнив команду (на мастер-ноде):

```
$ openssl x509 -pubkey -in /etc/kubernetes/pki/ca.crt | \
   openssl rsa -pubin -outform der 2>/dev/null | \
   openssl dgst -sha256 -hex | sed 's/^.* //'

8914081137bae4e13c741066a6b4394b68f62ab915735c4c4c92fc14b02fa5a3
```

Для ввода IPv6-адреса в параметр `<control-plane-host>:<control-plane-port>`, адрес должен быть заключен в квадратные скобки:

```
[fd00::101]:2073
```

Проверить наличие нод (на мастер-ноде):

```
$ kubectl get nodes
```

```
NAME       STATUS   ROLES                  AGE    VERSION
kube01     Ready    control-plane,master   42m     v1.28.14
kube02     Ready    <none>                 2m43s   v1.28.14
kube03     Ready    <none>                 24s     v1.28.14
```

или:

```
$ kubectl get nodes -o wide
```

Информация о кластере:

```
$ kubectl cluster-info
```

```
Kubernetes control plane is running at https://192.168.0.103:6443
KubeDNS is running at
https://192.168.0.103:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.
```

Посмотреть подробную информацию о ноде:

```
$ kubectl describe node node03
```

6.2.2 Тестовый запуск nginx

Deployment – это объект Kubernetes, представляющий работающее приложение в кластере.

Создать Deployment с nginx:

```
$ kubectl apply -f https://k8s.io/examples/application/deployment.yaml
```

```
deployment.apps/nginx-deployment created
```

Список подов:

```
$ kubectl get pods
```

```
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-66b6c48dd5-89lq4   1/1     Running   1          9s
nginx-deployment-66b6c48dd5-bt7zp   1/1     Running   1          9s
```

Создать сервис, с помощью которого можно получить доступ к приложению из внешней сети. Для этого создать файл `nginx-service.yaml`, со следующим содержимым:

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  type: NodePort
  ports:
```

```
  - port: 80
    targetPort: 80
  selector:
    app: nginx
```

Запустить новый сервис:

```
$ kubectl apply -f nginx-service.yaml

service/nginx created
```

Просмотреть порт сервиса nginx:

```
$ kubectl get svc nginx

NAME    TYPE       CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
nginx   NodePort   10.98.167.146   <none>        80:31868/TCP   17s
```

Проверить работу nginx, выполнив команду (сервер должен вернуть код 200):

```
$ curl -I <ip адрес>:<порт>
```

где <ip адрес> – это адрес любой из нод (не мастер-ноды), а <порт> – это порт сервиса, полученный с помощью предыдущей команды. В данном кластере возможна команда:

```
$ curl -I 192.168.0.102:31868

HTTP/1.1 200 OK
Server: nginx/1.14.2
```

6.3 Кластер высокой доступности Kubernetes

Kubernetes предлагает два основных способа реализации HA-кластера:

- со стековой топологией (узлы etcd размещаются вместе с узлами плоскости управления);
- с внешней etcd-топологией (etcd работает на узлах, отдельных от плоскости управления).

6.3.1 Стековая (составная) топология etcd

Стековая (составная) топология etcd – это топология, в которой распределенный кластер хранения данных, предоставляемый etcd, размещается на вершине кластера, образованного узлами, управляемыми kubeadm, которые запускают компоненты плоскости управления.

Каждый узел плоскости управления запускает экземпляр kube-apiserver, kube-scheduler и kube-controller-manager. Kube-apiserver предоставляется рабочим узлам с помощью балансировщика нагрузки.

Каждый узел уровня управления создает локальный etcd, и этот etcd взаимодействует только с kube-apiserver этого узла. То же самое относится к локальным экземплярам kube-controller-manager и kube-scheduler.

Эту топологию проще настроить, чем кластер с внешними узлами etcd, и проще управлять репликацией. Но если один узел выходит из строя, будут потеряны и etcd, и экземпляр уровня управления, и избыточность нарушится. Этот риск можно снизить, добавив больше узлов

плоскости управления. Поэтому для HA-кластера следует запустить как минимум три сгруппированных узла плоскости управления.

Это топология используется по умолчанию в kubeadm. Локальный etcd создается автоматически на узлах плоскости управления при использовании kubeadm init и kubeadm join --control-plane.

6.3.2 Внешняя etcd-топология

Внешняя etcd-топология – это топология, в которой кластер распределенного хранения данных, предоставляемый etcd, является внешним по отношению к кластеру, сформированному узлами, на которых выполняются компоненты плоскости управления.

Каждый узел плоскости управления во внешней топологии etcd запускает экземпляр kube-apiserver, kube-scheduler и kube-controller-manager. И kube-apiserver предоставляется рабочим узлам с помощью балансировщика нагрузки. Однако etcd работают на отдельных хостах, и каждый хост etcd взаимодействует с kube-apiserver каждого узла плоскости управления.

Эта топология разделяет плоскость управления и элемент etcd. Таким образом обеспечивается настройка HA, при которой потеря экземпляра уровня управления или etcd оказывает меньшее влияние и не влияет на избыточность кластера в такой степени, как многослойная топология.

Но для этой топологии требуется вдвое больше хостов, чем для многослойной топологии. Для HA-кластера с этой топологией требуется как минимум три хоста для узлов плоскости управления и три хоста для узлов etcd.

6.3.3 Создание HA-кластера с помощью kubeadm

Рекомендации:

- три или более управляющих узла;
- три или более вычислительных узла;
- все узлы должны быть доступны по сети друг для друга;
- на всех узлах должны быть установлены kubeadm, kubelet и среда выполнения контейнера;
- каждый узел должен иметь доступ к реестру образов контейнера Kubernetes (k8s.gcr.io);
- возможность доступа по ssh с одного узла ко всем узлам в системе.

Для создания HA-кластера etcd к вышеперечисленным требованиям дополнительно требуется три или более узла, которые станут членами кластера etcd.

Примечание. В данных примерах рассмотрена настройка сети с использованием Flannel.

6.3.3.1 Настройка HA-кластера etcd с помощью kubeadm

На первом управляющем узле необходимо выполнить следующие действия:

1) инициализировать кластер, выполнив команду:

```
# kubeadm init --control-plane-endpoint 192.168.0.201:6443 --upload-certs \
--pod-network-cidr=10.244.0.0/16
```

где:

- `--control-plane-endpoint` – указывает адрес и порт балансировщика нагрузки;
- `--upload-certs` – используется для загрузки в кластер сертификатов, которые должны быть общими для всех управляющих узлов;
- `--pod-network-cidr=10.244.0.0/16` – адрес внутренней (разворачиваемой Kubernetes) сети, рекомендуется оставить данное значение для правильной работы Flannel.

Если все сделано правильно, на экране отобразится команда, позволяющая присоединить остальные узлы кластера к управляющему узлу:

```
…
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

mkdir -p $HOME/.kube
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
https://kubernetes.io/docs/concepts/cluster-administration/addons/

You can now join any number of the control-plane node running the following command
on each as root:

kubeadm join 192.168.0.201:6443 --token cvvui8.lz82ufip6cz89ar9 \
      --discovery-token-ca-cert-hash
sha256:3ee0c550746a4a8e0abb6b59311f0fc301cdfeec00af8b26ed4598116c4d8184 \
      --control-plane --certificate-key
e0cbf1dc4e282bf517e23887dace30b411cd739b1aab037b056f0c23e5b0a222
```

```
Please note that the certificate-key gives access to cluster sensitive data, keep it
secret!
As a safeguard, uploaded-certs will be deleted in two hours; If necessary, you can
use
"kubeadm init phase upload-certs --upload-certs" to reload certs afterward.

Then you can join any number of worker nodes by running the following on each as
root:

kubeadm join 192.168.0.201:6443 --token cvvui8.lz82ufip6cz89ar9 \
      --discovery-token-ca-cert-hash
sha256:3ee0c550746a4a8e0abb6b59311f0fc301cdfeec00af8b26ed4598116c4d8184
```

2) настроить kubernetes для работы от пользователя:

- создать каталог ~/.kube (с правами пользователя):

```
$ mkdir ~/.kube
```

- скопировать конфигурацию (с правами администратора):

```
# cp /etc/kubernetes/admin.conf /home/<пользователь>/.kube/config
```

- изменить владельца конфигурационного файла (с правами администратора):

```
# chown <пользователь>: /home/<пользователь>/.kube/config
```

3) развернуть сеть (CNI):

```
$ kubectl apply -f
https://gitea.basealt.ru/alt/flannel-manifests/raw/branch/main/p10/latest/kube-
flannel.yml
```

4) проверить, что всё работает:

```
$ kubectl get pod -n kube-system –w

NAME                                READY   STATUS              RESTARTS   AGE
coredns-78fcd69978-c5swn            1/1     Running             0          11m
coredns-78fcd69978-zdbp8            1/1     Running             0          11m
etcd-master01                       1/1     Running             0          11m
kube-apiserver-master01             1/1     Running             0          11m
kube-controller-manager-master01    1/1     Running             0          11m
kube-flannel-ds-qfzbw               1/1     Running             0          116s
kube-proxy-r6kj9                    1/1     Running             0          11m
kube-scheduler-master01             1/1     Running             0          11m
```

На остальных управляющих узлах выполнить команду подключения узла к кластеру (данная команда была выведена при выполнении команды `kubeadm init` на первом управляющем узле):

```
# kubeadm join 192.168.0.201:6443 --token cvvui8.lz82ufip6cz89ar9 \
--discovery-token-ca-cert-hash
sha256:3ee0c550746a4a8e0abb6b59311f0fc301cdfeec00af8b26ed4598116c4d8184 \
```

```
--control-plane --certificate-key
e0cbf1dc4e282bf517e23887dace30b411cd739b1aab037b056f0c23e5b0a222
```

Подключить вычислительные узлы к кластеру (данная команда была выведена при выполнении команды `kubeadm init` на первом управляющем узле):

```
# kubeadm join 192.168.0.201:6443 --token cvvui8.lz82ufip6cz89ar9 \
--discovery-token-ca-cert-hash
sha256:3ee0c550746a4a8e0abb6b59311f0fc301cdfeec00af8b26ed4598116c4d8184
```

Проверить наличие нод (на управляющем узле):

```
$ kubectl get nodes

NAME       STATUS   ROLES                  AGE     VERSION
kube01     Ready    <none>                 23m     v1.28.14
kube02     Ready    <none>                 15m     v1.28.14
kube03     Ready    <none>                 2m30s   v1.28.14
master01   Ready    control-plane,master   82m     v1.28.14
master02   Ready    control-plane,master   66m     v1.28.14
master03   Ready    control-plane,master   39m     v1.28.14
```

6.3.3.2 Настройка HA-кластера etcd с помощью kubeadm

Настройка HA-кластера с внешней etcd-топологией аналогична процедуре, используемой для стековой топологии etcd, за исключением того, что предварительно необходимо настроить etcd и передать информацию etcd в конфигурационный файл kubeadm.

В данном примере рассматривается процесс создания HA-кластера etcd состоящего из трех узлов. Узлы должны иметь возможность общаться друг с другом через порты 2379 и 2380.

Основная идея при таком способе настройки кластера, состоит в том, чтобы генерировать все сертификаты на одном узле и распространять только необходимые файлы на другие узлы.

Примечание. kubeadm содержит все необходимое криптографические механизмы для создания сертификатов, никаких других криптографических инструментов для данного примера не требуется.

Настройка etcd кластера:

1) настроить kubelet в качестве диспетчера служб для etcd. Для этого на всех etcd узлах нужно добавить новый файл конфигурации systemd для модуля kubelet с более высоким приоритетом, чем предоставленный kubeadm файл модуля kubelet:

```
# cat << EOF > /etc/systemd/system/kubelet.service.d/kubelet.conf
# Replace "systemd" with the cgroup driver of your container runtime. The default
value in the kubelet is "cgroupfs".
# Replace the value of "containerRuntimeEndpoint" for a different container runtime
if needed.
#
```

```
apiVersion: kubelet.config.k8s.io/v1beta1
kind: KubeletConfiguration
authentication:
  anonymous:
    enabled: false
  webhook:
    enabled: false
authorization:
  mode: AlwaysAllow
cgroupDriver: systemd
address: 127.0.0.1
containerRuntimeEndpoint: unix:///var/run/crio/crio.sock
staticPodPath: /etc/kubernetes/manifests
EOF

# cat << EOF > /etc/systemd/system/kubelet.service.d/20-etcd-service-manager.conf
[Service]
ExecStart=
ExecStart=/usr/bin/kubelet
--config=/etc/systemd/system/kubelet.service.d/kubelet.conf
Restart=always
EOF

# systemctl daemon-reload
# systemctl restart kubelet
```

Убедиться, что kubelet запущен:

```
# systemctl status kubelet
```

2) на первом узле etcd создать файлы конфигурации kubeadm для всех узлов etcd. Для этого создать и запустить скрипт:

```
#!/bin/sh
# HOST0, HOST1, и HOST2 - IP-адреса узлов
export HOST0=192.168.0.205
export HOST1=192.168.0.206
export HOST2=192.168.0.207

# NAME0, NAME1 и NAME2 - имена узлов
export NAME0="etc01"
export NAME1="etc02"
export NAME2="etc03"

# Создать временные каталоги
mkdir -p /tmp/${HOST0}/ /tmp/${HOST1}/ /tmp/${HOST2}/
```

```
HOSTS=(${HOST0} ${HOST1} ${HOST2})
NAMES=(${NAME0} ${NAME1} ${NAME2})

for i in "${!HOSTS[@]}"; do
HOST=${HOSTS[$i]}
NAME=${NAMES[$i]}
cat << EOF > /tmp/${HOST}/kubeadmcfg.yaml
---
apiVersion: "kubeadm.k8s.io/v1beta3"
kind: InitConfiguration
nodeRegistration:
    name: ${NAME}
localAPIEndpoint:
    advertiseAddress: ${HOST}
---
apiVersion: "kubeadm.k8s.io/v1beta3"
kind: ClusterConfiguration
etcd:
    local:
        serverCertSANs:
        - "${HOST}"
        peerCertSANs:
        - "${HOST}"
        extraArgs:
            initial-cluster: ${NAMES[0]}=https://${HOSTS[0]}:2380,${NAMES[1]}
=https://${HOSTS[1]}:2380,${NAMES[2]}=https://${HOSTS[2]}:2380
            initial-cluster-state: new
            name: ${NAME}
            listen-peer-urls: https://${HOST}:2380
            listen-client-urls: https://${HOST}:2379
            advertise-client-urls: https://${HOST}:2379
            initial-advertise-peer-urls: https://${HOST}:2380
EOF
done
```

3) создать центр сертификации (CA).

Примечание. Если у вас уже есть CA, то необходимо скопировать сертификат (crt) и ключ CA в /etc/kubernetes/pki/etcd/ca.crt и /etc/kubernetes/pki/etcd/ca.key. После этого можно перейти к следующему шагу.

Если у вас еще нет CA, следует на узле, где были сгенерированы файлы конфигурации kubeadm, запустить команду:

```
# kubeadm init phase certs etcd-ca

[certs] Generating "etcd/ca" certificate and key
```

Эта команда создаст два файла: /etc/kubernetes/pki/etcd/ca.crt и /etc/kubernetes/pki/etcd/ca.key.

4) сгенерировать сертификаты для всех etcd узлов. Для этого создать и запустить скрипт (на первом etcd узле):

```
#!/bin/sh
# HOST0, HOST1, и HOST2 - IP-адреса узлов
export HOST0=192.168.0.205
export HOST1=192.168.0.206
export HOST2=192.168.0.207
kubeadm init phase certs etcd-server --config=/tmp/${HOST2}/kubeadmcfg.yaml
kubeadm init phase certs etcd-peer --config=/tmp/${HOST2}/kubeadmcfg.yaml
kubeadm init phase certs etcd-healthcheck-client
--config=/tmp/${HOST2}/kubeadmcfg.yaml
kubeadm init phase certs apiserver-etcd-client --config=/tmp/${HOST2}/kubeadmcfg.yaml
cp -R /etc/kubernetes/pki /tmp/${HOST2}/
# cleanup non-reusable certificates
find /etc/kubernetes/pki -not -name ca.crt -not -name ca.key -type f -delete

kubeadm init phase certs etcd-server --config=/tmp/${HOST1}/kubeadmcfg.yaml
kubeadm init phase certs etcd-peer --config=/tmp/${HOST1}/kubeadmcfg.yaml
kubeadm init phase certs etcd-healthcheck-client
--config=/tmp/${HOST1}/kubeadmcfg.yaml
kubeadm init phase certs apiserver-etcd-client --config=/tmp/${HOST1}/kubeadmcfg.yaml
cp -R /etc/kubernetes/pki /tmp/${HOST1}/
find /etc/kubernetes/pki -not -name ca.crt -not -name ca.key -type f -delete

kubeadm init phase certs etcd-server --config=/tmp/${HOST0}/kubeadmcfg.yaml
kubeadm init phase certs etcd-peer --config=/tmp/${HOST0}/kubeadmcfg.yaml
kubeadm init phase certs etcd-healthcheck-client
--config=/tmp/${HOST0}/kubeadmcfg.yaml
kubeadm init phase certs apiserver-etcd-client --config=/tmp/${HOST0}/kubeadmcfg.yaml
# No need to move the certs because they are for HOST0

# clean up certs that should not be copied off this host
find /tmp/${HOST2} -name ca.key -type f -delete
find /tmp/${HOST1} -name ca.key -type f -delete
```

5) скопировать сертификаты и файлы конфигурации kubeadm на второй и третий узлы etcd:

```
HOST1=192.168.0.206
HOST2=192.168.0.207
```

```
USER=user
# scp -r /tmp/${HOST1}/* ${USER}@${HOST1}:
# ssh ${USER}@${HOST1}
$ su -
# chown -R root:root /home/user/pki
# mv /home/user/pki /etc/kubernetes/
# exit
$ exit
# scp -r /tmp/${HOST2}/* ${USER}@${HOST2}:
# ssh ${USER}@${HOST2}
$ su -
# chown -R root:root /home/user/pki
# mv /home/user/pki /etc/kubernetes/
# exit
$ exit
```

6) в итоге должны существовать следующие файлы:

- на первом узле etcd (там, где были сгенерированы файлы конфигурации для kubeadm и сертификаты):

```
/tmp/${HOST0}
└── kubeadmcfg.yaml
---
/etc/kubernetes/pki
├── apiserver-etcd-client.crt
├── apiserver-etcd-client.key
└── etcd
    ├── ca.crt
    ├── ca.key
    ├── healthcheck-client.crt
    ├── healthcheck-client.key
    ├── peer.crt
    ├── peer.key
    ├── server.crt
    └── server.key
```

- на втором узле etcd:

```
$HOME
└── kubeadmcfg.yaml
---
/etc/kubernetes/pki
├── apiserver-etcd-client.crt
├── apiserver-etcd-client.key
└── etcd
```

```
        ├── ca.crt
        ├── healthcheck-client.crt
        ├── healthcheck-client.key
        ├── peer.crt
        ├── peer.key
        ├── server.crt
        └── server.key
```

- на третьем узле etcd:

```
$HOME
└── kubeadmcfg.yaml
---
/etc/kubernetes/pki
├── apiserver-etcd-client.crt
├── apiserver-etcd-client.key
└── etcd
    ├── ca.crt
    ├── healthcheck-client.crt
    ├── healthcheck-client.key
    ├── peer.crt
    ├── peer.key
    ├── server.crt
    └── server.key
```

7) на каждом etcd узле запустить команду kubeadm, чтобы сгенерировать статический манифест для etcd:

- на первом узле etcd (там, где были сгенерированы файлы конфигурации kubeadm и сертификаты):

```
# kubeadm init phase etcd local --config=/tmp/192.168.0.205/kubeadmcfg.yaml

[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests
```

- на втором и третьем узлах etcd:

```
# kubeadm init phase etcd local --config=/home/user/kubeadmcfg.yaml

[etcd] Creating static Pod manifest for local etcd in "/etc/kubernetes/manifests"
```

Настроить первый управляющий узел кластера:

1) скопировать сертификаты и ключ с первого узла etcd на первый управляющий узел:

```
export CONTROL_PLANE="user@192.168.0.201"
# scp /etc/kubernetes/pki/etcd/ca.crt "${CONTROL_PLANE}":
# scp /etc/kubernetes/pki/apiserver-etcd-client.crt "${CONTROL_PLANE}":
# scp /etc/kubernetes/pki/apiserver-etcd-client.key "${CONTROL_PLANE}":
```

2) создать на первом управляющем узле файл kubeadm-config.yaml:

```
---
```

```
apiVersion: kubeadm.k8s.io/v1beta3
kind: ClusterConfiguration
kubernetesVersion: stable
networking:
    podSubnet: "10.244.0.0/16"
controlPlaneEndpoint: "192.168.0.201:6443" # IP-адрес, порт балансировщика нагрузки
etcd:
   external:
      endpoints:
          - https://192.168.0.205:2379 # IP-адрес ETCD01
          - https://192.168.0.206:2379 # IP-адрес ETCD02
          - https://192.168.0.207:2379 # IP-адрес ETCD03
     caFile: /etc/kubernetes/pki/etcd/ca.crt
     certFile: /etc/kubernetes/pki/apiserver-etcd-client.crt
     keyFile: /etc/kubernetes/pki/apiserver-etcd-client.key
```

3) переместить ранее скопированные сертификаты и ключ в соответствующий каталог на первом управляющем узле:

```
# mkdir -p /etc/kubernetes/pki/etcd/
# cp /home/user/ca.crt /etc/kubernetes/pki/etcd/
# cp /home/user/apiserver-etcd-client.* /etc/kubernetes/pki/
```

4) создать первый управляющий узел:

```
# kubeadm init --config kubeadm-config.yaml --upload-certs
…
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

You can now join any number of the control-plane node running the following command
on each as root:
```

```
  kubeadm join 192.168.0.201:6443 --token 7onha1.afzqd41s8dzr1wj1 \
      --discovery-token-ca-cert-hash
sha256:ec2be69db54b2ae13c175765ddd058801fd70054508c0e118020896a1d4c9ec3 \
      --control-plane --certificate-key
eb1fabf70e994c061f749f13c0f26baef64764e813d5f0eaa7b09d5279a492c4

Please note that the certificate-key gives access to cluster sensitive data, keep it
secret!
As a safeguard, uploaded-certs will be deleted in two hours; If necessary, you can
use
"kubeadm init phase upload-certs --upload-certs" to reload certs afterward.

