Альт Виртуализация

ДОКУМЕНТАЦИЯ ALT LINUX TEAM

Альт Виртуализация редакци 11.0

Документация

Руководство пользователя

Редакция январь, 2026

Елена Мишина
ООО «Базальт СПО»
lepata@basealt.ru

Аннотация

Добро пожаловать в документацию дистрибутива Альт Виртуализация редакция ONE. Данное руководство предназначено как для начинающих, так и для опытных пользователей. Руководство описывает подготовку системы для установки, процесс установки дистрибутива, а также процедуру настройки и использования системы.

Данное руководство соответствует текущему состоянию сведений, но какие-либо окончательные правки могли не попасть в него. В случае обнаружения ошибок и неточностей в руководство вносятся изменения.

Часть I. Что такое Альт Виртуализация редакция ONE?

В этой части рассматривается что такое Linux и Альт Виртуализация редакция ONE.

Содержание

Глава 1. Что такое Альт Виртуализация редакция ONE

Операционная система Альт Виртуализация редакция ONE — это комплексное решение для управления виртуализированной инфраструктурой и частными облаками на базе открытого программного обеспечения. Платформа построена на системе управления облаками OpenNebula и предназначена для централизованного развёртывания и эксплуатации виртуальных машин в корпоративных средах.

Решение объединяет технологии аппаратной виртуализации KVM, гибкие механизмы управления вычислительными ресурсами, сетью и хранилищами, а также предоставляет удобные средства администрирования через веб-интерфейс, командную строку и API.

Основные возможности OpenNebula:

- Виртуализация на базе KVM — запуск и управление полноценными виртуальными машинами с поддержкой современных процессоров и аппаратного ускорения;
- Централизованное управление инфраструктурой — объединение физических узлов в единый пул ресурсов с распределением нагрузок;
- Высокая доступность (HA) — автоматический перезапуск виртуальных машин при отказе узла (при соответствующей конфигурации кластера);
- Интеграция с хранилищами — поддержка локальных и сетевых хранилищ, включая LVM, NFS, iSCSI, Ceph, а также файловые хранилища;
- Гибкая сетевая настройка — мосты, VLAN, VXLAN, виртуальные сети;
- Резервное копирование и восстановление — средства создания резервных копий виртуальных машин и их последующего восстановления;
- Многоарендность и ролевая модель — разграничение прав между пользователями и проектами.

В OpenNebula доступны несколько способов управления виртуальной инфраструктурой, обеспечивающих гибкость администрирования и интеграцию с внешними системами:

- Веб-интерфейс OpenNebula (Sunstone) — графический интерфейс для управления виртуальными машинами, сетями, хранилищами и пользователями. Позволяет выполнять основные операции: создание, запуск, остановку, миграцию и удаление ВМ, а также мониторинг состояния ресурсов;
- Командная строка (CLI) — управление с помощью утилит OpenNebula (onevm, onehost, oneimage, onetemplate и др.) и стандартных средств Linux;
- REST API — программный доступ к функциям платформы через HTTP-запросы. Используется для автоматизации и интеграции с внешними системами (Ansible, Terraform, системы мониторинга).

Альт Виртуализация редакция ONE подходит для построения частных облаков, виртуальных серверов приложений, тестовых и учебных сред, а также для централизованного управления виртуальной инфраструктурой.

1.1. Системные требования

Минимальные требования к узлу виртуализации (без учёта ресурсов виртуальных машин):

- Оперативная память — не менее 1 ГБ (рекомендуется от 2 ГБ);
- Процессор — 1 CPU с поддержкой аппаратной виртуализации;
- Дисковое пространство — от 10 ГБ.

1.2. Ограничения виртуализации

Примечание

Ограничения определяются гипервизором (KVM/QEMU) и аппаратной платформой, а не OpenNebula.

- Лимиты для узла виртуализации:
  - CPU — 8192 логических ядер. На двух-процессорном сервере, с включенным HT, смогут работать (8192/2/2=) два 2048 ядерных процессора (x86_64);
  - RAM — 32 ТБ;
  - HDD — нет явного ограничения;
  - NIC — нет ограничений.
- Лимиты для виртуальных машин:
  - CPU — 240;
  - RAM — 4 ТБ. Поддерживает максимальную память которую вы можете выделить ВМ. 32-х разрядные ВМ с поддержкой расширения физических адресов (PAE) могут получить доступ только к 64 ГБ — это ограничение виртуального оборудования;

- HDD — нет ограничений (зависит от файловой системы);
- NIC — 32;
- PCI — 16.

Глава 2. Что такое Linux

2.1. Свободные программы

Операционная система (далее — ОС) Linux — ядро, основные компоненты системы и большинство её пользовательских приложений — свободные программы. Свободные программы можно:

- запускать на любом количестве компьютеров;
- распространять бесплатно или за деньги без каких-либо ограничений;
- получать исходные тексты этих программ и вносить в них любые изменения.

Свобода программ обеспечила их широкое использование и интерес к ним со стороны тысяч разработчиков. Основные программы для Linux выходят под лицензией GNU General Public License (далее — GPL). Лицензия GNU не только гарантирует свободу, но и защищает её. Она допускает дальнейшее распространение программ только под той же лицензией, поэтому исходный код ядра Linux, компиляторов, библиотеки glibc, пользовательских графических оболочек не может быть использован для создания приложений с закрытым кодом. В этом принципиальное отличие Linux от свободных ОС семейства BSD (FreeBSD, NetBSD, OpenBSD), фрагменты которых вошли в Microsoft Windows и даже стали основой OS X. Linux включает в себя многие разработки BSD, но его компиляторы и системные библиотеки разработаны в рамках проекта GNU (http://www.gnu.org/home.ru.html).

2.2. Разработка Linux

В отличие от распространённых несвободных ОС, Linux не имеет географического центра разработки. Нет фирмы, которая владела бы этой ОС, нет и единого координационного центра. Программы для Linux — результат работы тысяч проектов. Большинство из них объединяет программистов из разных стран, связанных друг с другом только перепиской. Лишь некоторые проекты централизованы и сосредоточены в фирмах. Создать свой проект или присоединиться к уже существующему может любой программист, и, в случае успеха,

результаты этой работы станут известны миллионам пользователей. Пользователи принимают участие в тестировании свободных программ, общаются с разработчиками напрямую. Это позволяет за короткий срок добавлять в программное обеспечение новые возможности, оперативно находить ошибки и исправлять их.

Именно гибкая и динамичная система разработки, невозможная для проектов с закрытым кодом, определяет исключительную экономическую эффективность Linux. Низкая стоимость свободных разработок, отлаженные механизмы тестирования и распространения, привлечение независимых специалистов, обладающих индивидуальным, самостоятельным видением проблем, защита исходного текста программ лицензией GPL — всё это стало причиной успеха свободных программ.

Такая высокая эффективность разработки не могла не заинтересовать крупные фирмы. Они стали создавать свои свободные проекты, основывающиеся на тех же принципах. Так появились Mozilla, LibreOffice, свободный клон Interbase, SAP DB. IBM способствовала переносу Linux на свои мейнфреймы.

Открытый код программ значительно снизил себестоимость разработки закрытых систем для Linux и позволил снизить цену решения для пользователя. Вот почему Linux стала платформой, часто рекомендуемой для таких продуктов, как Oracle, DB2, Informix, Sybase, SAP ERP, Lotus Domino.

2.3. Защищённость

ОС Linux унаследовала от UNIX надёжность и отличную систему защиты. Система разграничения доступа к файлам позволяет не бояться вирусов. Но всё же, программ без ошибок не бывает, и Linux не исключение. Благодаря открытости исходного кода программ, аудит системы может осуществить любой специалист без подписок о неразглашении и без необходимости работы в стенах нанявшей его компании. Сообщества разработчиков и пользователей свободных программ создали множество механизмов оповещения об ошибках и их исправления. Сообщить об ошибке и принять участие в её исправлении независимому программисту или пользователю так же просто, как специалисту фирмы-разработчика или автору проекта. Благодаря этому ошибки защиты эффективно выявляются и быстро исправляются.

2.4. Дистрибутивы Linux

Большинство пользователей для установки Linux используют дистрибутивы. Дистрибутив — это не просто набор программ, а готовое решение для выполнения различных задач пользователя, обладающее идентичностью установки, управления, обновления, а также едиными системами настройки и поддержки.

2.5. Новичку

Linux — самостоятельная операционная система. Все операционные системы разные: Linux — не Windows, не OS X и не FreeBSD. В Linux свои правила, их необходимо изучить и к ним необходимо привыкнуть. Терпение и настойчивость в изучении Linux обернётся значительным повышением эффективности и безопасности вашей работы. То, что сегодня кажется странным и непривычным, завтра понравится и станет нормой.

Не стесняйтесь задавать вопросы, ведь самый простой способ найти ответ — совет опытного специалиста. Взаимопомощь и общение — традиция в мире Linux. Всегда можно обратиться за помощью к сообществу пользователей и разработчиков Linux. Большинство вопросов повторяются, поэтому для начала стоит поискать ответ на свой вопрос в документации, затем в сети Интернет. Если вы не нашли ответа в перечисленных источниках, не стесняйтесь, пишите на форум или в списки рассылки так, как писали бы своим друзьям, и вам обязательно помогут.

Глава 3. Что такое системы Альт

3.1. ALT Linux Team

Команда ALT Linux (https://www.altlinux.org/ALT_Linux_Team) — это интернациональное сообщество, насчитывающее более 300 разработчиков свободного программного обеспечения.

3.2. Сизиф

Sisyphus (https://packages.altlinux.org) — наш ежедневно обновляемый банк программ (часто называемый репозиторием). Поддерживаемая ALT Linux Team целостность Sisyphus, оригинальная технология сборки программ и утилита **`apt-get`** позволяют пользователям легко обновлять свои системы и быть в курсе актуальных новинок мира свободных программ.

Ежедневно изменяющийся репозиторий содержит самое новое программное обеспечение со всеми его преимуществами и недостатками (иногда ещё неизвестными). Поэтому, перед обновлением вашей системы из Sisyphus, мы советуем взвесить преимущества новых возможностей, реализованных в последних версиях программ, и вероятность возникновения неожиданностей в работе с ними (https://www.altlinux.org/Sisyphus_changes).

Разработка Sisyphus полностью открыта. У нас нет секретных изменений кода и закрытого тестирования с подписками о неразглашении. Всё, что мы сделали сегодня, завтра вы найдёте в сети. По сравнению с другими аналогичными банками программ (Debian unstable, Mandriva Cooker, PLD, Fedora), в Sisyphus есть немало самобытного. Особое внимание уделяется защите системы, локализации на русский язык, полноте и корректности зависимостей.

Название Sisyphus (Сизиф) заимствовано из греческой мифологии. С кропотливым Сизифом, непрерывно закатывающим в гору камни, команду ALT Linux Team объединяет постоянная работа над усовершенствованием технологий, заложенных в репозиторий.

Sisyphus, в первую очередь, — открытая лаборатория решений. Если вам это интересно, если вы хотите дополнить Sisyphus новыми решениями, если вы считаете, что можете собрать какую-то программу лучше — присоединяйтесь к проекту ALT Linux Team (https://www.altlinux.org/Join).

3.3. Что такое одиннадцатая платформа

Как уже говорилось ранее, Sisyphus — это часто обновляемый репозиторий, ориентированный прежде всего для разработчиков. Для пользователей, которым стабильность и предсказуемость работы системы важнее расширенной функциональности (а это в первую очередь начинающие и корпоративные пользователи), предназначены дистрибутивы Альт. Такие дистрибутивы базируются на стабильном срезе репозитория Sisyphus. Эти срезы называются платформами.

Одиннадцатая платформа (p11) была создана в июне 2024 года и её поддержка продлится до июля 2027 года.

3.3.1. Основные новшества одиннадцатой платформы

- Одиннадцатая платформа основана на ядре Linux 6.12 (LTS) с расширенной поддержкой современного оборудования: процессорных архитектур Intel, включая Intel Meteor Lake, Intel Xeon Sapphire Rapids, AMD Ryzen 7000 (Zen 4) и EPYC Genoa; аппаратных интерфейсов — PCI Express Gen5, USB4, Thunderbolt 4, Wi-Fi 6/6E, NVMe 1.4/2.0; улучшенной поддержкой виртуализации;
- Программное обеспечение на одиннадцатой платформе использует обновленный OpenSSL 3.1. Платформа сохраняет поддержку OpenSSL 1.1 для совместимости с устаревшим ПО;
- Произошел переход на Python 3.12;
- Добавлены PHP 8.3 и 8.4;
- Системный интерпретатор сценариев /bin/sh теперь основан на Bash 5.2;
- Обновлены основные системные библиотеки и компиляторы: glibc 2.38, компилятор GCC 13 и LLVM/Clang 19;
- Пакет systemd обновлён до версии 255;
- Подсистема начальной загрузки установщика propagator заменена на altboot;
- Основным фрэймворком приложений графической подсистемы стал Qt6, с поддержкой Qt5 для обратной совместимости приложений. Qt6 в качестве основного стека также использует системный установщик;
- Включена поддержка нового Kerberos 1.21, полностью совместимого с Samba 4.20+. В дистрибутивах 11 платформы также доступны и ключевые изменения Samba 4.20+;
- Существенно обновлён Альтератор в качестве Центра Управления Системой — новый интерфейс, взаимодействие с D-Bus, модульная архитектура. Новый Альтератор поддерживает модули предыдущих версий;
- ALT Diagnostic Tool — графическая утилита диагностики ОС. ADT использует заранее подготовленный набор проверок, предоставляет возможность пользователю выполнить тесты без дополнительных привилегий и единый вид отчета по проверкам;
- Копидел — средство тиражирования установленной системы (*alterator-kopidel*);
- Платформа доступна для архитектур x86_64 и ARM64.

Часть II. Установка дистрибутива

В этой части рассматривается процесс установки дистрибутива.

Содержание

Глава 4. Подготовка установочного диска

Наиболее частый способ установки операционной системы на компьютер представляет собой установку с установочного DVD-диска. В этой главе описываются различные способы записи дистрибутива на DVD-диск.

Установочные образы являются гибридными, что позволяет производить установку, записав такой образ на USB Flash. О записи установочного образа на USB Flash также рассказано в этой главе.

4.1. Запись ISO-образа дистрибутива на DVD

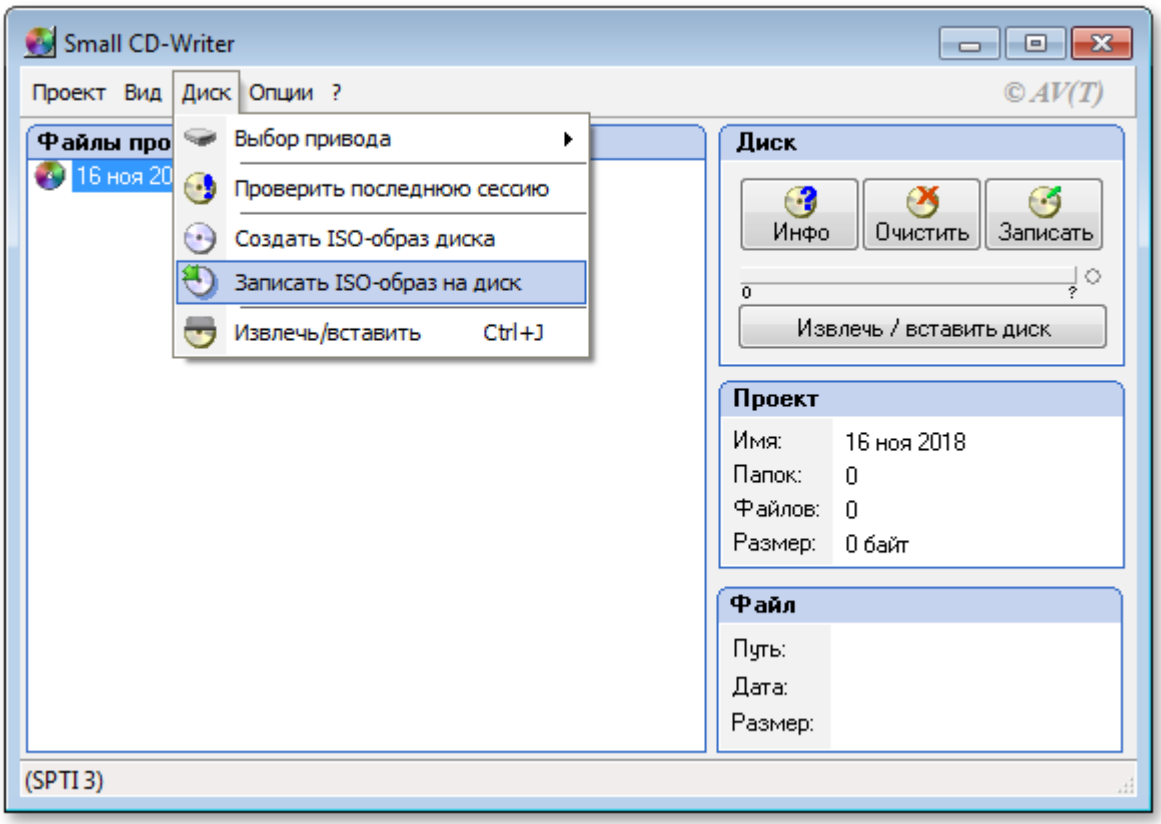
4.1.1. Запись образа диска под операционной системой MS Windows

Файл ISO-образа диска — это файл специального формата, подготовленный для записи на диск. Для записи ISO-образа под операционной системой MS Windows используйте специальные программы: SCDWriter, Nero BurningROM и другие. Рекомендуем для записи использовать новые диски от известных производителей, таких как: Verbatim, TDK. Записанный на плохой диск образ может вызвать неразрешимые проблемы при установке.

4.1.1.1. Запись образа диска с помощью Small CD-Writer

Весь процесс записи установочного диска при помощи **Small CD-Writer** состоит из следующих шагов:

- скачать образ дистрибутива;
- скачать архив программы Small CD-Writer http://gluek.info/wiki/_media/software/scdwriter14.zip;
- распаковать файлы программы из архива в любой каталог;
- вставить чистый диск в привод;
- войти в распакованный каталог и запустить программу **`SCDWriter.exe`**;
- открыть пункт меню **Диск** → **Записать ISO-образ на диск** и, в появившемся окне, указать путь к образу диска;
- нажать кнопку **`Записать`**.



4.1.1.2. Запись образа диска с помощью Nero BurningROM

Процесс записи установочного диска при помощи **Nero BurningROM** состоит из следующих шагов:

- скачать образ дистрибутива;
- скачать программу **Nero BurningROM** с сайта производителя http://www.nero.com и установить её;
- запустить программу и выбрать в списке устройств необходимый для записи CD/DVD дисковод;
- нажать кнопку **Открыть** в главном окне. В появившемся окне выбрать необходимый ISO-образ для записи и нажать кнопку **Открыть**;
- в окне **Записать проект** на вкладке **Запись** установить отметку в поле **Запись** и настроить необходимые параметры прожига;
- записать ISO-образ на диск, щёлкнув по кнопке **Прожиг**.

4.1.2. Запись образа диска под операционной системой Linux

Для записи ISO-образов можно использовать множество утилит и программ с графическим или текстовым интерфейсом. Наиболее удобно использовать программы **K3b** или **Brasero**, которые поставляются в комплекте любого дистрибутива операционной системы Linux.

4.1.2.1. Запись образа диска с помощью K3b

Весь процесс записи установочного диска при помощи **K3b** состоит из следующих шагов:

- если программа **k3b** отсутствует, необходимо установить её в систему, используя стандартные для вашего дистрибутива инструменты установки программ;
- запустить программу **k3b**. При правильных настройках программа сообщит об отсутствии проблем с системой и предложит перейти к записи на диск;
- в меню главного окна **Сервис (Service)** выбрать пункт **Записать образ DVD (Burn DVD image)**;
- в появившемся окне **Записать образ DVD (Burn DVD image)** нажать на кнопку **Выбор файла для записи**. Откроется диалог, в котором необходимо выбрать ISO-образ для записи и после выбора нажать кнопку **OK**;
- программа **k3b** покажет информацию о ISO-файле и начнёт вычислять контрольную сумму. Эта операция может занять несколько минут. Полученную контрольную сумму можно сравнить с MD5SUM суммой на странице дистрибутива;
- если контрольные суммы не совпадают, значит, для записи был выбран не тот файл или скачанный ISO-образ был испорчен во время передачи данных по сети;
- если контрольные суммы совпадают, вставить диск для записи в дисковод. Дождаться активации кнопки **Начать (Start)**;
- нажать на кнопку **Начать (Start)**.

4.2. Запись установочного образа на USB Flash

Предупреждение

Запись образа дистрибутива на flash-диск приведёт к изменению таблицы разделов на носителе, таким образом, если flash-диск выполнил функцию загрузочного\установочного устройства и требуется вернуть ему функцию переносного накопителя данных, то необходимо удалить все имеющиеся разделы на flash-диске и создать нужное их количество заново.

Для восстановления совместимости flash-диска с операционными системами семейства Windows может понадобиться также пересоздание таблицы разделов (например, при помощи parted). Нужно удалить таблицу GPT и создать таблицу типа msdos. Кроме того, должен быть только один раздел с FAT или NTFS.

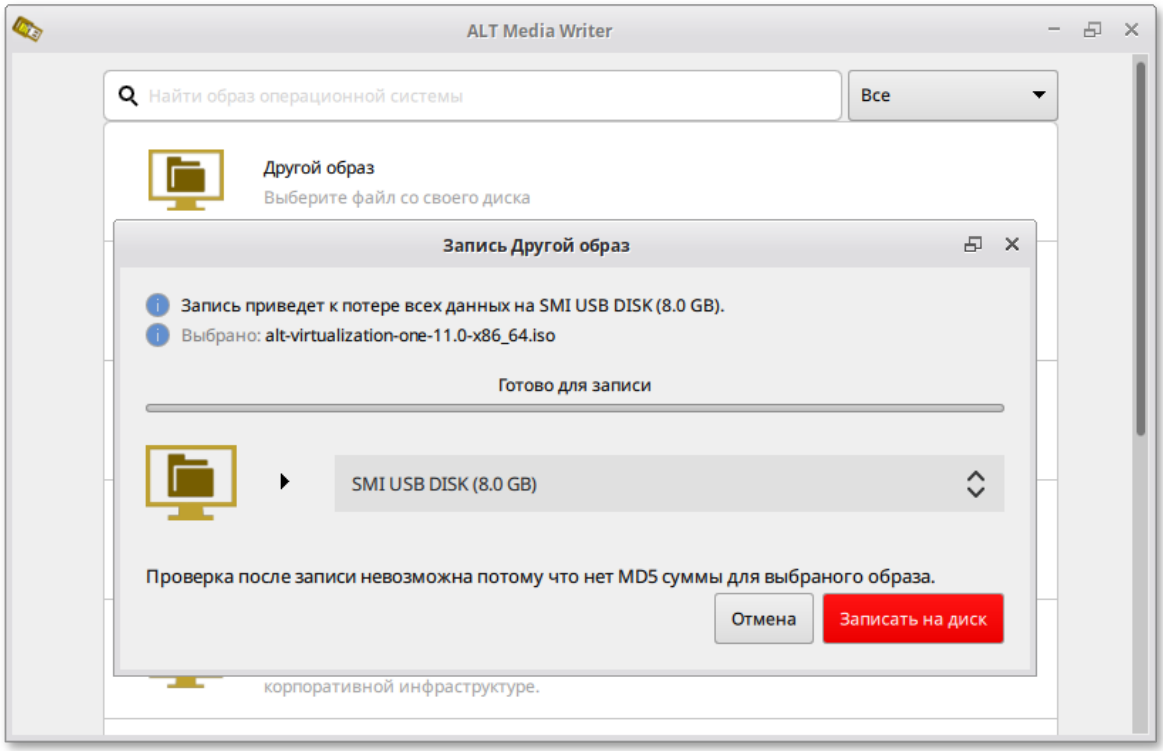
Для создания загрузочного flash-диска понадобится файл ISO-образа установочного диска с дистрибутивом. Установочные ISO-образы являются гибридными (Hybrid ISO/IMG), что позволяет производить установку, записав такой образ на flash-накопитель.

4.2.1. В операционной системе Windows

Для создания загрузочного flash-диска под операционной системой MS Windows используйте специальные программы: ALT Media Writer, Win32 Disk Imager, HDD Raw Copy Tool и другие.

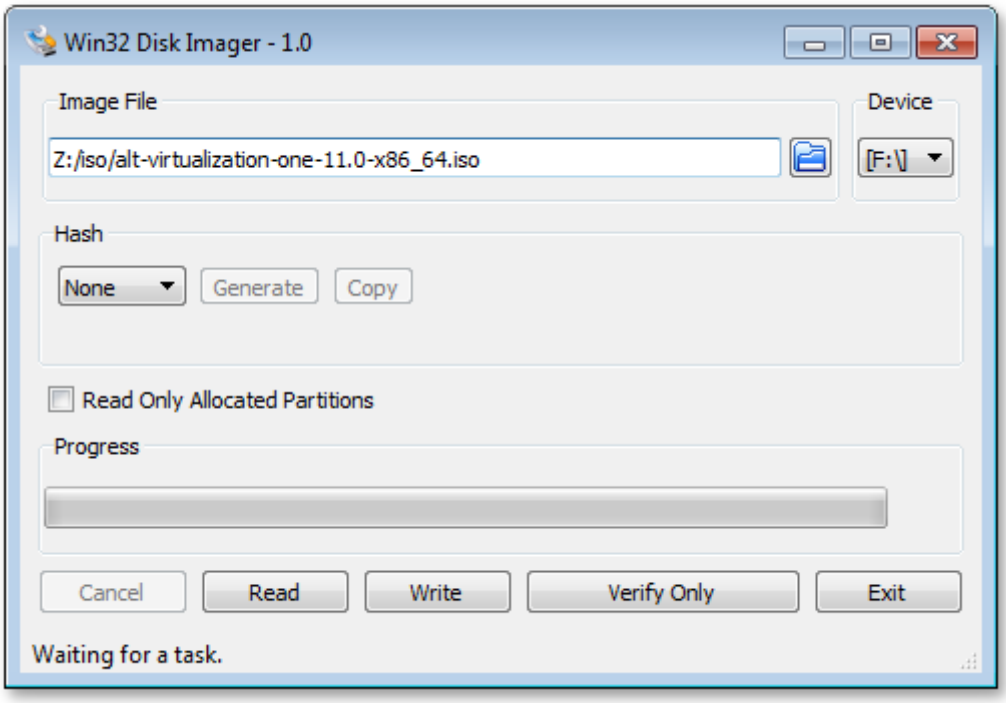
ALT Media Writer — это инструмент, который помогает записывать образы ALT на портативные накопители, такие как flash-диски. Он может автоматически загружать образы из интернета и записывать их. Для записи образа на flash-диск необходимо:

- скачать и установить **ALT Media Writer**;
- вставить flash-диск в USB-разъем;
- запустить **ALT Media Writer**;
- выбрать дистрибутив и нажать кнопку **`Создать Live USB…`**, начнётся загрузка образа из интернета, также можно выбрать локальный ISO-файл, нажав кнопку **`Другой образ`**;
- выбрать устройство (flash-диск);
- после окончания загрузки нажать кнопку **`Записать на диск`** (если был отмечен пункт **`Записать образ после загрузки`**, запись образа начнётся автоматически).