Then you can join any number of worker nodes by running the following on each as
root:

kubeadm join 192.168.0.201:6443 --token 7onha1.afzqd41s8dzr1wj1 \
      --discovery-token-ca-cert-hash
sha256:ec2be69db54b2ae13c175765ddd058801fd70054508c0e118020896a1d4c9ec3
```

Следует сохранить этот вывод, т.к. этот токен будет использоваться для присоединения к кластеру остальных управляющих и вычислительных узлов.

5) настроить kubernetes для работы от пользователя:

- создать каталог ~/.kube (с правами пользователя):

```
$ mkdir ~/.kube
```

- скопировать конфигурацию (с правами администратора):

```
# cp /etc/kubernetes/admin.conf /home/<пользователь>/.kube/config
```

- изменить владельца конфигурационного файла (с правами администратора):

```
# chown <пользователь>: /home/<пользователь>/.kube/config
```

6) развернуть сеть (CNI):

```
$ kubectl apply -f
https://gitea.basealt.ru/alt/flannel-manifests/raw/branch/main/p10/latest/kube-
flannel.yml
```

7) проверить, что всё работает:

```
$ kubectl get pod -n kube-system -w
```

На остальных управляющих узлах выполнить команду подключения узла к кластеру (данная команда была выведена при выполнении команды kubeadm init на первом управляющем узле):

```
# kubeadm join 192.168.0.201:6443 --token 7onha1.afzqd41s8dzr1wj1 \
      --discovery-token-ca-cert-hash
sha256:ec2be69db54b2ae13c175765ddd058801fd70054508c0e118020896a1d4c9ec3 \
```

```
      --control-plane --certificate-key
eb1fabf70e994c061f749f13c0f26baef64764e813d5f0eaa7b09d5279a492c4
```

Подключить вычислительные узлы к кластеру (данная команда была выведена при выполнении команды kubeadm init на первом управляющем узле):

```
# kubeadm join 192.168.0.201:6443 --token 7onha1.afzqd41s8dzr1wj1 \
      --discovery-token-ca-cert-hash
sha256:ec2be69db54b2ae13c175765ddd058801fd70054508c0e118020896a1d4c9ec3
```

Проверить наличие нод (на управляющем узле):

```
$ kubectl get nodes
```

7 НАСТРОЙКА СИСТЕМЫ

7.1 Центр управления системой

Для управления настройками установленной системы можно использовать Центр управления системой. Центр управления системой (ЦУС) представляет собой удобный интерфейс для выполнения наиболее востребованных административных задач: добавление и удаление пользователей, настройка сетевых подключений, просмотр информации о состоянии системы и т.п. ЦУС имеет веб-ориентированный интерфейс, позволяющий управлять сервером с любого компьютера сети.

Центр управления системой состоит из нескольких независимых диалогов-модулей. Каждый модуль отвечает за настройку определённой функции или свойства системы. Модули центра управления системой имеют справочную информацию.

7.1.1 Применение ЦУС

ЦУС можно использовать для разных целей, например:

- настройки даты и времени;
- управления системными службами;
- просмотра системных журналов;
- управления выключением удаленного компьютера;
- настройки ограничений выделяемых ресурсов памяти пользователям (квоты);
- настройки ограничений на использование внешних носителей;
- конфигурирования сетевых интерфейсов;
- настройки межсетевого экрана;
- изменения пароля администратора системы (root);
- создания, удаления и редактирования учётных записей пользователей.

7.1.2 Использование веб-ориентированного ЦУС

ЦУС имеет веб-ориентированный интерфейс, позволяющий управлять данным компьютером с любого другого компьютера сети.

Для запуска веб-ориентированного интерфейса должны быть запущены сервисы ahttpd и alteratord:

```
# systemctl enable --now ahttpd
# systemctl enable --now alteratord
```

Работа с ЦУС может происходить из любого веб-браузера. Для начала работы необходимо перейти по адресу https://ip-адрес:8080/**.**

Если для сервера задан IP-адрес 192.168.0.122, то интерфейс управления будет доступен по адресу: https://192.168.0.122:8080/.

Примечание. IP-адрес сервера можно узнать, введя на сервере команду:

```
$ ip addr
```

При запуске ЦУС необходимо ввести в соответствующие поля имя пользователя (root) и пароль пользователя root (Рис. 383).

Запуск веб-ориентированного центра управления системой

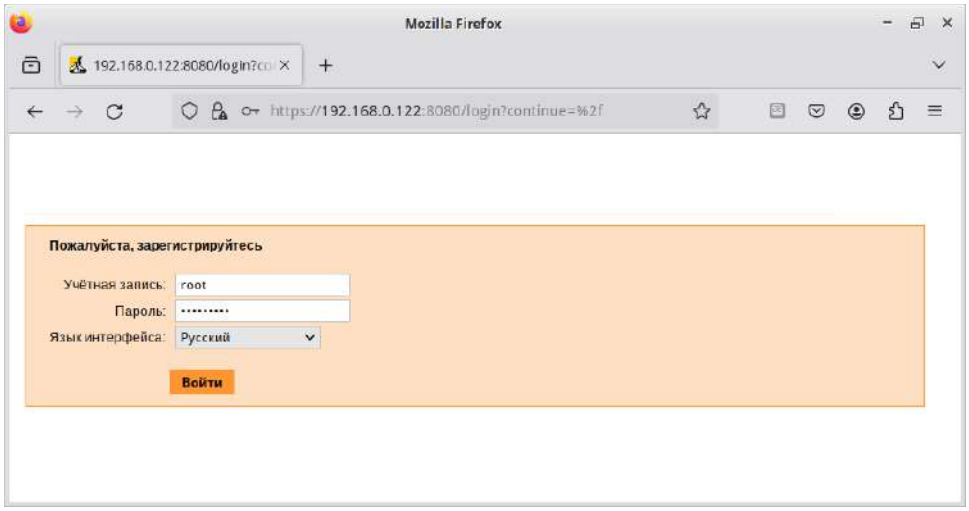


Рис. 383

После этого будут доступны все возможности ЦУС на той машине, к которой было произведено подключение через веб-интерфейс (Рис. 384).

Веб-ориентированный центр управления системой

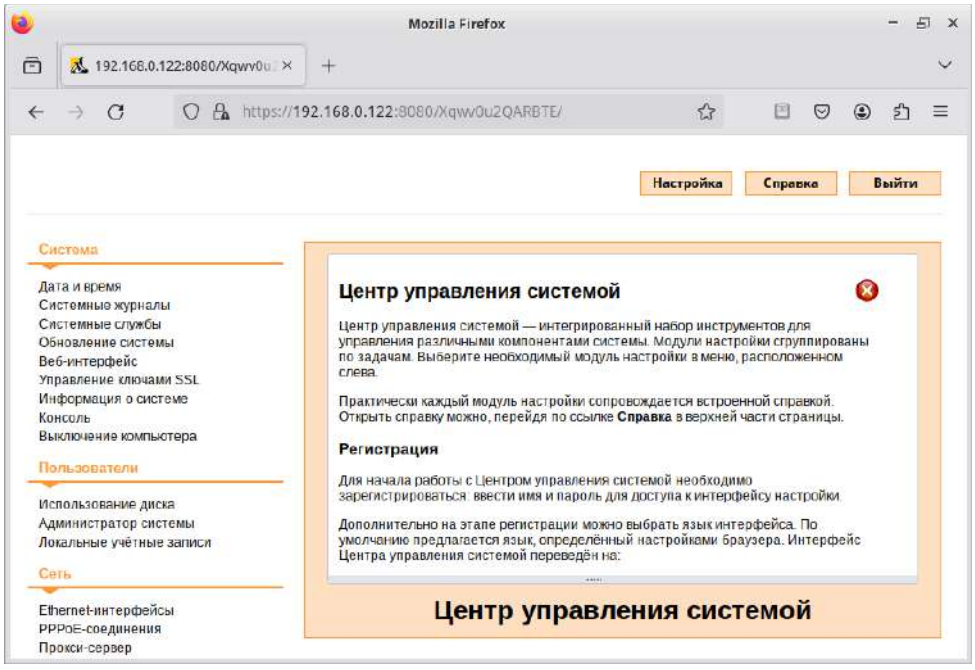


Рис. 384

Веб-интерфейс ЦУС можно настроить (кнопка «Режим эксперта»), выбрав один из режимов:

- основной режим;
- режим эксперта.

Выбор режима влияет на количество отображаемых модулей. В режиме эксперта отображаются все установленные модули, а в основном режиме только наиболее используемые.

ЦУС содержит справочную информацию по включённым в него модулям. Об использовании самого интерфейса системы управления можно прочитать, нажав на кнопку «Справка» на начальной странице центра управления системой (Рис. 384).

После работы с ЦУС, в целях безопасности, не следует оставлять открытым браузер. Необходимо обязательно выйти из сеанса работы с ЦУС, нажав на кнопку «Выйти».

Дальнейшие разделы описывают некоторые возможности использования ОС «Альт Сервер Виртуализации», настраиваемые в ЦУС.

7.2 Конфигурирование сетевых интерфейсов

Конфигурирование сетевых интерфейсов осуществляется в модуле ЦУС «Ethernet-интерфейсы» (пакет alterator-net-eth) из раздела раздел «Сеть» (Рис. 385).

Настройка модуля «Ethernet-интерфейсы»

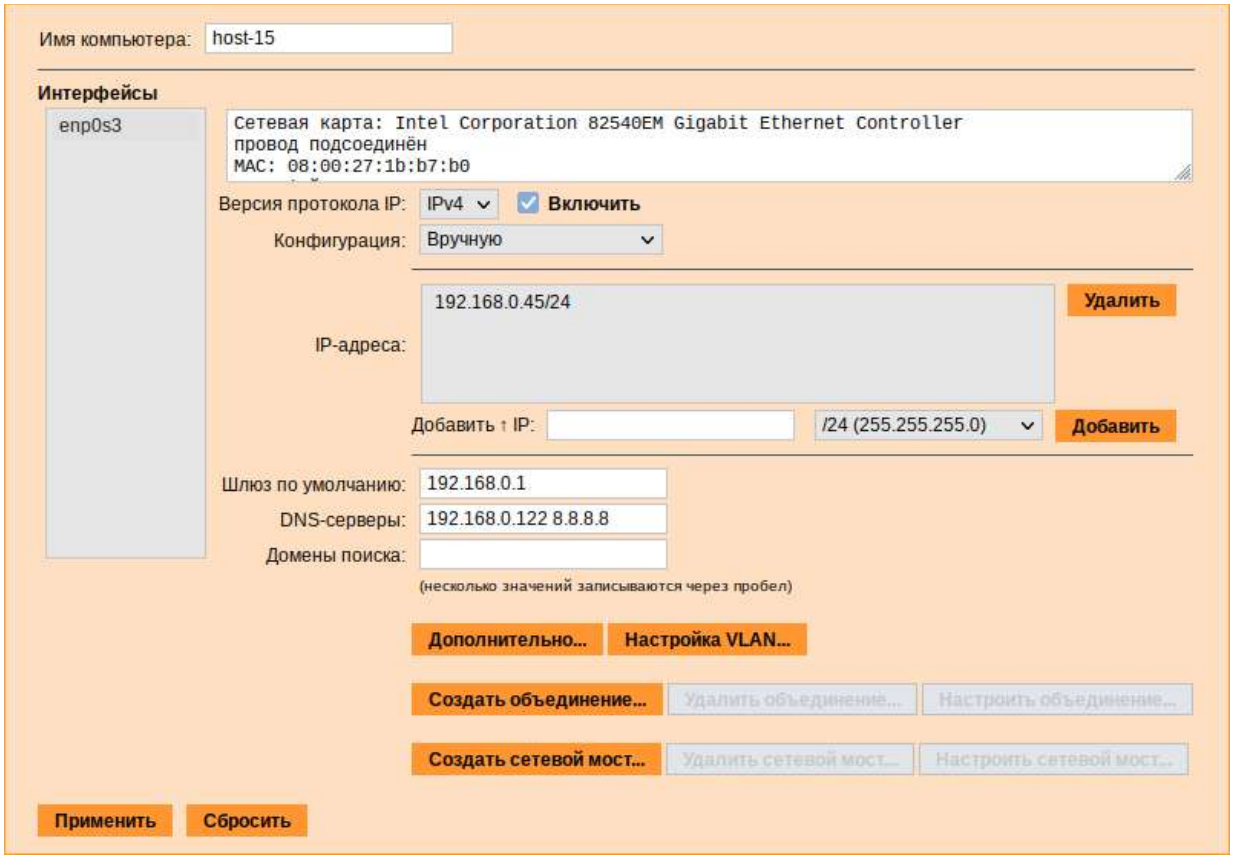


Рис. 385

В модуле «Ethernet-интерфейсы» можно заполнить следующие поля:

- «Имя компьютера» – указать сетевое имя ПЭВМ в поле для ввода имени компьютера (это общий сетевой параметр, не привязанный к какому либо конкретному интерфейсу). Имя компьютера, в отличие от традиционного имени хоста в Unix (hostname), не содержит названия сетевого домена;
- «Интерфейсы» – выбрать доступный сетевой интерфейс, для которого будут выполняться настройки;
- «Версия протокола IP» – указать в выпадающем списке версию используемого протокола IP (IPv4, IPv6) и убедиться, что пункт «Включить», обеспечивающий поддержку работы протокола, отмечен;
- «Конфигурация» – выбрать способ назначения IP-адресов (службы DHCP, Zeroconf, вручную);
- «IP-адреса» – пул назначенных IP-адресов из поля «IP», выбранные адреса можно удалить нажатием кнопки «Удалить»;
- «Добавить ↑ IP» – ввести IP-адрес вручную и выбрать в выпадающем поле предпочтительную маску сети, затем нажать кнопку «Добавить» для переноса адреса в пул поля «IP-адреса»;
- «Шлюз по умолчанию» – в поле для ввода необходимо ввести адрес шлюза, который будет использоваться сетью по умолчанию;
- «DNS-серверы» – в поле для ввода необходимо ввести список предпочтительных DNS-серверов, которые будут получать информацию о доменах, выполнять маршрутизацию почты и управлять обслуживающими узлами для протоколов в домене;
- «Домены поиска» – в поле для ввода необходимо ввести список предпочтительных доменов, по которым будет выполняться поиск.

«IP-адрес» и «Маска сети» – обязательные параметры каждого узла IP-сети. Первый параметр – уникальный идентификатор машины, от второго напрямую зависит, к каким машинам локальной сети данная машина будет иметь доступ. Если требуется выход во внешнюю сеть, то необходимо указать параметр «Шлюз по умолчанию».

В случае наличия DHCP-сервера можно все вышеперечисленные параметры получить автоматически – выбрав в списке «Конфигурация» пункт «Использовать DHCP» (Рис. 386).

Автоматическое получение настроек от DHCP-сервера

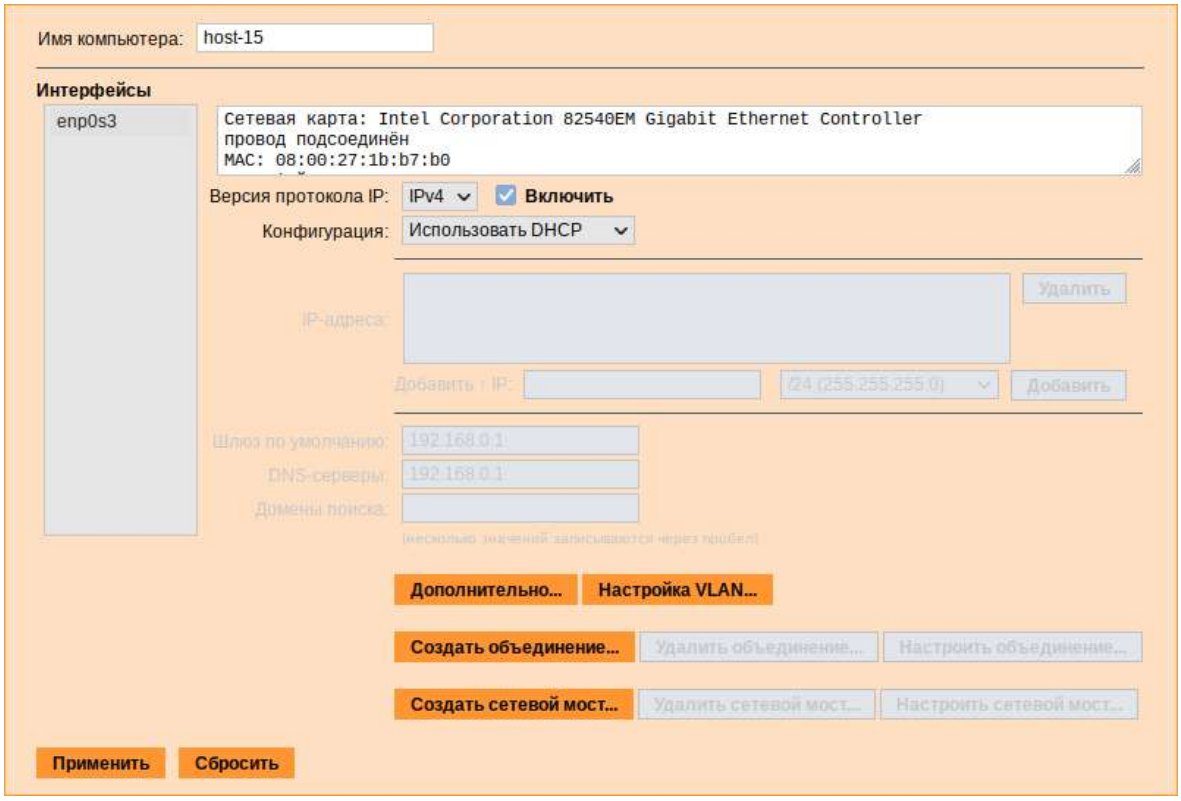


Рис. 386

Если в компьютере имеется несколько сетевых карт, то возможна ситуация, когда при очередной загрузке ядро присвоит имена интерфейсов (enp0s3, enp0s8) в другом порядке. В результате интерфейсы получат не свои настройки. Чтобы этого не происходило, можно привязать интерфейс к имени по его аппаратному адресу (MAC) или по местоположению на системной шине.

Дополнительно для каждого интерфейса можно настроить сетевую подсистему, а также указать должен ли запускаться данный интерфейс при загрузке системы (Рис. 387).

Выбор сетевой подсистемы

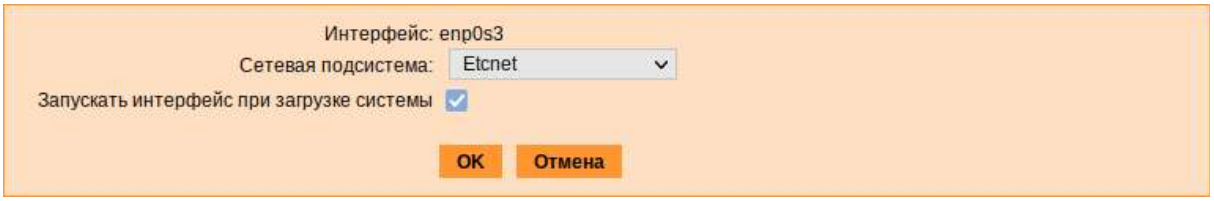


Рис. 387

В списке «Сетевая подсистема» можно выбрать следующие режимы:

- «Etcnet» – в этом режиме настройки берутся исключительно из файлов находящихся в каталоге настраиваемого интерфейса `/etc/net/ifaces/<интерфейс>`. Настройки сети могут изменяться либо в ЦУС в данном модуле, либо напрямую через редактирование файлов `/etc/net/ifaces/<интерфейс>`;

- «NetworkManager (etcnet)» – в этом режиме NetworkManager сам инициирует сеть, используя в качестве параметров – настройки из файлов Etcnet. Настройки сети могут изменяться либо в ЦУС в данном модуле, либо напрямую через редактирование файлов `/etc/net/ifaces/<интерфейс>`. В этом режиме можно просмотреть настройки сети, например, полученный по DHCP IP-адрес, через графический интерфейс NetworkManager;
- «NetworkManager (native)» – в данном режиме управление настройками интерфейса передаётся NetworkManager и не зависит от файлов Etcnet. Управлять настройками можно через графический интерфейс NetworkManager. Файлы с настройками находятся в каталоге `/etc/NetworkManager/system-connections`. Этот режим особенно актуален для задач настройки сети на клиенте, когда IP-адрес необходимо получать динамически с помощью DHCP, а DNS-сервер указать явно. Через ЦУС так настроить невозможно, так как при включении DHCP отключаются настройки, которые можно задавать вручную;
- «system-networkd» – в данном режиме управление настройками интерфейса передаётся службе `systemd-networkd`. Данный режим доступен, если установлен пакет `systemd-networkd`. Настройки сети могут изменяться либо в ЦУС в данном модуле (только настройки физического интерфейса), либо напрямую через редактирование файлов `/etc/systemd/network/<имя_файла>.network`, `/etc/systemd/network/<имя_файла>.netdev`, `/etc/systemd/network/<имя_файла>.link`;
- «Не контролируется» – в этом режиме интерфейс находится в состоянии DOWN (выключен).

7.2.1 Объединение сетевых интерфейсов

Модуль «Объединение интерфейсов» (пакет alterator-net-bond) позволяет объединить несколько физических сетевых интерфейсов в один логический. Это позволяет достичь отказоустойчивости, увеличения скорости и балансировки нагрузки.

Для создания объединения интерфейсов необходимо выполнить следующие действия:

1) нажать кнопку «Создать объединение…» (Рис. 388);

2) переместить сетевые интерфейсы, которые будут входить в объединение, из списка «Доступные интерфейсы» в список «Используемые интерфейсы» (Рис. 389);

3) в списке «Политика» выбрать режим объединения:

- «Round-robin» – режим циклического выбора активного интерфейса для исходящего трафика;
- «Активный-резервный» – активен только один интерфейс, остальные находятся в режиме горячей замены;

- «XOR» – один и тот же интерфейс работает с определённым получателем, передача пакетов распределяется между интерфейсами на основе формулы *((MAC-адрес источника) XOR (MAC-адрес получателя)) % число интерфейсов*;
- «Широковещательная» – трафик идёт через все интерфейсы одновременно;
- «Агрегирование каналов по стандарту IEEE 802.3ad» – в группу объединяются одинаковые по скорости и режиму интерфейсы, все физические интерфейсы используются одновременно в соответствии со спецификацией IEEE 802.3ad. Для реализации этого режима необходима поддержка на уровне драйверов сетевых карт и коммутатор, поддерживающий стандарт IEEE 802.3ad (коммутатор требует отдельной настройки);
- «Адаптивная балансировка нагрузки передачи» – исходящий трафик распределяется в соответствии с текущей нагрузкой (с учётом скорости) на интерфейсах (для данного режима необходима его поддержка в драйверах сетевых карт). Входящие пакеты принимаются только активным сетевым интерфейсом;
- «Адаптивная балансировка нагрузки» – включает в себя балансировку исходящего трафика и балансировку на приём (rlb) для IPv4 трафика и не требует применения специальных коммутаторов. Балансировка на приём достигается на уровне протокола ARP путём перехвата ARP ответов локальной системы и перезаписи физического адреса на адрес одного из сетевых интерфейсов (в зависимости от загрузки);

4) указать, если это необходимо, параметры объединения в поле «Параметры объединения»;

5) нажать кнопку «Назад»;

6) в результате будет создан агрегированный интерфейс bond0. Для данного интерфейса можно задать IP-адрес и, если необходимо, дополнительные параметры (Рис. 390);

7) нажать кнопку «Применить».

Объединение интерфейсов в веб-интерфейсе alterator-net-eth

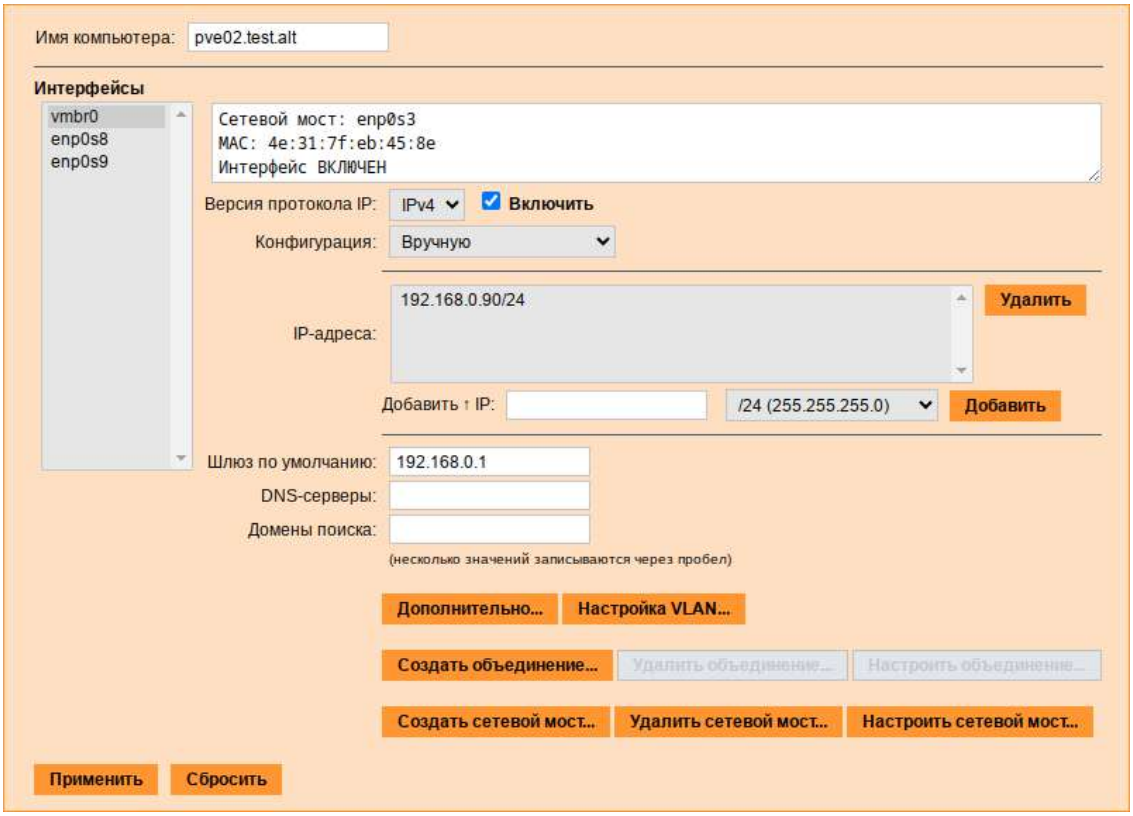


Рис. 388

Выбор сетевых интерфейсов для объединения

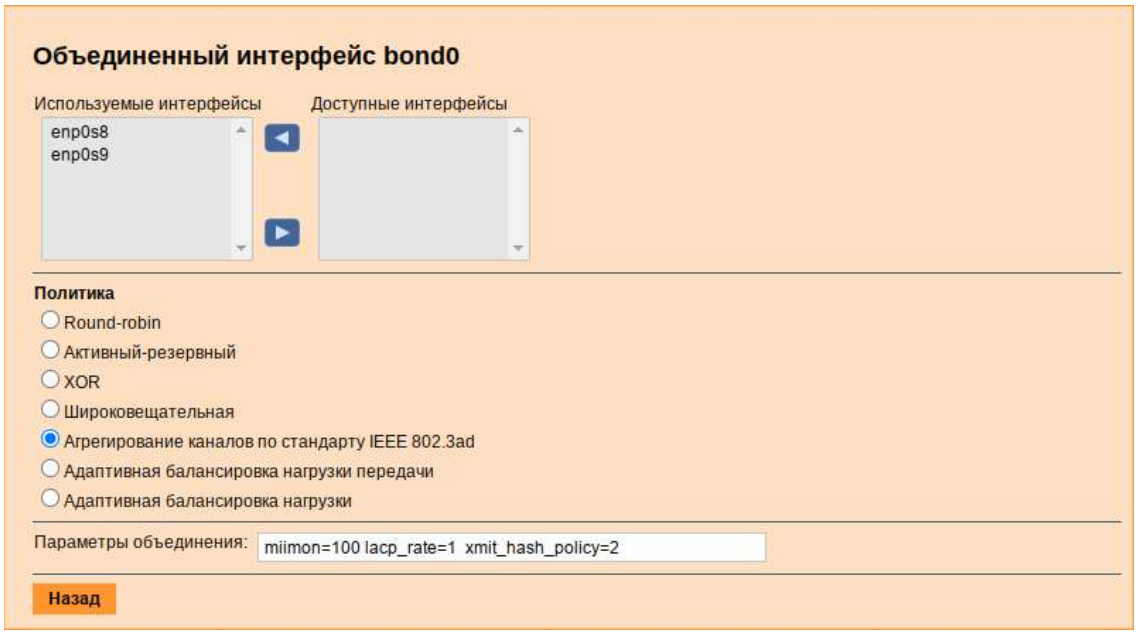


Рис. 389

Настройки интерфейса bond0

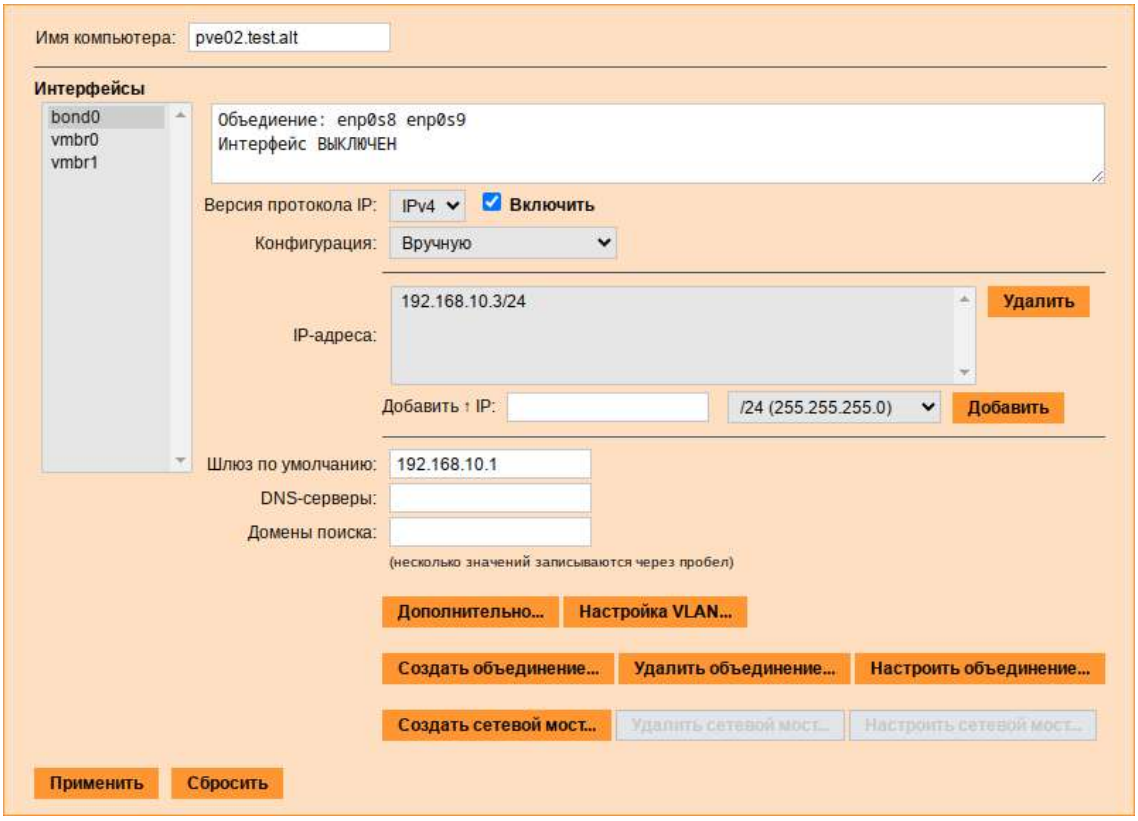


Рис. 390

Информацию о получившемся агрегированном интерфейсе можно посмотреть в `/proc/net/bonding/bond0`.

Для удаления агрегированного интерфейса необходимо выбрать его в списке «Интерфейсы» и нажать кнопку «Удалить объединение…».

7.2.2 Сетевые мосты

Модуль «Сетевые мосты» (пакет alterator-net-bridge) позволяет организовать виртуальный сетевой мост.

Примечание. Если интерфейсы, входящие в состав моста, являются единственными физически подключенными и настройка моста происходит с удалённого узла через эти интерфейсы, то требуется соблюдать осторожность, т.к. эти интерфейсы перестанут быть доступны.

Для создания Ethernet-моста необходимо выполнить следующие действия:

1) у интерфейсов, которые будут входить в мост, удалить IP-адреса и шлюз по умолчанию (если они были установлены);

2) нажать кнопку «Создать сетевой мост…» (Рис. 391);

3) в окне «Сетевые мосты» в поле «Интерфейс-мост» ввести имя моста;

4) в выпадающем списке «Тип моста» выбрать тип моста: «Linux Bridge» (по умолчанию) или «Open vSwitch»;

5) переместить сетевые интерфейсы, которые будут входить в мост, из списка «Доступные интерфейсы» в список «Члены»;

6) нажать кнопку «Ок» (Рис. 392);

7) в результате будет создан сетевой интерфейс моста (в примере vmbr0). Для данного интерфейса можно задать IP-адрес и, если необходимо, дополнительные параметры (Рис. 393);

8) нажать кнопку «Применить».

Настройка сети в веб-интерфейсе

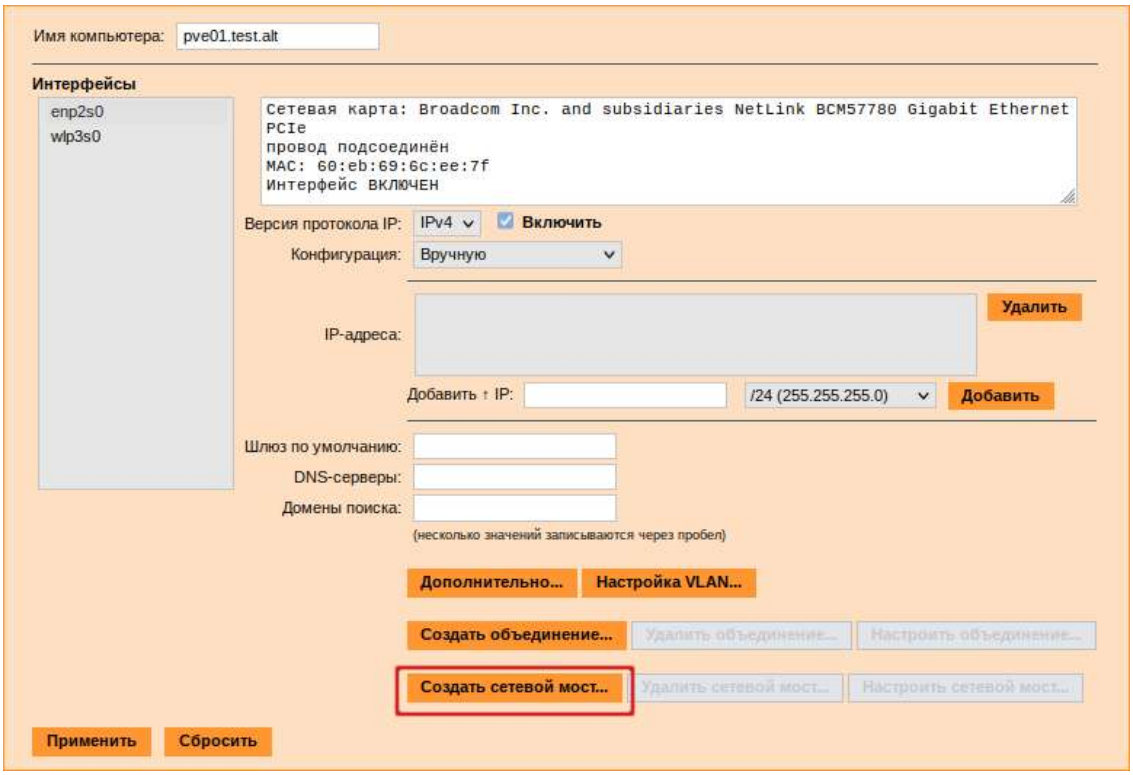


Рис. 391

Выбор сетевого интерфейса

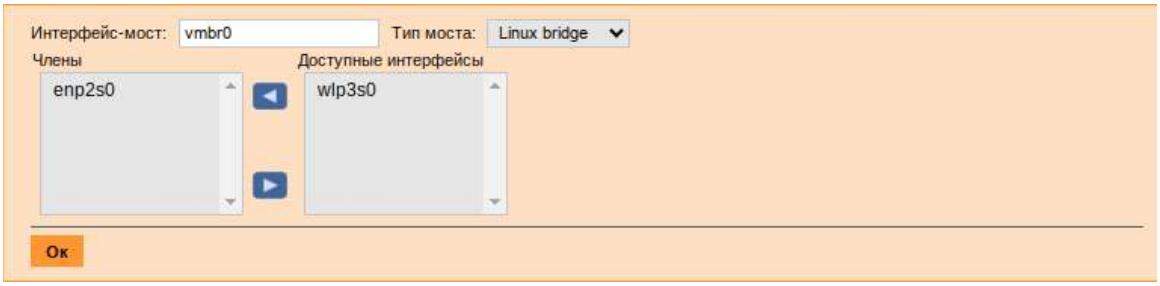


Рис. 392

Настройка параметров сетевого интерфейса vmbr0

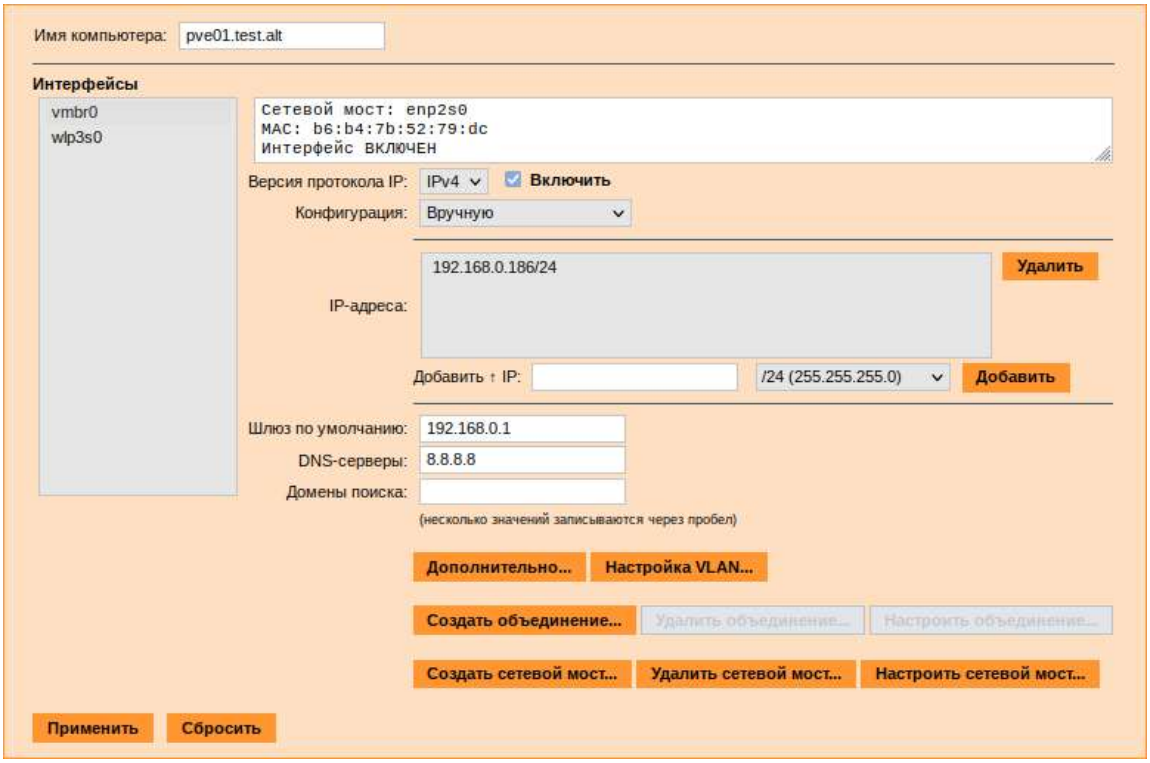


Рис. 393

Для удаления интерфейса моста необходимо выбрать его в списке «Интерфейсы» и нажать кнопку «Удалить сетевой мост…».

7.2.3 VLAN интерфейсы

Модуль «VLAN интерфейсы» (пакет alterator-net-vlan) предназначен для настройки 802.1Q VLAN.

Для создания интерфейсов VLAN необходимо выполнить следующие действия:

1) в списке «Интерфейсы» выбрать сетевой интерфейс и нажать кнопку «Настройка VLAN…» (Рис. 388);

2) ввести VLAN ID (число от 1 до 4095) в поле «VID» и нажать кнопку «Добавить VLAN» (Рис. 394);

Примечание. Следует обратить внимание, что 4094 является верхней допустимой границей идентификатора VLAN, а 4095 используется технически в процессе отбрасывания трафика по неверным VLAN.

3) для того чтобы вернуться к основным настройкам, нажать кнопку «Назад»;

4) в результате будут созданы виртуальные интерфейсы с именем, содержащим VLAN ID. Для данных интерфейсов можно задать IP-адрес и, если необходимо, дополнительные параметры (Рис. 395);

5) нажать кнопку «Применить».

Создание интерфейса VLAN

Рис. 394

Настройки интерфейса enp0s8.100

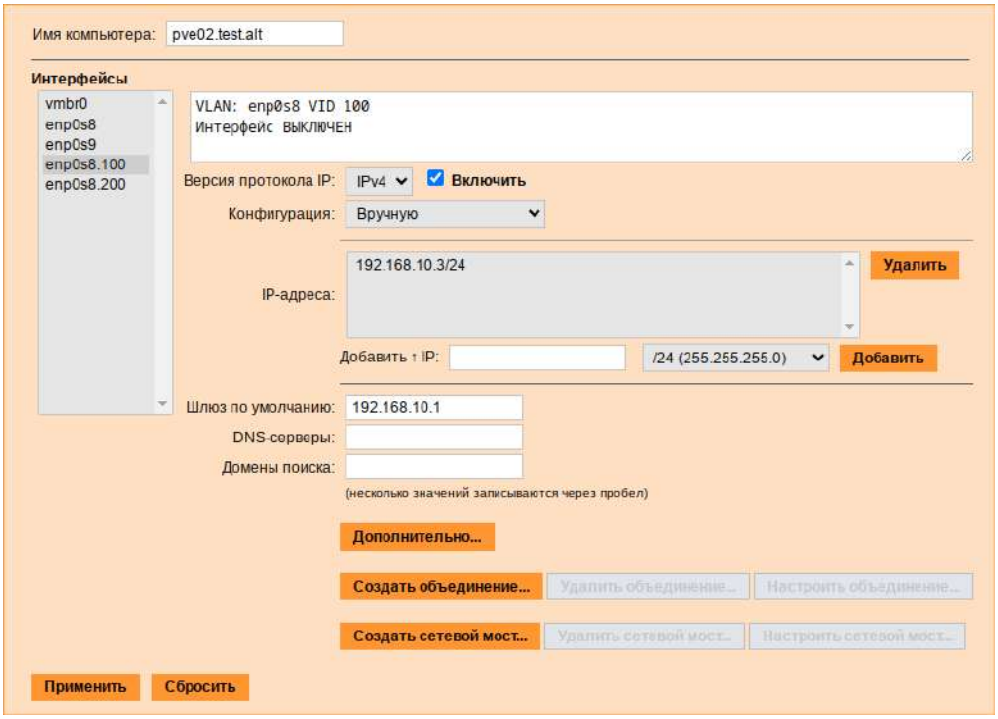


Рис. 395

Для удаления интерфейса VLAN следует в списке «Интерфейсы» выбрать «родительский» сетевой интерфейс и нажать кнопку «Настройка VLAN…». Затем в открывшемся окне выбрать VLAN интерфейс и нажать кнопку «Delete» («Удалить») (Рис. 396).

Удаление интерфейса VLAN

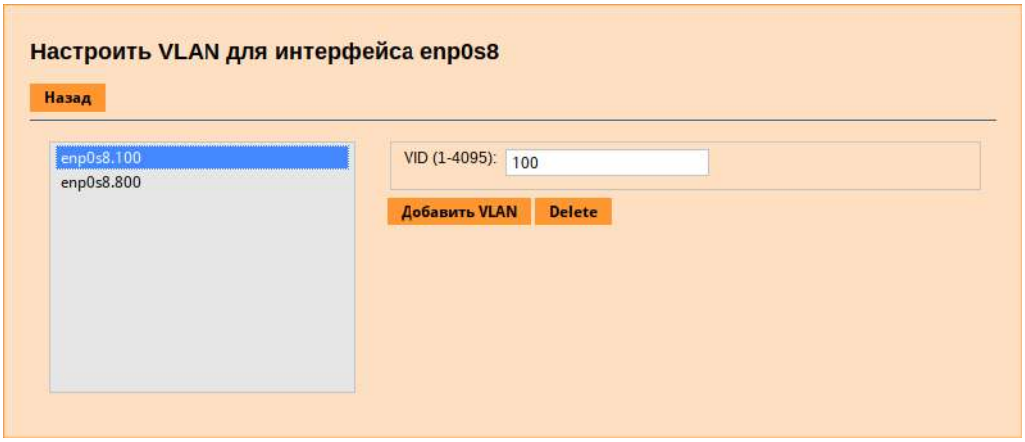


Рис. 396

7.3 Доступ к службам сервера из сети Интернет

7.3.1 Внешние сети

ОС предоставляет возможность организовать доступ к своим службам извне. Для обеспечения такой возможности необходимо разрешить входящие соединения на внешних интерфейсах. По умолчанию такие соединения блокируются.

Для разрешения внешних и внутренних входящих соединений предусмотрен раздел ЦУС «Брандмауэр». В списке «Разрешить входящие соединения на внешних интерфейсах» модуля «Внешние сети» (пакет alterator-net-iptables) перечислены наиболее часто используемые службы, отметив которые, можно сделать их доступными для соединений на внешних сетевых интерфейсах (Рис. 397). Если необходимо предоставить доступ к службе, отсутствующей в списке, то нужно задать используемые этой службой порты в соответствующих полях.

Можно выбрать один из трех режимов работы:

- роутер – перенаправление пакетов между сетевыми интерфейсами происходит без трансляции сетевых адресов;
- шлюз (NAT) – в этом режиме будет настроена трансляция сетевых адресов (NAT) при перенаправлении пакетов на внешние интерфейсы. Использование этого режима имеет смысл, если на компьютере настроен, по крайней мере, один внешний и один внутренний интерфейс;
- Хост (Рабочая станция) – в этом режиме можно для всех интерфейсов открыть или закрыть порт. Внешними автоматически выбираются все интерфейсы, кроме lo и специальных исключений (virbr*, docker*).

Модуль «Внешние сети»

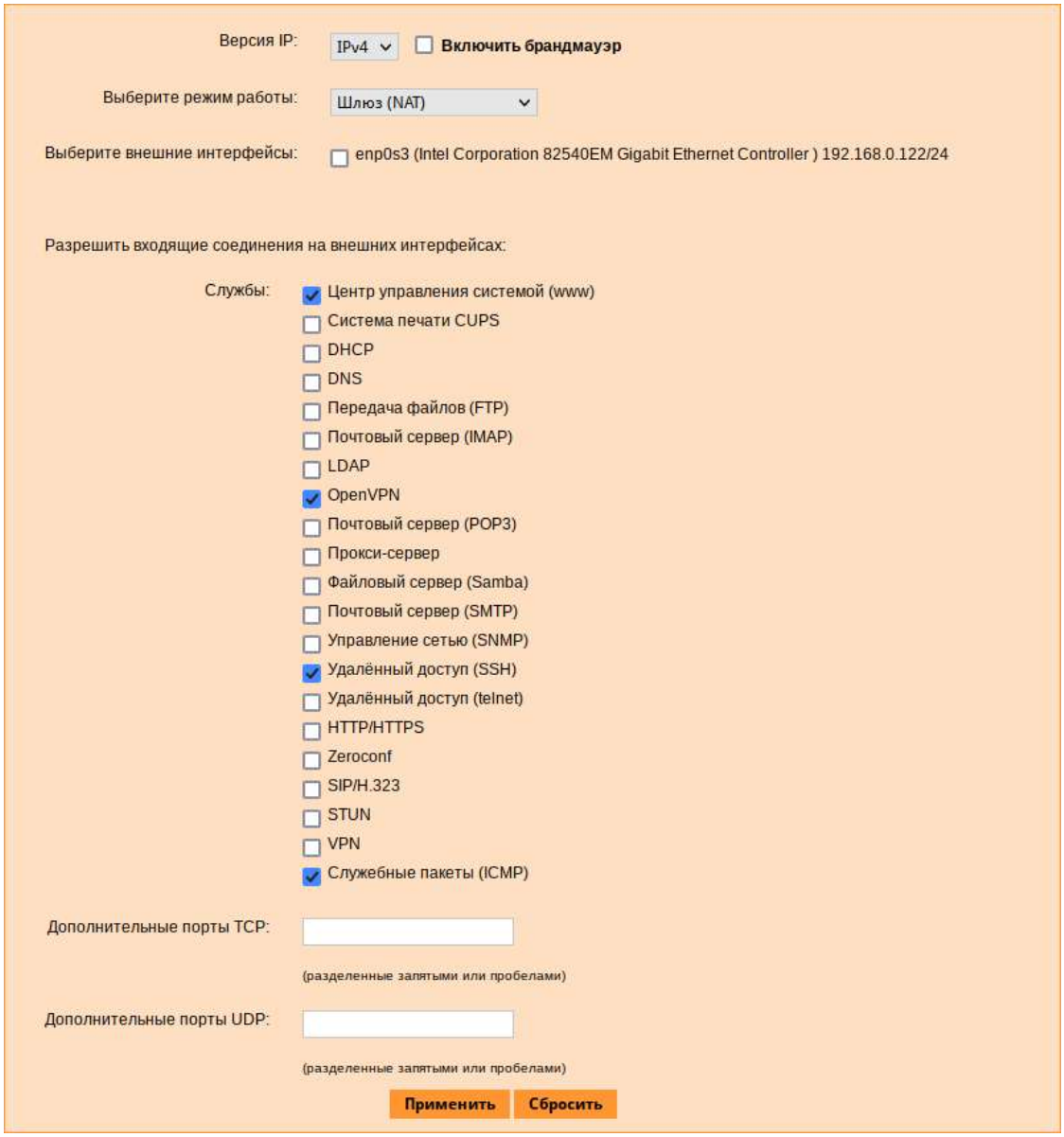


Рис. 397

В любом режиме включено только перенаправление пакетов с внутренних интерфейсов. Перенаправление пакетов с внешних интерфейсов всегда выключено. Все внутренние интерфейсы открыты для любых входящих соединений.

7.3.2 Список блокируемых хостов

Модуль «Список блокируемых хостов» (пакет alterator-net-iptables) позволяет настроить блокировку любого сетевого трафика с указанных в списке узлов (входящий, исходящий и пересылаемый).

Блокирование трафика с указанных в списке узлов начинается после установки флажка «Использовать чёрный список» (Рис. 398).

Модуль «Список блокируемых хостов»

Рис. 398

Для добавления блокируемого узла необходимо ввести IP-адрес в поле «Добавить IP-адрес сети или хоста» и нажать кнопку «Добавить».

Для удаления узла необходимо выбрать его из списка и нажать кнопку «Удалить».

7.4 Обслуживание сервера

Регулярный мониторинг состояния системы, своевременное резервное копирование, обновление установленного ПО, являются важной частью комплекса работ по обслуживанию сервера.

7.4.1 Мониторинг состояния системы

Для обеспечения бесперебойной работы сервера крайне важно производить постоянный мониторинг его состояния. Все события, происходящие с сервером, записываются в журналы, анализ которых помогает избежать сбоев в работе сервера и предоставляет возможность разобраться в причинах некорректной работы сервера.

Для просмотра журналов предназначен модуль ЦУС «Системные журналы» (пакет alterator-logs) из раздела «Система». Интерфейс позволяет просмотреть различные типы журналов с возможностью перехода к более старым или более новым записям.

Различные журналы могут быть выбраны из списка «Журналы» (Рис. 399).

Доступны следующие виды журналов:

- «Брандмауэр» – отображаются события безопасности, связанные с работой межсетевого экрана ОС;
- «Системные сообщения (Journald)» – отображаются события процессов ядра и пользовательской области. У каждого сообщения в этом журнале есть приоритет, который используется для пометки важности сообщений. Сообщения в зависимости от уровня приоритета подсвечиваются цветом.

Модуль «Системные журналы»

Рис. 399

Каждый журнал может содержать довольно большое количество сообщений. Уменьшить либо увеличить количество выводимых строк можно, выбрав нужное значение в списке «Показывать».

7.4.2 Системные службы

Для изменения состояния служб можно использовать модуль ЦУС «Системные службы» (пакет alterator-services) из раздела «Система». Интерфейс позволяет изменять текущее состояние службы и, если необходимо, применить опцию запуска службы при загрузке системы (Рис. 400).

Модуль «Системные службы»

Рис. 400

После выбора названия службы из списка отображается описание данной службы, а также текущее состояние: «Работает»/«Остановлена»/«Неизвестно».

7.4.3 Обновление системы

После установки системы крайне важно следить за обновлениями ПО. Обновления для ОС «Альт Сервер Виртуализации» могут содержать как исправления, связанные с безопасностью, так и новый функционал или просто улучшение и ускорение алгоритмов. В любом случае настоятельно рекомендуется регулярно обновлять систему для повышения надёжности работы сервера.

Для автоматизации процесса установки обновлений предусмотрен модуль ЦУС «Обновление системы» (пакет alterator-updates) из раздела «Система». Здесь можно включить автоматическое обновление через Интернет с одного из предлагаемых серверов или задать собственные настройки (Рис. 401).

Источник обновлений указывается явно (при выбранном режиме «Обновлять систему автоматически из сети Интернет») или вычисляется автоматически (при выбранном режиме «Обновление системы управляемое сервером» и наличии в локальной сети настроенного сервера обновлений).

Процесс обновления системы будет запускаться автоматически согласно заданному расписанию.

Модуль «Обновление системы»

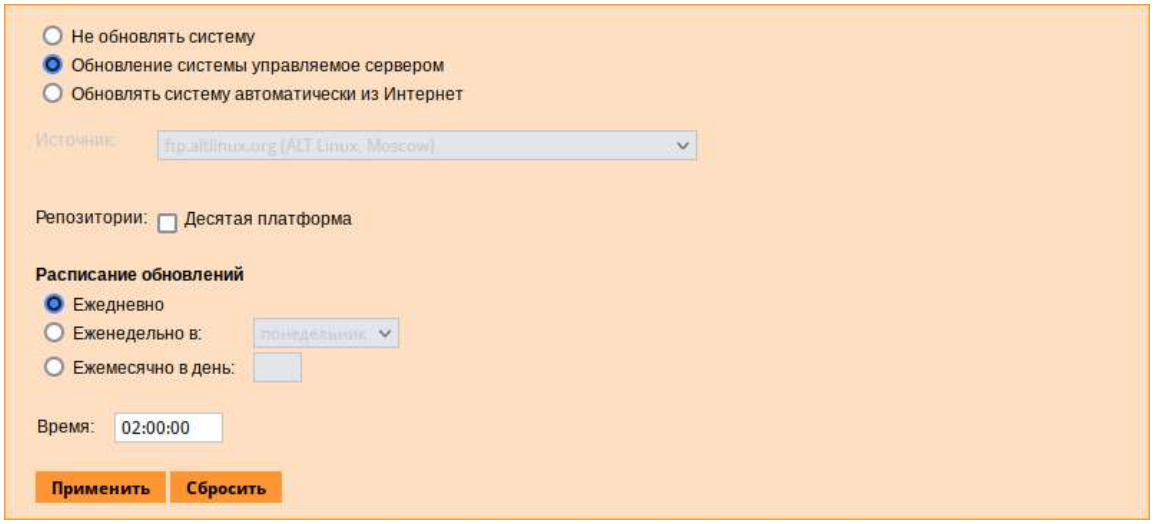


Рис. 401

7.4.4 Консоль

Модуль «Консоль» (пакет alterator-console) предназначен для запуска произвольных команд (Рис. 402).

Модуль «Консоль»

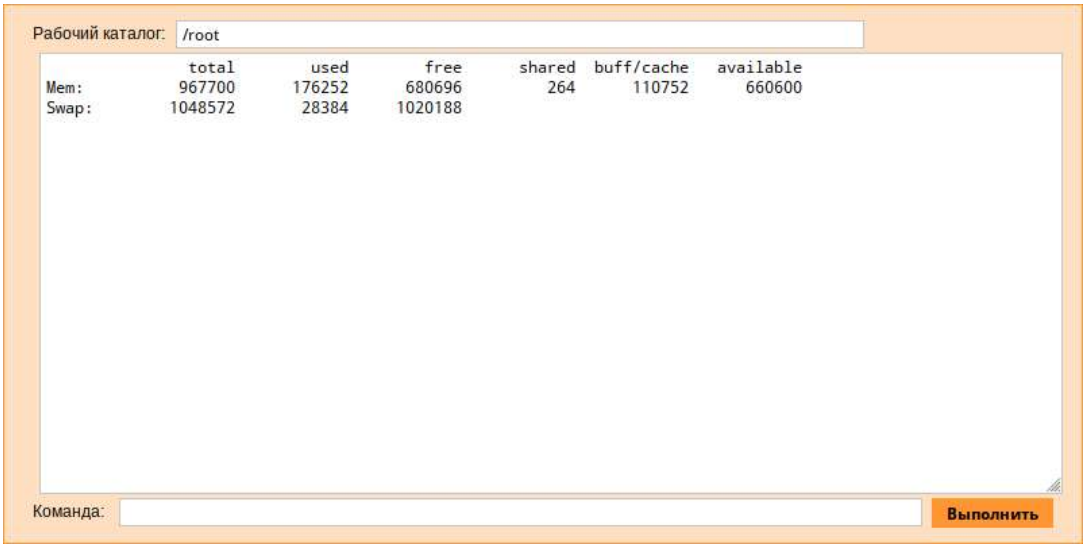


Рис. 402

Для запуска команды необходимо ввести команду в поле «Команда» и нажать кнопку «Выполнить». В рабочем поле будет выведен результат выполнения команды. Команда будет выполнена в указанном рабочем каталоге (по умолчанию `/root`).

7.4.5 Информация о системе

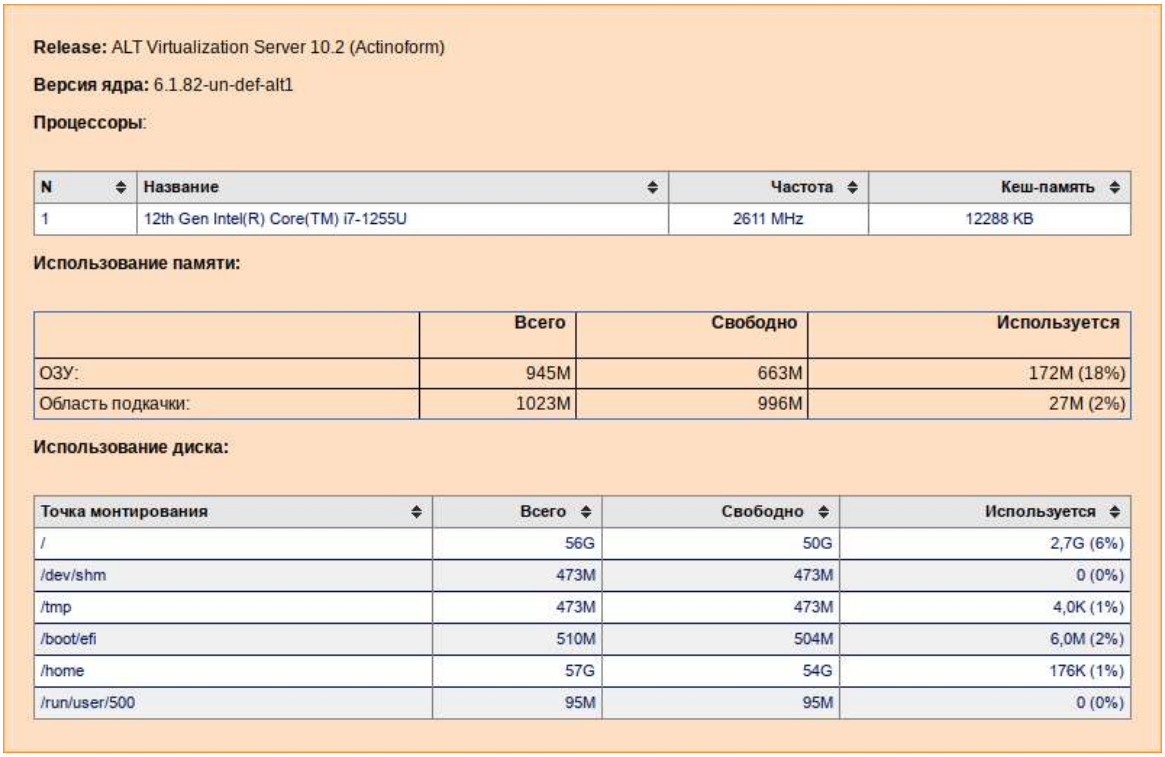
Модуль «Информация о системе» (пакет alterator-sysinfo) предназначен для отображения информации о системе (Рис. 403):

- версии загруженного ядра;
- информации о процессорах;
- использование памяти;
- использование дискового пространства.

7.4.6 Веб-интерфейс

Модуль «Веб-интерфейс» предназначен для управления настройками веб-сервера, обеспечивающего работоспособность ЦУС. В поле «Порт» (Рис. 404) указывается номер TCP-порта, на котором сервер принимает соединения (порт по умолчанию 8080), в поле «Адрес» указывается IP-адрес сетевого интерфейса, на котором будет доступен ЦУС, в списке «Протоколирование» можно выбрать степень подробности протоколирования.

Модуль «Информация о системе»



Release: ALT Virtualization Server 10.2 (Actinoform)

Версия ядра: 6.1.82-un-def-alt1

Процессоры:

N	Название	Частота	Кеш-память
1	12th Gen Intel(R) Core(TM) i7-1255U	2611 MHz	12288 KB

Использование памяти:

	Всего	Свободно	Используется
ОЗУ:	945M	663M	172M (18%)
Область подкачки:	1023M	996M	27M (2%)

Использование диска:

Точка монтирования	Всего	Свободно	Используется
/	56G	50G	2,7G (6%)
/dev/shm	473M	473M	0 (0%)
/tmp	473M	473M	4,0K (1%)
/boot/efi	510M	504M	6,0M (2%)
/home	57G	54G	176K (1%)
/run/user/500	95M	95M	0 (0%)

Рис. 403

Модуль «Веб-интерфейс»

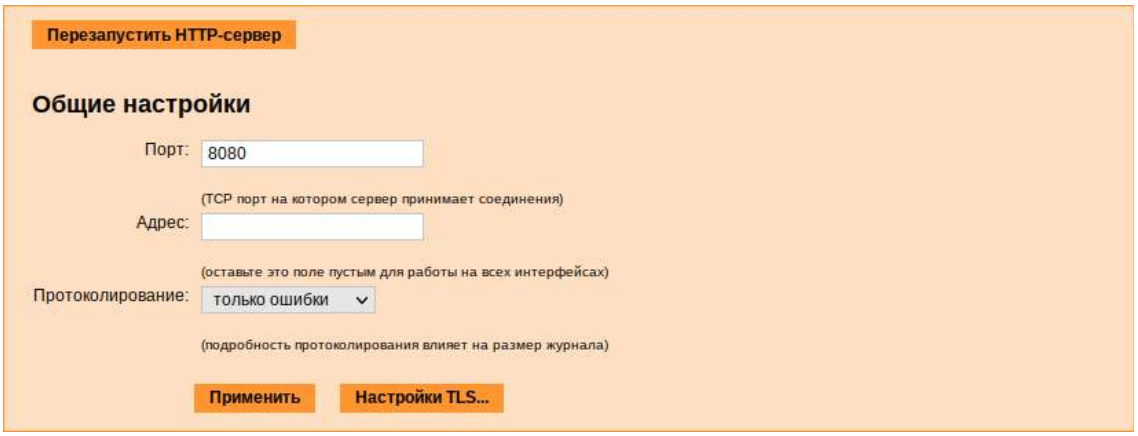


Рис. 404

7.4.7 Локальные учётные записи

Модуль «Локальные учётные записи» (пакет alterator-users) из раздела «Пользователи» предназначен для администрирования системных пользователей (Рис. 405).

Веб-интерфейс модуля alterator-users

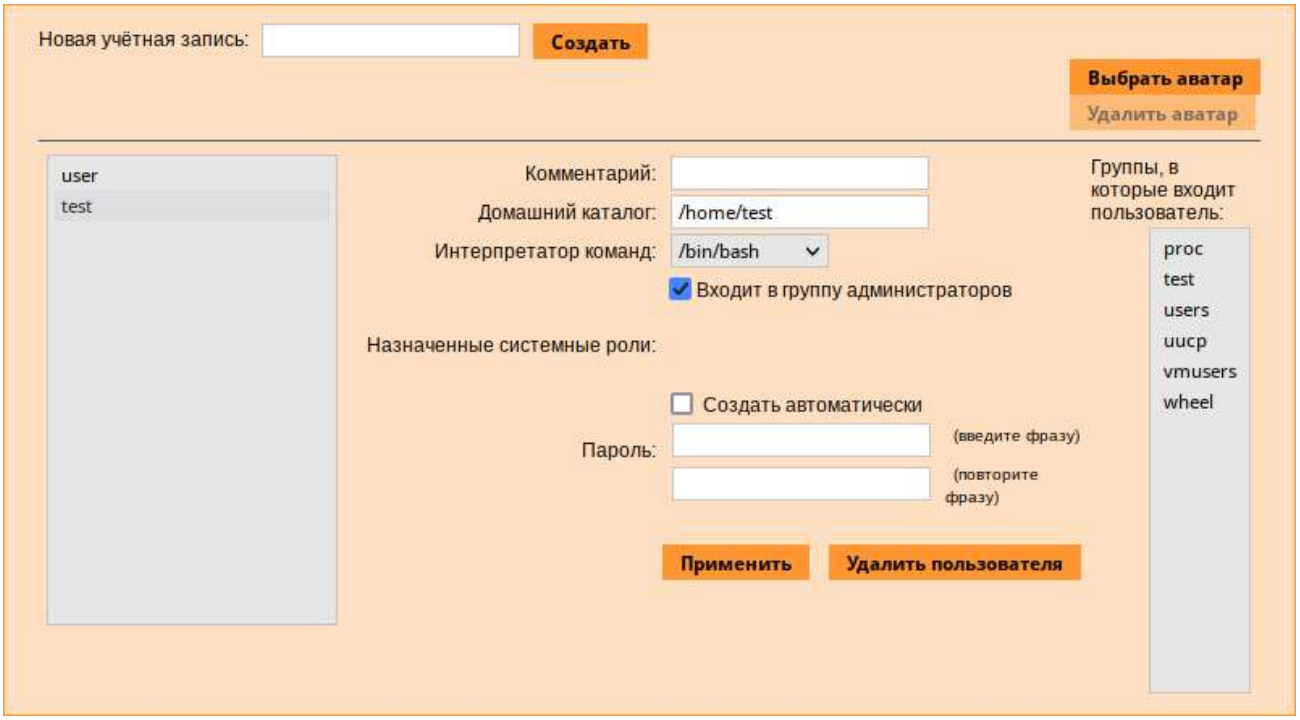


Рис. 405

Для создания новой учётной записи необходимо ввести имя новой учётной записи и нажать кнопку «Создать», после чего имя отобразится в списке слева.

Для дополнительных настроек необходимо выделить добавленное имя, либо, если необходимо изменить существующую учётную запись, выбрать её из списка.

7.4.8 Администратор системы

В модуле «Администратор системы» (пакет alterator-root) из раздела «Пользователи» можно изменить пароль суперпользователя (root), заданный при начальной настройке системы (Рис. 406).

Модуль «Администратор системы»

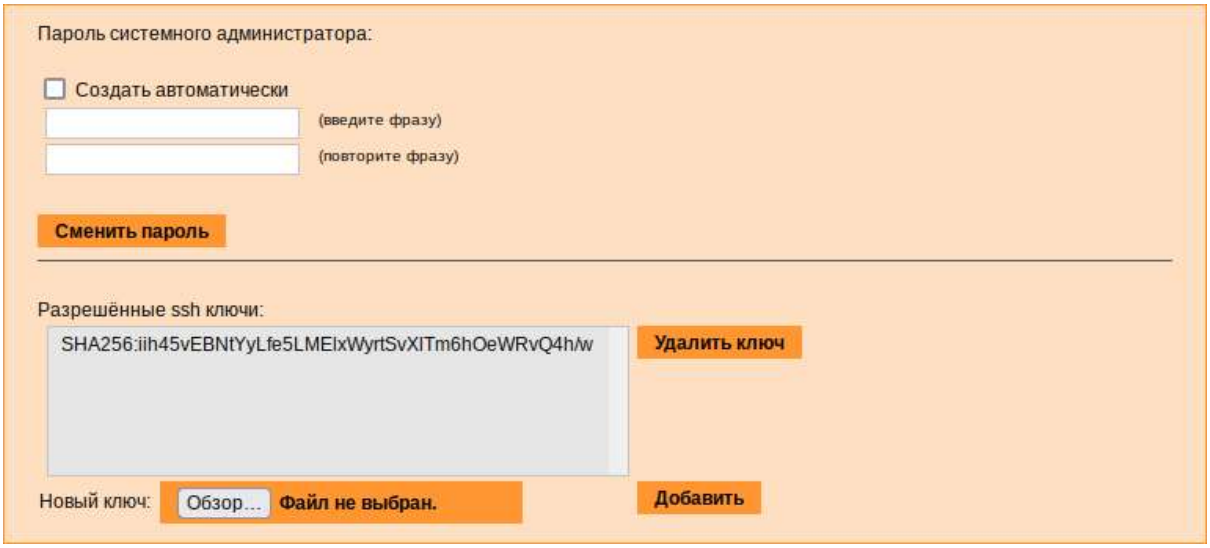


Рис. 406

В данном модуле можно добавить публичную часть ключа RSA или DSA для доступа к серверу по протоколу SSH.

7.4.9 Дата и время

В модуле «Дата и время» (пакет alterator-datetime) из раздела «Система» можно изменить дату и время на сервере, сменить часовой пояс, а также настроить автоматическую синхронизацию часов на самом сервере по протоколу NTP и предоставление точного времени по этому протоколу для рабочих станций локальной сети (Рис. 407).

Модуль «Дата и время»

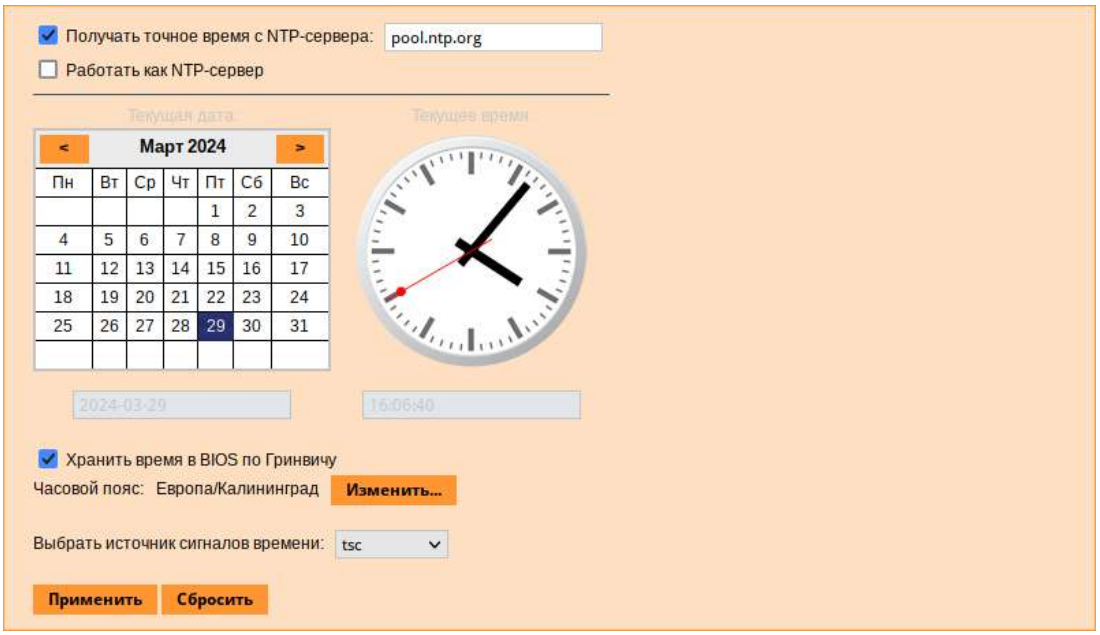


Рис. 407

Системное время зависит от следующих факторов:

- часы в BIOS – часы, встроенные в компьютер; они работают, даже если он выключен;
- системное время – часы в ядре операционной системы. Во время работы системы все процессы пользуются именно этими часами;
- часовые пояса – регионы Земли, в каждом из которых принято единое местное время.

При запуске системы происходит активация системных часов и их синхронизация с аппаратными, кроме того, в определённых случаях учитывается значение часового пояса. При завершении работы системы происходит обратный процесс.

Если настроена синхронизация времени с NTP-сервером, то сервер сможет сам работать как сервер точного времени. Для этого достаточно отметить соответствующий пункт «Работать как NTP-сервер».

Примечание. Выбор источника сигналов времени (источника тактовой частоты) доступен в режиме эксперта.

7.4.10 Ограничение использования диска

Модуль «Использование диска» (пакет `alterator-quota`) из раздела «Пользователи» позволяет ограничить использование дискового пространства пользователями, заведёнными на сервере в модуле «Пользователи».

Модуль позволяет задать ограничения (квоты) для пользователя при использовании определённого раздела диска. Ограничить можно как суммарное количество килобайт, занятых файлами пользователя, так и количество этих файлов (Рис. 408).

Модуль «Использование диска»

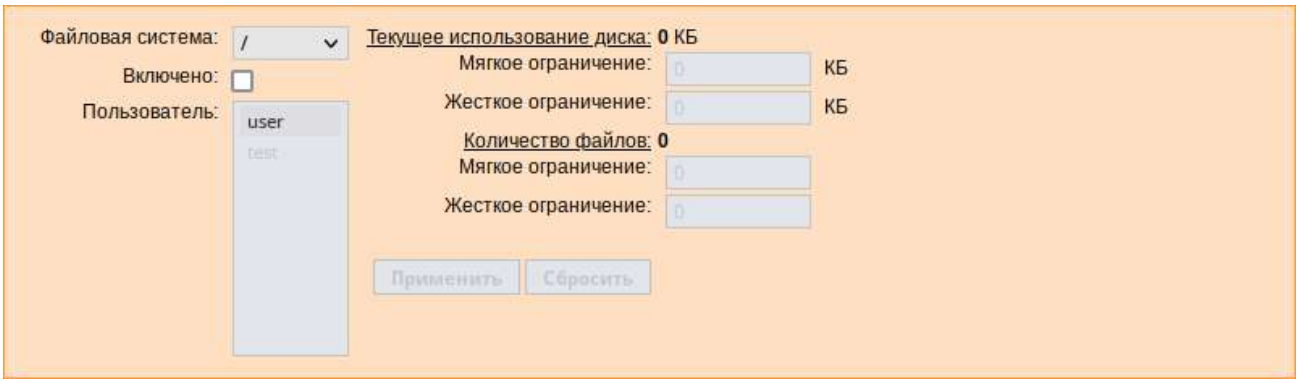


Рис. 408

Для управления квотами файловая система должна быть подключена с параметрами usrquota, grpquota. Для этого следует выбрать нужный раздел в списке «Файловая система» и установить отметку в поле «Включено» (Рис. 409).

Для того чтобы задать ограничения для пользователя, необходимо выбрать пользователя в списке «Пользователь», установить ограничения и нажать кнопку «Применить».

Модуль «Использование диска»

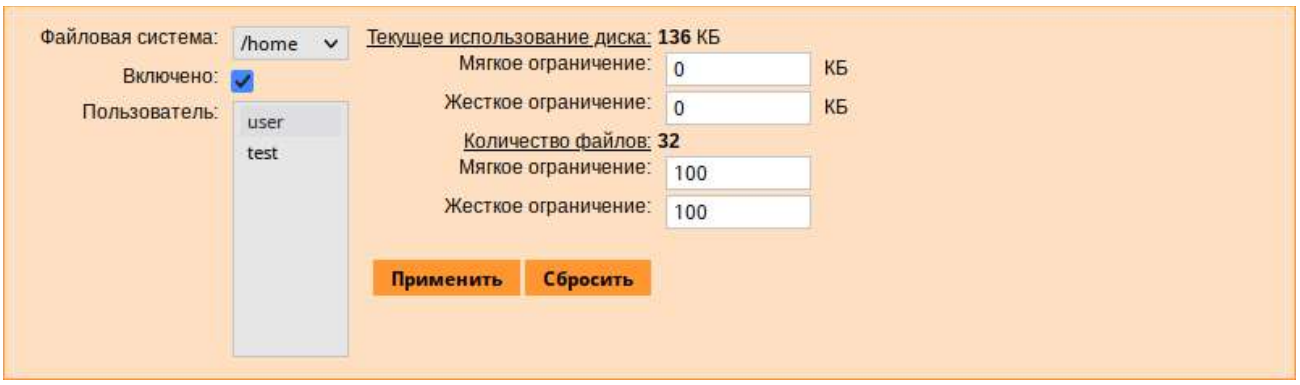


Рис. 409

При задании ограничений различают жёсткие и мягкие ограничения:

- мягкое ограничение: нижняя граница ограничения, которая может быть временно превышена. Временное ограничение – одна неделя;

- жёсткое ограничение: использование диска, которое не может быть превышено ни при каких условиях.

Значение 0 при задании ограничений означает отсутствие ограничений.

7.4.11 Выключение и перезагрузка компьютера

Иногда, в целях обслуживания или по организационным причинам необходимо корректно выключить или перезагрузить сервер. Для этого можно воспользоваться модулем ЦУС «Выключение компьютера» в разделе «Система».

Модуль ЦУС «Выключение компьютера» позволяет:

- выключить компьютер;
- перезагрузить компьютер;
- приостановить работу компьютера;
- погрузить компьютер в сон.

Возможна настройка ежедневного применения данных действий в заданное время.

Так как выключение и перезагрузка – критичные для функционирования компьютера операции, то по умолчанию настройка выставлена в значение «Продолжить работу» (Рис. 410). Для выключения, перезагрузки или перехода в энергосберегающие режимы нужно отметить соответствующий пункт и нажать «Применить».

Для ежедневного автоматического выключения компьютера, перезагрузки, а также перехода в энергосберегающие режимы необходимо отметить соответствующий пункт и задать желаемое время. Например, для выключения компьютера следует отметить пункт «Выключать компьютер каждый день в», задать время выключения в поле ввода слева от этого флажка и нажать кнопку «Применить».

Модуль «Выключение компьютера»

Рис. 410

Примечание. Для возможности настройки оповещений на e-mail, должен быть установлен пакет `state-change-notify-postfix` (из репозитория p10):

```
# apt-get install state-change-notify-postfix
```

Для настройки оповещений необходимо отметить пункт «При изменении состояния системы отправлять электронное письмо по адресу», ввести e-mail адрес и нажать кнопку «Применить» (Рис. 411).

Модуль «Выключение компьютера». Настройка оповещений

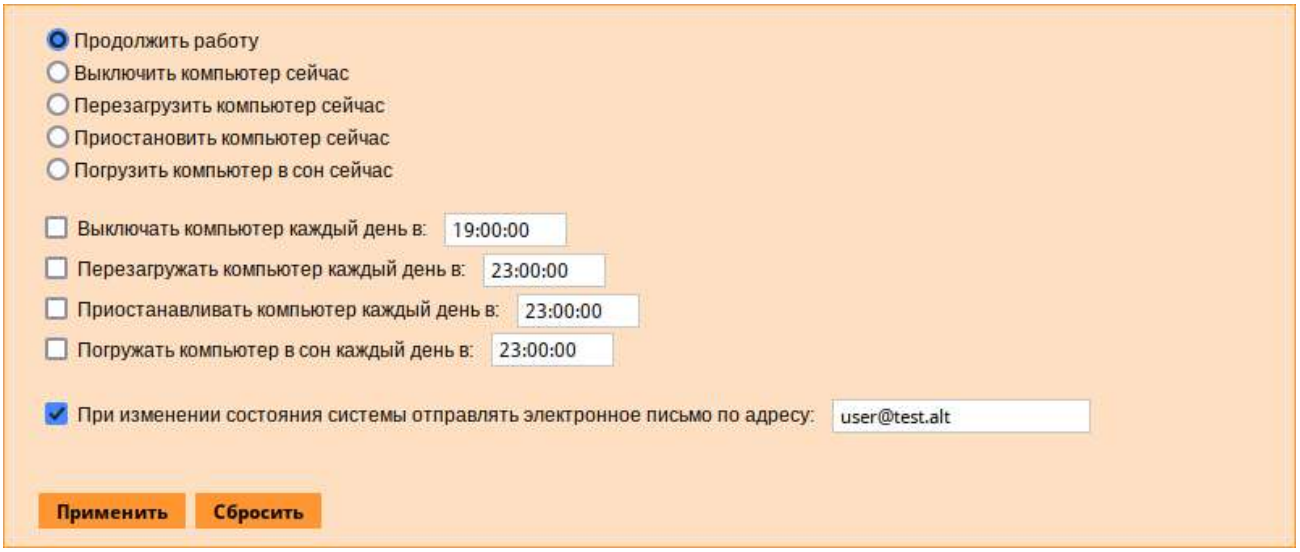


Рис. 411

По указанному адресу, при изменении состоянии системы будут приходить электронные письма. Например, при включении компьютера, содержание письма будет следующее:

```
Fri Mar 29 11:46:59 EET 2024: The server.test.alt is about to start.
```

При выключении:

```
Fri Mar 29 12:27:02 EET 2024: The server.test.alt is about to shutdown.
```

Кнопка «Сбросить» возвращает сделанный выбор к безопасному значению по умолчанию: «Продолжить работу», перечитывает расписания и выставляет отметки для ежедневного автоматического действия в соответствии с прочитанным.

7.5 Прочие возможности ЦУС

Возможности ЦУС ОС «Альт Сервер Виртуализации» не ограничиваются только теми, что были описаны выше.

Установленные пакеты, которые относятся к ЦУС, можно посмотреть, выполнив команду:

```
rpm -qa | grep alterator*
```

Прочие пакеты для ЦУС можно найти, выполнив команду:

```
apt-cache search alterator*
```

Модули можно дополнительно загружать и удалять как обычные программы:

```
# apt-get install alterator-net-openvpn
```

```
# apt-get remove alterator-net-openvpn
```

Примечание. После установки модуля, у которого есть веб-интерфейс, для того чтобы он отобразился в веб-интерфейсе, необходимо перезапустить сервис ahttpd:

```
# systemctl restart ahttpd
```

7.6 Права доступа к модулям ЦУС

Администратор системы (root) имеет доступ ко всем модулям, установленным в системе, и может назначать права доступа для пользователей к определенным модулям.

Для разрешения доступа пользователю к конкретному модулю, администратору в веб-интерфейсе ЦУС необходимо выбрать нужный модуль и нажать ссылку «Параметры доступа к модулю», расположенную в нижней части окна модуля (Рис. 412).

Ссылка «Параметры доступа к модулю»

Рис. 412

В открывшемся окне, в списке «Новый пользователь», необходимо выбрать пользователя, который получит доступ к данному модулю, и нажать кнопку «Добавить» (Рис. 413). Для сохранения настроек необходимо перезапустить HTTP-сервер, для этого достаточно нажать кнопку «Перезапустить HTTP-сервер».

Параметры доступа к модулю

Рис. 413

Для удаления доступа пользователя к определенному модулю, администратору, в окне этого модуля необходимо нажать ссылку «Параметры доступа к модулю», в открывшемся окне в списке пользователей которым разрешен доступ, выбрать пользователя, нажать кнопку «Удалить» (Рис. 413) и перезапустить HTTP-сервер.

Системный пользователь, пройдя процедуру аутентификации, может просматривать и вызывать модули, к которым он имеет доступ.

8 УСТАНОВКА ДОПОЛНИТЕЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

После установки ОС «Альт Сервер Виртуализации», при первом запуске, доступен тот или иной набор программного обеспечения. Количество предустановленных программ зависит от выбора, сделанного при установке системы. Имеется возможность доустановить программы, которых не хватает в системе, из разных источников.

Дополнительное программное обеспечение может находиться на установочном диске и/или в специальных банках программ (репозиториях), расположенных в сети Интернет и/или в локальной сети. Программы, размещённые в указанных источниках, имеют вид подготовленных для установки пакетов.

Примечание. В установочный комплект ОС «Альт Сервер Виртуализации» включено наиболее употребительное программное обеспечение. В то же время вы можете использовать репозиторий продукта (p10) для установки дополнительных программных пакетов.

Для установки, удаления и обновления программ и поддержания целостности системы в ОС семейства Linux используются менеджеры пакетов типа «rpm». Для автоматизации этого процесса и применяется Усовершенствованная система управления программными пакетами APT (Advanced Packaging Tool).

Автоматизация достигается созданием одного или нескольких внешних репозиториев, в которых хранятся пакеты программ и относительно которых производится сверка пакетов, установленных в системе. Репозитории могут содержать как официальную версию дистрибутива, обновляемую его разработчиками по мере выхода новых версий программ, так и локальные наработки, например, пакеты, разработанные внутри компании.

Таким образом, в распоряжении APT находятся две базы данных: одна описывает установленные в системе пакеты, вторая – внешний репозиторий. APT отслеживает целостность установленной системы и, в случае обнаружения противоречий в зависимостях пакетов, руководствуется сведениями о внешнем репозитории для разрешения конфликтов и поиска корректного пути их устранения.

Система APT состоит из нескольких утилит. Чаще всего используется утилита управления пакетами apt-get, которая автоматически определяет зависимости между пакетами и строго следит за их соблюдением при выполнении любой из следующих операций: установка, удаление или обновление пакетов.

8.1 Источники программ (репозитории)

Репозитории, с которыми работает APT, отличаются от обычного набора пакетов наличием

мета информации – индексов пакетов, содержащихся в репозитории, и сведений о них. Поэтому, чтобы получить всю информацию о репозитории, APT достаточно получить его индексы.

APT может работать с любым количеством репозиториев одновременно, формируя единую информационную базу обо всех содержащихся в них пакетах. При установке пакетов APT обращает внимание только на название пакета, его версию и зависимости, а расположение в том или ином репозитории не имеет значения. Если потребуется, APT в рамках одной операции установки группы пакетов может пользоваться несколькими репозиториями.

Подключая одновременно несколько репозиториев, нужно следить за тем, чтобы они были совместимы друг с другом по пакетной базе – отражали один определенный этап разработки. Совместимыми являются основной репозиторий дистрибутива и репозиторий обновлений по безопасности к данному дистрибутиву. В то же время смешение среди источников APT репозиториев, относящихся к разным дистрибутивам, или смешение стабильного репозитория с нестабильной веткой разработки (Sisyphus) чревато различными неожиданными трудностями при обновлении пакетов.

APT позволяет взаимодействовать с репозиторием с помощью различных протоколов доступа. Наиболее популярные – HTTP и FTP, однако существуют и некоторые дополнительные методы.

Для того чтобы APT мог использовать тот или иной репозиторий, информацию о нем необходимо поместить в файл `/etc/apt/sources.list`, либо в любой файл `.list` (например, `mysources.list`) в каталоге `/etc/apt/sources.list.d/`. Описания репозиториев заносятся в эти файлы в следующем виде:

```
rpm [подпись] метод: путь база название
rpm-src [подпись] метод: путь база название
```

где:

- rpm или rpm-src – тип репозитория (скомпилированные программы или исходные тексты);
- [подпись] – необязательная строка-указатель на электронную подпись разработчиков. Наличие этого поля подразумевает, что каждый пакет из данного репозитория должен быть подписан соответствующей электронной подписью. Подписи описываются в файле `/etc/apt/vendor.list`;
- метод – способ доступа к репозиторию: ftp, http, file, cdrom, copy;
- путь – путь к репозиторию в терминах выбранного метода;
- база – относительный путь к базе данных репозитория;
- название – название репозитория.

При выборе пакетов для установки APT руководствуется всеми доступными репозиториями вне зависимости от способа доступа к ним. Таким образом, если в репозитории, доступном по

сети Интернет, обнаружена более новая версия программы, чем на CD (DVD)-носителе информации, APT начнет загружать данный пакет по сети.

8.1.1 Добавление репозиториев

Непосредственно после установки дистрибутива «Альт Сервер Виртуализации» в `/etc/apt/sources.list`, а также в файлах `/etc/apt/sources.list.d/*.list` обычно указывается несколько репозиториев:

- репозиторий с установочного диска дистрибутива;
- интернет-репозиторий, совместимый с установленным дистрибутивом.

8.1.1.1 Утилита apt-repo для работы с репозиториями

Для добавления репозиториев можно воспользоваться утилитой `apt-repo`.

Примечание. Для выполнения большинства команд необходимы права администратора.

Просмотреть список активных репозиториев можно, выполнив команду:

```
$ apt-repo list
```

Команда добавления репозитория в список активных репозиториев:

```
apt-repo add <репозиторий>
```

Команда удаления или выключения репозитория:

```
apt-repo rm <репозиторий>
```

Команда удаления всех репозиториев:

```
apt-repo clean
```

Обновление информации о репозиториях:

```
apt-repo update
```

Вывод справки:

```
man apt-repo
```

или

```
apt-repo –help
```

Типичный пример использования: удалить все источники и добавить стандартный репозиторий p10 (архитектура выбирается автоматически):

```
# apt-repo rm all
# apt-repo add p10
```

Или то же самое одной командой:

```
# apt-repo set p10
```

8.1.1.2 Добавление репозитория на сменном носителе

Для добавления в `sources.list` репозитория на сменном носителе в APT предусмотрена специальная утилита – `apt-cdrom`.

Чтобы добавить запись о репозитории на сменном носителе необходимо:

1) создать каталог для монтирования. Точка монтирования указывается в параметре `Acquire::CDROM::mount` в файле конфигурации APT (/etc/apt/apt.conf), по умолчанию это `/media/ALTLinux`:

```
# mkdir /media/ALTLinux
```

2) примонтировать носитель в указанную точку:

```
# mount /dev/носитель /media/ALTLinux
```

где `/dev/носитель` – соответствующее блочное устройство (например, `/dev/dvd` – для CD/DVD-диска).

3) добавить носитель, выполнив команду:

```
# apt-cdrom -m add
```

После этого в `sources.list` появится запись о подключенном носителе примерно такого вида:

```
rpm cdrom:[ALT Server-V 10.4 x86_64 build 2024-10-28]/ ALTLinux main
```

Примечание. Команду `mount /dev/носитель /media/ALTLinux` необходимо выполнять перед каждой командой `apt-get install имя_пакета`.

8.1.1.3 Добавление репозиториев вручную

Для редактирования списка репозиториев можно отредактировать в любом текстовом редакторе файлы из папки `/etc/apt/sources.list.d/`. Для изменения этих файлов необходимы права администратора. В файле `alt.list` может содержаться такая информация:

```
rpm [alt] http://ftp.altlinux.org/pub/distributions/ALTLinux p10/x86_64 classic
rpm [alt] http://ftp.altlinux.org/pub/distributions/ALTLinux p10/x86_64-i586 classic
rpm [alt] http://ftp.altlinux.org/pub/distributions/ALTLinux p10/noarch classic
```

По сути, каждая строчка соответствует некому репозиторию. Для выключения репозитория достаточно закомментировать соответствующую строку (дописать символ решётки перед строкой). Для добавления нового репозитория необходимо дописать его вниз этого или любого другого файла.

8.1.2 Обновление информации о репозиториях

Практически любое действие с системой apt начинается с обновления данных от активированных источников. Список источников необходимо обновлять при поиске новой версии пакета, установке пакетов или обновлении установленных пакетов новыми версиями.

Обновление данных осуществляется командой:

```
# apt-get update
```

После выполнения этой команды, apt обновит свой кэш новой информацией.

8.2 Поиск пакетов

Утилита `apt-cache` предназначена для поиска программных пакетов, в репозитории, и позволяет искать не только по имени пакета, но и по его описанию.

Команда `apt-cache search <подстрока>` позволяет найти все пакеты, в именах или описании которых присутствует указанная подстрока. Пример поиска может выглядеть следующим образом:

```
$ apt-cache search ^telegraf
ceph-mgr-telegraf - Telegraf module for Ceph Manager Daemon
telegraf - The plugin-driven server agent for collecting and reporting metrics
```

Символ «^» в поисковом выражении, указывает на то, что необходимо найти совпадения только в начале строки (в данном случае – в начале имени пакета).

Для того чтобы подробнее узнать о каждом из найденных пакетов и прочитать его описание, можно воспользоваться командой `apt-cache show`, которая покажет информацию о пакете из репозитория:

```
$ apt-cache show telegraf
Package: telegraf
Section: Development/Other
Installed Size: 132855876
Maintainer: Alexey Shabalin (ALT Team) <shaba@altlinux.org>
Version: 1.19.2-alt1:p10+281572.100.1.1@1627678454
Pre-Depends: /bin/sh, /usr/sbin/groupadd, /usr/sbin/useradd, /usr/sbin/usermod, /usr/sbin/post_service, /usr/sbin/preun_service, rpmlib(PayloadIsXz)
Depends: /bin/kill, /bin/sh, /etc/logrotate.d, /etc/rc.d/init.d, /etc/rc.d/init.d(SourceIfNotEmpty), /etc/rc.d/init.d(msg_reloading), /etc/rc.d/init.d(msg_usage), /etc/rc.d/init.d(start_daemon), /etc/rc.d/init.d(status), /etc/rc.d/init.d(stop_daemon), /etc/rc.d/init.d/functions
Provides: telegraf (= 1.19.2-alt1:p10+281572.100.1.1)
Architecture: x86_64
Size: 22968033
MD5Sum: d9d6ecaba627d86436ddfdffc243f2cd
Filename: telegraf-1.19.2-alt1.x86_64.rpm
Description: The plugin-driven server agent for collecting and reporting metrics
 Telegraf is an agent written in Go for collecting, processing, aggregating, and writing metrics.

 Design goals are to have a minimal memory footprint with a plugin system so that developers
 in the community can easily add support for collecting metrics from well known services
 (like Hadoop, Postgres, or Redis) and third party APIs (like Mailchimp, AWS CloudWatch,
 or Google Analytics).
```

При поиске с помощью `apt-cache` можно использовать русскую подстроку. В этом случае будут найдены пакеты, имеющие описание на русском языке.

8.3 Установка или обновление пакета

Установка пакета с помощью APT выполняется командой:

```
# apt-get install <имя_пакета>
```

Примечание. Перед установкой и обновлением пакетов необходимо выполнить команду обновления индексов пакетов:

```
# apt-get update
```

Если пакет уже установлен и в подключенном репозитории нет обновлений для данного пакета, система сообщит об уже установленном пакете последней версии. Если в репозитории присутствует более новая версия или новое обновление – программа начнет процесс установки.

`apt-get` позволяет устанавливать в систему пакеты, требующие для работы другие, пока еще не установленные. В этом случае он определяет, какие пакеты необходимо установить, и устанавливает их, пользуясь всеми доступными репозиториями.

Установка пакета `telegraf` командой `apt-get install telegraf` приведет к следующему диалогу с APT (если пакет еще не установлен):

```
# apt-get install telegraf
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие НОВЫЕ пакеты будут установлены:
  telegraf
0 будет обновлено, 1 новых установлено, 0 пакетов будет удалено и 0 не будет обновлено.
Необходимо получить 29,1MB архивов.
После распаковки потребуется дополнительно 154MB дискового пространства.
Получено: 1 http://ftp.altlinux.org p10/branch/x86_64/classic telegraf 1.24.2-alt1:p10+326352.200.3.1@1691242931 [29,1MB]
Получено 29,1MB за 2s (11,0MB/s).
Совершаем изменения...
Подготовка...                          ################################## [100%]
Обновление / установка...
1: telegraf-1.24.2-alt1                ################################## [100%]
Завершено.
```

Команда `apt-get install <имя_пакета>` используется и для обновления уже установленного пакета или группы пакетов. В этом случае `apt-get` дополнительно проверяет, не обновилась ли версия пакета в репозитории по сравнению с установленным в системе.

Например, если пакет gimp установлен и в репозитории нет обновлённой версии этого пакета, то вывод команды `apt-get install telegraf` будет таким:

```
# apt-get install telegraf
Чтение списков пакетов... Завершено
```

```
Построение дерева зависимостей... Завершено
Последняя версия telegraf уже установлена.
0 будет обновлено, 0 новых установлено, 0 пакетов будет удалено и 2262 не будет обновлено.
```

При помощи APT можно установить и отдельный бинарный rpm-пакет, не входящий ни в один из репозиториев. Для этого достаточно выполнить команду `apt-get install путь_к_файлу.rpm`. При этом APT проведет стандартную процедуру проверки зависимостей и конфликтов с уже установленными пакетами.

В результате операций с пакетами без использования APT может нарушиться целостность ОС «Альт Сервер Виртуализации», и `apt-get` в таком случае откажется выполнять операции установки, удаления или обновления.

Для восстановления целостности ОС «Альт Сервер Виртуализации» необходимо повторить операцию, задав опцию `-f`, заставляющую `apt-get` исправить нарушенные зависимости, удалить или заменить конфликтующие пакеты. Любые действия в этом режиме обязательно требуют подтверждения со стороны пользователя.

При установке пакетов происходит запись в системный журнал вида:

```
apt-get: имя-пакета installed
```

8.4 Удаление установленного пакета

Для удаления пакета используется команда `apt-get remove <имя_пакета>`. Удаление пакета с сохранением его файлов настройки производится при помощи следующей команды:

```
# apt-get remove <значимая_часть_имени_пакета>
```

В случае, если при этом необходимо полностью очистить систему от всех компонент удаляемого пакета, то применяется команда:

```
# apt-get remove --purge <значимая_часть_имени_пакета>
```

Для того чтобы не нарушать целостность системы, будут удалены и все пакеты, зависящие от удаляемого.

В случае удаления с помощью `apt-get` базового компонента системы появится запрос на подтверждение операции:

```
# apt-get remove filesystem
Обработка файловых зависимостей... Завершено
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие пакеты будут УДАЛЕНЫ:
basesystem filesystem ppp sudo
Внимание: следующие базовые пакеты будут удалены:
В обычных условиях этого не должно было произойти, надеемся, вы точно
представляете, чего требуете!
```

```
basesystem filesystem (по причине basesystem)
0 пакетов будет обновлено, 0 будет добавлено новых, 4 будет
удалено(заменено) и 0 не будет обновлено.
Необходимо получить 0B архивов. После распаковки 588kБ будет
освобождено.
Вы делаете нечто потенциально опасное!
Введите фразу 'Yes, do as I say!' чтобы продолжить.
```

Каждую ситуацию, в которой APT выдает такое сообщение, необходимо рассматривать отдельно. Однако, вероятность того, что после выполнения этой команды система окажется неработоспособной, очень велика.

При удалении пакетов происходит запись в системный журнал вида:

```
apt-get: имя-пакета removed
```

8.5 Обновление всех установленных пакетов

Полное обновление всех установленных в системе пакетов производится при помощи команд:

```
# apt-get update && apt-get dist-upgrade
```

Первая команда (`apt-get update`) обновит индексы пакетов. Вторая команда (`apt-get dist-upgrade`) позволяет обновить только те установленные пакеты, для которых в репозиториях, перечисленных в `/etc/apt/sources.list`, имеются новые версии.

В случае обновления всего дистрибутива APT проведёт сравнение системы с репозиторием и удалит устаревшие пакеты, установит новые версии присутствующих в системе пакетов, отследит ситуации с переименованиями пакетов или изменения зависимостей между старыми и новыми версиями программ. Всё, что потребуется поставить (или удалить) дополнительно к уже имеющемуся в системе, будет указано в отчете apt-get, которым APT предварит само обновление.

Примечание. Команда `apt-get dist-upgrade` обновит систему, но ядро ОС не будет обновлено.

8.6 Обновление ядра

Для обновления ядра ОС необходимо выполнить команду:

```
# update-kernel
```

Примечание. Если индексы сегодня еще не обновлялись перед выполнением команды update-kernel необходимо выполнить команду `apt-get update`.

Команда `update-kernel` обновляет и модули ядра, если в репозитории обновилось что-то из модулей без обновления ядра.

Новое ядро загрузится только после перезагрузки системы.

9 КОРПОРАТИВНАЯ ИНФРАСТРУКТУРА

9.1 Zabbix

Zabbix – система мониторинга и отслеживания статусов разнообразных сервисов компьютерной сети, серверов и сетевого оборудования.

Для управления системой мониторинга и чтения данных используется веб-интерфейс.

Перед установкой должен быть установлен и запущен сервер PostgreSQL, с созданным пользователем zabbix и созданной базой zabbix.

9.1.1 Установка клиента Zabbix

Установить необходимый пакет:

```
# apt-get install zabbix-agent
```

Добавить Zabbix agent в автозапуск и запустить его:

```
# systemctl enable --now zabbix_agentd
```

Адрес сервера, которому разрешено обращаться к агенту задается в конфигурационном файле `/etc/zabbix/zabbix_agentd.conf` параметрами:

```
Server=127.0.0.1
ServerActive=127.0.0.1
```

10 ОБЩИЕ ПРИНЦИПЫ РАБОТЫ ОС

Работа с операционной средой заключается во вводе определенных команд (запросов) к операционной среде и получению на них ответов в виде текстового отображения.

Основой операционной среды является операционная система.

Операционная система (ОС) – совокупность программных средств, организующих согласованную работу операционной среды с аппаратными устройствами компьютера (процессор, память, устройства ввода-вывода и т. д.).

Диалог с ОС осуществляется посредством командных интерпретаторов и системных библиотек.

Каждая системная библиотека представляет собой набор программ, динамически вызываемых операционной системой.

Командные интерпретаторы – особый род специализированных программ, позволяющих осуществлять диалог с ОС посредством команд.

Для удобства пользователей при работе с командными интерпретаторами используются интерактивные рабочие среды (далее – ИРС), предоставляющие пользователю удобный интерфейс для работы с ОС.

В самом центре ОС изделия находится управляющая программа, называемая ядром. В ОС изделия используется новейшая модификация «устойчивого» ядра Linux – версия 5.4.

Ядро взаимодействует с компьютером и периферией (дисками, принтерами и т. д.), распределяет ресурсы и выполняет фоновое планирование заданий.

Другими словами, ядро ОС изолирует вас от сложностей аппаратуры компьютера, командный интерпретатор от ядра, а ИРС от командного интерпретатора.

ОС «Альт Сервер Виртуализации» является многопользовательской интегрированной системой. Это значит, что она разработана в расчете на одновременную работу нескольких пользователей.

Пользователь может либо сам работать в системе, выполняя некоторую последовательность команд, либо от его имени могут выполняться прикладные процессы.

Пользователь взаимодействует с системой через командный интерпретатор, который представляет собой, как было сказано выше, прикладную программу, которая принимает от пользователя команды или набор команд и транслирует их в системные вызовы к ядру системы. Интерпретатор позволяет пользователю просматривать файлы, передвигаться по дереву файловой системы, запускать прикладные процессы. Все командные интерпретаторы UNIX имеют развитый командный язык и позволяют писать достаточно сложные программы, упрощающие процесс администрирования системы и работы с ней.

10.1 Процессы функционирования ОС

Все программы, которые выполняются в текущий момент времени, называются процессами. Процессы можно разделить на два основных класса: системные процессы и пользовательские процессы. Системные процессы – программы, решающие внутренние задачи ОС, например, организацию виртуальной памяти на диске или предоставляющие пользователям те или иные сервисы (процессы-службы).

Пользовательские процессы – процессы, запускаемые пользователем из командного интерпретатора для решения задач пользователя или управления системными процессами. Linux изначально разрабатывался как многозадачная система. Он использует технологии, опробованные и отработанные другими реализациями UNIX, которые существовали ранее.

Фоновый режим работы процесса – режим, когда программа может работать без взаимодействия с пользователем. В случае необходимости интерактивной работы с пользователем (в общем случае) процесс будет «остановлен» ядром, и работа его продолжится только после переведения его в «нормальный» режим работы.

10.2 Файловая система ОС

В ОС использована файловая система Linux, которая в отличие от файловых систем DOS и Windows(™) является единым деревом. Корень этого дерева – каталог, называемый root (рут), и обозначаемый «/». Части дерева файловой системы могут физически располагаться в разных разделах разных дисков или вообще на других компьютерах, – для пользователя это прозрачно. Процесс присоединения файловой системы раздела к дереву называется монтированием, удаление – размонтированием. Например, файловая система CD-ROM в изделии монтируется по умолчанию в каталог /media/cdrom (путь в изделии обозначается с использованием «/», а не «\», как в DOS/ Windows). Текущий каталог обозначается «./».

Файловая система изделия содержит каталоги первого уровня:

- /bin (командные оболочки (shell), основные утилиты);
- /boot (содержит ядро системы);
- /dev (псевдофайлы устройств, позволяющие работать с ними напрямую);
- /etc (файлы конфигурации);
- /home (личные каталоги пользователей);
- /lib (системные библиотеки, модули ядра);
- /lib64 (64-битные системные библиотеки);
- /media (каталоги для монтирования файловых систем сменных устройств);
- /mnt (каталоги для монтирования файловых систем сменных устройств и внешних файловых систем);

- /proc (файловая система на виртуальном устройстве, ее файлы содержат информацию о текущем состоянии системы);
- /root (личный каталог администратора системы);
- /sbin (системные утилиты);
- /sys (файловая система, содержащая информацию о текущем состоянии системы);
- /usr (программы и библиотеки, доступные пользователю);
- /var (рабочие файлы программ, очереди, журналы);
- /tmp (временные файлы).

10.3 Организация файловой структуры

Система домашних каталогов пользователей помогает организовывать безопасную работу пользователей в многопользовательской системе. Вне своего домашнего каталога пользователь обладает минимальными правами (обычно чтение и выполнение файлов) и не может нанести ущерб системе, например, удалив или изменив файл.

Кроме файлов, созданных пользователем, в его домашнем каталоге обычно содержатся персональные конфигурационные файлы некоторых программ.

Маршрут (путь) – это последовательность имён каталогов, представляющий собой путь в файловой системе к данному файлу, где каждое следующее имя отделяется от предыдущего наклонной чертой (слэшем). Если название маршрута начинается со слэша, то путь в искомый файл начинается от корневого каталога всего дерева системы. В обратном случае, если название маршрута начинается непосредственно с имени файла, то путь к искомому файлу должен начаться от текущего каталога (рабочего каталога).

Имя файла может содержать любые символы за исключением косой черты (/). Однако следует избегать применения в именах файлов большинства знаков препинания и непечатаемых символов. При выборе имен файлов рекомендуется ограничиться следующими символами:

- строчные и ПРОПИСНЫЕ буквы. Следует обратить внимание на то, что регистр всегда имеет значение;
- цифры;
- символ подчеркивания (_);
- точка (.).

Для удобства работы можно использовать точку (.) для отделения имени файла от расширения файла. Данная возможность может быть необходима пользователям или некоторым программам, но не имеет значение для shell.

10.3.1 Иерархическая организация файловой системы

Каталог /:

/boot – место, где хранятся файлы необходимые для загрузки ядра системы;

/lib – здесь располагаются файлы динамических библиотек, необходимых для работы большей части приложений и подгружаемые модули ядра;

/lib64 – здесь располагаются файлы 64-битных динамических библиотек, необходимых для работы большей части приложений;

/bin – минимальный набор программ необходимых для работы в системе;

/sbin – набор программ для административной работы с системой (программы необходимые только суперпользователю);

/home – здесь располагаются домашние каталоги пользователей;

/etc – в данном каталоге обычно хранятся общесистемные конфигурационные файлы для большинства программ в системе;

/etc/rc?.d,/etc/init.d,/etc/rc.boot,/etc/rc.d – директории, где расположены командные файлы системы инициализации SysVinit;

/etc/passwd – база данных пользователей, в которой содержится информация об имени пользователя, его настоящем имени, личном каталоге, закодированный пароль и другие данные;

/etc/shadow – теневая база данных пользователей. При этом информация из файла /etc/passwd перемещается в /etc/shadow, который недоступен по чтению всем, кроме пользователя root. В случае использования альтернативной схемы управления теневыми паролями (TCB) все теневые пароли для каждого пользователя располагаются в директории /etc/tcb/<имя пользователя>/shadow;

/dev – в этом каталоге находятся файлы устройств. Файлы в /dev создаются сервисом udev;

/usr – обычно файловая система /usr достаточно большая по объему, так как все программы установлены именно здесь. Вся информация в каталоге /usr помещается туда во время установки системы. Отдельно устанавливаемые пакеты программ и другие файлы размещаются в каталоге /usr/local. Некоторые подкаталоги системы /usr рассмотрены ниже;

/usr/bin – практически все команды, хотя некоторые находятся в /bin или в /usr/local/bin;

/usr/sbin – команды, используемые при администрировании системы и не предназначенные для размещения в файловой системе root;

/usr/local – здесь рекомендуется размещать файлы, установленные без использования пакетных менеджеров, внутренняя организация каталогов практически такая же, как и корневого каталога;

/usr/man – каталог, где хранятся файлы справочного руководства man;

/usr/share – каталог для размещения общедоступных файлов большей части приложений.

Каталог /var:

/var/log – место, где хранятся файлы аудита работы системы и приложений;

/var/spool – каталог для хранения файлов находящих в очереди на обработку для того или иного процесса (очередь на печать, отправку почты и т. д.);

/tmp – временный каталог необходимый некоторым приложениям;

/proc – файловая система /proc является виртуальной и в действительности она не существует на диске. Ядро создает её в памяти компьютера. Система /proc предоставляет информацию о системе.

10.3.2 Имена дисков и разделов

Все физические устройства вашего компьютера отображаются в каталог /dev файловой системы изделия (об этом – ниже). Диски (в том числе IDE/SATA/SCSI жёсткие диски, USB-диски) имеют имена:

/dev/sda – первый диск;

/dev/sdb – второй диск;

и т. д.

Диски обозначаются /dev/sdX, где X – a,b,c,d,e,... в порядке обнаружения системой.

Раздел диска обозначается числом после его имени. Например, /dev/sdb4 – четвертый раздел второго диска.

10.4 Разделы, необходимые для работы ОС

Для работы ОС необходимо создать на жестком диске (дисках) по крайней мере, два раздела: корневой (то есть тот, который будет содержать каталог /) и раздел подкачки (swap). Размер последнего, как правило, составляет от однократной до двукратной величины оперативной памяти компьютера. Если свободного места на диске много, то можно создать отдельные разделы для каталогов /usr, /home, /var.

10.5 Управление системными сервисами и командами

10.5.1 Сервисы

Сервисы – это программы, которые запускаются и останавливаются через инициализированные скрипты, расположенные в каталоге /etc/init.d. Многие из этих сервисов запускаются на этапе старта ОС «Альт Сервер Виртуализации». В ОС существует шесть системных уровней выполнения, каждый из которых определяет список служб (сервисов), запускаемых на данном уровне. Уровни 0 и 6 соответствуют выключению и перезагрузке системы.

При загрузке системы процесс init запускает все сервисы, указанные в каталоге /etc/rc (0-6).d/ для уровня по умолчанию. Поменять его можно в конфигурационном файле /etc/inittab. Следующая строка соответствует второму уровню выполнения:

```
id:2:initdefault:
```

Для тестирования изменений, внесенных в файл inittab, применяется команда telinit. При указании аргумента -q процесс init повторно читает inittab.

Для перехода ОС «Альт Сервер Виртуализации» на нужный уровень выполнения можно воспользоваться командой init, например:

```
init 3
```

Данная команда переведет систему на третий уровень выполнения, запустив все сервисы, указанные в каталоге /etc/rc3.d/.

10.5.2 Команды

Далее приведены основные команды, использующиеся в ОС «Альт Сервер Виртуализации»:

- ar – создание и работа с библиотечными архивами;
- at – формирование или удаление отложенного задания;
- awk – язык обработки строковых шаблонов;
- batch – планирование команд в очереди загрузки;
- bc – строковый калькулятор;
- chfn – управление информацией учетной записи (имя, описание);
- chsh – управление выбором командного интерпретатора (по умолчанию – для учетной записи);
- cut – разбивка файла на секции, задаваемые контекстными разделителями;
- df – вывод отчета об использовании дискового пространства;
- dmesg – вывод содержимого системного буфера сообщений;
- du – вычисление количества использованного пространства элементов ФС;
- echo – вывод содержимого аргументов на стандартный вывод;
- egrep – поиск в файлах содержимого согласно регулярным выражениям;
- fgrep – поиск в файлах содержимого согласно фиксированным шаблонам;
- file – определение типа файла;
- find – поиск файла по различным признакам в иерархии каталогов;
- gettext – получение строки интернационализации из каталогов перевода;
- grep – вывод строки, содержащей шаблон поиска;
- groupadd – создание новой учетной записи группы;
- groupdel – удаление учетной записи группы;
- groupmod – изменение учетной записи группы;
- groups – вывод списка групп;
- gunzip – распаковка файла;
- gzip – упаковка файла;

- hostname – вывод и задание имени хоста;
- install – копирование файла с установкой атрибутов;
- ipcrm – удаление ресурса IPC;
- ipcs – вывод характеристик ресурса IPC;
- kill – прекращение выполнения процесса;
- killall – удаление процессов по имени;
- lpr – система печати;
- ls – вывод содержимого каталога;
- lsb_release – вывод информации о дистрибутиве;
- m4 – запуск макропроцессора;
- md5sum – генерация и проверка MD5-сообщения;
- mknod – создание файла специального типа;
- mktemp – генерация уникального имени файла;
- more – постраничный вывод содержимого файла;
- mount – монтирование ФС;
- msgfmt – создание объектного файла сообщений из файла сообщений;
- newgrp – смена идентификатора группы;
- nice – изменение приоритета процесса перед его запуском;
- nohup – работа процесса после выхода из системы;
- od – вывод содержимого файла в восьмеричном и других видах;
- passwd – смена пароля учетной записи;
- patch – применение файла описания изменений к оригинальному файлу;
- pidof – вывод идентификатора процесса по его имени;
- ps – вывод информации о процессах;
- renice – изменение уровня приоритета процесса;
- sed – строковый редактор;
- sendmail – транспорт системы электронных сообщений;
- sh – командный интерпретатор;
- shutdown – команда останова системы;
- su – изменение идентификатора запускаемого процесса;
- sync – сброс системных буферов на носители;
- tar – файловый архиватор;
- umount – размонтирование ФС;
- useradd – создание новой учетной записи или обновление существующей;
- userdel – удаление учетной записи и соответствующих файлов окружения;

- usermod – модификация информации об учетной записи;
- w – список пользователей, кто в настоящий момент работает в системе и с какими файлами;
- who – вывод списка пользователей системы.

Узнать об опциях команд можно с помощью команды man.

11 РАБОТА С НАИБОЛЕЕ ЧАСТО ИСПОЛЬЗУЕМЫМИ КОМПОНЕНТАМИ

11.1 Командные оболочки (интерпретаторы)

Для управления ОС используется командные интерпретаторы (shell).

Зайдя в систему, можно увидеть приглашение – строку, содержащую символ «$» (далее, этот символ будет обозначать командную строку). Программа ожидает ввода команд. Роль командного интерпретатора – передавать команды пользователя операционной системе. При помощи командных интерпретаторов можно писать небольшие программы – сценарии (скрипты). В Linux доступны следующие командные оболочки:

bash – самая распространённая оболочка под linux. Она ведет историю команд и предоставляет возможность их редактирования.

pdksh – клон korn shell, хорошо известной оболочки в UNIX(™) системах.

Оболочкой по умолчанию является «Bash» (Bourne Again Shell) Проверить, какая оболочка используется можно, выполнив команду:

```
$ echo $SHELL
```

У каждой оболочки свой синтаксис. Все примеры в дальнейшем построены с использованием оболочки Bash.

11.1.1 Командная оболочка Bash

В Bash имеется несколько приемов для работы со строкой команд. Например, используя клавиатуру, можно:

<Ctrl> + <A> – перейти на начало строки;

<Ctrl> + <U> – удалить текущую строку;

<Ctrl> + <C> – остановить текущую задачу.

Для ввода нескольких команд одной строкой можно использовать разделитель «;». По истории команд можно перемещаться с помощью клавиш <↑> и <↓>. Чтобы найти конкретную команду в списке набранных, не пролистывая всю историю, необходимо набрать <Ctrl> + <R> и начать вводить символы ранее введенной команды.

Для просмотра истории команд можно воспользоваться командой history. Команды, присутствующие в истории, отображаются в списке пронумерованными. Чтобы запустить конкретную команду необходимо набрать:

```
!номер команды
```

Если ввести:

```
!!
```

запустится последняя, из набранных команд.

В Bash имеется возможность самостоятельного завершения имен команд из общего списка команд, что облегчает работу при вводе команд, в случае, если имена программ и команд слишком длинны. При нажатии клавиши <Tab> Bash завершает имя команды, программы или каталога, если не существует нескольких альтернативных вариантов. Например, чтобы использовать программу декомпрессии gunzip, можно набрать следующую команду:

```
$ gu
```

Затем нажать <Tab>. Так как в данном случае существует несколько возможных вариантов завершения команды, то необходимо повторно нажать клавишу <Tab>, чтобы получить список имен, начинающихся с gu.

В предложенном примере можно получить следующий список:

```
$ gu
guile gunzip gupnp-binding-tool
```

Если набрать: n (gunzip – это единственное имя, третьей буквой которого является «n»), а затем нажать клавишу <Tab>, то оболочка самостоятельно дополнит имя. Чтобы запустить команду нужно нажать <Enter>.

Программы, вызываемые из командной строки, Bash ищет в каталогах, определяемых в системной переменной PATH. По умолчанию в этот перечень каталогов не входит текущий каталог, обозначаемый ./ (точка слеш) (если только не выбран один из двух самых слабых уровней защиты). Поэтому, для запуска программы из текущего каталога, необходимо использовать команду (в примере запускается команда prog):

```
./prog
```

11.1.2 Базовые команды оболочки Bash

Все команды, приведенные ниже, могут быть запущены в режиме консоли. Для получения более подробной информации следует использовать команду `man`. Пример:

```
$ man ls
```

Примечание. Параметры команд обычно начинаются с символа «-», и обычно после одного символа «-» можно указать сразу несколько опций. Например, вместо команды `ls -l -F` можно ввести команду `ls -lF`

11.1.2.1 Учетные записи пользователей

Команда su

Команда `su` позволяет изменить «владельца» текущего сеанса (сессии) без необходимости завершать сеанс и открывать новый.

Синтаксис:

```
su [ОПЦИИ...] [ПОЛЬЗОВАТЕЛЬ]
```

Команду можно применять для замены текущего пользователя на любого другого, но чаще всего она используется для получения пользователем прав суперпользователя (root).

При вводе команды `su -` будет запрошен пароль суперпользователя (root), и, в случае ввода корректного пароля, пользователь получит привилегии суперпользователя. Чтобы вернуться к правам пользователя, необходимо ввести команду:

```
exit
```

Команда id

Команда `id` выводит информацию о пользователе и группах, в которых он состоит, для заданного пользователя или о текущем пользователе (если ничего не указано).

Синтаксис:

```
id [параметры] [ПОЛЬЗОВАТЕЛЬ]
```

Команда passwd

Команда `passwd` меняет (или устанавливает) пароль, связанный с входным_именем пользователя.

Обычный пользователь может менять только пароль, связанный с его собственным входным_именем.

Команда запрашивает у обычных пользователей старый пароль (если он был), а затем дважды запрашивает новый. Новый пароль должен соответствовать техническим требованиям к паролям, заданным администратором системы.

11.1.2.2 Основные операции с файлами и каталогами

Команда ls

Команда `ls` (list) выдает список файлов каталога.

Синтаксис:

```
ls [опции...] [файл...]
```

Основные опции:

`-a` – просмотр всех файлов, включая скрытые;

`-l` – отображение более подробной информации;

`-R` – выводить рекурсивно информацию о подкаталогах.

Команда cd

Команда `cd` предназначена для смены каталога. Команда работает как с абсолютными, так и с относительными путями. Если каталог не указан, используется значение переменной окружения `$HOME` (домашний каталог пользователя). Если каталог задан полным маршрутным именем, он становится текущим. По отношению к новому каталогу нужно иметь право на выполнение, которое в данном случае трактуется как разрешение на поиск.

Синтаксис:

```
cd [-L|-P] [каталог]
```

Если в качестве аргумента задано «-», то это эквивалентно `$OLDPWD`. Если переход был осуществлен по переменной окружения `$CDPATH` или в качестве аргумента был задан «-» и смена каталога была успешной, то абсолютный путь нового рабочего каталога будет выведен на стандартный вывод.

Примеры:

- находясь в домашнем каталоге перейти в его подкаталог `docs/` (относительный путь):

```
$ cd docs/
```

- сделать текущим каталог `/usr/bin` (абсолютный путь):

```
$ cd /usr/bin/
```

- сделать текущим родительский каталог:

```
$ cd ..
```

- вернуться в предыдущий каталог:

```
$ cd -
```

- сделать текущим домашний каталог:

```
$ cd
```

Команда pwd

Команда `pwd` выводит абсолютный путь текущего (рабочего) каталога.

Синтаксис:

```
pwd [-L|-P]
```

Опции:

`-P` – не выводить символические ссылки;

`-L` – выводить символические ссылки.

Команда rm

Команда `rm` служит для удаления записей о файлах. Если заданное имя было последней ссылкой на файл, то файл уничтожается.

Синтаксис:

```
rm [опции...] имя_файла
```

Основные опции:

`-f` – не запрашивать подтверждения;

`-i` – запрашивать подтверждение;

`-r`, `-R` – рекурсивно удалять содержимое указанных каталогов.

Пример. Удалить все файлы `html` в каталоге `~/html`:

```
$ rm -i ~/html/*.html
```

Команда mkdir

Команда mkdir позволяет создать каталог.

Синтаксис:

```
mkdir [-p] [-m права] [каталог...]
```

Команда rmdir

Команда rmdir удаляет записи, соответствующие указанным пустым каталогам.

Синтаксис:

```
rmdir [-p] [каталог...]
```

Основные опции:

-p – удалить каталог и его потомки.

Команда rmdir часто заменяется командой rm -rf, которая позволяет удалять каталоги, даже если они не пусты.

Команда cp

Команда cp предназначена для копирования файлов из одного в другие каталоги.

Синтаксис:

```
cp [-fip] [исх_файл] [цел_файл]
cp [-fip] [исх_файл...] [каталог]
cp [-R] [[-H] | [-L] | [-P]] [-fip] [исх_файл...] [каталог]
```

Основные опции:

-p – сохранять по возможности времена изменения и доступа к файлу, владельца и группу, права доступа;

-i – запрашивать подтверждение перед копированием в существующие файлы;

-r, -R – рекурсивно копировать содержимое каталогов.

Команда mv

Команда mv предназначена для перемещения файлов.

Синтаксис:

```
mv [-fi] [исх_файл...] [цел_файл]
mv [-fi] [исх_файл...] [каталог]
```

В первой синтаксической форме, характеризующейся тем, что последний операнд не является ни каталогом, ни символической ссылкой на каталог, mv перемещает исх_файл в цел_файл (происходит переименование файла).

Во второй синтаксической форме mv перемещает исходные файлы в указанный каталог под именами, совпадающими с краткими именами исходных файлов.

Основные опции:

-f – не запрашивать подтверждения перезаписи существующих файлов;

-i – запрашивать подтверждение перезаписи существующих файлов.

Команда cat

Команда `cat` последовательно выводит содержимое файлов.

Синтаксис:

```
cat [опции...] [файл...]
```

Основные опции:

`-n`, `--number` – нумеровать все строки при выводе;

`-E`, `--show-ends` – показывать $ в конце каждой строки.

Если файл не указан, читается стандартный ввод. Если в списке файлов присутствует имя «-», вместо этого файла читается стандартный ввод.

Команда head

Команда `head` выводит первые 10 строк каждого файла на стандартный вывод.

Синтаксис:

```
head [опции...] [файл...]
```

Основные опции:

`-n`, `--lines=[-]K` – вывести первые К строк каждого файла, а не первые 10;

`-q`, `--quiet` – не печатать заголовки с именами файлов.

Команда less

Команда `less` позволяет постранично просматривать текст (для выхода необходимо нажать <q>).

Синтаксис:

```
less имя_файла
```

Команда grep

Команда `grep` имеет много опций и предоставляет возможности поиска символьной строки в файле.

Синтаксис:

```
grep шаблон_поиска файл
```

11.1.2.3 Поиск файлов

Команда find

Команда `find` предназначена для поиска всех файлов, начиная с корневого каталога. Поиск может осуществляться по имени, типу или владельцу файла.

Синтаксис:

```
find [-H] [-L] [-P] [-Оуровень] [-D help|tree|search|stat|rates|opt|
exec] [путь…] [выражение]
```

Ключи для поиска:

`-name` – поиск по имени файла;

`-type` – поиск по типу f=файл, d=каталог, l=ссылка(lnk);

-user – поиск по владельцу (имя или UID).

Когда выполняется команда find, можно выполнять различные действия над найденными файлами. Основные действия:

-exec команда \; – выполнить команду. Запись команды должна заканчиваться экранированной точкой с запятой. Строка «{}» заменяется текущим маршрутным именем файла;

-execdir команда \; – то же самое что и exec, но команда вызывается из подкаталога, содержащего текущий файл;

-ok команда – эквивалентно -exec за исключением того, что перед выполнением команды запрашивается подтверждение (в виде сгенерированной командной строки со знаком вопроса в конце) и она выполняется только при ответе: y;

-print – вывод имени файла на экран.

Путем по умолчанию является текущий подкаталог. Выражение по умолчанию -print.

Примеры:

- найти в текущем каталоге обычные файлы (не каталоги), имя которых начинается с символа «~»:

```
$ find . -type f -name "~*" -print
```

- найти в текущем каталоге файлы, измененные позже, чем файл file.bak:

```
$ find . -newer file.bak -type f -print
```

- удалить все файлы с именами a.out или *.o, доступ к которым не производился в течение недели:

```
$ find / \( -name a.out -o -name '*.o' \) \ -atime +7 -exec rm {} \;
```

- удалить из текущего каталога и его подкаталогов все файлы нулевого размера, запрашивая подтверждение:

```
$ find . -size 0c -ok rm {} \;
```

Команда whereis

Команда whereis сообщает путь к исполняемому файлу программы, ее исходным файлам (если есть) и соответствующим страницам справочного руководства.

Синтаксис:

```
whereis [опции...] имя_файла
```

Опции:

-b – вывод информации только об исполняемых файлах;

-m – вывод информации только о страницах справочного руководства;

-s – вывод информации только об исходных файлах.

11.1.2.4 Мониторинг и управление процессами

Команда ps

Команда `ps` отображает список текущих процессов.

Синтаксис:

```
ps [-aA] [-defl] [-G список] [-o формат...] [-p список] [-t список] [-U список] [-g список] [-n список] [-u список]
```

По умолчанию выводится информация о процессах с теми же действующим UID и управляющим терминалом, что и у подающего команду пользователя.

Основные опции:

`-a` – вывести информацию о процессах, ассоциированных с терминалами;

`-f` – вывести «полный» список;

`-l` – вывести «длинный» список;

`-p список` – вывести информацию о процессах с перечисленными в списке PID;

`-u список` – вывести информацию о процессах с перечисленными идентификаторами или именами пользователей.

Команда kill

Команда `kill` позволяет прекратить исполнение процесса или передать ему сигнал.

Синтаксис:

```
kill [-s] [сигнал] [идентификатор] [...]
kill [-l] [статус_завершения]
kill [-номер_сигнала] [идентификатор] [...]
```

Идентификатор – PID ведущего процесса задания или номер задания, предварённый знаком «%».

Основные опции:

`-l` – вывести список поддерживаемых сигналов;

`-s сигнал`, `-сигнал` – послать сигнал с указанным именем.

Если обычная команда `kill` не дает желательного эффекта, необходимо использовать команду `kill` с параметром `-9`:

```
$ kill -9 PID_номер
```

Команда df

Команда `df` показывает количество доступного дискового пространства в файловой системе, в которой содержится файл, переданный как аргумент. Если ни один файл не указан, показывается доступное место на всех смонтированных файловых системах. Размеры по умолчанию указаны в блоках по 1КБ по умолчанию.

Синтаксис:

```
df [опция...] [файл...]
```

Основные опции:

`-total` – подсчитать общий объем в конце;

`-h`, `--human-readable` – печатать размеры в удобочитаемом формате (например, 1K, 234M, 2G).

Команда du

Команда `du` подсчитывает использование диска каждым файлом, для каталогов подсчет происходит рекурсивно.

Синтаксис:

```
du [опции][файл...]
```

Основные опции:

`-a`, `--all` – выводить общую сумму для каждого заданного файла, а не только для каталогов;

`-c`, `--total` – подсчитать общий объем в конце. Может быть использовано для выяснения суммарного использования дискового пространства для всего списка заданных файлов;

`-d`, `--max-depth=N` – выводить объем для каталога (или файлов, если указано `--all`) только если она на N или менее уровней ниже аргументов командной строки;

`-S`, `--separate-dirs` – выдавать отдельно размер каждого каталога, не включая размеры подкаталогов;

`-s`, `--summarize` – отобразить только сумму для каждого аргумента.

Команда which

Команда `which` отображает полный путь к указанным командам или сценариям.

Синтаксис:

```
which [опции] [--] имя_программы [...]
```

Основные опции:

`-a`, `--all` – выводит все совпавшие исполняемые файлы по содержимому в переменной окружения `$PATH`, а не только первый из них;

`-c`, `--total` – подсчитать общий объем в конце. Может быть использовано для выяснения суммарного использования дискового пространства для всего списка заданных файлов;

`-d`, `--max-depth=N` – выводить объем для каталога (или файлов, если указано `--all`) только если она на N или менее уровней ниже аргументов командной строки;

`-S`, `--separate-dirs` – выдавать отдельно размер каждого каталога, не включая размеры подкаталогов;

`--skip-dot` – пропускает все каталоги из переменной окружения `$PATH`, которые начинаются с точки.

11.1.2.5 Использование многозадачности

ОС «Альт Сервер Виртуализации» – многозадачная система.

Для того чтобы запустить программу в фоновом режиме, необходимо набрать «&» после имени программы. После этого оболочка дает возможность запускать другие приложения.

Так как некоторые программы интерактивны – их запуск в фоновом режиме бессмысленен. Подобные программы просто остановятся, если их запустить в фоновом режиме.

Можно также запускать нескольких независимых сеансов. Для этого в консоли необходимо набрать <Alt> и одну из клавиш, находящихся в интервале от <F1> до <F6>. На экране появится новое приглашение системы, и можно открыть новый сеанс.

Команда bg

Команда `bg` используется для того, чтобы перевести задание на задний план.

Синтаксис:

```
bg [идентификатор ...]
```

Идентификатор – PID ведущего процесса задания или номер задания, предварённый знаком «%».

Команда fg

Команда `fg` позволяет перевести задание на передний план.

Синтаксис:

```
fg [идентификатор ...]
```

Идентификатор – PID ведущего процесса задания или номер задания, предварённый знаком «%».

11.1.2.6 Сжатие и упаковка файлов

Команда tar

Сжатие и упаковка файлов выполняется с помощью команды `tar`, которая преобразует файл или группу файлов в архив без сжатия (tarfile).

Упаковка файлов в архив чаще всего выполняется следующей командой:

```
$ tar -cf [имя создаваемого файла архива] [упаковываемые файлы и/или каталоги]
```

Пример использования команды упаковки архива:

```
$ tar -cf moi_dokumenti.tar Docs project.tex
```

Распаковка содержимого архива в текущий каталог выполняется командой:

```
$ tar -xf [имя файла архива]
```

Для сжатия файлов используются специальные программы сжатия: gzip, bzip2 и 7z.

11.2 Стыкование команд в системе

11.2.1 Стандартный ввод и стандартный вывод

Многие команды системы имеют так называемые стандартный ввод (standard input) и стандартный вывод (standard output), часто сокращаемые до stdin и stdout. Ввод и вывод здесь – это входная и выходная информация для данной команды. Программная оболочка делает так, что стандартным вводом является клавиатура, а стандартным выводом – экран монитора.

Пример с использованием команды `cat`. По умолчанию команда `cat` читает данные из всех файлов, которые указаны в командной строке, и посылает эту информацию непосредственно в стандартный вывод (stdout). Следовательно, команда:

```
$ cat history-final masters-thesis
```

выведет на экран сначала содержимое файла history-final, а затем – файла masters-thesis.

Если имя файла не указано, команда `cat` читает входные данные из stdin и возвращает их в stdout. Пример:

```
$ cat
Hello there.
Hello there.
Bye.
Bye.
<Ctrl>-<D>
```

Каждую строку, вводимую с клавиатуры, команда `cat` немедленно возвращает на экран. При вводе информации со стандартного ввода конец текста сигнализируется вводом специальной комбинации клавиш, как правило, <Ctrl>-<D>. Сокращённое название сигнала конца текста – EOT (end of text).

11.2.2 Перенаправление ввода и вывода

При необходимости можно перенаправить стандартный вывод, используя символ «>» и стандартный ввод, используя символ «<».

Фильтр (filter) – программа, которая читает данные из стандартного ввода, некоторым образом их обрабатывает и результат направляет на стандартный вывод. Когда применяется перенаправление, в качестве стандартного ввода и вывода могут выступать файлы. Как указывалось выше, по умолчанию, stdin и stdout относятся к клавиатуре и к экрану соответственно. Команда `sort` является простым фильтром – она сортирует входные данные и посылает результат на стандартный вывод. Совсем простым фильтром является команда `cat` – она ничего не делает с входными данными, а просто пересылает их на выход.

11.2.3 Использование состыкованных команд

Стыковку команд (pipelines) осуществляет командная оболочка, которая stdout первой команды направляет на stdin второй команды. Для стыковки используется символ «|». Направить stdout команды `ls` на stdin команды `sort`:

```
$ ls | sort -r
notes
masters-thesis
history-final
english-list
```

Вывод списка файлов частями:

```
$ ls /usr/bin | more
```

Пример стыкования нескольких команд. Команда `head` является фильтром следующего свойства: она выводит первые строки из входного потока (в примере на вход будет подан выход от нескольких состыкованных команд). Если необходимо вывести на экран последнее по алфавиту имя файла в текущем каталоге, можно использовать следующую команду:

```
$ ls | sort -r | head -1 notes
```

где команда `head -1` выводит на экран первую строку получаемого ей входного потока строк (в примере поток состоит из данных от команды `ls`), отсортированных в обратном алфавитном порядке.

11.2.4 Не деструктивное перенаправление вывода

Эффект от использования символа «>» для перенаправления вывода файла является деструктивным; то есть, команда

```
$ ls > file-list
```

уничтожит содержимое файла file-list, если этот файл ранее существовал, и создаст на его месте новый файл. Если вместо этого перенаправление будет сделано с помощью символов «>>», то вывод будет приписан в конец указанного файла, при этом исходное содержимое файла не будет уничтожено.

Примечание. Перенаправление ввода и вывода и стыкование команд осуществляется командными оболочками, которые поддерживают использование символов «>», «>>» и «|». Сами команды не способны воспринимать и интерпретировать эти символы.

11.3 Средства управления дискреционными правами доступа

11.3.1 Команда chmod

Команда `chmod` предназначена для изменения прав доступа файлов и каталогов.

Синтаксис:

`chmod [ОПЦИЯ]... РЕЖИМ[,РЕЖИМ]... [`*`Файл`*`...]`

`chmod [ОПЦИЯ]... --reference=ИФАЙЛ ФАЙЛ...`

Основные опции:

`-R` – рекурсивно изменять режим доступа к файлам, расположенным в указанных каталогах;

`--reference=ИФАЙЛ` – использовать режим файла ИФАЙЛ.

Команда `chmod` изменяет права доступа каждого указанного файла в соответствии с правами доступа, указанными в параметре режим, который может быть представлен как в символьном виде, так и в виде восьмеричного, представляющего битовую маску новых прав доступа.

Формат символьного режима следующий:

`[ugoa...][[+-=][разрешения...]...]`

Здесь разрешения – это ноль или более букв из набора «rwxXst» или одна из букв из набора «ugo».

Каждый аргумент – это список символьных команд изменения прав доступа, разделеных запятыми. Каждая такая команда начинается с нуля или более букв «ugoa», комбинация которых указывает, чьи права доступа к файлу будут изменены: пользователя, владеющего файлом (u), пользователей, входящих в группу, к которой принадлежит файл (g), остальных пользователей (o) или всех пользователей (a). Если не задана ни одна буква, то автоматически будет использована буква «a», но биты, установленные в umask, не будут затронуты.

Оператор «+» добавляет выбранные права доступа к уже имеющимся у каждого файла, «-» удаляет эти права, «=» присваивает только эти права каждому указанному файлу.

Буквы «rwxXst» задают биты доступа для пользователей: «r» – чтение, «w» – запись, «x» – выполнение (или поиск для каталогов), «X» – выполнение/поиск, только если это каталог или же файл с уже установленным битом выполнения, «s» – задать ID пользователя и группы при выполнении, «t» – запрет удаления.

Числовой режим состоит из не более четырех восьмеричных цифр (от нуля до семи), которые складываются из битовых масок с разрядами «4», «2» и «1». Любые пропущенные разряды дополняются лидирующими нулями:

- первый разряд выбирает установку идентификатора пользователя (setuid) (4) или идентификатора группы (setgid) (2) или sticky-бита (1);
- второй разряд выбирает права доступа для пользователя, владеющего данным файлом: чтение (4), запись (2) и исполнение (1);
- третий разряд выбирает права доступа для пользователей, входящих в данную группу, с тем же смыслом, что и у второго разряда;

- четвертый разряд выбирает права доступа для остальных пользователей (не входящих в данную группу), опять с тем же смыслом.

Примеры:

- установить права, позволяющие владельцу читать и писать в файл `f1`, а членам группы и прочим пользователям только читать. Команду можно записать двумя способами:

```
$ chmod 644 f1
$ chmod u=rw,go=r f1
```

- позволить всем выполнять файл `f2`:

```
$ chmod +x f2
```

- запретить удаление файла `f3`:

```
$ chmod+t f3
```

- дать всем права на чтение запись и выполнение, а также на переустановку идентификатора группы при выполнении файла `f4`:

```
$ chmod =rwx,g+s f4
$ chmod 2777 f4
```

11.3.2 Команда chown

Команда `chown` изменяет владельца и/или группу для каждого заданного файла.

Синтаксис:

```
chown [КЛЮЧ]…[ВЛАДЕЛЕЦ][:[ГРУППА]] ФАЙЛ
```

Изменить владельца может только владелец файла или суперпользователь. Владелец не изменяется, если он не задан в аргументе. Группа также не изменяется, если не задана, но если после символьного ВЛАДЕЛЬЦА стоит символ «:», подразумевается изменение группы на основную группу текущего пользователя. Поля ВЛАДЕЛЕЦ и ГРУППА могут быть как числовыми, так и символьными.

Примеры:

- поменять владельца `/u` на пользователя `test`:

```
$ chown test /u
```

- поменять владельца и группу `/u`:

```
$ chown test:staff /u
```

- поменять владельца `/u` и вложенных файлов на `test`:

```
$ chown -hR test /u
```

11.3.3 Команда chgrp

Команда `chgrp` изменяет группу для каждого заданного файла.

Синтаксис:

```
chgrp [ОПЦИИ] ГРУППА ФАЙЛ
```

Основные опции:

`-R` – рекурсивно изменять файлы и каталоги;

`--reference=ИФАЙЛ` – использовать группу файла ИФАЙЛ.

11.3.4 Команда umask

Команда `umask` задает маску режима создания файла в текущей среде командного интерпретатора равной значению, задаваемому операндом режим. Эта маска влияет на начальное значение битов прав доступа всех создаваемых далее файлов.

Синтаксис:

```
umask [-p] [-S] [режим]
```

Пользовательской маске режима создания файлов присваивается указанное восьмеричное значение. Три восьмеричные цифры соответствуют правам на чтение/запись/выполнение для владельца, членов группы и прочих пользователей соответственно. Значение каждой заданной в маске цифры вычитается из соответствующей «цифры», определенной системой при создании файла. Например, umask 022 удаляет права на запись для членов группы и прочих пользователей (у файлов, создававшихся с режимом 777, он оказывается равным 755; а режим 666 преобразуется в 644).

Если маска не указана, выдается ее текущее значение:

```
$ umask
0022
```

или то же самое в символьном режиме:

```
$ umask -S
u=rwx,g=rx,o=rx
```

Команда `umask` распознается и выполняется командным интерпретатором bash.

11.3.5 Команда chattr

Команда `chattr` изменяет атрибуты файлов на файловых системах ext3, ext4.

Синтаксис:

```
chattr [ -RVf ] [+-=aAcCdDeFijmPsStTux] [ -v версия ] <ФАЙЛЫ> …
```

Основные опции:

`-R` – рекурсивно изменять атрибуты каталогов и их содержимого. Символические ссылки игнорируются;

`-V` – выводит расширенную информацию и версию программы;

`-f` – подавлять вывод сообщений об ошибках;

`-v версия` – установить номер версии/генерации файла.

Формат символьного режима:

```
+-=aAcCdDeFijmPsStTux
```

Оператор «+» означает добавление выбранных атрибутов к существующим атрибутам; «-» означает их снятие; «=» означает определение только этих указанных атрибутов для файлов.

Символы «aAcCdDeFijmPsStTux» указывают на новые атрибуты файлов:

- a – только добавление к файлу;
- A – не обновлять время последнего доступа (atime) к файлу;
- c – сжатый файл;
- C – отключение режима «Copy-on-write» для указанного файла;
- d – не архивировать (отключает создание архивной копии файла командой `dump`);
- D – синхронное обновление каталогов;
- e – включает использование extent при выделении места на устройстве (атрибут не может быть отключён с помощью `chattr`);
- F – регистронезависимый поиск в каталогах;
- i – неизменяемый файл (файл защищен от изменений: не может быть удалён или переименован, к этому файлу не могут быть созданы ссылки, и никакие данные не могут быть записаны в этот файл);
- j – ведение журнала данных (данные файла перед записью будут записаны в журнал ext3/ ext4);
- m – не сжимать;
- P – каталог с вложенными файлами является иерархической структурой проекта;
- s – безопасное удаление (перед удалением все содержимое файла полностью затирается «00»);
- S – синхронное обновление (аналогичен опции монтирования «sync» файловой системы);
- t – отключает метод tail-merging для файлов;
- T – вершина иерархии каталогов;
- u – неудаляемый (при удалении файла его содержимое сохраняется, это позволяет пользователю восстановить файл);

x – прямой доступ к файлам (атрибут не может быть установлен с помощью `chattr`).

11.3.6 Команда lsattr

Команда `lsattr` выводит атрибуты файла расширенной файловой системы.

Синтаксис:

```
lsattr [ -RVadlpv ] <ФАЙЛЫ> …
```

Опции:

`-R` – рекурсивно изменять атрибуты каталогов и их содержимого. Символические ссылки игнорируются;

`-V` – выводит расширенную информацию и версию программы;

`-a` – просматривает все файлы в каталоге, включая скрытые файлы (имена которых начинаются с «.»);

`-d` – отображает каталоги так же, как и файлы вместо того, чтобы просматривать их содержимое;

`-l` – отображает параметры, используя длинные имена вместо одного символа;

`-p` – выводит номер проекта файла;

`-v` – выводит номер версии/генерации файла.

11.3.7 Команда getfacl

Команда `getfacl` выводит атрибуты файла расширенной файловой системы.

Синтаксис:

```
getfacl [ --aceEsRLPtpndvh ] <ФАЙЛ> …
```

Опции:

`-a` – вывести только ACL файла;

`-d` – вывести только ACL по умолчанию;

`-c` – не показывать заголовок (имя файла);

`-e` – показывать все эффективные права;

`-E` – не показывать эффективные права;

`-s` – пропускать файлы, имеющие только основные записи;

`-R` – для подкаталогов рекурсивно;

`-L` – следовать по символическим ссылкам, даже если они не указаны в командной строке;

`-P` – не следовать по символическим ссылкам, даже если они указаны в командной строке;

`-t` – использовать табулированный формат вывода;

`-p` – не удалять ведущие «/» из пути файла;

`-n` – показывать числовые значения пользователя/группы.

Формат вывода:

```
1: # file: somedir/
2: # owner: lisa
3: # group: staff
4: # flags: -s-
```

```
5: user::rwx
6: user:joe:rwx                 #effective:r-x
7: group::rwx             #effective:r-x
8: group:cool:r-x
9: mask:r-x
10: other:r-x
11: default:user::rwx
12: default:user:joe:rwx #effective:r-x
13: default:group::r-x
14: default:mask:r-x
15: default:oter:---
```

Строки 1 – 3 указывают имя файла, владельца и группу владельцев.

В строке 4 указаны биты setuid (s), setgid (s) и sticky (t): либо буква, обозначающая бит, либо тире (-). Эта строка включается, если какой-либо из этих битов установлен, и опускается в противном случае, поэтому она не будет отображаться для большинства файлов.

Строки 5, 7 и 10 относятся к традиционным битам прав доступа к файлу, соответственно, для владельца, группы-владельца и всех остальных. Эти три элемента являются базовыми. Строки 6 и 8 являются элементами для отдельных пользователя и группы. Строка 9 – маска эффективных прав. Этот элемент ограничивает эффективные права, предоставляемые всем группам и отдельным пользователям. Маска не влияет на права для владельца файла и всех других. Строки 11 – 15 показывают ACL по умолчанию, ассоциированный с данным каталогом.

11.3.8 Команда setfacl

Команда `setfacl` изменяет ACL к файлам или каталогам. В командной строке за последовательностью команд идет последовательность файлов (за которой, в свою очередь, также может идти последовательность команд и так далее).

Синтаксис:

```
setfacl [-bkndRLPvh] [{-m|-x} acl_spec] [{-M|-X} acl_file] <ФАЙЛ> …
setfacl --restore=file
```

Опции:

`-b` – удалить все разрешенные записи ACL;

`-k` – удалить ACL по умолчанию;

`-n` – не пересчитывать маску эффективных прав, обычно `setfacl` пересчитывает маску (кроме случая явного задания маски) для того, чтобы включить ее в максимальный набор прав доступа элементов, на которые воздействует маска (для всех групп и отдельных пользователей);

`-d` – применить ACL по умолчанию;

`-R` – для подкаталогов рекурсивно;

`-L` – переходить по символическим ссылкам на каталоги (имеет смысл только в сочетании с опцией `-R`);

`-P` – не переходить по символическим ссылкам на каталоги (имеет смысл только в сочетании с опцией `-R`);

`-L` – следовать по символическим ссылкам, даже если они не указаны в командной строке;

`-P` – не следовать по символическим ссылкам, даже если они указаны в командной строке;

`--mask` – пересчитать маску эффективных прав;

`-m` – изменить текущий ACL для файла;

`-M` – прочитать записи ACL для модификации из файла;

`-x` – удалить записи из ACL файла;

`-X` – прочитать записи ACL для удаления из файла;

`--restore=file` – восстановить резервную копию прав доступа, созданную командой `getfacl -R` или ей подобной. Все права доступа дерева каталогов восстанавливаются, используя этот механизм. В случае если вводимые данные содержат элементы для владельца или группы-владельца, и команда `setfacl` выполняется пользователем с именем root, то владелец и группа-владелец всех файлов также восстанавливаются. Эта опция не может использоваться совместно с другими опциями за исключением опции `--test`;

`--set=acl` – установить ACL для файла, заменив текущий ACL;

`--set-file=file` – прочитать записи ACL для установления из файла;

`--test` – режим тестирования (ACL не изменяются).

При использовании опций `--set`, `-m` и `-x` должны быть перечислены записи ACL в командной строке. Элементы ACL разделяются одинарными кавычками.

При чтении ACL из файла при помощи опций `--set-file`, `-M` и `-X` команда `setfacl` принимает множество элементов в формате вывода команды `getfacl`. В строке обычно содержится не больше одного элемента ACL.

Команда `setfacl` использует следующие форматы элементов ACL:

- права доступа отдельного пользователя (если не задан UID, то права доступа владельца файла):

```
[d[efault]:] [u[ser]:]uid [:perms]
```

- права доступа отдельной группы (если не задан GID, то права доступа группы-владельца):

```
[d[efault]:] g[roup]:gid [:perms]
```

- маска эффективных прав:

`[d[efault]:] m[ask][:] [:perms]`

- права доступа всех остальных:

`[d[efault]:] o[ther][:] [:perms]`

Элемент ACL является абсолютным, если он содержит поле `perms` и является относительным, если он включает один из модификаторов: «+» или «^». Абсолютные элементы могут использоваться в операциях установки или модификации ACL. Относительные элементы могут использоваться только в операции модификации ACL. Права доступа для отдельных пользователей, группы, не содержащие никаких полей после значений UID, GID (поле `perms` при этом отсутствует), используются только для удаления элементов.

Значения UID и GID задаются именем или числом. Поле `perms` может быть представлено комбинацией символов «r», «w», «x», «-» или цифр (0 – 7).

Изначально файлы и каталоги содержат только три базовых элемента ACL: для владельца, группы-владельца и всех остальных пользователей. Существует ряд правил, которые следует учитывать при установке прав доступа:

- не могут быть удалены сразу три базовых элемента, должен присутствовать хотя бы один;
- если ACL содержит права доступа для отдельного пользователя или группы, то ACL также должен содержать маску эффективных прав;
- если ACL содержит какие-либо элементы ACL по умолчанию, то в последнем должны также присутствовать три базовых элемента (т. е. права доступа по умолчанию для владельца, группы-владельца и всех остальных);
- если ACL по умолчанию содержит права доступа для всех отдельных пользователей или групп, то в ACL также должна присутствовать маска эффективных прав.

Для того чтобы помочь пользователю выполнять эти правила, команда `setfacl` создает права доступа, используя уже существующие, согласно следующим условиям:

- если права доступа для отдельного пользователя или группы добавлены в ACL, а маски прав не существует, то создается маска с правами доступа группы-владельца;
- если создан элемент ACL по умолчанию, а трех базовых элементов не было, тогда делается их копия и они добавляются в ACL по умолчанию;
- если ACL по умолчанию содержит какие-либо права доступа для конкретных пользователя или группы и не содержит маску прав доступа по умолчанию, то при создании эта маска будет иметь те же права, что и группа по умолчанию.

Пример. Изменить разрешения для файла `test.txt`, принадлежащего пользователю liza и группе docs, так, чтобы:

- пользователь ivan имел права на чтение и запись в этот файл;

- пользователь misha не имел никаких прав на этот файл.

Исходные данные:

```
$ ls -l test.txt
-rw-r--r-- 1 liza docs 8 янв 22 15:54 test.txt
$ getfacl test.txt
# file: test.txt
# owner: liza
# group: docs
user::rw-
group::r--
other::r--
```

Установить разрешения (от пользователя liza):

```
$ setfacl -m u:ivan:rw- test.txt
$ setfacl -m u:misha:--- test.txt
```

Просмотреть разрешения (от пользователя liza):

```
$ getfacl test.txt
# file: test.txt
# owner: liza
# group: docs
user::rw-
user:ivan:rw-
user:misha:---
group::r--
mask::rw-
other::r—
```

Примечание. Символ «+» (плюс) после прав доступа в выводе команды ls -l указывает на использование ACL:

```
$ ls -l test.txt
-rw-rw-r--+ 1 liza docs 8 янв 22 15:54 test.txt
```

11.4 Управление пользователями

11.4.1 Общая информация

Пользователи и группы внутри системы обозначаются цифровыми идентификаторами – UID и GID, соответственно.

Пользователь может входить в одну или несколько групп. По умолчанию он входит в группу, совпадающую с его именем. Чтобы узнать, в какие еще группы входит пользователь, можно использовать команду `id`, вывод её может быть примерно следующим:

```
uid=500(test) gid=500(test) группы=500(test),16(rpm)
```

Такая запись означает, что пользователь test (цифровой идентификатор 500) входит в группы test и rpm. Разные группы могут иметь разный уровень доступа к тем или иным каталогам; чем в большее количество групп входит пользователь, тем больше прав он имеет в системе.

Примечание. В связи с тем, что большинство привилегированных системных утилит в дистрибутивах «Альт» имеют не SUID-, а SGID-бит, необходимо быть предельно внимательным и осторожным в переназначении групповых прав на системные каталоги.

11.4.2 Команда useradd

Команда `useradd` регистрирует нового пользователя или изменяет информацию по умолчанию о новых пользователях.

Синтаксис:

```
useradd [ОПЦИИ...] <ИМЯ ПОЛЬЗОВАТЕЛЯ>
useradd -D [ОПЦИИ...
```

Возможные опции:

`-b каталог` – базовый каталог для домашнего каталога новой учётной записи;

`-c комментарий` – текстовая строка (обычно используется для указания фамилии и мени);

`-d каталог` – домашний каталог новой учётной записи;

`-D` – показать или изменить настройки по умолчанию для useradd;

`-e дата` – дата устаревания новой учётной записи;

`-g группа` – имя или ID первичной группы новой учётной записи;

`-G группы` – список дополнительных групп (через запятую) новой учётной записи;

`-m` – создать домашний каталог пользователя;

`-M` – не создавать домашний каталог пользователя;

`-p пароль` – зашифрованный пароль новой учётной записи (не рекомендуется);

`-s оболочка` – регистрационная оболочка новой учётной записи (по умолчанию /bin/bash);

`-u UID` – пользовательский ID новой учётной записи.

Команда `useradd` имеет множество параметров, которые позволяют менять её поведение по умолчанию. Например, можно принудительно указать, какой будет UID или какой группе будет принадлежать пользователь:

```
# useradd -u 1500 -G usershares new_user
```

11.4.3 Команда passwd

Команда `passwd` поддерживает традиционные опции passwd и утилит shadow.

Синтаксис:

`passwd [ОПЦИИ...] [ИМЯ ПОЛЬЗОВАТЕЛЯ]`

Возможные опции:

`-d`, `--delete` – удалить пароль для указанной записи;

`-f`, `--force` – форсировать операцию;

`-k`, `--keep-tokens` – сохранить не устаревшие пароли;

`-l`, `--lock` – блокировать указанную запись;

`--stdin` – прочитать новые пароли из стандартного ввода;

`-S`, `--status` – дать отчет о статусе пароля в указанной записи;

`-u`, `--unlock` – разблокировать указанную запись;

`-?`, `--help` – показать справку и выйти;

`--usage` – дать короткую справку по использованию;

`-V`, `--version` – показать версию программы и выйти.

Код выхода: при успешном завершении `passwd` заканчивает работу с кодом выхода 0. Код выхода 1 означает, что произошла ошибка. Текстовое описание ошибки выводится на стандартный поток ошибок.

Пользователь может в любой момент поменять свой пароль. Единственное, что требуется для смены пароля – знать текущий пароль.

Только суперпользователь может обновить пароль другого пользователя.

11.4.4 Добавление нового пользователя

Для добавления нового пользователя используйте команды `useradd` и `passwd`:

```
# useradd test1

# passwd test1
passwd: updating all authentication tokens for user test1.

You can now choose the new password or passphrase.

A valid password should be a mix of upper and lower case letters, digits, and
other characters.  You can use a password containing at least 7 characters
from all of these classes, or a password containing at least 8 characters
from just 3 of these 4 classes.
An upper case letter that begins the password and a digit that ends it do not
count towards the number of character classes used.

A passphrase should be of at least 3 words, 11 to 72 characters long, and
contain enough different characters.
```

```
Alternatively, if no one else can see your terminal now, you can pick this as
your password: "Burst*texas$Flow".

Enter new password:
Weak password: too short.
Re-type new password:
passwd: all authentication tokens updated successfully.
```

В результате описанных действий в системе появился пользователь test1 с некоторым паролем. Если пароль оказался слишком слабым с точки зрения системы, она об этом предупредит (как в примере выше). Пользователь в дальнейшем может поменять свой пароль при помощи команды `passwd` – но если он попытается поставить слабый пароль, система откажет ему (в отличие от *root*) в изменении.

В ОС «Альт Сервер Виртуализации» для проверки паролей на слабость используется модуль PAM `passwdqc`.

11.4.5 Настройка парольных ограничений

Настройка парольных ограничений производится в файле `/etc/passwdqc.conf`.

Файл `passwdqc.conf` состоит из 0 или более строк следующего формата:

`опция=значение`

Пустые строки и строки, начинающиеся со знака решетка («#»), игнорируются. Символы пробела между опцией и значением не допускаются.

Опции, которые могут быть переданы в модуль (в скобках указаны значения по умолчанию): min=N0,N1,N2,N3,N4 (min=disabled,24,11,8,7) – минимально допустимая длина пароля.

Используемые типы паролей по классам символов (алфавит, число, спецсимвол, верхний и нижний регистр) определяются следующим образом:

- тип N0 используется для паролей, состоящих из символов только одного класса;
- тип N1 используется для паролей, состоящих из символов двух классов;
- тип N2 используется для парольных фраз, кроме этого требования длины, парольная фраза должна также состоять из достаточного количества слов;
- типы N3 и N4 используются для паролей, состоящих из символов трех и четырех классов, соответственно.

Ключевое слово disabled используется для запрета паролей выбранного типа N0 – N4 независимо от их длины.

Примечание. Каждое следующее число в настройке «min» должно быть не больше, чем предыдущее.

При расчете количества классов символов, заглавные буквы, используемые в качестве первого символа и цифр, используемых в качестве последнего символа пароля, не учитываются.

max=N (max=40) – максимально допустимая длина пароля. Эта опция может быть использована для того, чтобы запретить пользователям устанавливать пароли, которые могут быть слишком длинными для некоторых системных служб. Значение 8 обрабатывается особым образом: пароли длиннее 8 символов, не отклоняются, а обрезаются до 8 символов для проверки надежности (пользователь при этом предупреждается).

passphrase=N (passphrase=3) – число слов, необходимых для ключевой фразы (значение 0 отключает поддержку парольных фраз).

match=N (match=4) – длина общей подстроки, необходимой для вывода, что пароль хотя бы частично основан на информации, найденной в символьной строке (значение 0 отключает поиск подстроки). Если найдена слабая подстрока пароль не будет отклонен; вместо этого он будет подвергаться обычным требованиям к прочности при удалении слабой подстроки. Поиск подстроки нечувствителен к регистру и может обнаружить и удалить общую подстроку, написанную в обратном направлении.

similar=permit|deny (similar=deny) – параметр similar=permit разрешает задать новый пароль, если он похож на старый (параметр similar=deny – запрещает). Пароли считаются похожими, если есть достаточно длинная общая подстрока, и при этом новый пароль с частично удаленной подстрокой будет слабым.

random=N[,only] (random=42) – размер случайно сгенерированных парольных фраз в битах (от 26 до 81) или 0, чтобы отключить эту функцию. Любая парольная фраза, которая содержит предложенную случайно сгенерированную строку, будет разрешена вне зависимости от других возможных ограничений. Значение only используется для запрета выбранных пользователем паролей.

enforce=none|users|everyone (enforce=users) – параметр enforce=users задает ограничение задания паролей в passwd на пользователей без полномочий root. Параметр enforce=everyone задает ограничение задания паролей в passwd и на пользователей, и на суперпользователя root. При значении none модуль PAM будет только предупреждать о слабых паролях.

retry=N (retry=3) – количество запросов нового пароля, если пользователь с первого раза не сможет ввести достаточно надежный пароль и повторить его ввод.

Далее приводится пример задания следующих значений в файле /etc/passwdqc.conf:

```
min=8,7,4,4,4
enforce=everyone
```

В указанном примере пользователям, включая суперпользователя root, будет невозможно задать пароли:

- типа N0 (символы одного класса) – длиной меньше восьми символов;
- типа N1 (символы двух классов) – длиной меньше семи символов;
- типа N2 (парольные фразы), типа N3 (символы трех классов) и N4 (символы четырех классов) – длиной меньше четырех символов.

11.4.6 Управление сроком действия пароля

Для управления сроком действия паролей используется команда `chage`.

Примечание. Должен быть установлен пакет `shadow-change`:

```
# apt-get install shadow-change
```

`chage` изменяет количество дней между сменой пароля и датой последнего изменения пароля.

Синтаксис команды:

```
chage [опции] логин
```

Основные опции:

`-d, --lastday LAST_DAY` – установить последний день смены пароля в LAST_DAY (число дней с 1 января 1970). Дата также может быть указана в формате ГГГГ-ММ-ДД;

`-E, --expiredate EXPIRE_DAYS` – установить дату окончания действия учётной записи в EXPIRE_DAYS (число дней с 1 января 1970). Дата также может быть указана в формате ГГГГ-ММ-ДД. Значение `-1` удаляет дату окончания действия учётной записи;

`-I, --inactive INACTIVE` – используется для задания количества дней «неактивности», то есть дней, когда пользователь вообще не входил в систему, после которых его учетная запись будет заблокирована. Пользователь, чья учетная запись заблокирована, должен обратиться к системному администратору, прежде чем снова сможет использовать систему. Значение `-1` отключает этот режим;

`-l,--list` – просмотр информации о «возрасте» учётной записи пользователя;

`-m, --mindays MIN_DAYS` – установить минимальное число дней перед сменой пароля. Значение 0 в этом поле обозначает, что пользователь может изменять свой пароль, когда угодно;

`-M, --maxdays MAX_DAYS` – установить максимальное число дней перед сменой пароля. Когда сумма MAX_DAYS и LAST_DAY меньше, чем текущий день, у пользователя будет запрошен новый пароль до начала работы в системе. Эта операция может предваряться предупреждением (параметр `-W`). При установке значения `-1`, проверка действительности пароля не будет выполняться;

`-W, --warndays WARN_DAYS` – установить число дней до истечения срока действия пароля, начиная с которых пользователю будет выдаваться предупреждение о необходимости смены пароля.

Пример настройки времени действия пароля для пользователя test:

```
# chage -M 5 test
```

Получить информацию о «возрасте» учётной записи пользователя test:

```
# chage -l test
Последний раз пароль был изменён                    : дек 27, 2023
Срок действия пароля истекает                       : янв 01, 2024
Пароль будет деактивирован через                    : янв 11, 2024
Срок действия учётной записи истекает                     : никогда
Минимальное количество дней между сменой пароля     : -1
Максимальное количество дней между сменой пароля          : 5
Количество дней с предупреждением перед деактивацией пароля      : -1
```

Примечание. Задать время действия пароля для вновь создаваемых пользователей можно, изменив параметр PASS_MAX_DAYS в файле /etc/login.defs.

11.4.7 Настройка неповторяемости пароля

Для настройки неповторяемости паролей используется модуль pam_pwhistory, который сохраняет последние пароли каждого пользователя и не позволяет пользователю при смене пароля чередовать один и тот же пароль слишком часто.

Примечание. В данном случае системный каталог станет доступным для записи пользователям группы `pw_users` (следует создать эту группу и включить в неё пользователей).

Примечание. База используемых паролей ведется в файле `/etc/security/opasswd`, в который пользователи должны иметь доступ на чтение и запись. При этом они могут читать хэши паролей остальных пользователей. Не рекомендуется использовать на многопользовательских системах.

Создайте файл `/etc/security/opasswd` и дайте права на запись пользователям:

```
# install -Dm0660 -gpw_users /dev/null /etc/security/opasswd
# chgrp pw_users /etc/security
# chmod g+w /etc/security
```

Для настройки этого ограничения необходимо изменить файл `/etc/pam.d/system-auth-local-only` таким образом, чтобы он включал модуль `pam_pwhistory` после первого появления строки с паролем:

```
password        required        pam_passwdqc.so config=/etc/passwdqc.conf
password        required        pam_pwhistory.so debug use_authtok remember=10
retry=3
```

После добавления этой строки в файле `/etc/security/opasswd` будут храниться последние 10 паролей пользователя (содержит хэши паролей всех учетных записей пользователей) и при попытке использования пароля из этого списка будет выведена ошибка:

```
Password has been already used. Choose another.
```

В случае если необходимо, чтобы проверка выполнялась и для суперпользователя root, в настройки нужно добавить параметр enforce_for_root:

```
password        required        pam_pwhistory.so
use_authtok enforce_for_root remember=10 retry=3
```

11.4.8 Модификация пользовательских записей

Для модификации пользовательских записей применяется утилита `usermod`:

```
# usermod -G audio,rpm,test1 test1
```

Такая команда изменит список групп, в которые входит пользователь test1 – теперь это audio, rpm, test1.

```
# usermod -l test2 test1
```

Будет произведена смена имени пользователя с test1 на test2.

Команды `usermod -L test2` и `usermod -U test2` соответственно временно блокируют возможность входа в систему пользователю test2 и возвращают всё на свои места.

Изменения вступят в силу только при следующем входе пользователя в систему.

При неинтерактивной смене или задании паролей для целой группы пользователей используется команда `chpasswd`. На стандартный вход ей следует подавать список, каждая строка которого будет выглядеть как `имя:пароль`.

11.4.9 Удаление пользователей

Для удаления пользователей используется команда `userdel`.

Команда `userdel test2` удалит пользователя test2 из системы. Если будет дополнительно задан параметр `-r`, то будет уничтожен и домашний каталог пользователя. Нельзя удалить пользователя, если в данный момент он еще работает в системе.

11.5 Режим суперпользователя

11.5.1 Какие бывают пользователи?

Linux – система многопользовательская, а потому пользователь – ключевое понятие для организации всей системы доступа в Linux. Файлы всех пользователей в Linux хранятся раздельно, у каждого пользователя есть собственный домашний каталог, в котором он может хранить свои данные. Доступ других пользователей к домашнему каталогу пользователя может быть ограничен.

Суперпользователь в Linux – это выделенный пользователь системы, на которого не распространяются ограничения прав доступа. Именно суперпользователь имеет возможность произвольно изменять владельца и группу файла. Ему открыт доступ на чтение и запись к любому файлу или каталогу системы.

Среди учётных записей Linux всегда есть учётная запись суперпользователя – root. Поэтому вместо «суперпользователь» часто говорят «root». Множество системных файлов принадлежат

root, множество файлов только ему доступны для чтения или записи. Пароль этой учётной записи – одна из самых больших драгоценностей системы. Именно с её помощью системные администраторы выполняют самую ответственную работу.

11.5.2 Для чего может понадобиться режим суперпользователя?

Системные утилиты, например, такие, как ЦУС или «Программа управления пакетами Synaptic» требуют для своей работы привилегий суперпользователя, потому что они вносят изменения в системные файлы. При их запуске выводится диалоговое окно с запросом пароля системного администратора.

11.5.3 Как получить права суперпользователя?

Для опытных пользователей, умеющих работать с командной строкой, существует два различных способа получить права суперпользователя.

Первый – это зарегистрироваться в системе под именем root.

Второй способ – воспользоваться специальной утилитой `su` (shell of user), которая позволяет выполнить одну или несколько команд от лица другого пользователя. По умолчанию эта утилита выполняет команду sh от пользователя root, то есть запускает командный интерпретатор. Отличие от предыдущего способа в том, что всегда известно, кто именно запускал `su`, а значит, ясно, кто выполнил определённое административное действие.

В некоторых случаях удобнее использовать не `su`, а утилиту `sudo`, которая позволяет выполнять только заранее заданные команды.

Примечание. Для того чтобы воспользоваться командами `su` и `sudo`, необходимо быть членом группы `wheel`. Пользователь, созданный при установке системы, по умолчанию уже включён в эту группу.

В дистрибутивах «Альт» для управления доступом к важным службам используется подсистема `control`. `control` – механизм переключения между неким набором фиксированных состояний для задач, допускающих такой набор.

Команда `control` доступна только для суперпользователя (root). Для того чтобы посмотреть, что означает та или иная политика `control` (разрешения выполнения конкретной команды, управляемой `control`), надо запустить команду с ключом `help`:

```
# control su help
```

Запустив `control` без параметров, можно увидеть полный список команд, управляемых командой (facilities) вместе с их текущим состоянием и набором допустимых состояний.

11.5.4 Как перейти в режим суперпользователя?

Для перехода в режим суперпользователя наберите в терминале команду (**минус важен!**):

```
su -
```

Если воспользоваться командой `su` без ключа, то происходит вызов командного интерпретатора с правами root. При этом значение переменных окружения, в частности $PATH, остаётся таким же, как у пользователя: в переменной $PATH не окажется каталогов `/sbin`, `/usr/sbin`, без указания полного имени будут недоступны команды `route`, `shutdown`, `mkswap` и другие. Более того, переменная $HOME будет указывать на каталог пользователя, все программы, запущенные в режиме суперпользователя, сохранят свои настройки с правами root в каталоге пользователя, что в дальнейшем может вызвать проблемы.

Чтобы избежать этого, следует использовать `su -`. В этом режиме `su` запустит командный интерпретатор в качестве login shell, и он будет вести себя в точности так, как если бы в системе зарегистрировался root.

12 ОБЩИЕ ПРАВИЛА ЭКСПЛУАТАЦИИ

12.1 Включение компьютера

Для включения компьютера необходимо:

- включить стабилизатор напряжения, если компьютер подключен через стабилизатор напряжения;
- включить принтер, если он нужен;
- включить монитор компьютера, если он не подключен к системному блоку кабелем питания;
- включить компьютер (переключателем на корпусе компьютера либо клавишей с клавиатуры).

После этого на экране компьютера появятся сообщения о ходе работы программ проверки и начальной загрузки компьютера.

12.2 Выключение компьютера

Для выключения компьютера надо:

- закончить работающие программы;
- выбрать функцию завершения работы и выключения компьютера, после чего ОС самостоятельно выключит компьютер, имеющий системный блок формата ATX;
- выключить компьютер (переключателем на корпусе AT системного блока);
- выключить принтер;
- выключить монитор компьютера (если питание монитора не от системного блока);
- выключить стабилизатор, если компьютер подключен через стабилизатор напряжения.