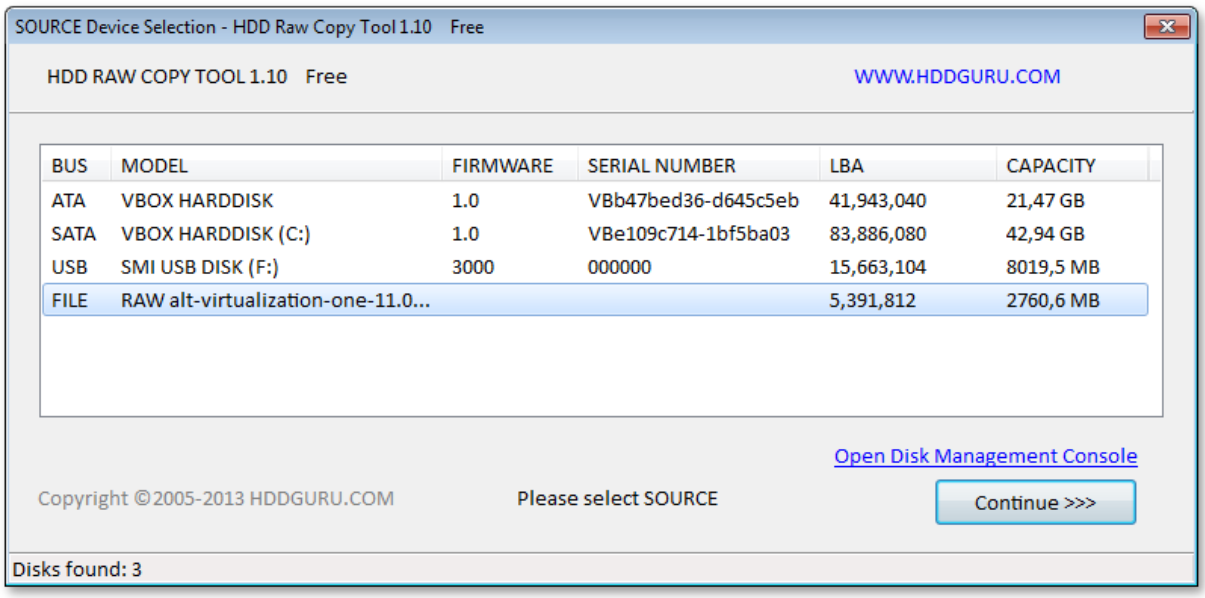
Инструкция для записи образа в программе **Win32 Disk Imager**:

- скачать и установить программу Win32 Disk Imager;
- скачать образ дистрибутива;
- вставить flash-диск в USB-разъем (размер flash-диска должен быть не меньше размера скачанного образа диска);
- запустить **Win32 Disk Imager**;
- в появившемся окне выбрать ISO-образ дистрибутива, выбрать устройство (flash-диск):

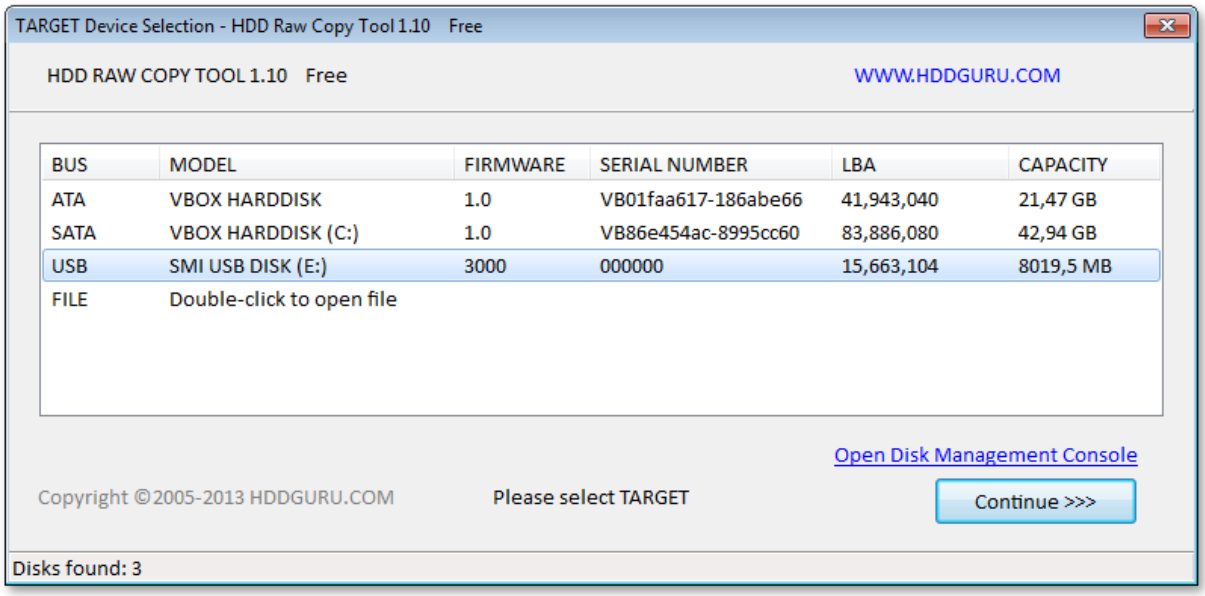


- нажать кнопку **`Write`** для записи образа на flash-диск.

Для записи образа на flash-диск подойдёт и утилита HDD Raw Copy Tool. На первом шаге нужно выбрать файл с образом диска:



На втором шаге нужно выбрать flash-диск, на который будет записан образ:



Предупреждение

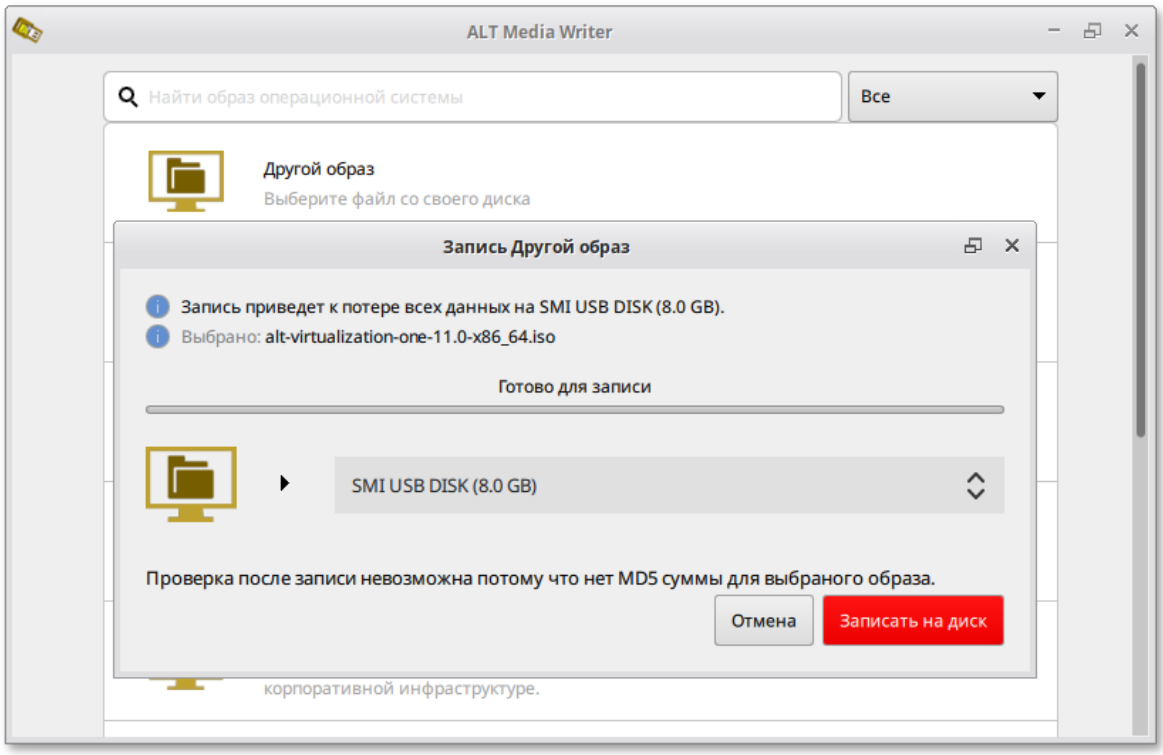
Будьте внимательны при указании имени USB-устройства — запись образа по ошибке на свой жёсткий диск приведёт к почти гарантированной потере данных на нём!

После проверки правильности выбранных параметров и нажатия кнопки **`Continue`** можно приступать к записи, нажав кнопку **`START`**. По успешному завершению записи окно с индикацией процесса записи закроется, после чего можно закрыть и окно самой программы.

4.2.2. В операционной системе Linux

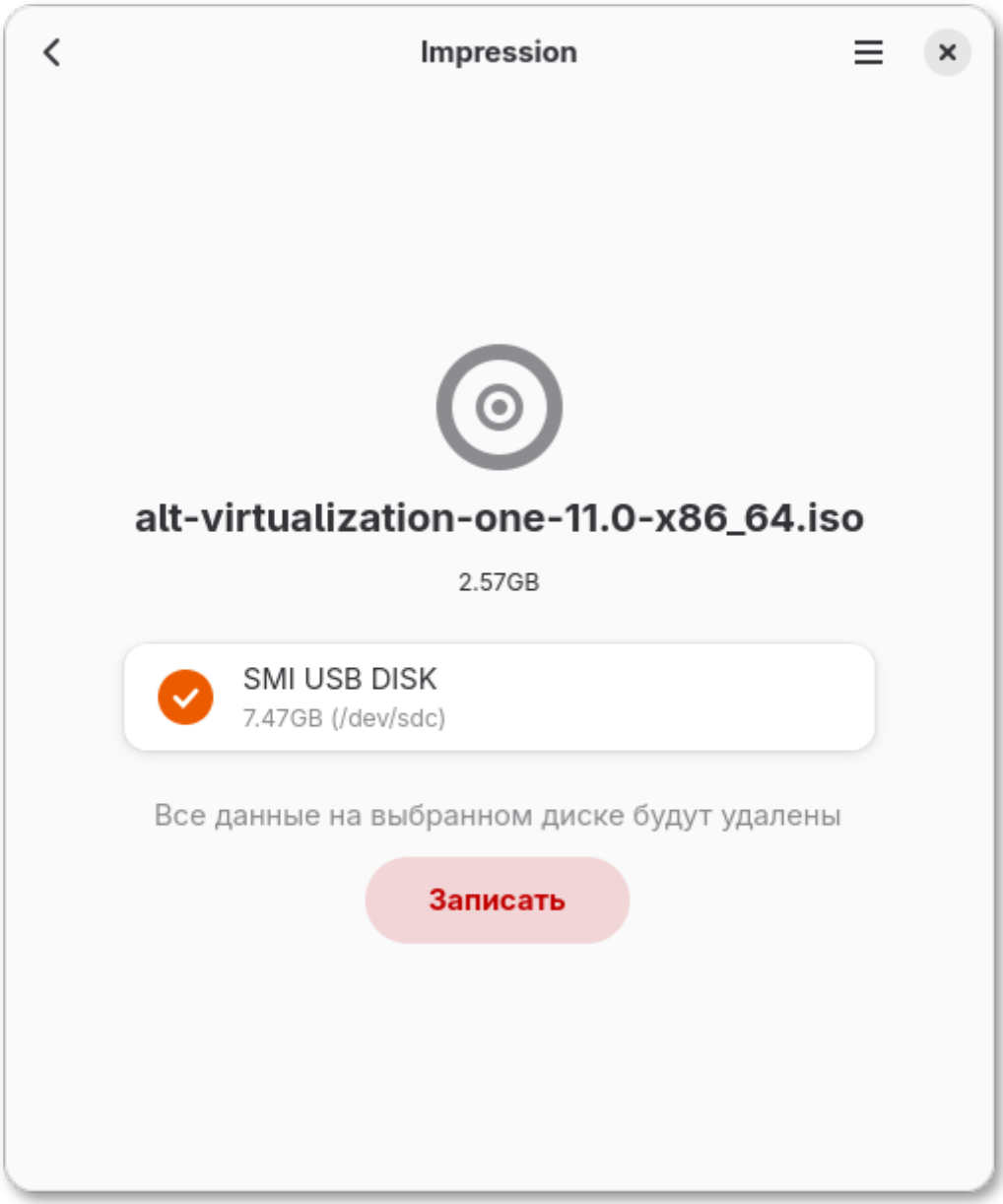
Для записи образа на flash-диск можно воспользоваться одной из программ с графическим интерфейсом:

- ALT Media Writer (*altmediawriter*):

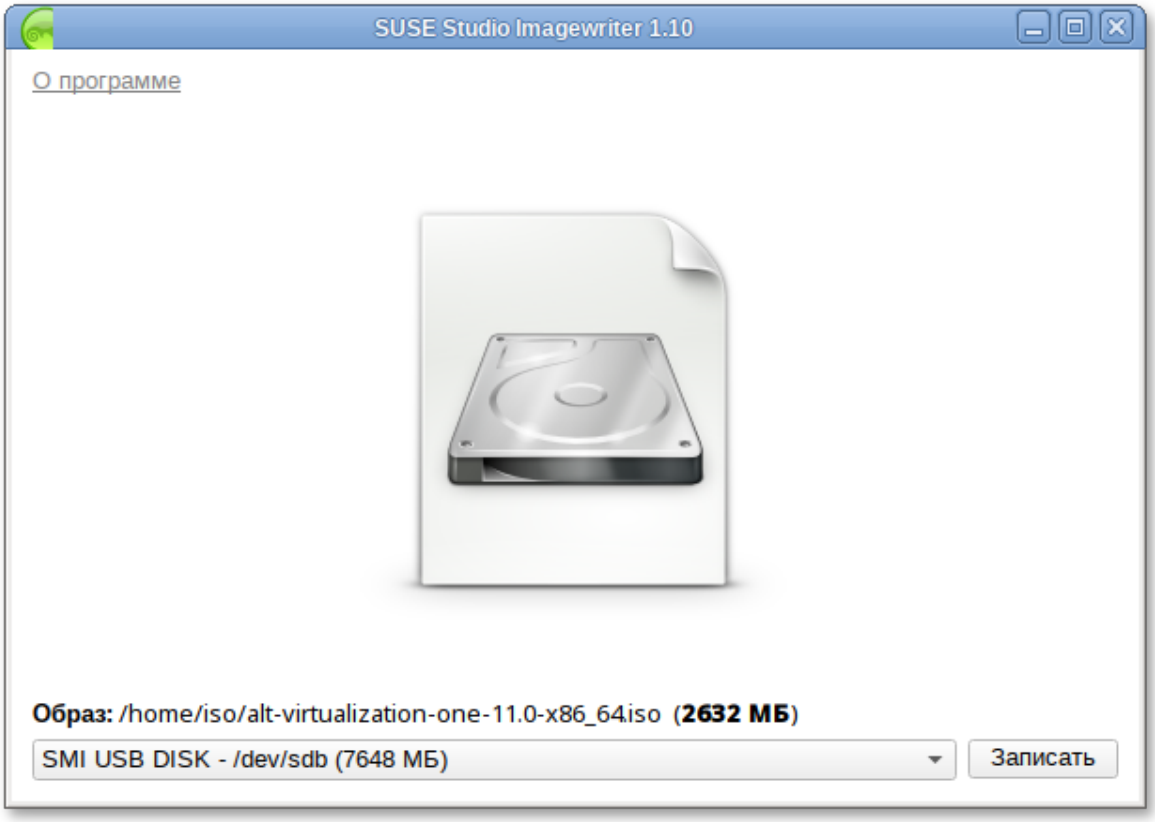


  ALT Media Writer может автоматически загружать образы из интернета и записывать их, при необходимости извлекая сжатые образы (img.xz).

- Impression (*impression*):



SUSE Studio Imagewriter (*imagewriter*):



Предупреждение

Будьте внимательны при указании имени USB-устройства — запись образа по ошибке на свой жёсткий диск приведёт к почти гарантированной потере данных на нём!

Предупреждение

Не добавляйте номер раздела, образ пишется на flash-диск с самого начала!

Для записи установочного образа можно воспользоваться утилитой командной строки **dd**:

```
# dd oflag=direct if=<файл-образа.iso> of=/dev/sdX bs=1M status=progress;sync
```

где <файл-образа.iso> — ISO-образ установочного диска с дистрибутивом, а **/dev/sdX** — устройство, соответствующее flash-диску.

Для удобства показа прогресса записи можно установить пакет *pv* и использовать команду:

```
# pv <файл-образа.iso> | dd oflag=direct of=/dev/sdX bs=1M;sync
```

где <файл-образа.iso> — ISO-образ диска, а **/dev/sdX** — устройство, соответствующее flash-диску.

Просмотреть список доступных устройств можно командой **lsblk** или (если такой команды нет) **blkid**.

Например, так можно определить имя flash-диска:

```
$ lsblk | grep disk
sda      8:0    0 931,5G  0 disk
sdb      8:16   0 931,5G  0 disk
sdc      8:32   1   7,4G  0 disk
```

USB-диск имеет имя устройства sdc.

Затем записать:

```
# dd oflag=direct if=/iso/alt-virtualization-one-11.0-x86_64.iso of=/dev/sdc
bs=1M status=progress; sync
```

или, например, так:

```
# pv /iso/alt-virtualization-one-11.0-x86_64.iso | dd oflag=direct of=/dev/
sdc bs=1M;sync
dd: warning: partial read (524288 bytes); suggest iflag=fullblock
3GiB 0:10:28 [4,61MiB/s] [==================================>  ] 72% ETA
0:04:07
```

Предупреждение

Не извлекайте USB-диск, пока образ не запишется до конца! Определить финал процесса можно по прекращению моргания индикатора USB-диска либо посредством виджета **Безопасное извлечение съемных устройств**.

4.2.3. В операционной системе OS X

В операционной системе OS X для создания загрузочного USB-диска можно использовать команду:

```
sudo dd if=alt-virtualization-one-11.0-x86_64.iso of=/dev/rdiskX bs=10M
sync
```

где **alt-virtualization-one-11.0-x86_64.iso** — ISO-образ диска, а /dev/rdiskX — flash-диск.

Просмотреть список доступных устройств можно командой:

```
diskutil list
```

Предупреждение

Будьте внимательны при указании имени USB-устройства — запись образа по ошибке на свой жёсткий диск приведёт к почти гарантированной потере данных на нём!

4.2.4. Проверка целостности записанного образа

Для проверки целостности записанного образа необходимо выполнить следующие шаги:

- определить длину образа в байтах:

```
$ du -b alt-virtualization-one-11.0-x86_64.iso | cut -f1
2760607744
```

- посчитать контрольную сумму образа (или просмотреть контрольную сумму образа из файла MD5SUM на сервере FTP):

```
$ md5sum alt-virtualization-one-11.0-x86_64.iso
e182bf50725dbdef572b9a61af83367e  alt-virtualization-one-11.0-x86_64.iso
```

- подсчитать контрольную сумму записанного образа на DVD или USB Flash (выполняется под правами пользователя root):

```
# head -c 2760607744 /dev/sdd | md5sum
e182bf50725dbdef572b9a61af83367e
```

  где размер после -c — вывод в п.1, а /dev/sdd — устройство DVD или USB Flash, на которое производилась запись.

Глава 5. Сохранение данных и меры предосторожности

Если необходимо установить ОС Альт Виртуализация редакция ONE и при этом сохранить уже установленную на компьютере операционную систему (например, другую версию GNU/Linux или Microsoft Windows), то нужно обязательно позаботиться о подготовке компьютера к установке второй системы и о сохранении ценных для вас данных.

Если у вас нет загрузочного диска для уже установленной системы, создайте его. В случае прерванной установки ОС Альт Виртуализация редакция ONE или неправильной настройки загрузчика, вы можете потерять возможность загрузиться в вашу предыдущую ОС.

Если на диске, выбранном для установки ОС Альт Виртуализация редакция ONE, не осталось свободного раздела, то программа установки должна будет изменить размер существующего раздела. От этой операции могут пострадать ваши данные, поэтому предварительно надо сделать следующие действия:

- Выполнить проверку раздела, который вы собираетесь уменьшать. Для этого воспользуйтесь соответствующим программным обеспечением (далее — ПО), входящим в состав уже установленной ОС. Программа установки
  Альт Виртуализация редакция ONE может обнаружить некоторые очевидные ошибки при изменении размера раздела, но специализированное ПО предустановленной ОС справится с этой задачей лучше.

- Выполнить дефрагментацию уменьшаемого раздела в целях повышения уровня безопасности данных. Это действие не является обязательным, но мы настоятельно рекомендуем его произвести: изменение размера раздела пройдёт легче и быстрее.

Предупреждение

Полной гарантией от проблем, связанных с потерей данных, является резервное копирование!

Глава 6. Начало установки: загрузка системы

6.1. Способы первоначальной загрузки

Для установки системы компьютер необходимо загрузить с загрузочного носителя.

Простейший способ запустить программу установки — загрузить компьютер с установочного DVD с дистрибутивом (при условии, что система поддерживает загрузку с оптического привода).

Программу установки можно также запустить с другого загрузочного носителя, например с USB-флеш-накопителя.

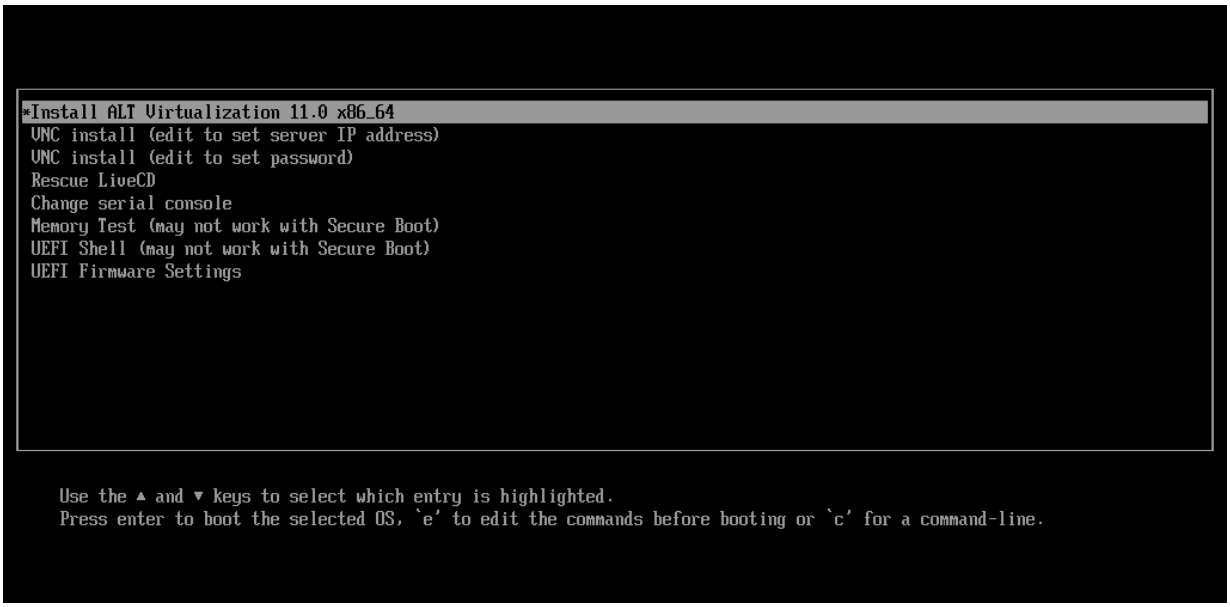
6.2. Загрузка системы

Для того чтобы начать установку ОС Альт Виртуализация редакция ONE, достаточно загрузиться с носителя, на котором записан дистрибутив.

Примечание

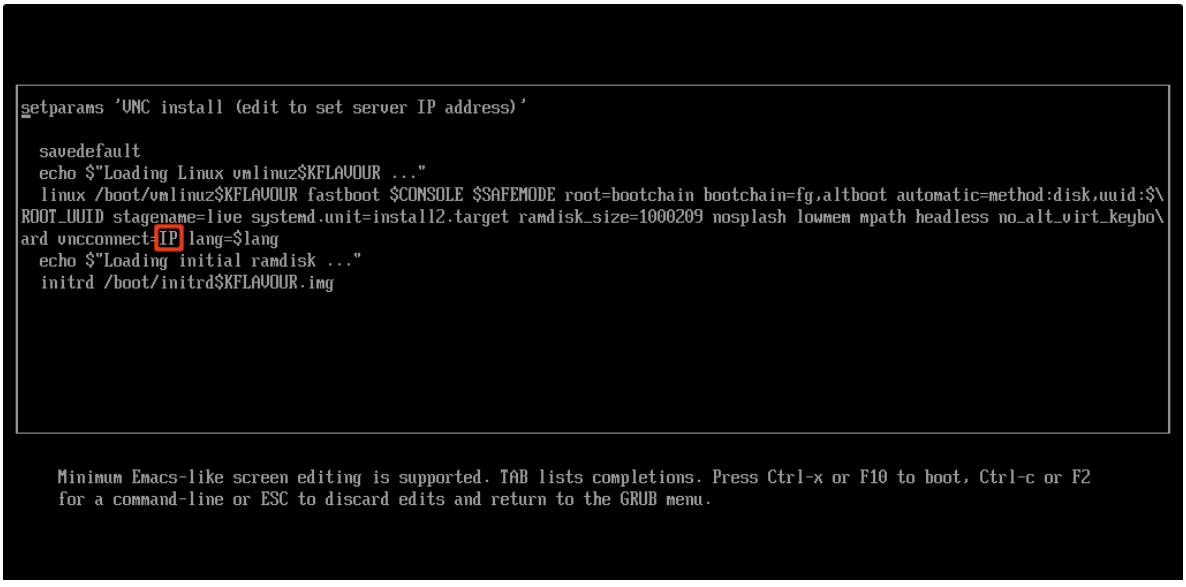
Предварительно следует включить в BIOS опцию загрузки с оптического привода или с USB-устройства.

В большинстве случаев указание способа входа в BIOS отображается на вашем мониторе непосредственно после включения компьютера. Способ входа в меню BIOS и информация о расположении настроек определяется производителем используемого оборудования. За информацией можно обратиться к документации на ваше оборудование.

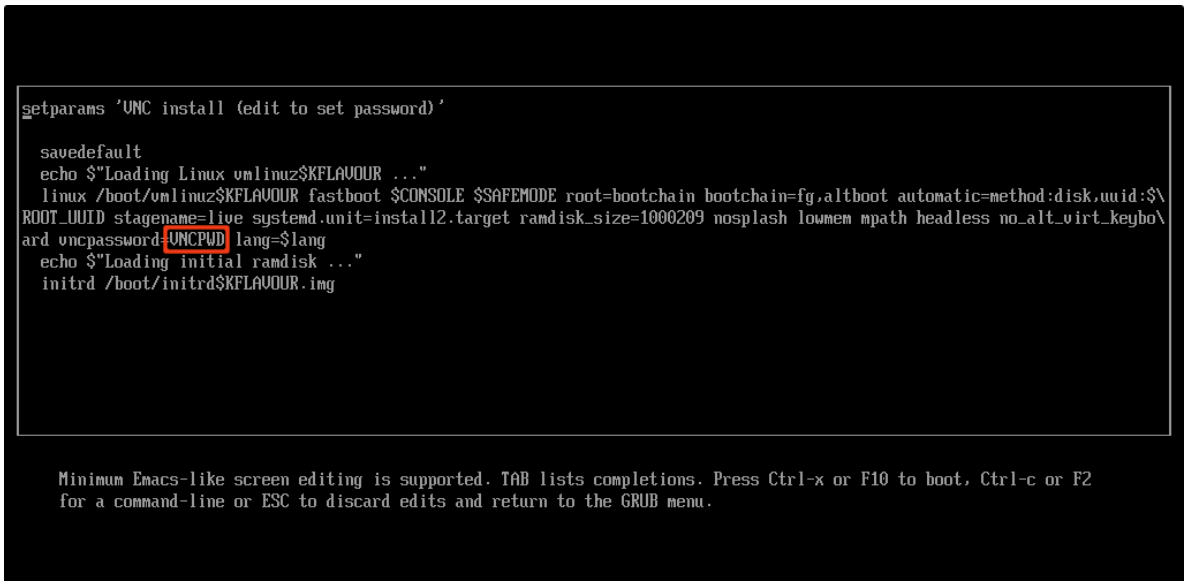


Загрузка с установочного диска или специально подготовленного USB-flash-накопителя начинается с меню, в котором перечислено несколько вариантов загрузки. Кроме установки системы с установочного диска, в данном меню доступны несколько вариантов сетевой установки:

- **Install ALT Virtualization 11.0** — установка операционной системы;
- **VNC install (edit to set server IP address)** — установка по VNC, при которой устанавливаемая машина подключается к серверу VNC с указанным IP-адресом. Параметры установки по VNC передаются как аргументы ядра. Нажатие клавиши **E** позволяет задать IP-адрес сервера. (чтобы принять подключение, на сервере VNC следует запустить, например, **`vncviewer --listen`**):



- **VNC install (edit to set password and connect here)** — установка по VNC при которой управляющий компьютер подключается к устанавливаемой машине. Параметры установки по VNC передаются как аргументы ядра. Нажатие клавиши **E** позволяет задать пароль (по умолчанию — VNCPWD):



- **Rescue LiveCD** — восстановление уже установленной, но так или иначе поврежденной ОС Linux за счёт запуска небольшого образа системы в оперативной памяти. Восстановление системы потребует некоторой квалификации. Этот режим также можно использовать для сбора информации об оборудовании компьютера в случае некорректной установки или работы ОС Альт Виртуализация редакция ONE. Собранную информацию можно отправить разработчикам. Загрузка восстановительного режима заканчивается запросом на ввод логина суперпользователя (root):

```
Log in as root
localhost login:
```

- **Change language (press F2)** — позволяет выбрать язык интерфейса загрузчика и программы установки (нажатие клавиши **F2** вызывает такое же действие);

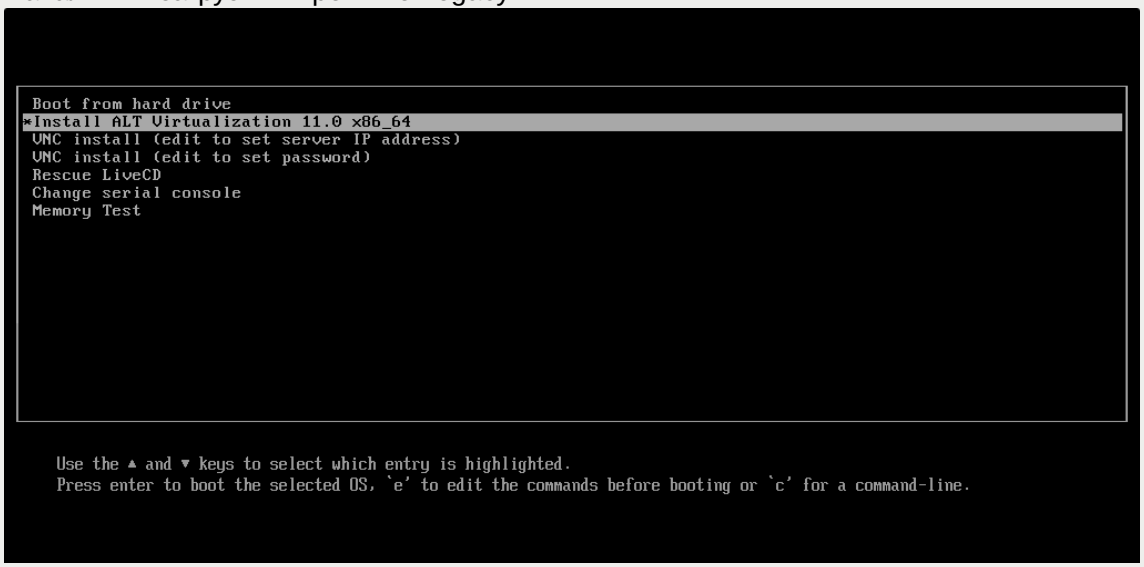
- **Change serial console** — позволяет выбрать последовательный порт для консольного подключения (например, COM1/ttyS0):

Подробнее об установке через последовательный порт см. Загрузка с помощью последовательного порта.

- **Memory Test (may not work with Secure Boot)** — проверка целостности оперативной памяти. Процесс диагностики заключается в проведении нескольких этапов тестирования каждого отдельного модуля ОЗУ (данный процесс будет выполняться бесконечно, пока его не остановят, необходимо дождаться окончания хотя бы одного цикла проверки);
- **UEFI Shell (may not work with Secure Boot)** — оболочка/терминал для прошивки, позволяющий запускать EFI-приложения, в том числе загрузчики UEFI;
- **UEFI Firmware Settings** — позволяет получить доступ к настройкам UEFI.

Примечание

Начальный загрузчик в режиме Legacy:



Пункт **Boot from hard drive** — позволяет запустить уже установленную на жёсткий диск операционную систему.

Примечание

Мышь на этом этапе установки не поддерживается. Для выбора опций установки и различных вариантов необходимо использовать клавиатуру.

Нажатием клавиши **E** можно вызвать редактор параметров текущего пункта загрузки. Если система настроена правильно, то редактировать их нет необходимости.

Чтобы начать процесс установки, нужно клавишами перемещения курсора **вверх** и **вниз** выбрать пункт меню **Install ALT Virtualization 11.0** и нажать **Enter**. Начальный этап установки не требует вмешательства пользователя: происходит автоматическое определение оборудования и запуск компонентов программы установки. Сообщения о происходящем на данном этапе можно просмотреть, нажав клавишу **ESC**.

Примечание

В начальном загрузчике установлено небольшое время ожидания: если в этот момент не предпринимать никаких действий, то будет загружена та система, которая уже установлена на жестком диске. Если вы пропустили нужный момент, перезагрузите компьютер и вовремя выберите пункт **Install ALT ALT Virtualization 11.0**.

6.3. Загрузка с помощью последовательного порта

Установка ОС Альт Виртуализация редакция ONE через последовательный порт:

1. В меню загрузчика выберите пункт **Change serial console** и укажите порт для консольного подключения:

2. Выберите скорость передачи данных:

3. Запустите установку, выбрав в меню вариантов загрузки пункт **VNC install (edit to set password)**.

Примечание

При необходимости можно изменить пароль для VNC-подключения (см. VNC install).

4. Подключитесь к последовательному порту с помощью терминала. В консоли отобразится процесс установки, а также IP-адрес, порт и пароль для VNC-подключения:

```
…
Running initinstall script [15-expert.sh] [ DONE ] 14:19:54
Running initinstall script [20-nodesign.sh] [ DONE ] 14:19:54
Running initinstall script [20-pts.sh] [ DONE ] 14:19:54
Running initinstall script [25-setup-dhcp.sh] [ DONE ] 14:19:54
Running initinstall script [26-metadata-autoinstall.sh] [ DONE ]
14:19:54
Running initinstall script [27-metadata-install-scripts.sh] [ DONE ]
14:19:54
Running initinstall script [30-ntp-client.sh] [ DONE ] 14:19:54
Running initinstall script [40-xorg.sh] [ DONE ] 14:19:54
Running initinstall script [50-removable] [ DONE ] 14:19:54
Running initinstall script [80-stop-md-dm.sh] [ DONE ] 11:19:59
Running initinstall script [85-start-multipath.sh] [ DONE ] 11:19:59
Running initinstall script [90-alteratord.sh] [ DONE ] 11:19:59
Running initinstall script [90-date.sh] [ DONE ] 11:19:59
Running initinstall script [90-pkg.sh] [ DONE ] 11:20:00
Running initinstall script [91-pkg-size.sh] [ DONE ] 11:20:01
Running initinstall script [95-virt-env.sh] [ DONE ] 11:20:01
Running initinstall script [98-oem.sh] [ DONE ] 11:20:01
Open debug shell on tty2 [ DONE ] 11:20:01
Open debug shell on tty3 [ DONE ] 11:20:01
** IP(s): 192.168.0.95
** VNC cmdline: vncpassword=Pa$$word

The VNC desktop is:      localhost.localdomain:0
PORT=5900
```

5. Для продолжения установки подключитесь по VNC, например:

```
$ vncviewer 192.168.0.95
```

Примечание

Установка через VNC выполняется так же, как описано в главе Последовательность установки.

После завершения графической части установки в окне последовательной консоли появится запрос на ввод логина:

```
Welcome to ALT Virtualization ONE Edition 11.0 (Nimbostratus)!

Hostname: one
IP: 192.168.0.185
one login:
```

Глава 7. Последовательность установки

До того как будет произведена установка базовой системы на жёсткий диск, программа установки работает с образом системы, загруженным в оперативную память компьютера.

Если инициализация оборудования завершилась успешно, будет запущен графический интерфейс программы-установщика. Процесс установки разделён на шаги. Каждый шаг посвящён настройке или установке определённого свойства системы. Шаги нужно проходить последовательно. Переход к следующему шагу происходит по нажатию кнопки **Далее**. При помощи кнопки **Назад**, при необходимости, можно вернуться к уже пройденному шагу и изменить настройки. Однако возможность перехода к предыдущему шагу ограничена теми шагами, в которых нет зависимости от данных, введённых ранее.

Если по каким-то причинам возникла необходимость прекратить установку, необходимо нажать кнопку <Reset> на корпусе системного блока компьютера.

Примечание

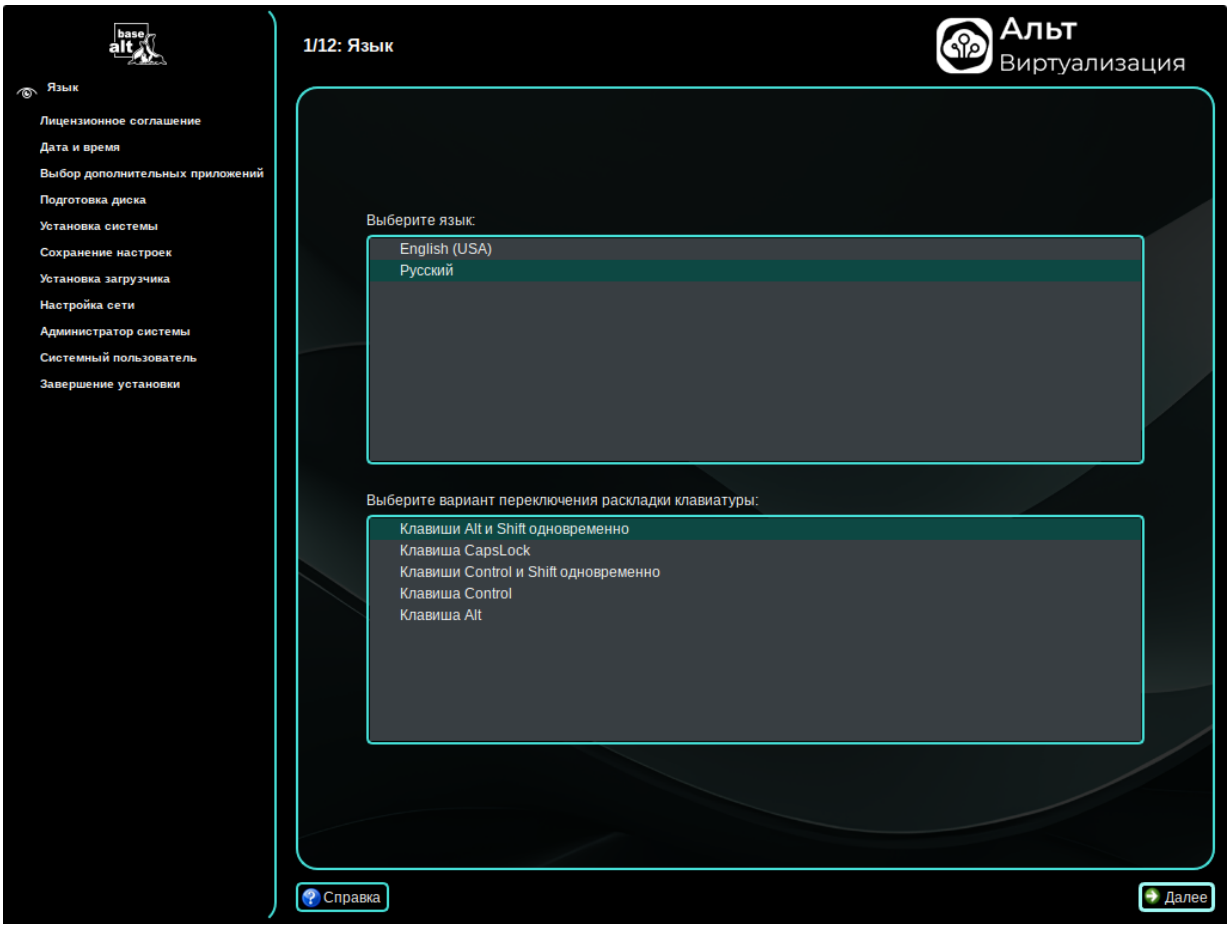
Совершенно безопасно выполнить отмену установки только до шага Подготовка диска, поскольку до этого момента не производится никаких изменений на жёстком диске. Если прервать установку между шагами Подготовка диска и Установка загрузчика, существует вероятность, что после этого с жёсткого диска не сможет загрузиться ни одна из установленных систем (если такие имеются).

Технические сведения о ходе установки можно посмотреть, нажав **Ctrl**+**Alt**+**F1**, вернуться к программе установки — **Ctrl**+**Alt**+**F7**. По нажатию **Ctrl**+**Alt**+**F2** откроется отладочная виртуальная консоль.

Каждый шаг сопровождается краткой справкой, которую можно вызвать, щёлкнув кнопку **Справка** или нажав клавишу **F1**.

Нажатие на кнопку

позволяет показать/скрыть панель со списком шагов установки:



Во время установки системы выполняются следующие шаги:

- Язык;
- Лицензионное соглашение;
- Дата и время;
- Выбор дополнительных приложений;
- Подготовка диска;
- Установка системы;
- Сохранение настроек;
- Установка загрузчика;
- Настройка сети;
- Администратор системы;
- Системный пользователь;
- Завершение установки.

Глава 8. Язык

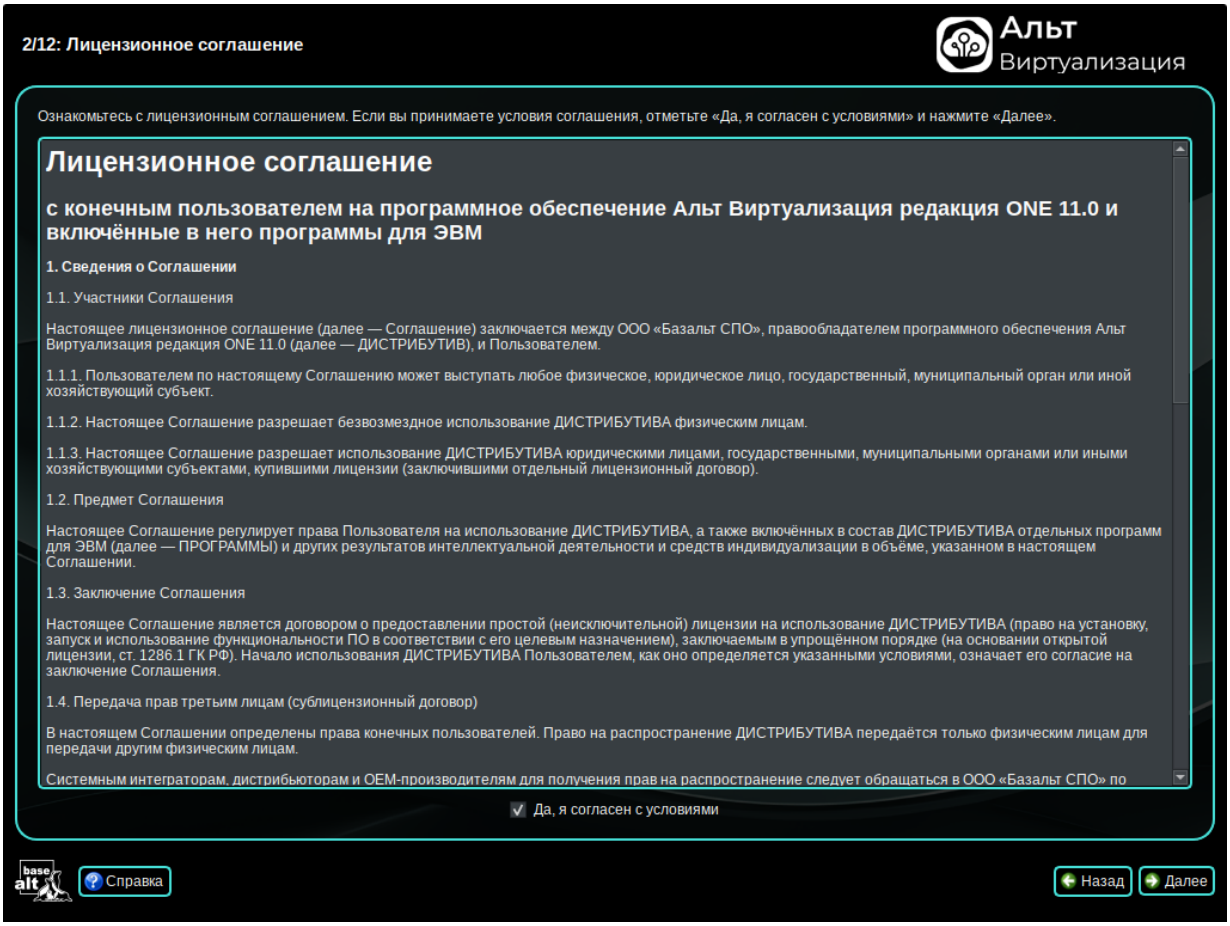
Установка Альт Виртуализация редакция ONE начинается с выбора основного языка — языка интерфейса программы установки и устанавливаемой системы. В списке, помимо доступных языков региона (выбранного на этапе начальной загрузки), указан и английский язык.

На этом же этапе выбирается вариант переключения раскладки клавиатуры. Раскладка клавиатуры — это привязка букв, цифр и специальных символов к клавишам на клавиатуре. Помимо ввода символов на основном языке, в любой системе Linux необходимо иметь возможность вводить латинские символы (имена команд, файлов и т.п.). Для этого обычно используется стандартная английская раскладка клавиатуры. Переключение между раскладками осуществляется при помощи специально зарезервированных для этого клавиш. Для русского языка доступны следующие варианты переключения раскладки:

- клавиши **`Alt`** и **`Shift`** одновременно;
- клавиша **`CapsLock`**;
- клавиши **`Control`** и **`Shift`** одновременно;
- клавиша **`Control`**;
- клавиша **`Alt`**.

Если выбранный основной язык имеет всего одну раскладку (например, при выборе английского языка в качестве основного), эта единственная раскладка будет принята автоматически.

Глава 9. Лицензионное соглашение



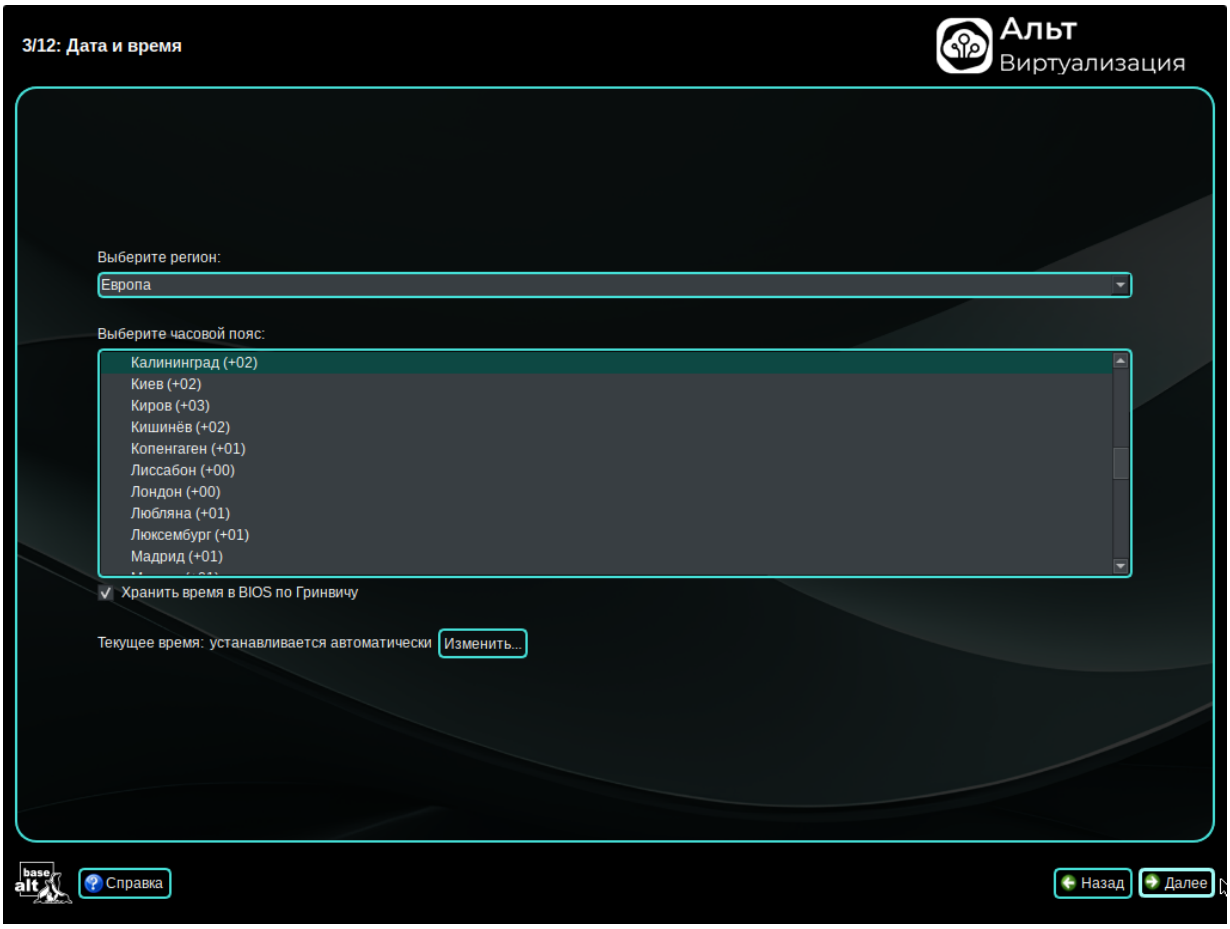
Перед продолжением установки следует внимательно прочитать условия лицензии. В лицензии говорится о ваших правах. В частности, за вами закрепляются права на:

- эксплуатацию программ на любом количестве компьютеров и в любых целях;
- распространение программ (сопровождая их копией авторского договора);
- получение исходных текстов программ.

Если вы приобрели дистрибутив, то данное лицензионное соглашение прилагается в печатном виде к вашей копии дистрибутива. Лицензия относится ко всему дистрибутиву Альт Виртуализация редакция ONE. Если вы согласны с условиями лицензии, отметьте пункт **`Да, я согласен с условиями`** и нажмите кнопку **Далее**.

Глава 10. Дата и время

На данном этапе выполняется выбор региона и города, по которым будет определен часовой пояс и установлены системные часы.



Для корректной установки даты и времени достаточно правильно указать часовой пояс и выставить желаемые значения для даты и времени.

На этом шаге следует выбрать часовой пояс, по которому нужно установить часы. Для этого в соответствующих списках выберите регион, а затем город. Поиск по списку можно ускорить, набирая на клавиатуре первые буквы искомого слова.

Пункт **`Хранить время в BIOS по Гринвичу`** выставляет настройки даты и времени в соответствии с часовыми поясами, установленными по Гринвичу, и добавляет к местному времени часовую поправку для выбранного региона.

После выбора часового пояса будут предложены системные дата и время по умолчанию.

Для ручной установки текущих даты и времени нужно нажать кнопку **`Изменить…`**. Откроется окно ручной настройки системных параметров даты и времени.

Для синхронизации системных часов с удалённым сервером времени (NTP) по локальной сети или по сети Интернет нужно отметить пункт **Получать точное время с NTP-сервера** и указать предпочитаемый NTP-сервер. В большинстве случаев можно указать сервер **pool.ntp.org**.

Если выбрана опция **Получать точное время с NTP-сервера**, то компьютер может и сам быть сервером точного времени. Например, использоваться как сервер точного времени машинами локальной сети. Для активации этой возможности необходимо отметить пункт **Работать как NTP-сервер**.

Для сохранения настроек и продолжения установки системы в окне ручной установки даты и времени необходимо нажать кнопку **OK** и затем в окне **Дата и время** нажать кнопку **Далее**.

Примечание

В случае если ОС Альт Виртуализация редакция ONE устанавливается как вторая ОС, необходимо снять отметку с пункта **Хранить время в BIOS по Гринвичу**, иначе время в уже установленной ОС может отображаться некорректно.

Глава 11. Выбор дополнительных приложений

На данном этапе программа установки предлагает выбрать дополнительные пакеты программ, которые будут включены в состав ОС Альт Виртуализация редакция ONE и установлены вместе с ней на диск.

В любом дистрибутиве Альт Виртуализация редакция ONE доступно значительное количество программ (до нескольких тысяч), часть из них составляет саму операционную систему, а остальные — это прикладные программы и утилиты.

В ОС Альт Виртуализация редакция ONE все операции установки и удаления производятся над пакетами — отдельными компонентами системы. Пакет и программа соотносятся неоднозначно: иногда одна программа состоит из нескольких пакетов, иногда один пакет включает несколько программ.

В процессе установки системы обычно не требуется детализированный выбор компонентов на уровне пакетов — это требует слишком много времени и знаний от проводящего установку, тем более, что комплектация дистрибутива подбирается таким образом, чтобы из имеющихся программ можно было составить полноценную рабочую среду для соответствующей аудитории пользователей. Поэтому в процессе установки системы пользователю предлагается выбрать из небольшого списка групп пакетов именно те, которые необходимы для решения наиболее распространённых задач.

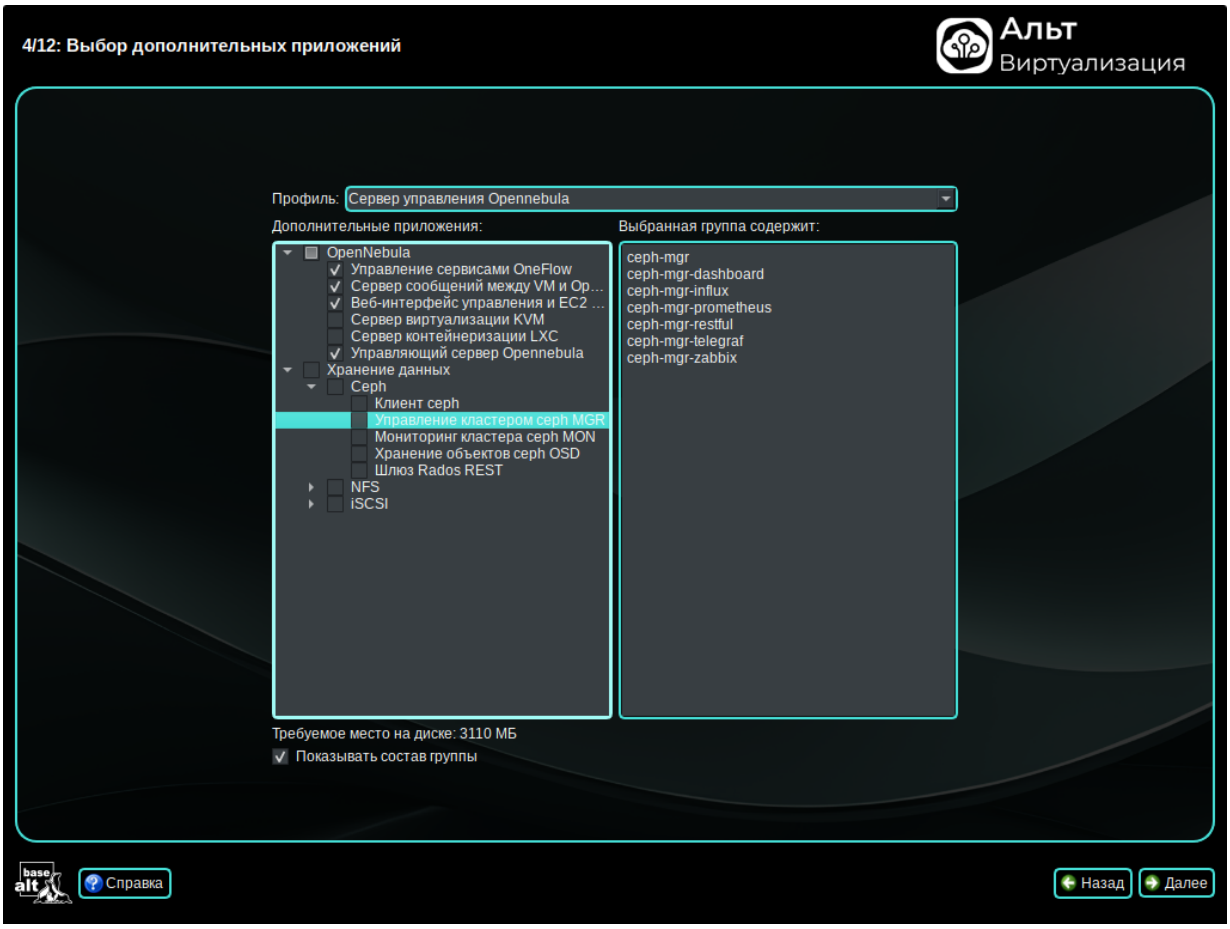
При установке доступны следующие профили:

- **Сервер управления Opennebula** — управляющий сервер Opennebula;
- **Вычислительный узел Opennebula KVM** — гипервизор с виртуальными машинами;
- **Вычислительный узел Opennebula LXC** — гипервизор LXC контейнеров.

После выбора профиля можно изменить состав устанавливаемых пакетов.

Под списком групп на экране отображается информация об объёме дискового пространства, которое будет занято после установки пакетов, входящих в выбранные группы.

При выборе группы пакетов будет показан список программных пакетов, входящих в состав этой группы:



Выбрав необходимые группы, следует нажать **Далее**.

Глава 12. Подготовка диска

На данном этапе установки необходимо выбрать конфигурацию хранилища будущей системы.

Возможные варианты разметки:

- установка на LVM (с тонким пулом под образы виртуальных машин);
- программный Btrfs RAID массив.

Профиль разбиения диска можно выбрать в поле **Схема разметки**:

12.1. Профиль LVM

LVM — соответствует пункту **`ext4 (LVM)`**. При данной схеме разбивки будет создана группа томов (VG) с именем one. Также будут созданы следующие логические тома (LV):

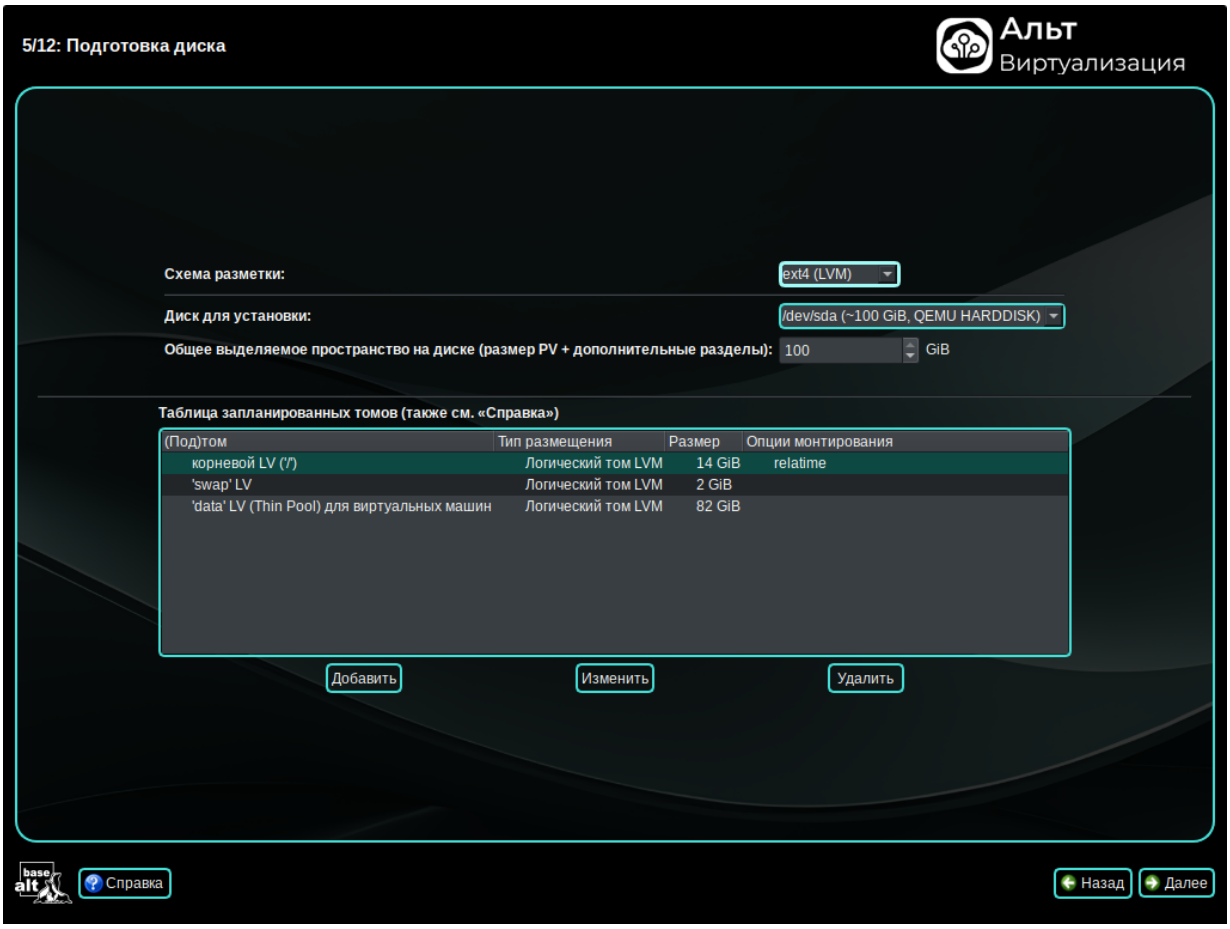
- **`root`** — корневой том ('/') с файловой системой ext4;
- **`swap`** — том для подкачки (рекомендуемый размер ≥ 2 ГиБ);
- **`data`** — том с тонким пулом (Thin Pool) для хранения образов виртуальных дисков и данных контейнеров.

Размеры перечисленных выше томов выставляются оптимально на основе общего выделяемого пространства.

По умолчанию используется весь диск, но при этом сохраняется 1-2 ГиБ свободного места в VG (оно может быть полезно при создании мгновенного снимка системы).

Примечание

При установке системы в режиме UEFI раздел EFI будет создан автоматически.



Для установки системы на LVM необходимо:

1. В списке **Схема разметки** выбрать файловую систему **`ext4 (LVM)`**.
2. В списке **Диск для установки** выбрать диск, на который будет установлена система.
3. В поле **`Общее выделяемое пространство на диске`** указать, если это необходимо, размер выделяемого пространства (по умолчанию весь диск).

В таблице будут показаны все запланированные логические тома.

Для добавления дополнительного тома необходимо:

1. Нажать кнопку **`Добавить`**.
2. В открывшемся окне:
   - в списке **Том (точка монтирования)** выбрать точку монтирования из списка или выбрать пункт **`Указать произвольную точку монтирования`** и указать точку монтирования в поле **`Путь монтирования`**;
   - в списке **Тип размещения** выбрать:
     - **`Логический том LVM`** — для размещения тома как еще один LV в VG;
     - **`Отдельный раздел`** — для размещения тома в отдельном разделе на диске (например, для **`/boot`**);
   - в поле **`Размер`** указать размер тома;

- в поле **Опции монтирования** при необходимости указать опции монтирования.

Примечание

Для добавления новых томов может потребоваться уменьшить размер уже созданного тома.

Все дополнительные тома будут использовать файловую систему ext4 (как корневой раздел).

Изменить или удалить том можно, выбрав его и нажав кнопку **Изменить** или **Удалить**.

Примечание

Корневой том удалить нельзя.

Для продолжения установки следует нажать кнопку **Далее**. Появится окно с предупреждением о том, что данные на выбранном диске будут уничтожены. Если вы уверены в том, что диск выбран верно, подтвердите переход к следующему шагу нажатием кнопки **OK**.

Предупреждение

С выбранного диска будут удалены все данные без возможности восстановления.

Примечание

Результат автоматического разбиения диска (100 ГБ) при выборе схемы разметки **`ext4 (LVM)`**:

```
# lsblk
NAME         MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda            8:0    0  100G  0 disk
├─sda1         8:1    0  600M  0 part /boot/efi
└─sda2         8:2    0 99,4G  0 part
  ├─one-swap 253:0    0    2G  0 lvm  [SWAP]
  └─one-root 253:1    0   14G  0 lvm  /
sr0           11:0    1  2,6G  0 rom
```

12.2. Профиль Btrfs RAID

При установке дистрибутива можно выбрать Btrfs в качестве корневой файловой системы и указать тип RAID:

- **`RAID 0`** — чередование без отказоустойчивости. Ёмкость тома равна сумме ёмкостей всех дисков. RAID0 не обеспечивает избыточности, поэтому отказ одного из дисков приведет к потере всех данных;
- **`RAID 1`** — зеркалированный массив. Данные записываются одинаково на все диски. Для работы требуется не менее двух дисков одинакового размера. Эффективная ёмкость равна ёмкости одного диска;
- **`RAID 10`** (RAID 1+ RAID0) — зеркалированный массив с чередованием. Данные записываются последовательно на несколько дисков, как в RAID 0, но каждый блок зеркалируется, как в RAID 1. Требуется не менее четырёх дисков;
- **`Single`** — одиночный режим. Используется один диск без отказоустойчивости.

Примечание

При установке системы в режиме UEFI раздел EFI будет создан автоматически.

Для установки системы на Btrfs RAID необходимо:

1. В списке **Схема разметки** выбрать файловую систему Btrfs и нужный тип RAID.

2. В списке **Диски для установки** выбрать диски, на которых будет создан RAID и установлена система.

В результате такой разбивки будет создан Btrfs-массив в выбранном режиме RAID с корневым подтомом (subvolume).

Для добавления дополнительного тома или подтома необходимо:

1. Нажать кнопку **Добавить**.

2. В открывшемся окне:

   - в списке **Том (точка монтирования)** выбрать точку монтирования из списка или выбрать пункт **Указать произвольную точку монтирования** и указать точку монтирования в поле **Путь монтирования**;

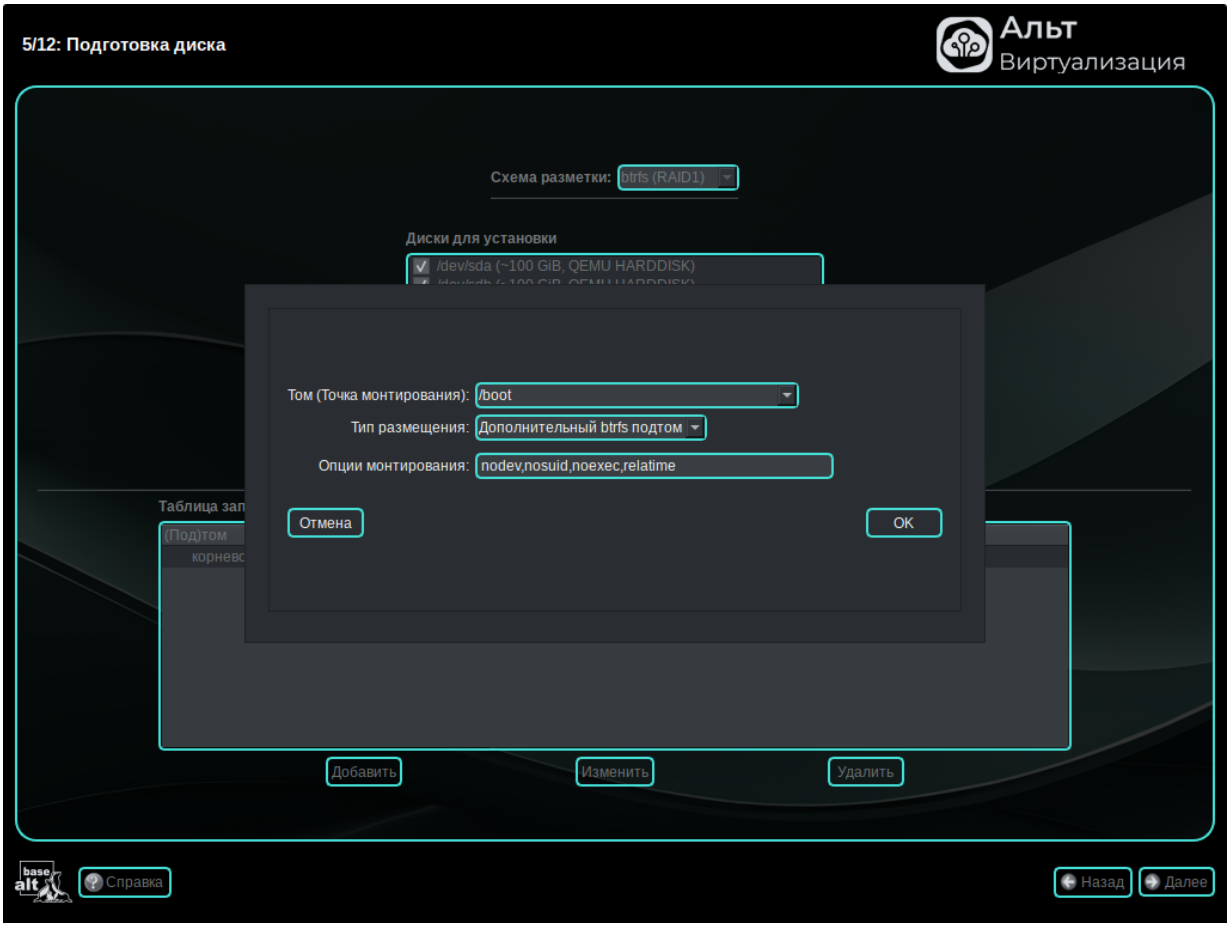
   - в списке **Тип размещения** выбрать:

     - **Дополнительный btrfs подтом** — новый подтом будет создан в корневом томе;

     - **Отдельный btrfs том** — будут созданы отдельные разделы на дисках, из которых будет собран соответствующий btrfs RAID том (это позволяет ограничить размер тома, сделав его независимым от корневого);

   - если выбран тип размещения **Отдельный btrfs том**, в поле **Размер** указать размер тома;

   - в поле **Опции монтирования** указать, если это необходимо, опции монтирования.



Примечание

Для добавления новых томов может потребоваться уменьшить размер корневого раздела.

Изменить или удалить том можно, выбрав его и нажав кнопку **Изменить** или **Удалить**.

Примечание

Корневой том удалить нельзя.

Примечание

После установки вывести список всех подтомов можно, выполнив команду:

```
# btrfs subvolume list /
ID 256 gen 23 top level 5 path root
ID 257 gen 15 top level 256 path var/lib/machines
```

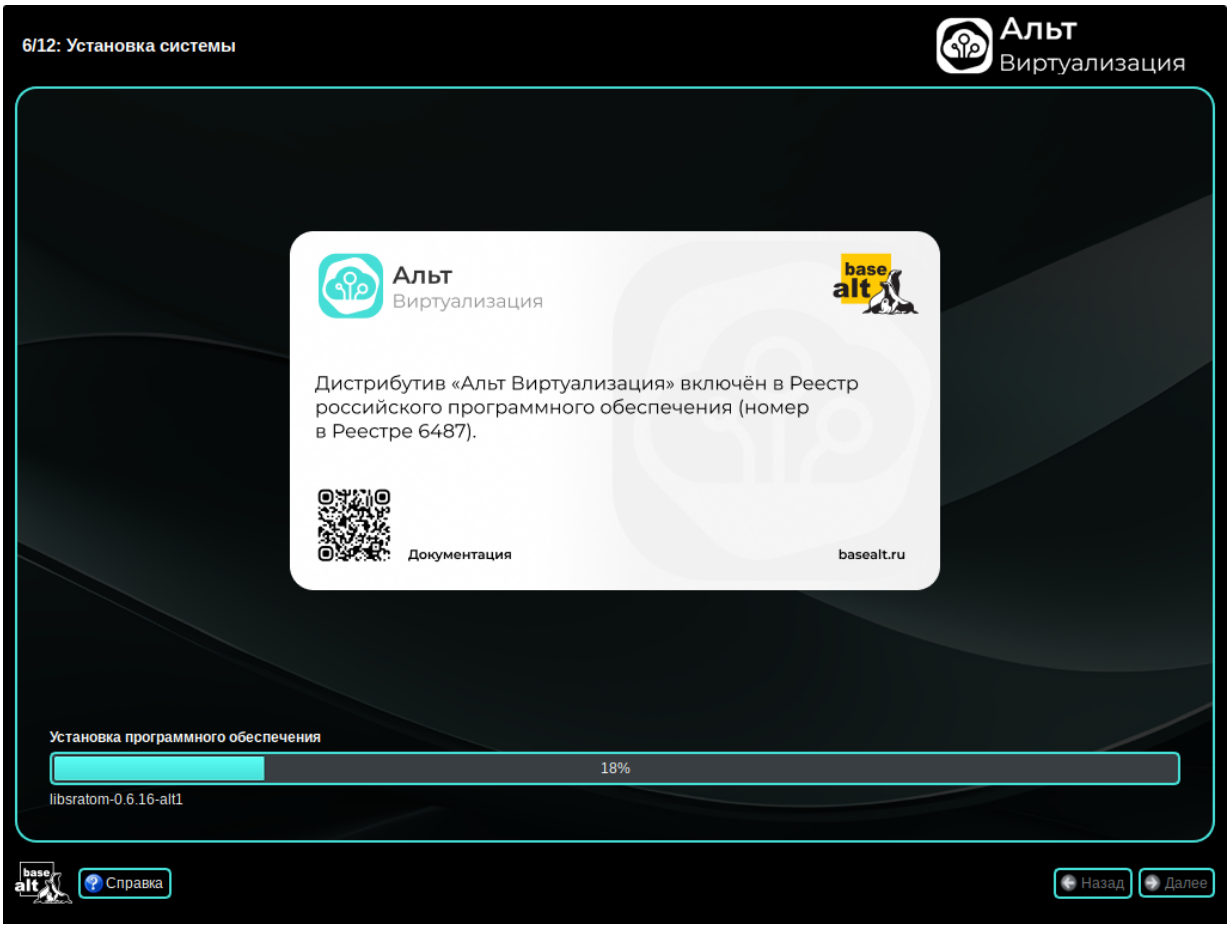
Для продолжения установки следует нажать кнопку **Далее**. Появится окно с предупреждением о том, что данные на выбранных дисках будут уничтожены. Если вы уверены в том, что диски выбраны верно, подтвердите переход к следующему шагу нажатием кнопки **OK**.

Предупреждение

С выбранных дисков будут удалены все данные без возможности восстановления.

Глава 13. Установка системы

На данном этапе происходит распаковка ядра и установка набора программ, необходимых для работы Альт Виртуализация редакция ONE.



Установка происходит автоматически в два этапа:

- получение пакетов;
- установка пакетов.

Получение пакетов осуществляется из источника, выбранного на этапе начальной загрузки. При сетевой установке время выполнения этого шага будет зависеть от скорости соединения и может быть значительно большим в сравнении с локальной установкой.

Глава 14. Сохранение настроек

Начиная с данного этапа, программа установки работает с файлами только что установленной базовой системы. Все последующие изменения можно будет совершить после завершения установки посредством редактирования соответствующих конфигурационных файлов или при помощи модулей управления, включенных в дистрибутив.

По завершении установки базовой системы начинается шаг сохранения настроек. Он проходит автоматически и не требует вмешательства пользователя. На экране отображается индикатор выполнения.

На этом шаге производится перенос настроек, выполненных на первых шагах установки, в только что установленную базовую систему. Производится также запись информации о соответствии разделов жесткого диска смонтированным на них файловым системам (заполняется конфигурационный файл **`/etc/fstab`**).

После сохранения настроек осуществляется автоматический переход к следующему шагу.

Глава 15. Установка загрузчика

Загрузчик ОС — это программа, которая позволяет загружать
Альт Виртуализация редакция ONE и другие ОС, если они установлены на данной машине.

При установке на EFI модуль установки загрузчика предложит вариант установить загрузчик в специальный раздел «**EFI**» (рекомендуется выбрать автоматическое разбиение на этапе разметки диска для создания необходимых разделов для загрузки с EFI):

Варианты установки загрузчика при установке в режиме EFI:

- **EFI (рекомендуемый)** — при установке загрузчика в NVRAM будет добавлена запись, без которой большинство компьютеров не смогут загрузиться во вновь установленную ОС;
- **EFI (сначала очистить NVRAM)** — перед добавлением записи в NVRAM её содержимое будет сохранено в **`/root/.install-log`**, после чего из неё будут удалены все загрузочные записи, что приведёт к восстановлению полностью заполненной NVRAM и гарантирует загрузку вновь установленной ОС;
- **EFI (запретить запись в NVRAM)** — этот вариант следует выбрать, только если инсталлятор не может создать запись в NVRAM или если заведомо известно, что запись в NVRAM может вывести компьютер из строя (вероятно, запись в NVRAM придётся создать после установки ОС средствами BIOS Setup);
- **EFI (для съёмных устройств)** — этот вариант следует выбрать, только если ОС устанавливается на съёмный накопитель. Этот вариант также можно использовать вместо варианта **EFI (запретить запись в NVRAM)** при условии, что это будет единственная ОС на данном накопителе. Создавать запись в NVRAM не потребуется.

Выбор варианта установки загрузчика, зависит от вашего оборудования. Если не работает один вариант, попробуйте другие.

Примечание

Установка загрузчика при установке в режиме Legacy:

Программа установки автоматически определяет, в каком разделе жёсткого диска следует располагать загрузчик для возможности корректного запуска ОС
Альт Виртуализация редакция ONE. Положение загрузчика, в случае необходимости, можно изменить в списке **Устройство**, выбрав другой раздел.

Если же вы планируете использовать и другие ОС, уже установленные на этом компьютере, тогда имеет значение, на каком жёстком диске или в каком разделе будет расположен загрузчик.

Для ограничения доступа к опциям загрузки можно установить пароль на загрузчик. Для этого необходимо отметить пункт **Установить или сбросить пароль** и задать пароль в появившихся полях для ввода.

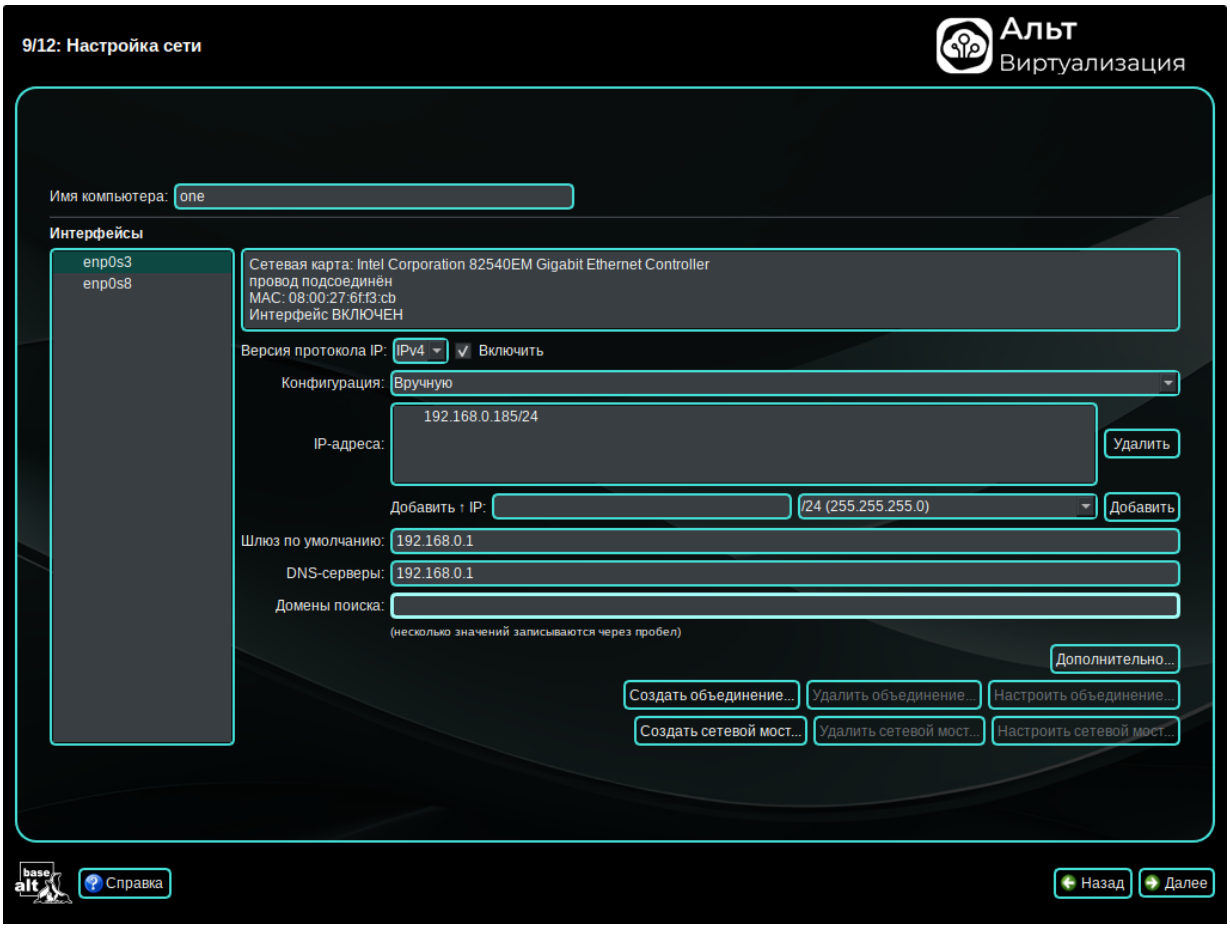
Примечание

При необходимости изменения опций загрузки при старте компьютера потребуется ввести имя пользователя «boot» и заданный на этом шаге пароль.

Для подтверждения выбора и продолжения работы программы установки необходимо нажать кнопку **Далее**.

Глава 16. Настройка сети

На этом этапе необходимо задать параметры работы сетевой карты и настройки сети: IP-адреса сетевых интерфейсов, DNS-сервер, шлюз и т.п. Конкретные значения будут зависеть от используемого вами сетевого окружения. Ручного введения настроек можно избежать при наличии в сети настроенного DHCP-сервера. В этом случае все необходимые сетевые настройки будут получены автоматически.



В окне **Настройка сети** доступны следующие поля:

- **Имя компьютера** — сетевое имя компьютера (это общий сетевой параметр, не привязанный к какому-либо конкретному интерфейсу);
- **Интерфейсы** — список доступных сетевых интерфейсов;
- **Версия протокола IP** — используемая версия IP-протокола (IPv4, IPv6);
- **Конфигурация** — способ назначения IP-адресов (**Использовать DHCP**, **Использовать Zeroconf**, **Вручную**);
- **IP-адреса** — пул назначенных IP-адресов (формируется из введённых в поле **Добавить ↑ IP**, для удаления адреса используется кнопка **Удалить**);
- **Добавить ↑ IP** — позволяет вручную ввести IP-адрес и выбрать маску сети из выпадающего списка. Для добавления адреса в пул IP-адресов нужно нажать кнопку **Добавить**;
- **Шлюз по умолчанию** — адрес маршрутизатора (шлюза), используемого по умолчанию;
- **DNS-серверы** — список DNS-серверов, используемых для разрешения доменных имён;
- **Домены поиска** — список доменов, по которым будет выполняться поиск (используется, например, при неполных DNS-запросах).

Для сохранения настроек сети и продолжения работы программы установки необходимо нажать кнопку **Далее**.

Глава 17. Администратор системы

На данном этапе загрузчик создает учетную запись администратора. В открывшемся окне необходимо ввести пароль учетной записи администратора (root). Чтобы исключить опечатки при вводе пароля, пароль учетной записи вводится дважды.

Примечание

Чтобы избежать последствий неверной раскладки клавиатуры можно просмотреть пароль, который будет сохранен. Для этого нажмите на значок глаза в поле ввода:

Для автоматической генерации пароля необходимо отметить пункт **`Создать автоматически`**. Система предложит пароль, сгенерированный автоматическим образом в соответствии с требованиями по стойкости паролей.

В любой системе Linux всегда присутствует один специальный пользователь — *администратор системы*, он же *суперпользователь*. Для него зарезервировано стандартное системное имя — root.

Администратор системы отличается от всех прочих пользователей тем, что ему позволено производить *любые*, в том числе самые разрушительные изменения в системе. Поэтому выбор пароля администратора системы — очень важный момент для *безопасности*. Любой, кто сможет ввести его правильно (узнать или подобрать), получит неограниченный доступ к системе. Даже ваши собственные неосторожные действия от имени root могут иметь катастрофические последствия для всей системы.

Важно

Стоит запомнить пароль root — его нужно будет вводить для получения права изменять настройки системы с помощью стандартных средств настройки Альт Виртуализация редакция ONE. Более подробную информацию о режиме суперпользователя вы можете прочитать в главе Режим суперпользователя.

Подтверждение введенного (или сгенерированного) пароля учетной записи администратора (root) и продолжение работы программы установки выполняется нажатием кнопки **Далее**.

Глава 18. Системный пользователь

На данном этапе программа установки создает учетную запись системного пользователя (пользователя) Альт Виртуализация редакция ONE.

Помимо администратора (root) в систему необходимо добавить, по меньшей мере, одного обычного *системного пользователя*. Работа от имени администратора системы считается опасной, поэтому повседневную работу в Linux следует выполнять от имени ограниченного в полномочиях системного пользователя.

При добавлении системного пользователя предлагается ввести имя учётной записи пользователя. Имя учётной записи всегда представляет собой одно слово, состоящее только из строчных латинских букв (заглавные запрещены), цифр и символа подчёркивания «_» (причём цифра и символ «_» не могут стоять в начале слова).

Для того чтобы исключить опечатки, пароль пользователя вводится дважды. Пароль пользователя можно создать автоматически, по аналогии с автоматическим созданием пароля суперпользователя.

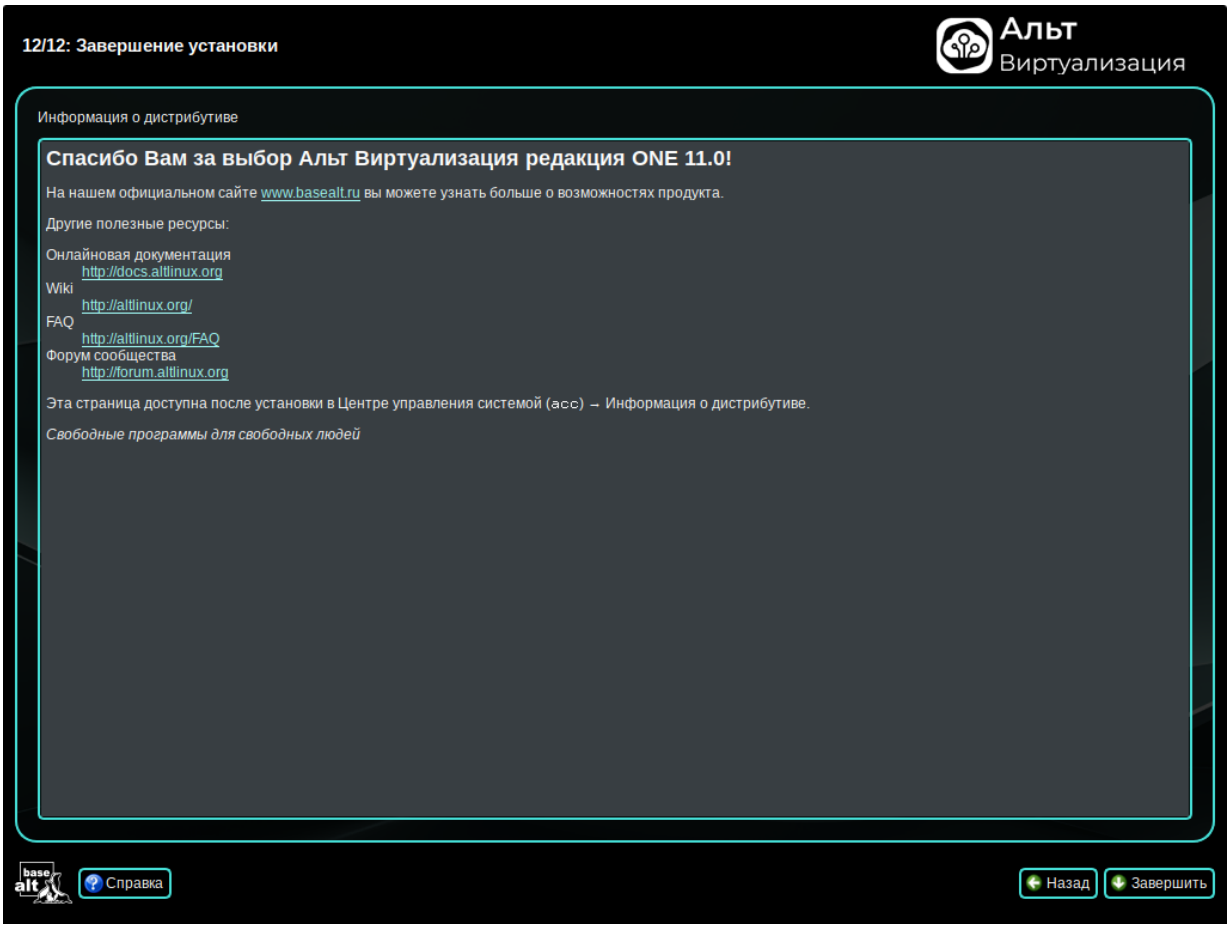
Для автоматической генерации пароля необходимо отметить пункт **Создать автоматически**. Система предложит пароль, сгенерированный автоматическим образом в соответствии с требованиями по стойкости паролей.

В процессе установки предлагается создать только одну учётную запись системного пользователя — от его имени можно выполнять задачи, не требующие привилегий суперпользователя. Учётные записи для всех прочих пользователей системы можно будет создать в любой момент после установки операционной системы.

Подтверждение введенного (или сгенерированного) пароля учетной записи системного пользователя и продолжение работы программы установки выполняется нажатием кнопки **Далее**.

Глава 19. Завершение установки

На экране последнего шага установки отображается информация о завершении установки Альт Виртуализация редакция ONE.



После нажатия кнопки **Завершить** автоматически начнется перезагрузка системы.

Не забудьте извлечь установочный DVD (если это не происходит автоматически). Далее можно загружать установленную систему в обычном режиме.

Глава 20. Обновление системы до актуального состояния

После установки системы, её лучше сразу обновить до актуального состояния. Можно не обновлять систему и сразу приступать к работе только в том случае, если вы не планируете подключаться к сети или Интернету, не собираетесь устанавливать дополнительных программ.

Для обновления системы необходимо выполнить команды (с правами администратора):

```
# apt-get update
# apt-get dist-upgrade
# update-kernel
# apt-get clean
# reboot
```

Примечание

Получить права администратора можно, выполнив в терминале команду:

```
$ su -
```

или зарегистрировавшись в системе (например, на второй консоли **Ctrl**+**Alt**+**F2**) под именем **root**. Про режим суперпользователя можно почитать в главе Режим суперпользователя.

Примечание

Подробнее про обновление пакетов можно прочитать в главах Обновление всех установленных пакетов и Обновление ядра.

Глава 21. Первая помощь

Важно

В случае возникновения каких-либо неприятностей не паникуйте, а спокойно разберитесь в сложившейся ситуации. Linux не так уж просто довести до полной неработоспособности и утраты ценных данных. Поспешные действия отчаявшегося пользователя могут привести к плачевным результатам. Помните, что решение есть, и оно обязательно найдётся!

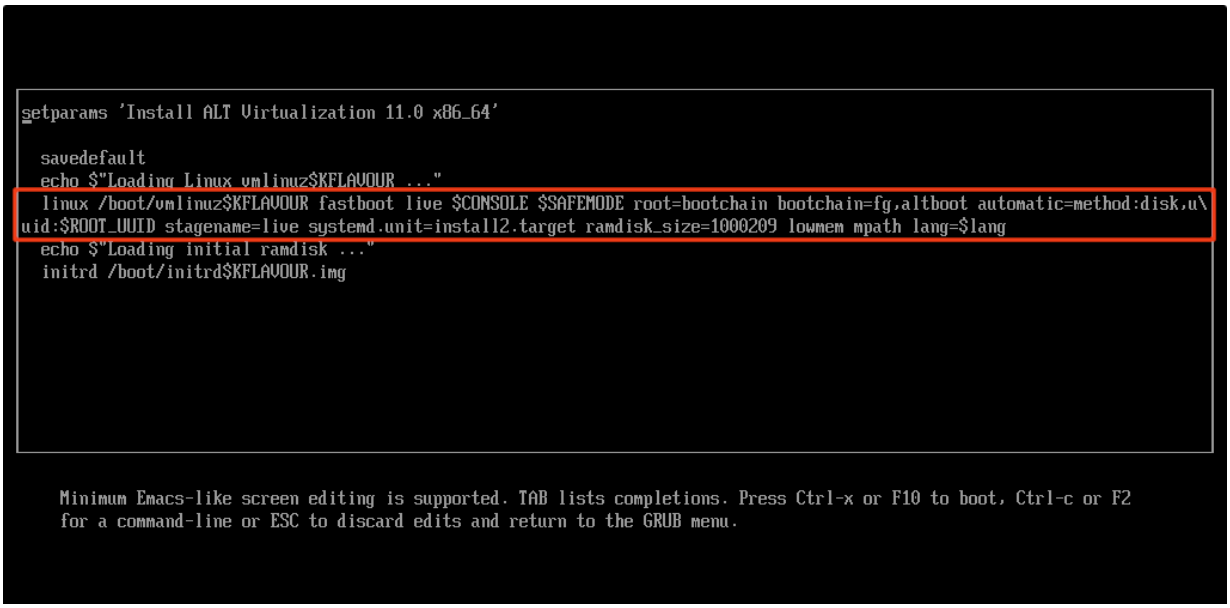
21.1. Проблемы при установке системы

Важно

При возникновении проблем с UEFI или Legacy/CSM рекомендуется изменить выбор используемого вида прошивки на другой. Не следует выбирать режим смешанной загрузки Legacy/UEFI! Рекомендуется отключить всевозможные оптимизации и ускорение UEFI-загрузки, а также отключить на время установки SecureBoot.

Если в системе не произошла настройка какого-либо компонента после стадии установки пакетов, не отчаивайтесь, доведите установку до конца, загрузитесь в систему и попытайтесь в спокойной обстановке повторить настройку.

Нажатием клавиши **E** можно вызвать редактор параметров текущего пункта загрузки. В открывшемся редакторе следует найти строку, начинающуюся с **linux /boot/vmlinuz**, в её конец дописать требуемые параметры, отделив пробелом и нажать **F10**.



Примеры параметров пункта загрузки:

- ***nomodeset*** — не использовать modeset-драйверы для видеокарты;
- ***vga=normal*** — отключить графический экран загрузки установщика;
- ***xdriver=vesa*** — явно использовать видеодрайвер vesa. Данным параметром можно явно указать нужный вариант драйвера;
- ***acpi=off noapic*** — отключение ACPI (управление питанием), если система не поддерживает ACPI полностью.

Если вы вообще не смогли установить систему (не произошла или не завершилась стадия установки пакетов), то сначала попробуйте повторить попытку в безопасном режиме (***apm=off acpi=off mce=off barrier=off vga=normal***). В безопасном режиме отключаются все параметры ядра, которые могут вызвать проблемы при загрузке. В этом режиме установка будет произведена без поддержки APIC. Возможно, у вас какое-то новое или нестандартное оборудование, но может оказаться, что оно отлично настраивается со старыми драйверами.

Если вы хотите получить точный ответ, то сообщите, пожалуйста, подробный состав вашего оборудования и подробное описание возникшей проблемы в нашей системе отслеживания ошибок.

21.2. Проблемы с загрузкой системы

Если не загружается ни одна из установленных операционных систем, то значит, есть проблема в начальном загрузчике. Такие проблемы могут возникнуть после установки системы, в случае если загрузчик все-таки не установлен или установлен с ошибкой. При установке или переустановке Windows на вашем компьютере загрузчик Linux будет перезаписан в принудительном порядке, и станет невозможно запускать Linux.

Повреждение или перезапись загрузчика никак не затрагивает остальные данные на жёстком диске, поэтому в такой ситуации очень легко вернуть работоспособность: для этого достаточно восстановить загрузчик.

Если у вас исчез загрузчик другой операционной системы или другого производителя, то внимательно почитайте соответствующее официальное руководство на предмет его восстановления. Но в большинстве случаев вам это не потребуется, так как загрузчик, входящий в состав Альт Виртуализация редакция ONE, поддерживает загрузку большинства известных операционных систем.

Для восстановления загрузчика достаточно любым доступным способом загрузить Linux и получить доступ к тому жёсткому диску, на котором находится повреждённый загрузчик. Для этого проще всего воспользоваться *восстановительным режимом*, который предусмотрен на установочном диске дистрибутива (пункт **Rescue LiveCD**).

Загрузка восстановительного режима заканчивается запросом на ввод логина суперпользователя (root):

```
Log in as root
localhost login:
```

После ввода логина логина суперпользователя (root) появится приглашение командной строки: **`[root@localhost /]#`** . Начиная с этого момента, система готова к вводу команд.

В большинстве случаев для восстановления загрузчика можно просто воспользоваться командой **`fixmbr`** без параметров. Программа попытается переустановить загрузчик в автоматическом режиме.

21.3. Полезные ссылки

Если у вас что-то не получается, вы всегда можете поискать решение на ресурсах, указанных в разделе Техническая поддержка продуктов «Базальт СПО».

Часть III. Начало использования Альт Виртуализация редакция ONE

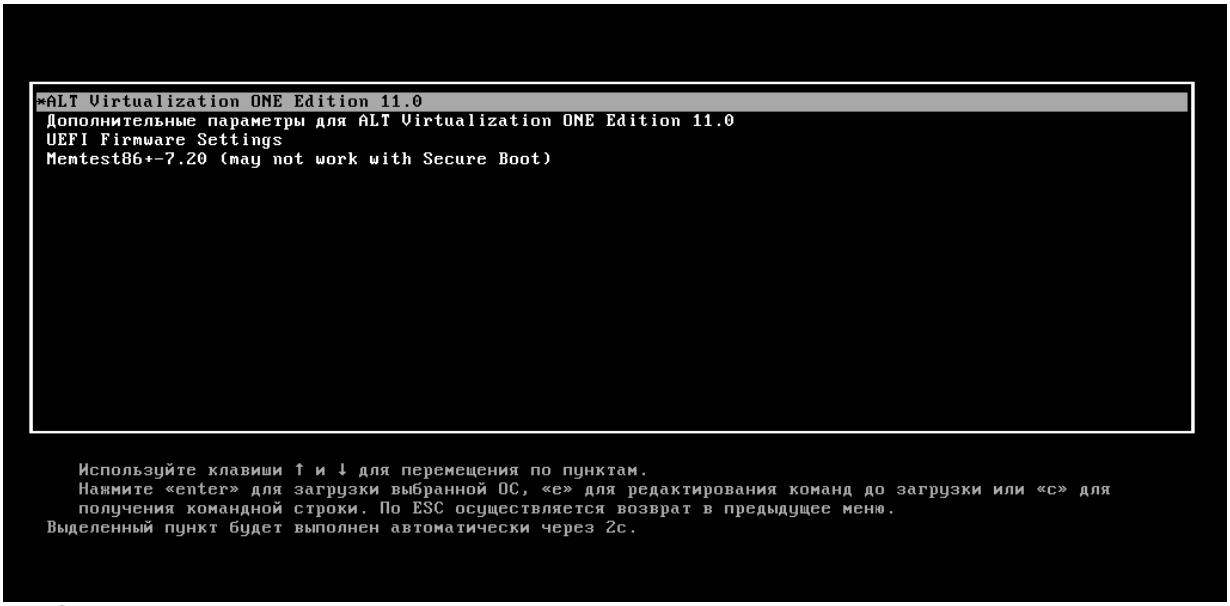
В этой части рассматривается загрузка установленной операционной системы и вход в среду рабочего стола.

Содержание

Глава 22. Загрузка системы

Запуск системы Альт Виртуализация редакция ONE выполняется автоматически после запуска компьютера и завершения инициализации BIOS.

На экране появляется меню, в котором перечислены возможные варианты загрузки операционной системы.



Важно

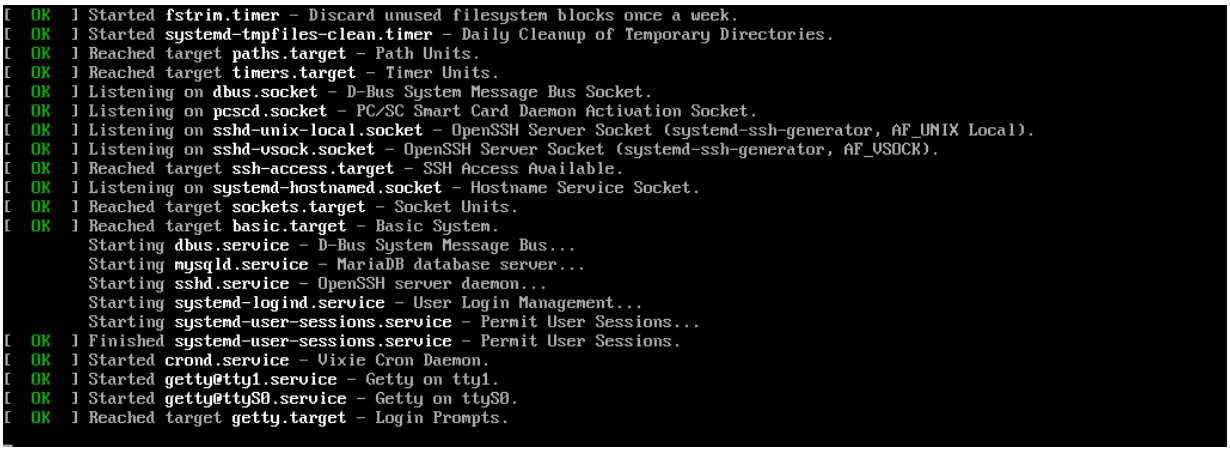
При первом запуске, если на компьютере установлено несколько операционных систем, некоторые из них могут отсутствовать в загрузочном меню. Они будут добавлены автоматически после перезагрузки. Все перечисленные в меню после перезагрузки варианты могут быть загружены загрузчиком системы.

Стрелками клавиатуры **Вверх** и **Вниз** выберите нужную операционную систему. Дополнительно к основным вариантам запуска ОС из этого меню можно загрузить Linux в безопасном режиме или запустить проверку памяти.

Загрузка операционной системы по умолчанию (первая в списке) начинается автоматически после небольшого времени ожидания (обычно несколько секунд). Нажав клавишу **Enter**, можно начать загрузку немедленно.

Нажатием клавиши **E** можно вызвать редактор параметров текущего пункта загрузки. Если система настроена правильно, то редактировать их нет необходимости.

В процессе загрузки Альт Виртуализация редакция ONE пользователь может следить за информацией процесса загрузки, которая отображает этапы запуска различных служб и программных серверов в виде отдельных строк, на экране монитора.



При этом каждая строка сопровождается индикатором вида *[OK]* или *[FAILED]*, указывающим на успешное или неудачное завершение этапа загрузки. Индикатор *[FAILED]* означает, что на данном этапе возникла ошибка, требующая внимания администратора.

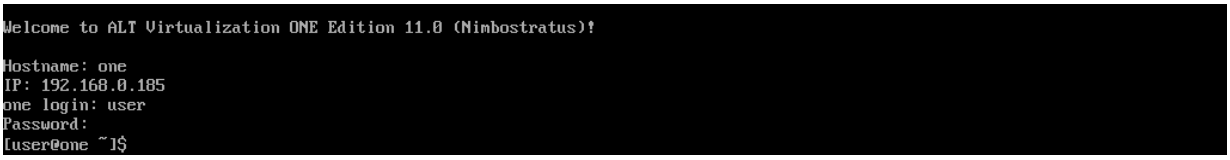
Загрузка операционной системы может занять некоторое время, в зависимости от производительности компьютера. Основные этапы загрузки операционной системы — загрузка ядра, подключение (монтирование) файловых систем, запуск системных служб — периодически могут дополняться проверкой файловых систем на наличие ошибок. В этом случае время ожидания может занять больше времени, чем обычно. Подробную информацию о шагах загрузки можно получить, нажав клавишу **Esc**.

Глава 23. Вход в систему

Стандартная установка Альт Виртуализация редакция ONE включает базовую систему, работающую в консольном режиме.

При загрузке в консольном режиме работа загрузчика Альт Виртуализация редакция ONE завершается запросом на ввод логина и пароля учетной записи.

После загрузки будут показаны имя и IP-адрес компьютера. Для дальнейшего входа в систему необходимо ввести логин и пароль учетной записи пользователя.



В случае успешного прохождения процедуры аутентификации и идентификации будет выполнен вход в систему. ОС Альт Виртуализация редакция ONE перейдет к штатному режиму работы и предоставит дальнейший доступ к консоли.

В процессе работы ОС Альт Виртуализация редакция ONE доступно 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.

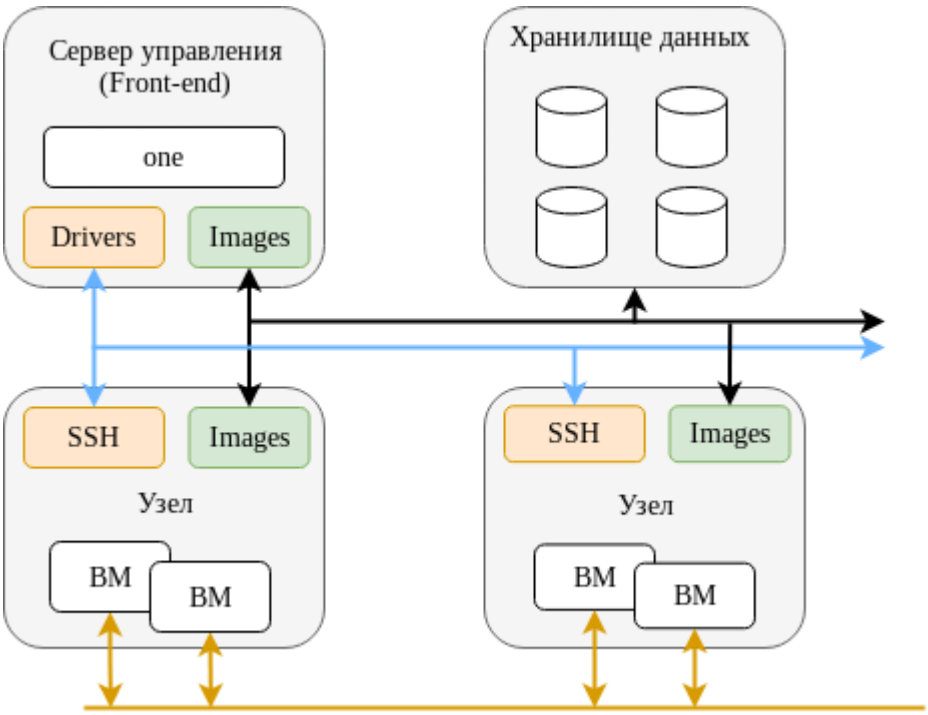
Часть IV. OpenNebula

OpenNebula — это платформа для управления гетерогенной виртуальной инфраструктурой в распределённых центрах обработки данных. Она поддерживает развёртывание частных, публичных и гибридных облаков класса Infrastructure as a Service (IaaS).

Облачная архитектура определяется тремя ключевыми элементами: системой хранения данных, сетевой инфраструктурой и системой виртуализации.

OpenNebula состоит из следующих компонентов:

- Сервер управления (Front-end) — центральный узел, выполняющий основные сервисы OpenNebula;
- Узлы виртуализации — физические серверы, на которых запускаются виртуальные машины;
- Хранилище данных — система хранения образов виртуальных машин, их дисков и метаданных;
- Физическая сеть — обеспечивает связность компонентов инфраструктуры и изоляцию трафика виртуальных машин (через VLAN, VXLAN и другие технологии).



Примечание

Компоненты OpenNebula устанавливаются в систему автоматически, если при установке дистрибутива выбрать один из следующих профилей: **Вычислительный узел Opennebula KVM**, **Вычислительный узел Opennebula LXC** или **Сервер управления Opennebula** (см. главу Установка системы).

Содержание

Глава 24. Планирование ресурсов

24.1. Сервер управления

Сервер управления (Front-end) — это «мозг» облачной инфраструктуры OpenNebula. Он отвечает за:

- управление жизненным циклом ВМ;

- планирование и распределение ресурсов;
- взаимодействие с узлами виртуализации и системами хранения;
- предоставление API и веб-интерфейса управления (Sunstone).

Таблица 24.1. Минимальные требования к серверу управления

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

Максимальное количество узлов виртуализации, управляемых одним сервером управления (экземпляром OpenNebula), зависит от конфигурации системы, в первую очередь от производительности базы данных, подсистемы хранения и сети.

Один экземпляр OpenNebula способен управлять до 2500 узлов виртуализации и до 10 000 виртуальных машин при корректной настройке компонентов (прежде всего базы данных, в которой хранится состояние облака).

Для масштабирования за пределы этих значений OpenNebula поддерживает федеративные развёртывания, объединяющие несколько зон в единую облачную инфраструктуру.

24.2. Узлы виртуализации

Узлы виртуализации — это физические серверы, на которых выполняются виртуальные машины (ВМ). Подсистема виртуализации в OpenNebula отвечает за взаимодействие с гипервизором и управление жизненным циклом ВМ.

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

- Процессор (CPU). Каждое ядро, выделяемое ВМ, должно соответствовать физическому ядру. Например, для размещения 40 ВМ по 2 ядра требуется 80 физических ядер. Они могут быть распределены, например, по 10 узлам с 8 ядрами или 5 узлам с 16 ядрами.

  В OpenNebula для управления вычислительными ресурсами используются атрибуты ***CPU*** и ***VCPU***:

  - ***CPU*** определяет количество физических ядер, выделяемых ВМ;
  - ***VCPU*** — количество виртуальных процессоров, видимых гостевой ОС.

- Память. По умолчанию OpenNebula не разрешает выделять гостевым системам больше оперативной памяти, чем физически доступно на узле. Рекомендуется резервировать не менее 10 % оперативной памяти каждого узла для нужд гипервизора и системных процессов.

Например, для 45 ВМ с 2 ГБ ОЗУ каждая требуется 90 ГБ оперативной памяти. Важным фактором является количество физических серверов: каждый сервер должен иметь 10% запас для работы гипервизора. Если использовать 10 узлов с 10 ГБ ОЗУ каждый, то после резервирования 1 ГБ (10 %) на нужды гипервизора каждый узел предоставит 9 ГБ для ВМ. В сумме это даёт 90 ГБ — достаточный объём для указанного сценария.

24.3. Хранилище данных

OpenNebula работает с двумя типами данных: виртуальных машин и дисками ВМ (подробнее см. Работа с хранилищами в OpenNebula).

Один из ключевых механизмов управления хранилищем является использование квот, ограничивающих максимальное количество виртуальных машин и объём оперативной памяти, который может запрашивать пользователь. Администратор также должен обеспечить достаточный объём системного хранилища для хранения образов и метаданных с учётом установленных квот. При необходимости возможно подключение дополнительных хранилищ.

Планирование подсистемы хранения является критически важным аспектом, так как напрямую влияет на производительность облака. Требуемый объём и конфигурация зависят от используемой технологии, например:

- Ceph — минимум 3 сервера со следующей конфигурацией: 5 дисков по 1 ТБ, 16 ГБ ОЗУ, два 4-ядерных процессора, не менее двух сетевых интерфейсов;
- NFS — выделенный файловый сервер с RAID-массивом и резервированием;
- LVM — SAN-хранилище с использованием iSCSI или Fibre Channel.

Важно

Недостаток свободного места в системном хранилище приведёт к невозможности запуска виртуальных машин, даже если в хранилище образов имеется доступное пространство.

24.4. Сетевая инфраструктура

Сетевая инфраструктура должна обеспечивать достаточную пропускную способность и низкую задержку между всеми компонентами облака.

Рекомендуемая конфигурация:

- 2 сетевых интерфейса на сервере управления (для управления и резервирования);
- 4 сетевых интерфейса на каждом узле виртуализации: для публичного трафика ВМ, внутренней коммуникации, управления и подключения к хранилищу.

Примечание

Рекомендуется использовать раздельные VLAN или физически изолированные сети для различных типов трафика, чтобы избежать конкуренции за пропускную способность и повысить отказоустойчивость.

Глава 25. Запуск сервера управления OpenNebula

25.1. Установка пароля для пользователя oneadmin

Во время установки OpenNebula автоматически создаётся пользователь oneadmin, от имени которого выполняются все дальнейшие операции по управлению системой.

При установке OpenNebula также формируется файл **`/var/lib/one/.one/one_auth`** со случайно сгенерированным паролем. Необходимо изменить этот пароль перед первым запуском OpenNebula:

1. Установите пароль для пользователя oneadmin:

   ```
   # passwd oneadmin
   ```

2. Войдите под пользователем oneadmin и замените содержимое **`/var/lib/one/.one/one_auth`**. Он должен содержать следующее: oneadmin:<пароль>. Например:

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

25.2. Настройка MySQL (MariaDB) для хранения конфигурации

По умолчанию OpenNebula использует встроенную базу SQLite. Для среды промышленной эксплуатации рекомендуется использовать MariaDB или MySQL. Настройку следует выполнить до первого запуска OpenNebula, чтобы избежать проблем с учётными данными oneadmin и serveradmin.

Примечание

Безопасность MySQL можно настроить с помощью:

```
# mysql_secure_installation
```

Создайте пользователя базы данных, предоставьте ему права доступа к базе данных opennebula (эта база данных будет создана при первом запуске OpenNebula) и настройте уровень изоляции:

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

MariaDB > CREATE USER 'oneadmin'@'localhost' IDENTIFIED BY '<пароль>';
Query OK, 0 rows affected (0,007 sec)

MariaDB > GRANT ALL PRIVILEGES ON opennebula.* TO 'oneadmin'@'localhost';
Query OK, 0 rows affected (0,004 sec)

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

MariaDB > FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)

MariaDB > exit
```

В конфигурационном файле **`/etc/one/oned.conf`** закомментируйте секцию sqlite, раскомментируйте и настройте секцию MySQL:

```
#DB = [ BACKEND = "sqlite",
#       TIMEOUT = 2500 ]

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

где:

- ***SERVER*** — IP-адрес/имя машины, на которой запущен сервер MySQL;
- ***PORT*** — порт для подключения к СУБД. Значение 0 означает использование порта по умолчанию (3306 для MySQL/MariaDB);
- ***USER*** — имя пользователя MySQL;
- ***PASSWD*** — пароль MySQL;

- ***DB_NAME*** — имя базы данных MySQL, которую будет использовать OpenNebula;
- ***CONNECTIONS*** — максимальное количество подключений;
- ***COMPARE_BINARY*** — включает учёт регистра при сравнении имён (используется оператор BINARY в SQL-запросах).

25.3. Запуск OpenNebula

Включите и запустите службы:

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

25.4. Проверка установки

После запуска OpenNebula следует проверить, что команды могут подключаться к демону OpenNebula. Это можно сделать в командной строке или в веб-интерфейсе Sunstone.

Убедиться, что CLI-клиент работает, можно выполнив команду (от имени пользователя oneadmin):

```
$ 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
```

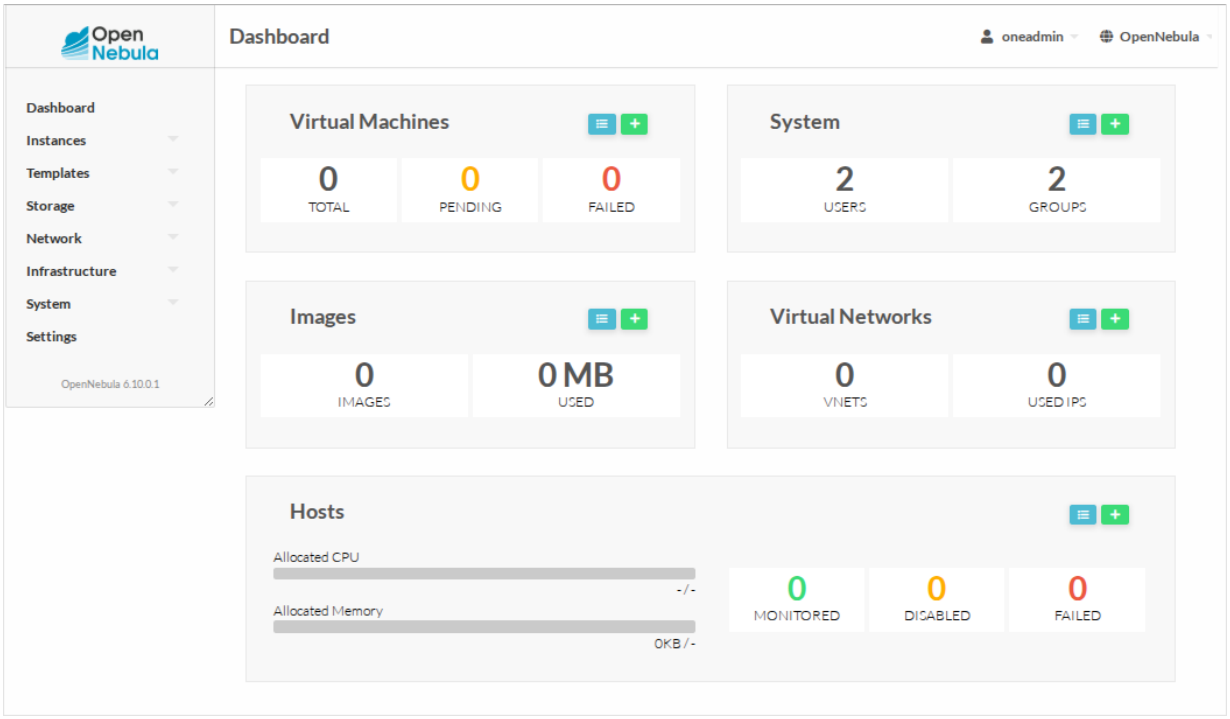
Также можно войти в веб-интерфейс Sunstone, перейдя по адресу:

```
http://<IP-адрес_сервера>:9869
```

Если всё в порядке, будет предложена страница входа.

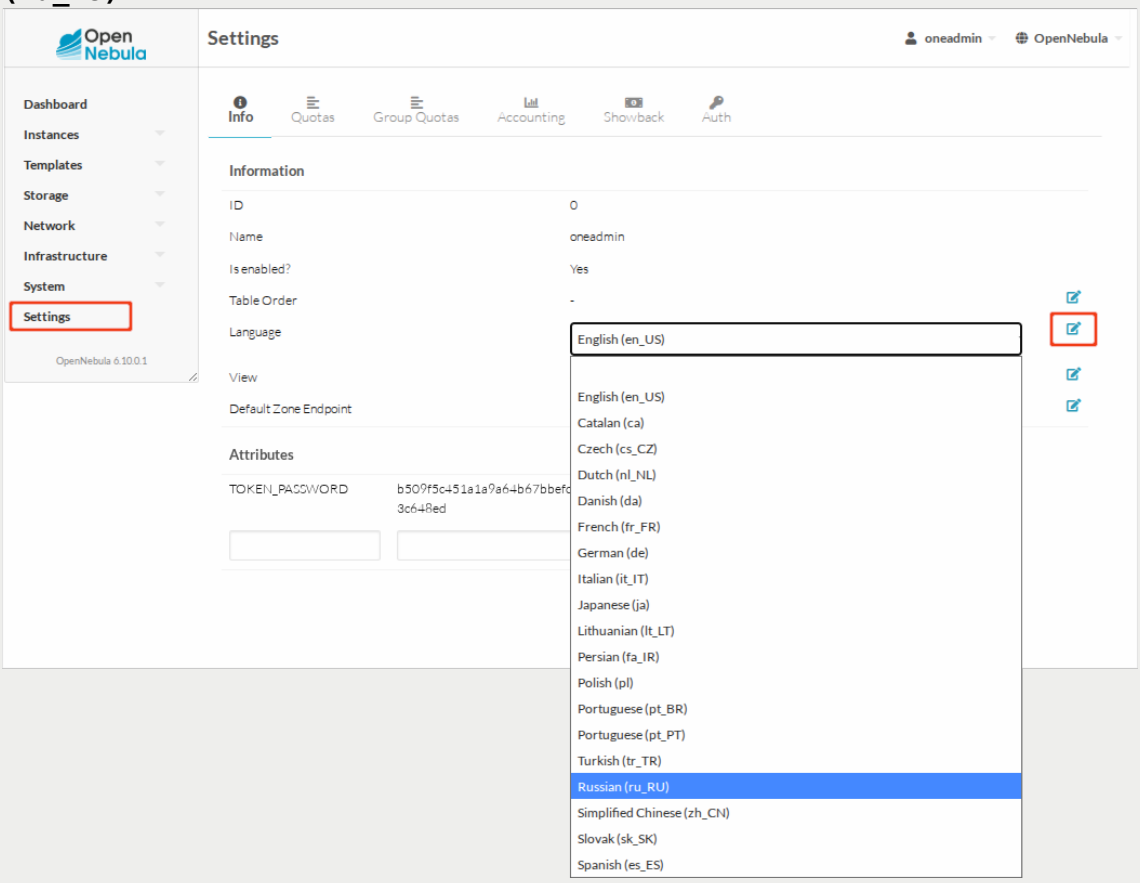
Введите в соответствующие поля имя пользователя (**oneadmin**) и его пользователя (из файла **`/var/lib/one/.one/one_auth`**):

После входа откроется панель управления Sunstone:

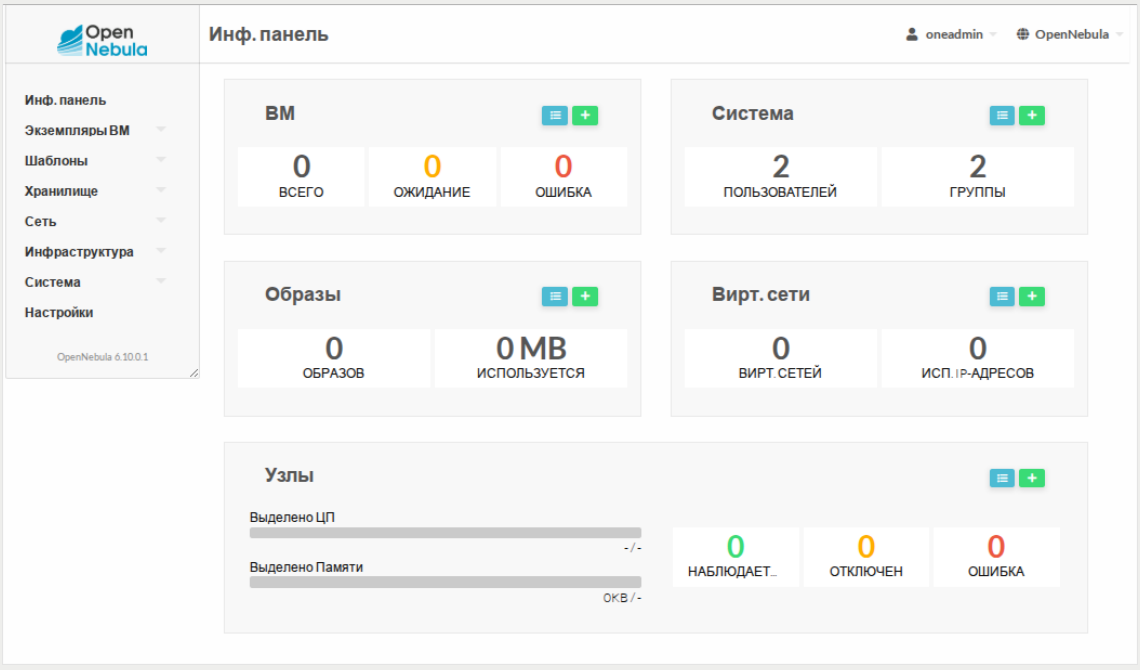


Примечание

Чтобы изменить язык интерфейса, в левом меню выберите пункт **Settings**, и на открывшейся странице в выпадающем списке **Language** выберите **Russian (ru_RU)**:



Язык интерфейса будет изменён на русский:



25.5. Ключи для доступа по SSH

Сервер управления OpenNebula (Front-end) должен иметь беспарольный SSH-доступ ко всем узлам гипервизора от имени пользователя oneadmin. Для корректной работы OpenNebula используются следующие типы соединений:

- от сервера управления к самому серверу управления;
- от сервера управления к каждому узлу гипервизора;
- от сервера управления к узлу гипервизора с последующим подключением к другому узлу гипервизора (для операций миграции ВМ);
- от сервера управления к узлу гипервизора с последующим подключением обратно к серверу управления (для операций копирования данных).

Важно

Необходимо убедиться, что сервер управления и все узлы гипервизора могут подключаться друг к другу по SSH без ручного вмешательства (без запроса пароля и подтверждения ключей).

Для включения беспарольных подключений необходимо распространить открытый SSH-ключ пользователя **oneadmin** со всех машин в файл **`/var/lib/one/.ssh/authorized_keys`** на всех узлах инфраструктуры.

При установке сервера управления OpenNebula автоматически генерируется пара SSH-ключей для пользователя oneadmin (**`/var/lib/one/.ssh/id_rsa`** и **`/var/lib/one/.ssh/id_rsa.pub`**), а также открытый ключ добавляется в файл **`/var/lib/one/.ssh/authorized_keys`**.

Примечание

На узлах виртуализации пара SSH-ключей не генерируется автоматически. При необходимости ключи можно создать вручную, выполнив команду:

```
$ ssh-keygen -t rsa -f /var/lib/one/.ssh/id_rsa -N ""
```

25.5.1. Управление списком доверенных узлов (known_hosts)

Необходимо подготовить и поддерживать актуальный список открытых SSH-ключей узлов (файл **`known_hosts`**), чтобы все взаимодействующие стороны могли идентифицировать друг друга без запроса подтверждения.

Файл располагается по пути **`/var/lib/one/.ssh/known_hosts`**. Для его создания и обновления используется команда **`ssh-keyscan`**, например:

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

Команду следует выполнить от имени пользователя oneadmin на сервере управления, после чего полученный файл необходимо скопировать на все узлы гипервизора.

Примечание

Команду **ssh-keyscan** необходимо выполнять как для имён узлов, так и для их IP-адресов:

```
$ ssh-keyscan <IP-узел1> <hostname-узел1> ... >> /var/lib/one/.ssh/
known_hosts
```

Например:

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

Важно

Файл **known_hosts** необходимо обновлять и распространять заново каждый раз, когда:

- переустанавливается какой-либо узел;
- SSH-ключи узла генерируются заново.

25.5.2. Включение входа без пароля

Для включения беспарольного входа в инфраструктуре необходимо скопировать конфигурацию аутентификации пользователя oneadmin с сервера управления на все узлы гипервизора.

В рамках этой процедуры требуется распространить:

- файл **known_hosts**, созданный на предыдущем шаге;
- открытый SSH-ключ пользователя oneadmin с сервера управления.

Для копирования открытого ключа необходимо выполнить следующие команды для каждого узла:

```
# su - oneadmin
$ ssh-copy-id -i /var/lib/one/.ssh/id_rsa.pub <узел1>
$ ssh-copy-id -i /var/lib/one/.ssh/id_rsa.pub <узел2>
$ ssh-copy-id -i /var/lib/one/.ssh/id_rsa.pub <узел3>
...
```

Распространение файла **known_hosts**:

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

Примечание

Распространять закрытый SSH-ключ пользователя oneadmin с сервера управления не требуется. Он безопасно используется сервером управления и может делегироваться на узлы гипервизора с помощью стандартного SSH-агента аутентификации, запущенного на сервере управления.

25.5.3. Проверка корректности SSH-подключений

После завершения настройки необходимо убедиться, что ни одно из следующих подключений, выполняемых от имени пользователя oneadmin, не запрашивает пароль и не завершается ошибкой:

- от сервера управления к самому серверу управления;
- от сервера управления ко всем узлам гипервизора;
- от каждого узла гипервизора ко всем другим узлам;
- от всех узлов гипервизора к серверу управления.

Пример проверки, выполняемой на сервере управления:

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

# от сервера управления к узлу1, обратно на сервер управления и к другим
узлам
ssh <узел1>
ssh <сервер_управления>
exit
ssh <узел2>
exit
ssh <узел3>
exit
exit
```

Аналогичную проверку следует выполнить для всех узлов гипервизора.

25.6. Конфигурация сети

Сервисам, работающим на сервере управления, необходим сетевой доступ к узлам гипервизора для управления и мониторинга узлов виртуализации, а также для передачи файлов образов ВМ. Для этих целей настоятельно рекомендуется выделить отдельную физическую или логическую сеть, изолированную от публичного трафика.

Можно использовать простейшую сетевую модель — сетевой драйвер **Bridge**. Для его работы необходимо на каждом узле гипервизора настроить Linux-мост (например, br0) и добавить в него физический сетевой интерфейс. При создании сетевого шаблона в OpenNebula следует указать имя этого моста. В этом случае OpenNebula будет подключать ВМ к указанному мосту, обеспечивая их сетевую связность с физической сетью, к которой подключён данный мост.

Примечание

Конфигурация виртуальных мостов требуется только на узлах гипервизора, но не на сервере управления. Точные имена ресурсов (br0, br1 и т. п.) не имеют принципиального значения, однако они должны совпадать на всех узлах, чтобы сетевые шаблоны OpenNebula работали корректно.

Глава 26. Установка и настройка узлов

26.1. Установка и настройка узла OpenNebula KVM

Перед добавлением узла типа KVM в OpenNebula необходимо предварительно настроить соответствующий сервер.

Для создания узла типа KVM при установке дистрибутива следует выбрать профиль **`Вычислительный узел Opennebula KVM`** (см. главу Установка системы):

Примечание

В уже установленной системе узел KVM можно настроить, установив пакет *opennebula-node-kvm*:

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

Затем включите и запустите службу libvirtd:

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

После установки узла:

1. Задайте пароль пользователю oneadmin:

   ```
   # passwd oneadmin
   ```

2. Настройте беспарольный SSH-доступ (см. раздел Ключи для доступа по SSH).

26.2. Установка и настройка узла OpenNebula LXC

LXD — это менеджер системных контейнеров на основе LXC, интегрируемый с OpenNebula.

Перед добавлением узла типа LXC в OpenNebula необходимо настроить соответствующий сервер.

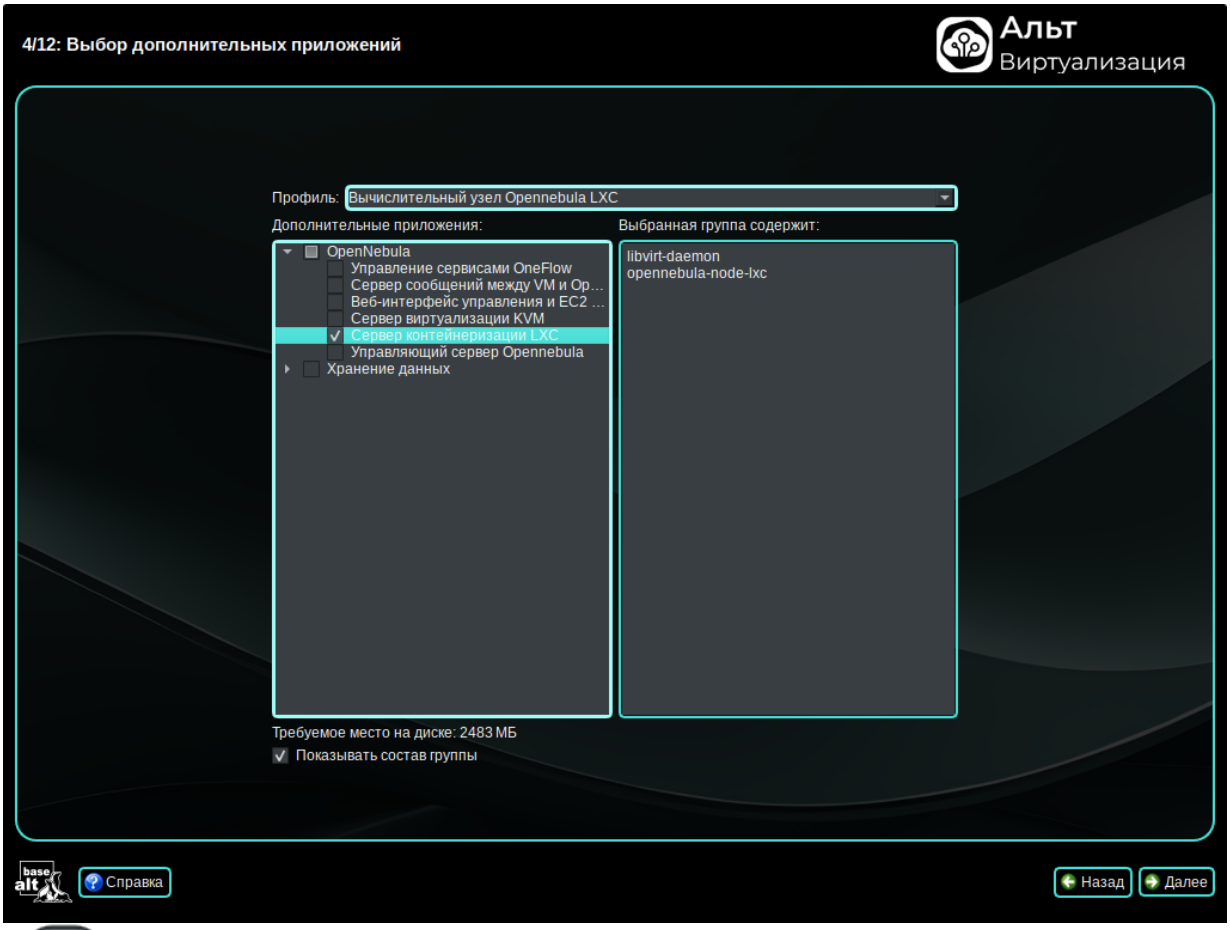
Важно

Для работы с LXC в OpenNebula требуется настроить два хранилища:

- хранилище образов;
- хранилище системных данных.

Эти хранилища не должны использовать формат образов qcow2. Рекомендуется применять хранилища типов shared, ssh или lvm.

Для создания узла типа LXC при установке дистрибутива нужно выбрать профиль **`Вычислительный узел Opennebula LXC`** (см. главу Установка системы):



Примечание

В уже установленной системе узел LXC можно настроить, установив пакет *opennebula-node-lxc*:

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

После создания узла:

1. Задайте пароль пользователю oneadmin:

```
# passwd oneadmin
```

2. Настройте беспарольный SSH-доступ (см. раздел Ключи для доступа по SSH).

Глава 27. Узлы в OpenNebula

Узлы — это серверы, управляемые OpenNebula и отвечающие за запуск ВМ. Для использования этих узлов в OpenNebula необходимо зарегистрировать их, чтобы они отслеживались и были доступны планировщику.

Управление узлами осуществляется с помощью:

- командной строки (**`onehost`**);
- веб-интерфейса Sunstone.

Примечание

Перед добавлением узла следует убедиться, что с сервера управления можно подключиться к нему по SSH без запроса пароля от имени пользователя oneadmin.

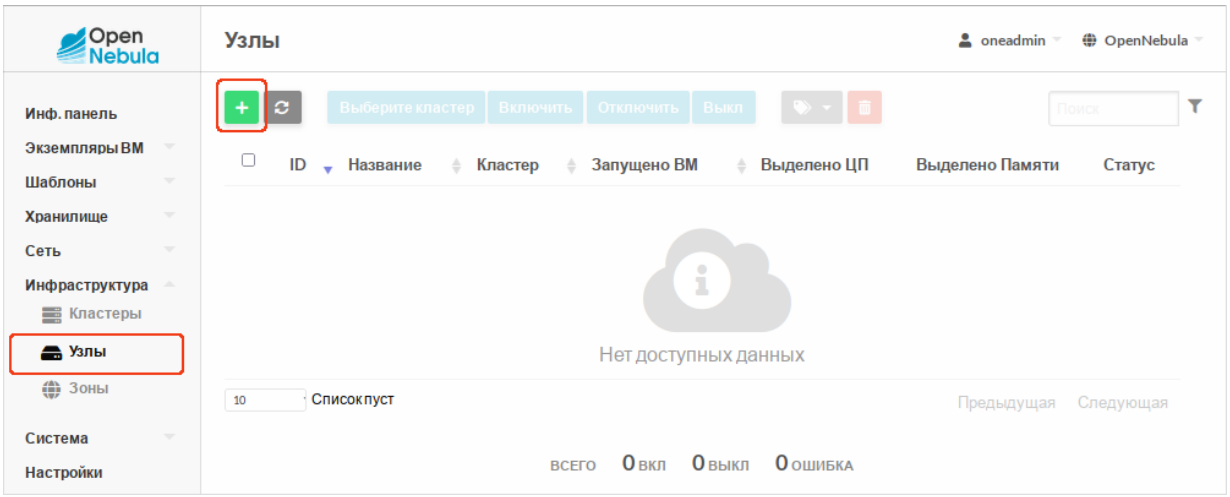
Примечание

Если узел переходит в состояние ошибки вместо состояния ВКЛ (on), проверьте журнал OpenNebula **`/var/log/one/oned.log`**. Проблема может быть связана с подключением по SSH.

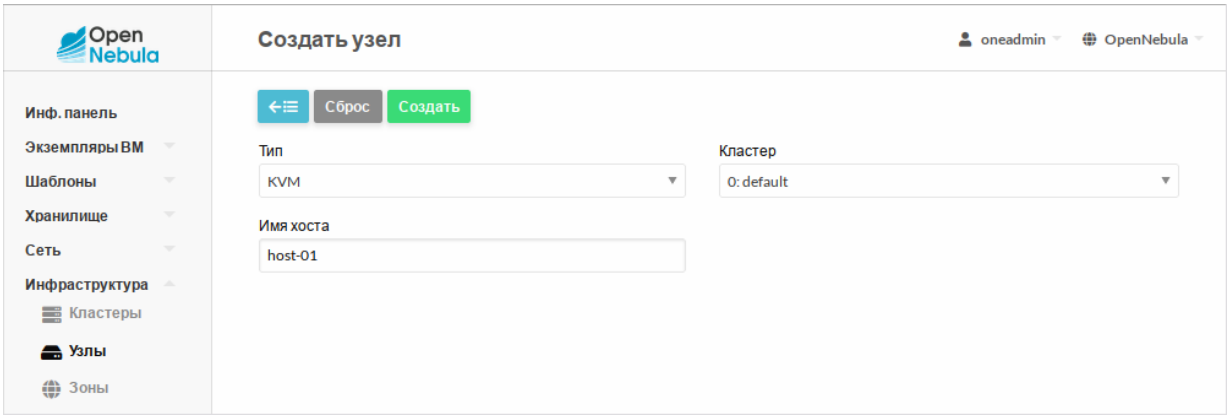
27.1. Управление узлами в Sunstone

27.1.1. Добавление узла типа KVM

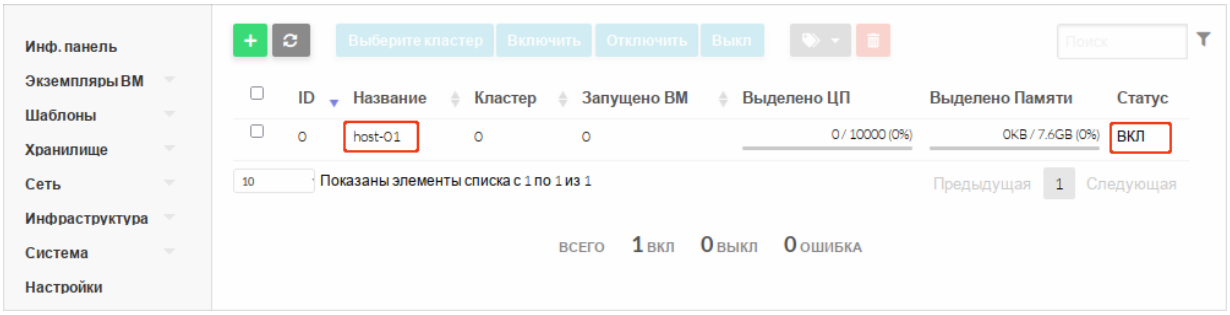
Для добавления узла необходимо в левом меню выбрать **Инфраструктура** → **Узлы** и на загруженной странице нажать кнопку **+**:



Далее необходимо указать тип виртуализации (KVM), в поле **`Имя хоста`** ввести IP-адрес или DNS-имя узла и нажать кнопку **`Создать`**:

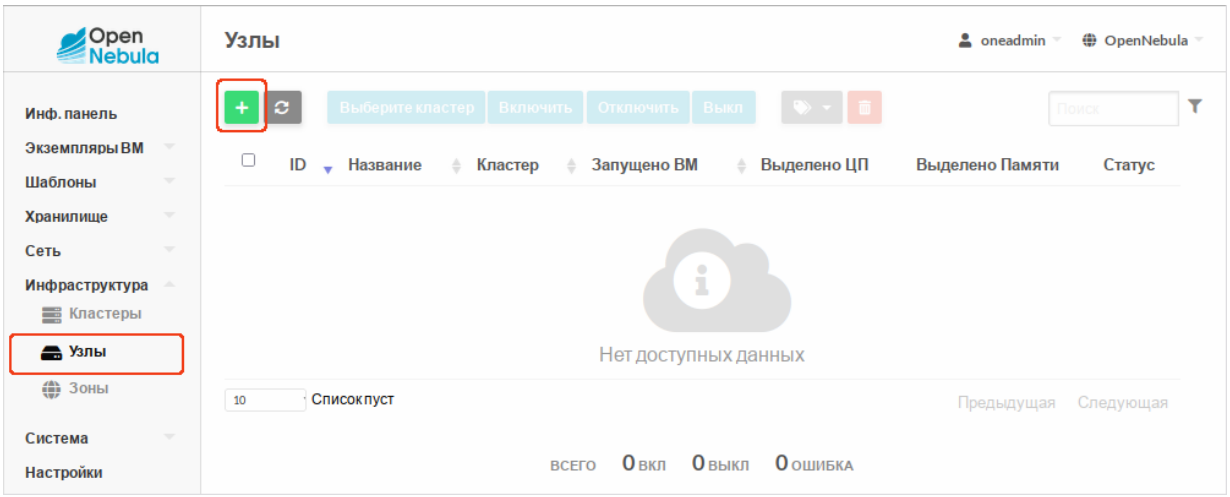


Затем следует вернуться к списку узлов и дождаться, пока статус узла изменится на **ВКЛ** (обычно занимает 20–60 секунд, для обновления списка можно нажать кнопку **`Обновить`**):

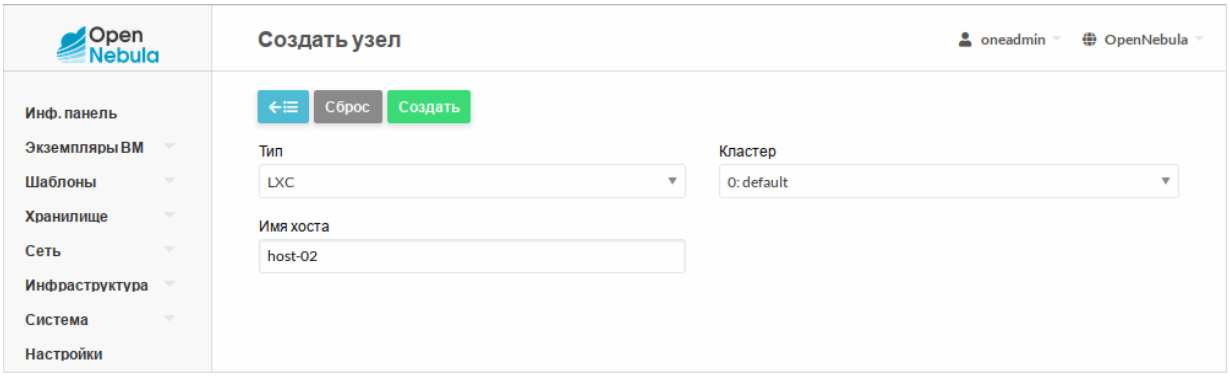


27.1.2. Добавление узла типа LXC

Для добавления узла типа LXC на сервере OpenNebula необходимо в левом меню выбрать **Инфраструктура** → **Узлы** и на загруженной странице нажать кнопку **+**:



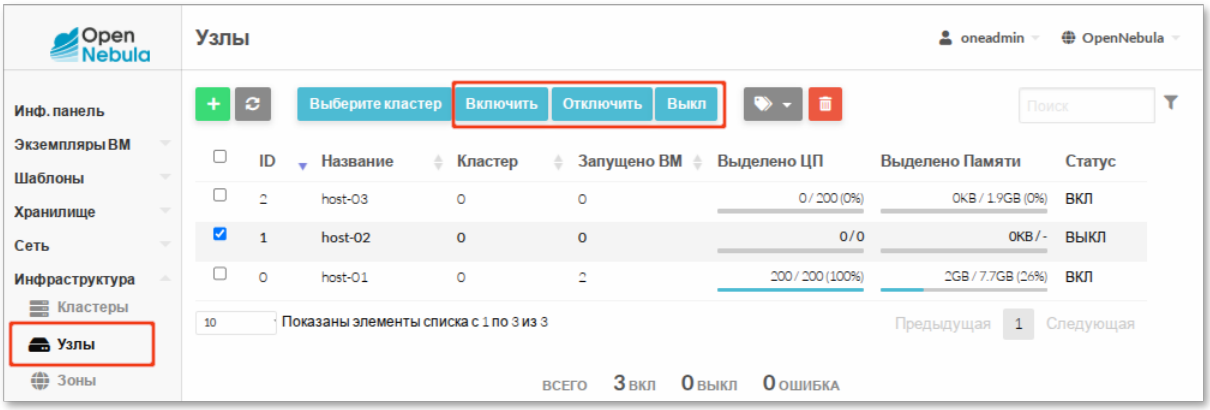
В открывшейся форме следует указать тип виртуализации — LXC, в поле **`Имя хоста`** ввести IP-адрес или DNS-имя узла и нажать кнопку **`Создать`**:



Затем следует вернуться к списку узлов и дождаться, пока статус узла изменится на **ВКЛ** (обычно занимает 20–60 секунд, для обновления списка можно нажать кнопку **`Обновить`**).

27.1.3. Управление состоянием узлов

На вкладке **Инфраструктура** → **Узлы** можно включать, отключать, удалять узлы, а также просматривать информацию о них.



27.2. Управление узлами через командную строку

Утилита **`onehost`** предоставляет полный контроль над узлами OpenNebula. Описание всех доступных опций утилиты **`onehost`** можно получить, выполнив команду:

```
$ man onehost
```

Добавление узла KVM:

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

где:

- ***--im*** — драйвер Information Manager;
- ***--vm*** — драйвер Virtual Machine Manager.

Добавление узла LXC:

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

Просмотр списка узлов:

```
$ 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
```

Узел можно указывать по ID или имени.

Удаление узла:

```
$ onehost delete 1 # с указанием ID
$ onehost delete host-02 # с указанием имени
```

Изменение состояния узла:

```
$ onehost disable host-01 // деактивировать (новые ВМ не запускаются)
$ onehost enable host-01 // активировать
$ onehost offline host-01 // перевести в автономный режим
```

Команды **`disable`** и **`offline`** не останавливают и не мигрируют уже запущенные ВМ. Чтобы безопасно вывести узел из эксплуатации, следует использовать команду **`flush`**:

```
$ onehost flush host-01
```

Эта команда мигрирует все активные ВМ на другие узлы и отключает указанный узел.

Поведение **`flush`** можно настроить в файле конфигурации CLI **`/etc/one/cli/onehost.yaml`**:

```
:default_actions:
  - :flush: resched # или delete-recreate
```

где:

- **`resched`** — перепланирование (миграция);

- **`delete-recreate`** — остановка и пересоздание ВМ на другом узле.

Просмотр информации об узле:

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

Вывод данной команды содержит:

- общие сведения об узле;
- ресурсы (CPU, память — раздел Host Shares);
- информацию о локальном хранилище (если используется);
- данные мониторинга;
- список запущенных виртуальных машин.

27.3. Режимы работы

Для управления жизненным циклом узла можно установить различные режимы работы: включён (on), отключён (dsbl) и офлайн (off).

Таблица 27.1. Режимы работы

Режим	Мониторинг	Запуск ВМ (ручной)	Запуск ВМ (через планировщик)	Описание
ENABLED (on)	Да	Да	Да	Узел полностью рабочий
DISABLED (dsbl)	Да	Да	Нет	Узел отключен от планировщика (для обслуживания), но можно запускать ВМ вручную
OFFLINE (off)	Нет	Нет	Нет	Узел полностью выключен из системы
ERROR (err)	Да	Да	Нет	Ошибка мониторинга
RETRY (retry)	Да	Да	Нет	Повторная попытка мониторинга после ошибки

Глава 28. Виртуальные сети

OpenNebula позволяет создавать виртуальные сети, абстрагируя их от физической инфраструктуры.

При запуске ВМ её сетевые интерфейсы (определённые в шаблоне с помощью атрибутов NIC) подключаются к соответствующим сетевым устройствам гипервизора в соответствии с настройками выбранной виртуальной сети. Это обеспечивает ВМ доступ как к частным, так и к публичным сетям.

Управление виртуальными сетями осуществляется через:

- командную строку (**`onevnet`**);
- веб-интерфейс Sunstone.

Примечание

Подробное описание команды **`onevnet`** доступно в справочном руководстве:

```
$ man onevnet
```

Примеры:

- получить список виртуальных сетей:

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

- вывести подробную информацию о сети:

```
$ onevnet show 0
```

Все операции с виртуальными сетями также доступны в веб-интерфейсе:

OpenNebula поддерживает следующие сетевые режимы:

- **Bridged** (режим *Сетевой мост*) — ВМ подключается напрямую к Linux-мосту на узле;
- **802.1Q** (режим *VLAN*) — изоляция трафика через VLAN-теги на общем мосту;

- **VXLAN** — изоляция через инкапсуляцию VXLAN поверх IP-сети;
- **Open vSwitch** — использование моста Open vSwitch с поддержкой VLAN;
- **Open vSwitch — VXLAN** — комбинированная изоляция с применением OVS и VXLAN.

Тип сетевого режима задаётся атрибутом ***VN_MAD*** в шаблоне виртуальной сети.

28.1. Режим Bridged

В этом режиме трафик ВМ передаётся напрямую через Linux-мост на узле гипервизора. OpenNebula поддерживает три варианта реализации Bridged:

- ***dummy*** — без настройки моста (только для тестирования);
- ***bridge*** (**Bridged**) — стандартный мост без фильтрации трафика;
- ***fw*** (**Bridged & Security Groups**) — мост с применением правил групп безопасности с использованием iptables.

Примечание

Для работы групп безопасности в режиме fw должен быть установлен пакет *ebtables*.

28.1.1. Глобальная конфигурация

Параметры поведения сетевого драйвера настраиваются в файле **`/var/lib/one/remotes/etc/vnm/OpenNebulaNetwork.conf`** (см. табл. Параметры конфигурации).

Таблица 28.1. Параметры конфигурации

Параметр	Описание
:ipset_maxelem	Максимальное число записей в IP-наборах (для групп безопасности)
:keep_empty_bridge	Сохранять мосты без подключённых интерфейсов (true/false)
:ip_bridge_conf	Хеш параметров, передаваемых команде **`ip`** при создании моста (**`ip link add <bridge> type bridge ...`**)

Примечание

После изменения конфигурации необходимо синхронизировать файлы со всеми узлами кластера:

```
$ onehost sync -f
```

Требования к инфраструктуре:

- на каждом узле виртуализации должен быть создан сетевой мост с именем, указанным в атрибуте ***BRIDGE***;

- имя моста должно совпадать на всех узлах, входящих в кластер.

Важно

Не используйте интерфейс управления (через который осуществляется SSH-доступ к узлу) в качестве ***PHYDEV***.

Если мост будет создан на этом интерфейсе, соединение с узлом будет потеряно. Всегда выделяйте отдельный физический интерфейс для виртуальных сетей (например, enp3s0).

Примечание

Если мост, указанный в атрибуте ***BRIDGE***, отсутствует, OpenNebula не создаёт его автоматически, кроме случаев, когда задан ***PHYDEV***. В этом случае мост будет создан при первом запуске ВМ.

28.1.2. Создание виртуальной сети

Пример создания виртуальной сети с использованием конфигурационного файла:

1. Создайте файл **`net-bridged.conf`**:

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

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

2. Загрузите сеть, выполнив команду:

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

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

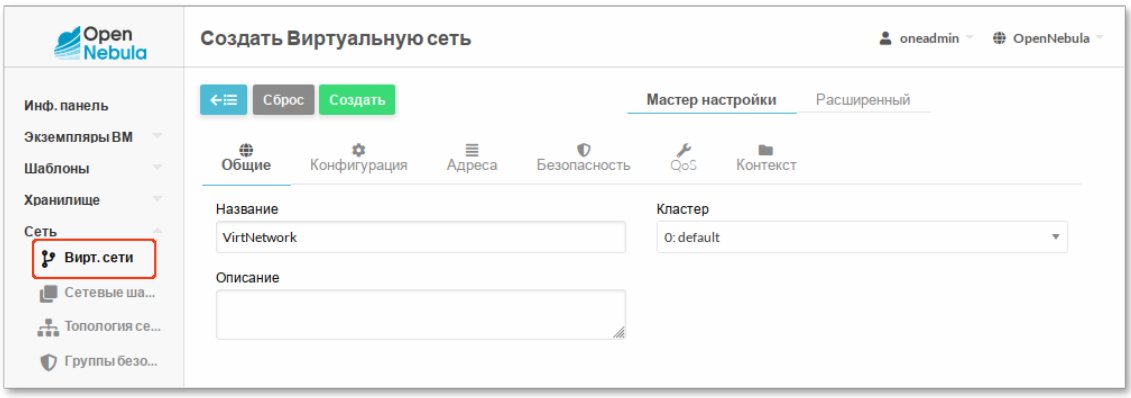
Таблица 28.2. Параметры виртуальной сети в режиме Bridged

Параметр	Значение	Обязательный
NAME	Имя виртуальной сети	Да
VN_MAD	Режим: - **dummy** — режим Dummy Bridged; - **bridge** — без фильтрации;	Да

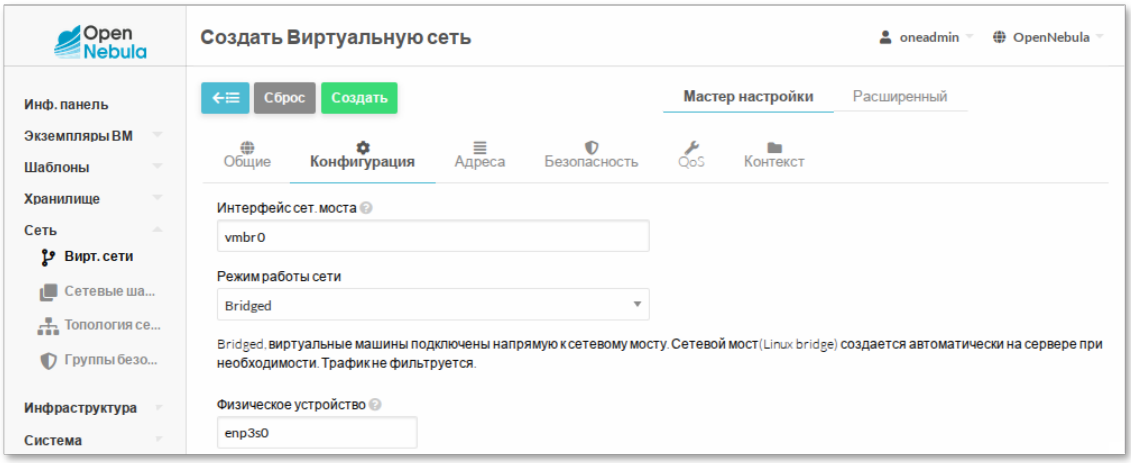
Параметр	Значение	Обязательный
	fw — фильтрация с группами безопасности;	
BRIDGE	Имя сетевого моста на узлах виртуализации	Нет (кроме dummy)
PHYDEV	Имя физического сетевого устройства на узле виртуализации, подключаемого к мосту (не применяется в режиме dummy)	Нет
AR	Диапазон адресов, доступных в виртуальной сети	Нет

Пример создания виртуальной сети в веб-интерфейсе:

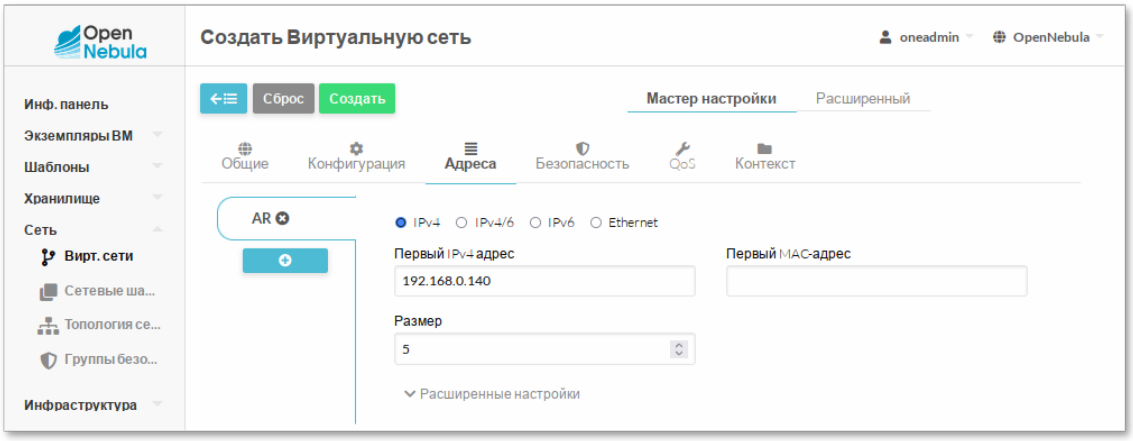
1. В левом меню выберите пункт **Сеть** → **Вирт. сети**.
2. На открывшейся странице нажмите кнопку **+** и выберите пункт **`Создать`**.
3. На вкладке **Общие** укажите название виртуальной сети:



4. На вкладке **Конфигурация** укажите интерфейс сетевого моста и выберите режим работы сети:



5. На вкладке **Адреса** задайте диапазон IP-адресов, который будет использоваться при выделении IP-адресов для ВМ:



6. Нажмите кнопку **Создать**.

28.1.3. Дополнительные возможности

По умолчанию мост не фильтрует VLAN-теги. Для ограничения разрешённых тегов используется атрибут ***VLAN_TAGGED_ID***. Он задаёт диапазон VLAN, разрешённых для трафика ВМ (поддерживаются списки тегов, разделённые запятыми, и диапазоны значений). Пример:

```
VLAN_TAGGED_ID = "100,105-107"
```

В этом случае ВМ будет разрешено использовать VLAN 100, 105, 106 и 107.

Примечание

Маркировка трафика выполняется самой ВМ. Мост лишь пробрасывает теги, но не добавляет их.

28.2. Режим 802.1Q (VLAN)

В режиме 802.1Q OpenNebula использует стандарт IEEE 802.1Q для изоляции трафика ВМ.

Для каждой виртуальной сети:

- на узле гипервизора создаётся Linux-мост;
- физический интерфейс (***PHYDEV***) подключается к мосту через VLAN-подынтерфейс (например, enp3s0.100);
- все ВМ, подключённые к данной сети, получают доступ к одному и тому же VLAN ID.

Важно

Коммутатор, к которому подключён ***PHYDEV***, должен быть настроен в режиме trunk и разрешать передачу соответствующих VLAN.

Идентификатор VLAN (VLAN ID) может быть:

- автоматически назначен OpenNebula из пула;
- задан вручную с помощью параметра ***VLAN_ID***.

Пул автоматических VLAN ID настраивается в файле **/etc/one/oned.conf**:

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

где:

- ***START*** — начальный VLAN ID для автоматического назначения;
- ***RESERVED*** — список или диапазоны VLAN ID, которые не будут использоваться.

28.2.1. Глобальная конфигурация

Параметры поведения сетевого драйвера настраиваются в файле **/var/lib/one/remotes/etc/vnm/OpenNebulaNetwork.conf** (см. табл. Параметры конфигурации).

Таблица 28.3. Параметры конфигурации

Параметр	Описание
:validate_vlan_id	Проверять, что к мосту не подключены другие VLAN (true/false)
:keep_empty_bridge	Сохранять мосты без подключённых интерфейсов (true/false)
:ip_bridge_conf	Хеш параметров, передаваемых команде **ip** при создании моста (**ip link add <bridge> type bridge …**)
:ip_link_conf	Хеш параметров, передаваемых команде **ip** при создании VLAN-интерфейса (**ip link add**)

Пример:

```
:ip_bridge_conf:
    :stp_state: on

:ip_link_conf:
    :udp6zerocsumrx:
    :tos: 3
```

Примечание

После изменения конфигурации необходимо синхронизировать файлы со всеми узлами кластера:

```
$ onehost sync -f
```

28.2.2. Создание виртуальной сети

Пример создания виртуальной сети с использованием конфигурационного файла:

1. Создайте файл **`net-vlan.conf`**:

```
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. Загрузите сеть, выполнив команду:

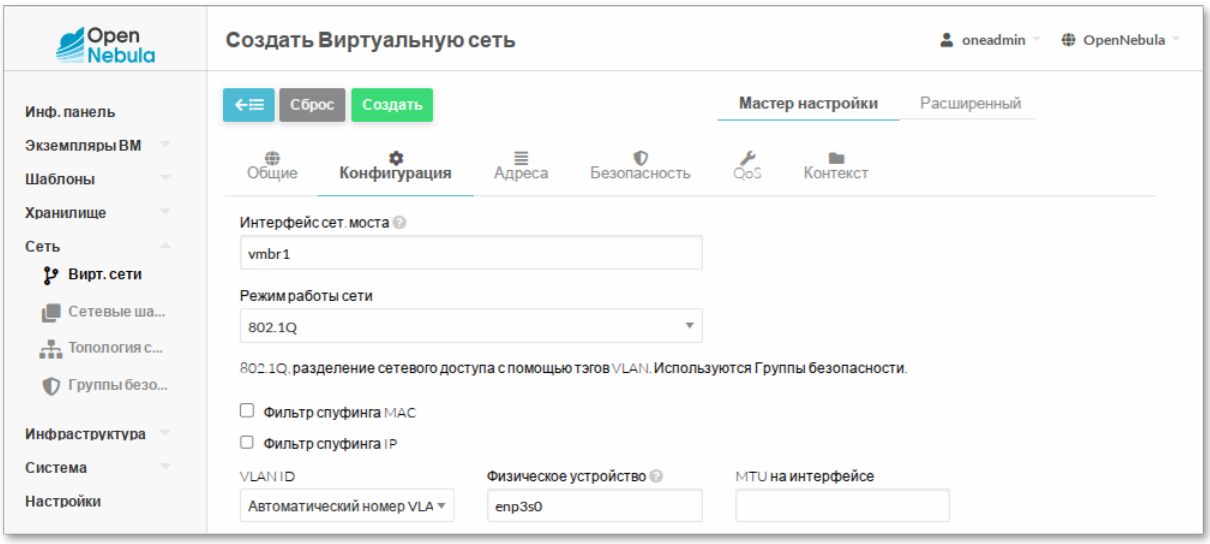
```
$ onevnet create net-vlan.conf
ID: 6
```

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

Таблица 28.4. Параметры виртуальной сети в режиме 802.1Q

Параметр	Значение	Обязательный
NAME	Имя виртуальной сети	Да
VN_MAD	802.1Q	Да
BRIDGE	Имя сетевого моста (по умолчанию onebr<net_id> или onebr.<vlan_id>)	Нет
PHYDEV	Имя физического сетевого устройства на узле виртуализации	Да
VLAN_ID	Идентификатор VLAN (если не указан и AUTOMATIC_VLAN_ID = "YES", генерируется автоматически)	Да (если AUTOMATIC_VLAN_ID = "NO")
AUTOMATIC_VLAN_ID	Автоматически генерировать VLAN_ID	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 в режиме 802.1Q в веб-интерфейсе:



Если мост vmbr1 отсутствует, он будет создан автоматически при первом запуске ВМ. Физический интерфейс enp3s0 будет помечен VLAN-тегом (например, enp3s0.7) и подключён к мосту vmbr1.

28.2.3. Поддержка Q-in-Q (Stacked VLAN)

Хотя нативный Q-in-Q не поддерживается Linux-мостами (в отличие от Open vSwitch), OpenNebula предоставляет ограниченную эмуляцию данной функциональности за счёт фильтрации VLAN.

При задании параметра ***CVLANS***:

- нетегированный трафик от ВМ помечается внешним VLAN (***VLAN_ID***);
- тегированный трафик с клиентским VLAN (CVLAN), входящим в список ***CVLANS***, инкапсулируется во внешний VLAN;
- весь прочий тегированный трафик отбрасывается.

Примечание

Ограничения режима:

- ВМ не видит внутренний тег (CVLAN) и отправляет нетегированный трафик;
- полноценный Q-in-Q, при котором ВМ самостоятельно управляет двумя VLAN-тегами, невозможен без использования Open vSwitch.

Поведение Q-in-Q контролируется дополнительными атрибутами, приведёнными в табл. Параметры виртуальной сети в режиме Q-in-Q.

Таблица 28.5. Параметры виртуальной сети в режиме Q-in-Q

Параметр	Значение	Обязательный
VLAN_ID	Внешний (сервисный) VLAN ID	Да (если AUTOMATIC_VLAN_ID = "NO")

Параметр	Значение	Обязательный
CVLANS	Список разрешённых клиентских VLAN (через запятую, поддерживаются диапазоны)	Да

Пример определения сети с поддержкой Q-in-Q:

```
NAME = "qinq_net"
VN_MAD = "802.1Q"
PHYDEV = eth0
VLAN_ID = 50 # Идентификатор сервисного VLAN
CVLANS = "101,103,110-113" # Список клиентских VLAN
```

Примечание

Параметр ***CVLANS*** можно изменять динамически — правила фильтрации будут обновлены без перезапуска ВМ.

Поведение моста при наличии ***CVLANS***:

- активируется фильтрация VLAN на мосту;
- на портах ВМ разрешаются только CVLAN из указанного списка;
- нетегированный трафик ассоциируется с внешним VLAN;
- при выходе в физическую сеть трафик инкапсулируется во внешний VLAN.

28.3. Режим VXLAN

Режим VXLAN (Virtual eXtensible Local Area Network) обеспечивает изоляцию трафика ВМ за счёт L2-инкапсуляции поверх IP-сети. В отличие от VLAN, VXLAN использует 24-битный идентификатор (VNI), что позволяет создавать до 16 миллионов изолированных сетей.

Примечание

VXLAN не использует теги 802.1Q. Трафик инкапсулируется в UDP-пакеты и передаётся через обычный IP-интерфейс.

Для каждой виртуальной сети:

- создаётся VXLAN-интерфейс (например, onevni50);
- создаётся Linux-мост (например, onebr.50);
- ВМ подключаются к этому мосту.

Трафик между узлами передаётся через UDP-порт 8472 (по умолчанию).

Примечание

Официальный порт IANA для VXLAN — UDP 4789. Если используется сетевое оборудование, соответствующее стандарту, порт следует указать явно через ***IP_LINK_CONF***.

Идентификатор VXLAN (VNI) может быть:

- автоматически назначен OpenNebula из пула;
- задан вручную с помощью параметра ***VLAN_ID***.

Для передачи широковещательного, многоадресного и неизвестного одноадресного (BUM) трафика используется многоадресная группа. По умолчанию:

- базовый multicast-адрес: 239.0.0.0/8;
- конкретная группа: 239.0.0.<VLAN_ID>.

Примечание

Физический сетевой интерфейс (***PHYDEV***), используемый для передачи трафика, должен иметь назначенный IP-адрес.

Будет создан мост вида ${PHYSDEV}.${VXLAN_ID}, где ***PHYDEV*** — это физический интерфейс, а ***VXLAN_ID*** — идентификатор VXLAN (VNI). К этому мосту подключаются сетевые интерфейсы.

Это имеет принципиальное значение, поскольку максимальная длина имени сетевого интерфейса в iproute2 составляет 15 символов.

Примечание

Если имя физического интерфейса и значение VNI в сумме превышают 15 символов, развёртывание ВМ, использующих такую сеть, завершится ошибкой.

Возможное решение — задать альтернативное имя (псевдоним) для физического интерфейса. Например, если используется интерфейс en0s0f0p0, можно выполнить:

```
# ip link set en0s0f0p0 alias vx
```

и затем использовать устройство vx в качестве ***PHYDEV***.

Назначение псевдонима не сохраняется после перезагрузки системы. Для постоянного применения необходимо повторять команду при старте системы или внести соответствующие изменения в конфигурационные файлы сети.

Каждый VXLAN связан с одной многоадресной группой (multicast group). В Linux существует ограничение на количество multicast-групп, членом которых может быть узел одновременно. Это ограничение также определяет максимальное число VXLAN, которые могут использоваться на одном физическом узле.

Значение по умолчанию — 20. Его можно изменить с помощью параметра ядра ***net.ipv4.igmp_max_memberships***. Для постоянного увеличения, например до 150, добавьте в файл **/etc/sysctl.conf**:

```
net.ipv4.igmp_max_memberships=150
```

и примените настройки:

```
# sysctl -p
```

28.3.1. Глобальная конфигурация

Начальный идентификатор VXLAN можно задать в файле **/etc/one/oned.conf**:

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

Параметры поведения сетевого драйвера настраиваются в файле **/var/lib/one/remotes/etc/vnm/OpenNebulaNetwork.conf** (см. табл. Параметры конфигурации).

Таблица 28.6. Параметры конфигурации

Параметр	Описание
:vxlan_mc	Базовый multicast-адрес (например, "239.0.0.0"). Фактический адрес вычисляется как ***vxlan_mc + VNI***
:vxlan_ttl	TTL для multicast-трафика (должен быть >1 при маршрутизации)
:validate_vlan_id	Проверять конфликты VLAN (true/false)
:keep_empty_bridge	Сохранять мосты без ВМ (true/false)
:ip_bridge_conf	Хеш параметров, передаваемых команде **ip** при создании моста (**ip link add <bridge> type bridge …**)
:ip_link_conf	Хеш параметров, передаваемых команде **ip** при создании VXLAN-интерфейса (**ip link add**)

Пример:

```
:ip_link_conf:
    :udp6zerocsumrx:
    :tos: 3
```

Примечание

После изменения конфигурации необходимо синхронизировать файлы со всеми узлами кластера:

```
$ onehost sync -f
```

28.3.2. Создание виртуальной сети

Пример создания виртуальной сети с использованием конфигурационного файла:

1. Создайте файл **`net-vxlan.conf`**:

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

2. Загрузите сеть:

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

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

Таблица 28.7. Параметры виртуальной сети в режиме VXLAN

Параметр	Значение	Обязательный
NAME	Имя виртуальной сети	Да
VN_MAD	vxlan	Да
PHYDEV	Имя физического сетевого устройства	Нет
BRIDGE	Имя сетевого моста (по умолчанию onebr<net_id> или onebr.<vlan_id>)	Нет
VLAN_ID	Идентификатор VXLAN (если не указан и AUTOMATIC_VLAN_ID = "YES", генерируется автоматически)	Да (если AUTOMATIC_VLAN_ID = "NO")
AUTOMATIC_VLAN_ID	Автоматически генерировать VLAN_ID	Да (если не указан VLAN_ID)
MTU	MTU для VXLAN-интерфейса и моста	Нет
VXLAN_MODE	Режим работы: multicast (по умолчанию) или evpn	Нет
VXLAN_TEP		Нет

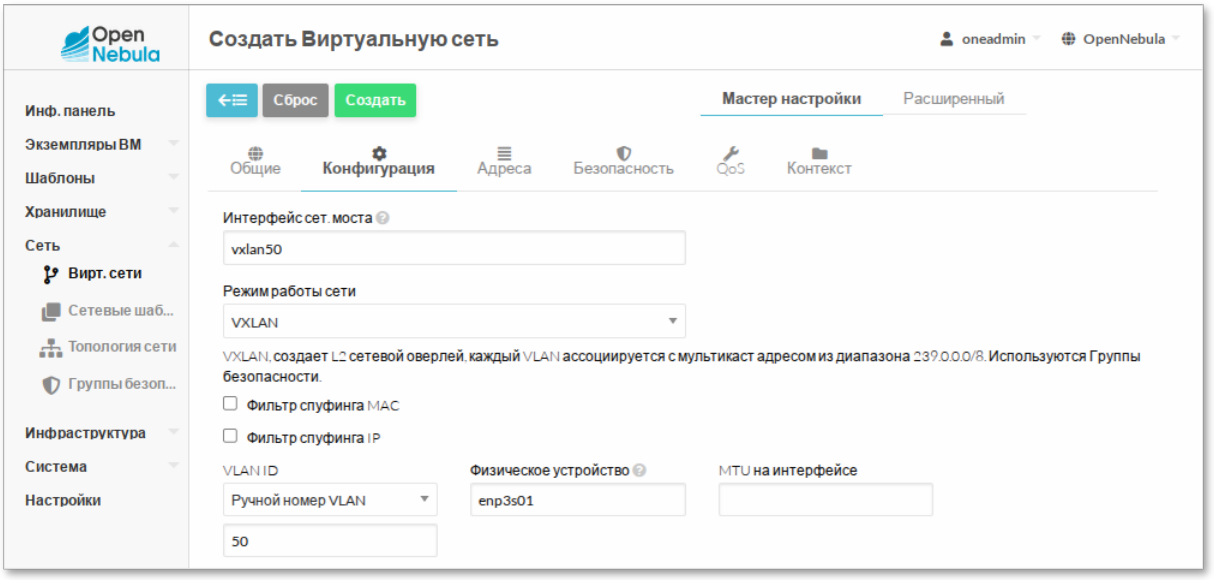
Параметр	Значение	Обязательный
	Тип адреса конечной точки туннеля (только для evpn): dev или local_ip	
VXLAN_MC	Базовый multicast-адрес (MC = vxlan_mc + vlan_id)	Нет
IP_LINK_CONF	Параметры команды **ip**, специфичные для данной сети. Формат: option=value (IP_LINK_CONF="option1=value1,option2=,option3=value3,...")	Нет
AR	Диапазон адресов виртуальной сети	Нет

Примечание

Параметры ***VXLAN_MODE***, ***VXLAN_TEP*** и ***VXLAN_MC*** можно задать глобально в файле **/var/lib/one/remotes/etc/vnm/OpenNebulaNetwork.conf**. Для переопределения на уровне отдельной сети используйте ***IP_LINK_CONF***, например:

```
IP_LINK_CONF = "dstport=4789,ttl=16"
```

Пример создания виртуальной сети в режиме VXLAN в веб-интерфейсе:



В этом примере драйвер проверит наличие моста vxlan50. Если мост отсутствует, он будет создан автоматически. Сетевой интерфейс enp3s0 будет использован для передачи VXLAN-трафика и подключён к мосту vxlan50.

Примечание

Если требуется дополнительная изоляция VXLAN-трафика с помощью VLAN, настройка VLAN должна выполняться на коммутаторе, а не на узле виртуализации. В этом случае OpenNebula использует уже тегированный интерфейс в качестве ***PHYDEV***.

28.4. Режим Open vSwitch

OpenNebula поддерживает использование программного коммутатора Open vSwitch (OVS) для создания виртуальных сетей. OVS предоставляет расширенные возможности по сравнению с Linux bridge: поддержка VLAN, VXLAN, QoS, групп безопасности и интеграция с SDN-контроллерами.

Примечание

На всех узлах виртуализации:

1. Установите пакет *openvswitch*:

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

2. Запустите службу Open vSwitch и добавьте её в автозагрузку:

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

28.4.1. Глобальная конфигурация

Идентификатор VLAN может быть:

- автоматически назначен OpenNebula из пула;
- задан вручную с помощью параметра ***VLAN_ID***.

Пул VLAN ID настраивается в файле **/etc/one/oned.conf**:

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

где

- ***START*** — начальный ID для автоматического назначения;
- ***RESERVED*** — список или диапазоны ID, которые не будут использоваться (например, 100:200).

Данный пул используется совместно для сетевых режимов 802.1Q и Open vSwitch.

Параметры поведения сетевого драйвера настраиваются в файле **/var/lib/one/remotes/etc/vnm/OpenNebulaNetwork.conf** (см. табл. Параметры конфигурации).

Таблица 28.8. Параметры конфигурации

Параметр	Описание
:arp_cache_poisoning	Включить защиту от ARP-spoofing (работает только при активных правилах IP/MAC anti-spoofing) (true/false)

Параметр	Описание
:keep_empty_bridge	Сохранять мосты без ВМ (true/false)
:ovs_bridge_conf	Хеш параметров для создания моста Open vSwitch (**ovs-vsctl add-br**)

Примечание

После изменения конфигурации необходимо синхронизировать файлы со всеми узлами кластера:

```
$ onehost sync -f
```

28.4.2. Создание виртуальной сети

Пример создания виртуальной сети с использованием конфигурационного файла:

1. Создайте файл **net-ovs.conf**:

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

2. Загрузите сеть:

```
$ onevnet create net-ovs.conf
ID: 8
```

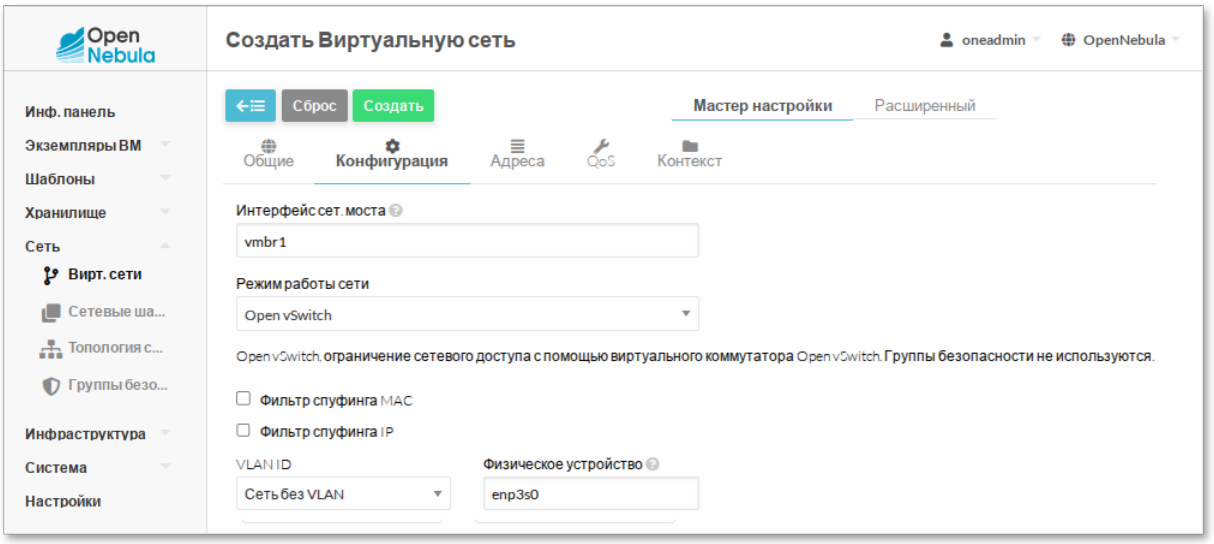
Таблица 28.9. Параметры виртуальной сети в режиме Open vSwitch

Параметр	Значение	Обязательный
NAME	Имя виртуальной сети	Да
VN_MAD	ovswitch	Да
BRIDGE	Имя моста Open vSwitch	Нет
PHYDEV	Физический интерфейс для привязки (обязателен при использовании VLAN)	Нет (если не используются VLAN)
VLAN_ID	Идентификатор VLAN (если не указан и AUTOMATIC_VLAN_ID = "YES", генерируется автоматически)	Нет
AUTOMATIC_VLAN_ID	Автоматически генерировать VLAN_ID (игнорируется, если задан VLAN_ID)	Нет
MTU	MTU для моста Open vSwitch	Нет
AR	Диапазон адресов виртуальной сети	Нет

Примечание

Если параметр ***PHYDEV*** не указан, мост создаётся изолированным и не имеет подключения к физической сети.

Пример создания виртуальной сети в режиме Open vSwitch в веб-интерфейсе:



28.4.3. Транкинг VLAN

Для разрешения нескольких VLAN на одном порту используйте атрибут ***VLAN_TAGGED_ID*** в шаблоне сети или ВМ:

```
VLAN_TAGGED_ID = "1,10,30,32,100-200"
```

Это настроит порт Open vSwitch в режиме trunk, разрешающем передачу указанных VLAN.

28.5. Использование Open vSwitch в сетях VXLAN

Режим ovswitch_vxlan комбинирует VXLAN-инкапсуляцию (для L3-изоляции между узлами) и VLAN-тегирование (для L2-сегментации внутри VXLAN-туннеля).

Принцип работы:

- внешний уровень — VXLAN-туннель с VNI, заданным параметром ***OUTER_VLAN_ID***;
- внутренний уровень — VLAN-теги (***VLAN_ID***) на портах ВМ.

Важно

Основная изоляция трафика обеспечивается VXLAN. VLAN используется только для логической сегментации внутри одного VXLAN-туннеля.

Пример создания виртуальной сети с использованием конфигурационного файла:

1. Создайте файл **`net-ovsx.conf`**:

```
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"
]
```

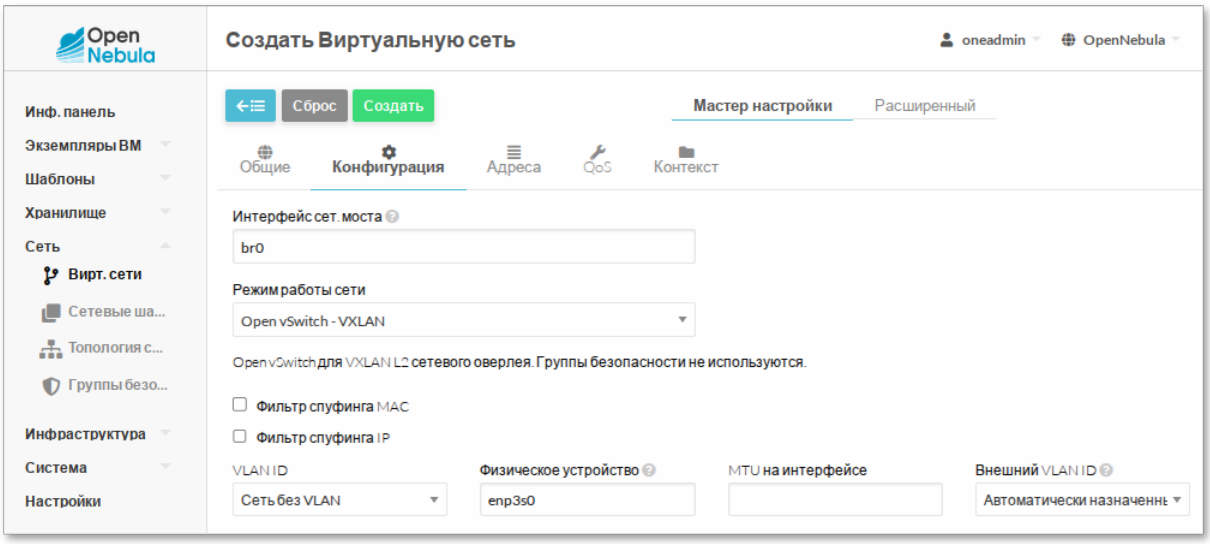
2. Загрузите сеть:

```
$ onevnet create net-ovsx.conf
ID: 11
```

Таблица 28.10. Параметры виртуальной сети в режиме Open vSwitch VXLAN

Параметр	Значение	Обязательный
NAME	Имя виртуальной сети	Да
VN_MAD	ovswitch_vxlan	Да
BRIDGE	Имя моста Open vSwitch	Нет
PHYDEV	Имя физического сетевого устройства	Да
OUTER_VLAN_ID	Идентификатор внешней сети VXLAN (если не указан и AUTOMATIC_OUTER_VLAN_ID = "YES", генерируется автоматически)	Да (если AUTOMATIC_OUTER_VLAN_ID = "NO")
AUTOMATIC_OUTER_VLAN_ID	Генерировать OUTER_VLAN_ID автоматически (игнорируется, если задан вручную)	Да (если не указан OUTER_VLAN_ID)
VLAN_ID	Внутренний VLAN ID 802.1Q (если не указан и AUTOMATIC_VLAN_ID = "YES", генерируется автоматически)	Нет
AUTOMATIC_VLAN_ID	Автоматически генерировать VLAN_ID (игнорируется, если определен VLAN_ID)	Нет
MTU	MTU для VXLAN-интерфейса и моста	Нет
AR	Диапазон адресов виртуальной сети	Нет

Пример создания виртуальной сети в режиме Open vSwitch VXLAN в веб-интерфейсе:



В данном примере драйвер проверит наличие моста ovsvxbr0.10000. Если мост отсутствует, он будет создан автоматически. Кроме того, создаётся VXLAN-интерфейс eth0.10000, который подключается к мосту Open vSwitch ovsvxbr0.10000. При запуске виртуальной машины её порты на мосту будут помечены VLAN-тегом 802.1Q с внутренним VLAN_ID, указанным в конфигурации (50 в данном примере).

Глава 29. Работа с хранилищами в OpenNebula

В OpenNebula система хранения данных построена на концепции хранилищ данных (datastores). Хранилище данных — это любой носитель информации, предназначенный для хранения образов дисков и вспомогательных файлов.

29.1. Типы хранилищ

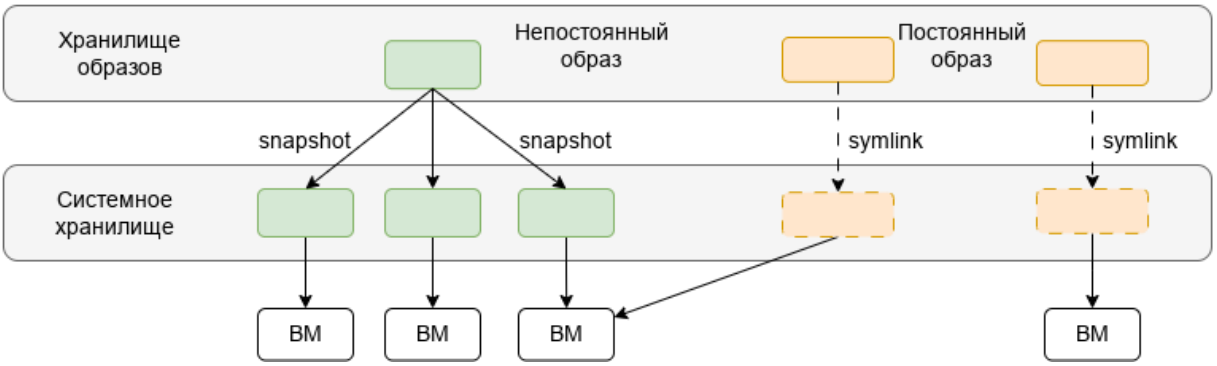
OpenNebula использует три типа хранилищ данных:

- Хранилище образов (Images Datastore) — используется для хранения образов операционных систем, постоянных томов данных и образов CD-ROM.
- Системное хранилище (System Datastore) — хранит диски работающих ВМ. При запуске ВМ её диски копируются из хранилища образов, а при остановке — удаляются (если образ непостоянный).
- Хранилище файлов и ядер (Files & Kernels Datastore) — хранит вспомогательные файлы, например, ядра Linux (vmlinuz), образы initrd/initramfs, а также файлы контекстуализации.

В зависимости от назначения выделяют два типа образов:

- Постоянные (persistent) — предназначены для хранения пользовательских данных (например, баз данных). Изменения, внесённые в такие образы, сохраняются после завершения работы ВМ. В каждый момент времени постоянный образ может быть подключён только к одной ВМ.

- Непостоянные (non-persistent) — используются для хранения дисков ВМ, работающих в текущий момент. Образы копируются или клонируются из хранилища образов при запуске ВМ и удаляются при её завершении, отключении диска или удалении ВМ. После удаления ВМ копия образа в системном хранилище также удаляется.



Передача образов дисков между хранилищем образов и системным хранилищем осуществляется с помощью драйверов Transfer Manager (TM). Эти драйверы представляют собой специализированные программные компоненты, выполняющие низкоуровневые операции хранения и передачи данных.

Образы хранятся в соответствующих каталогах хранилищ (**`/var/lib/one/datastores/<ID_хранилища>`**). Кроме того, для каждой запущенной ВМ в системном хранилище создаётся каталог **`/var/lib/one/datastores/<ID_хранилища>/<ID_ВМ>`**. В этих каталогах размещаются диски ВМ и дополнительные файлы, например контрольные точки и снимки.

Например, система с хранилищем образов с идентификатором 1, содержащим три образа, и тремя ВМ (ВМ 0 и 2 работают, ВМ 7 остановлена), развёрнутыми в системном хранилище с идентификатором 0, будет иметь следующую структуру:

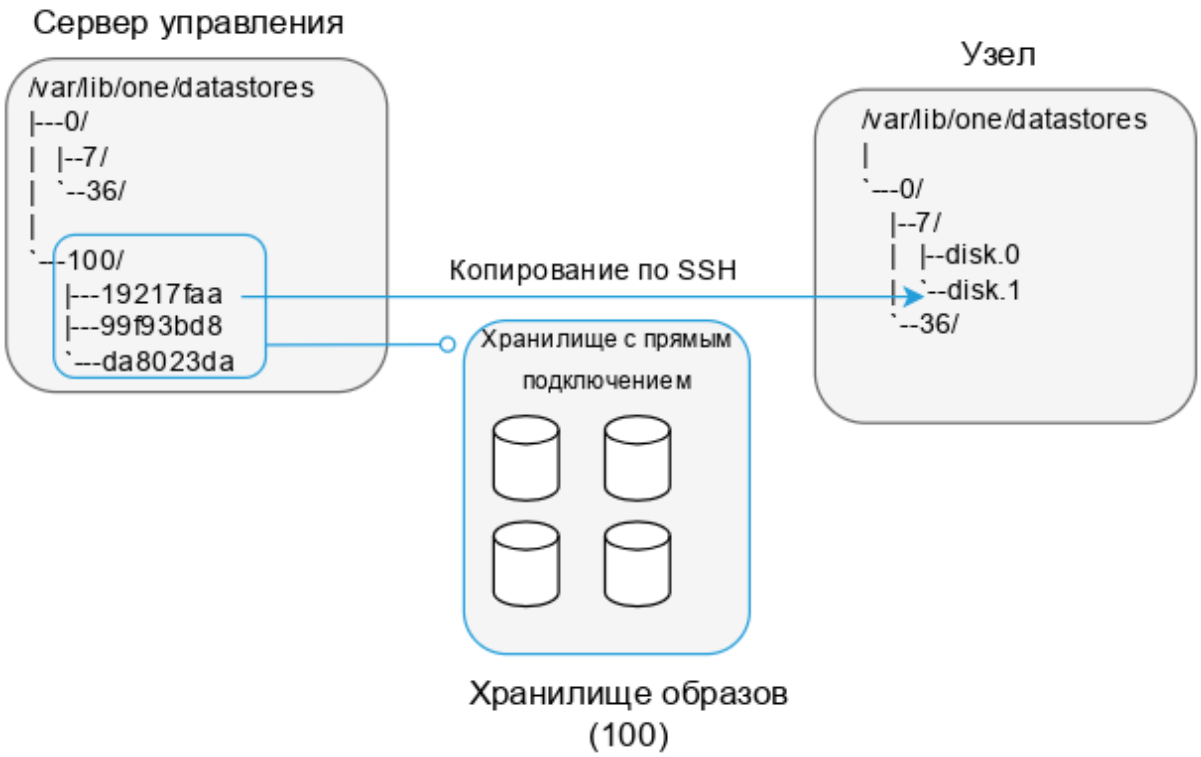
```
/var/lib/one/datastores
|-- 0/
|   |-- 0/
|   |   |-- disk.0
|   |   `-- disk.1
|   |-- 2/
|   |   `-- disk.0
|   `-- 7/
|       |-- checkpoint
|       `-- disk.0
`-- 1
|-- 19217fdaaa715b04f1c740557826514b
|-- 99f93bd825f8387144356143dc69787d
`-- da8023daf074d0de3c1204e562b8d8d2
```

Примечание

Базовый путь **`/var/lib/one/datastores`** можно изменить в файле **`/etc/one/oned.conf`** с помощью параметра ***`DATASTORE_LOCATION`***.

29.2. Драйверы передачи данных

Драйвер ***`ssh`*** использует локальные файловые системы узлов для размещения образов работающих ВМ. Все файловые операции выполняются локально, однако образы копируются на узлы по SSH при каждом запуске ВМ. Это может быть ресурсоёмкой операцией, особенно в крупных кластерах.



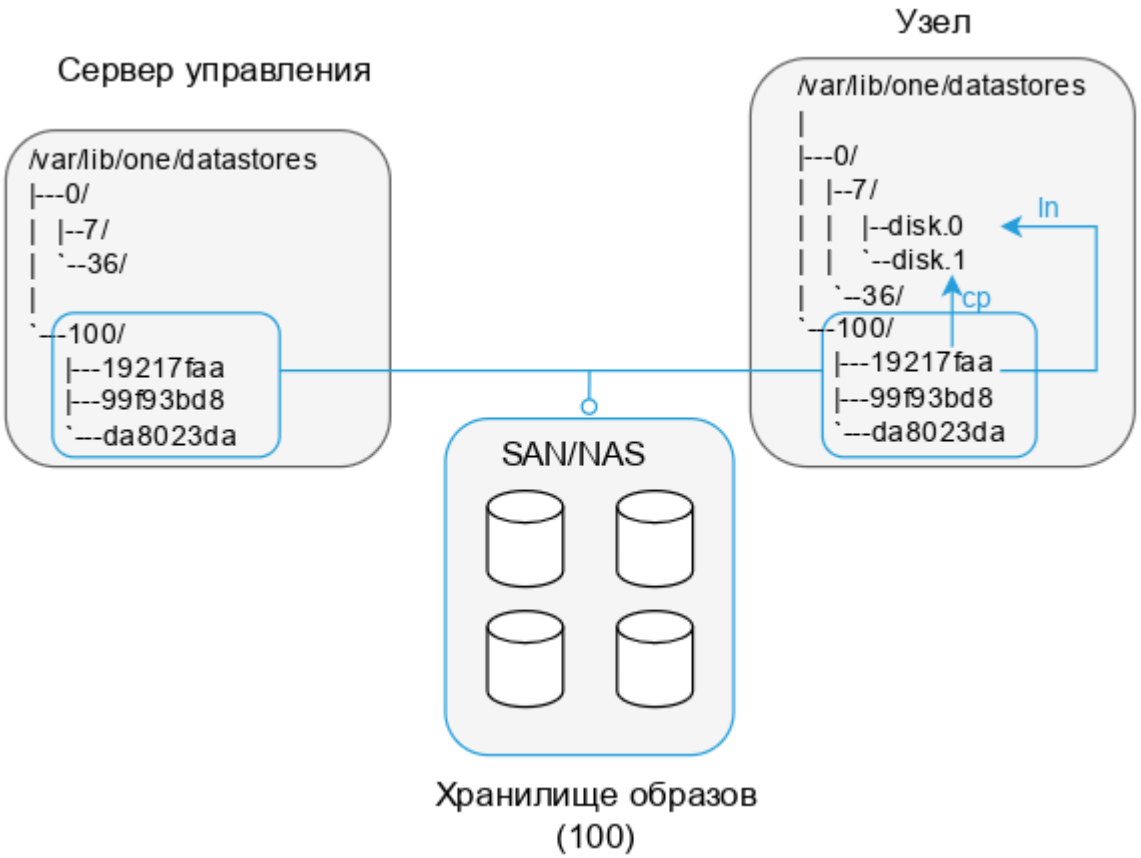
Драйвер ***`shared`*** предполагает, что хранилище данных смонтировано на всех узлах кластера (узле управления и вычислительных узлах). Обычно это реализуется с помощью общей файловой системы, например NFS или GlusterFS.

Все файловые операции (**`ln`**, **`cp`** и т. д.) выполняются локально на узле. Такой режим передачи, как правило, сокращает время развёртывания ВМ и обеспечивает возможность live-migration, однако может стать узким местом инфраструктуры и негативно повлиять на производительность ВМ при высоких дисковых нагрузках.

Рекомендации по оптимизации:

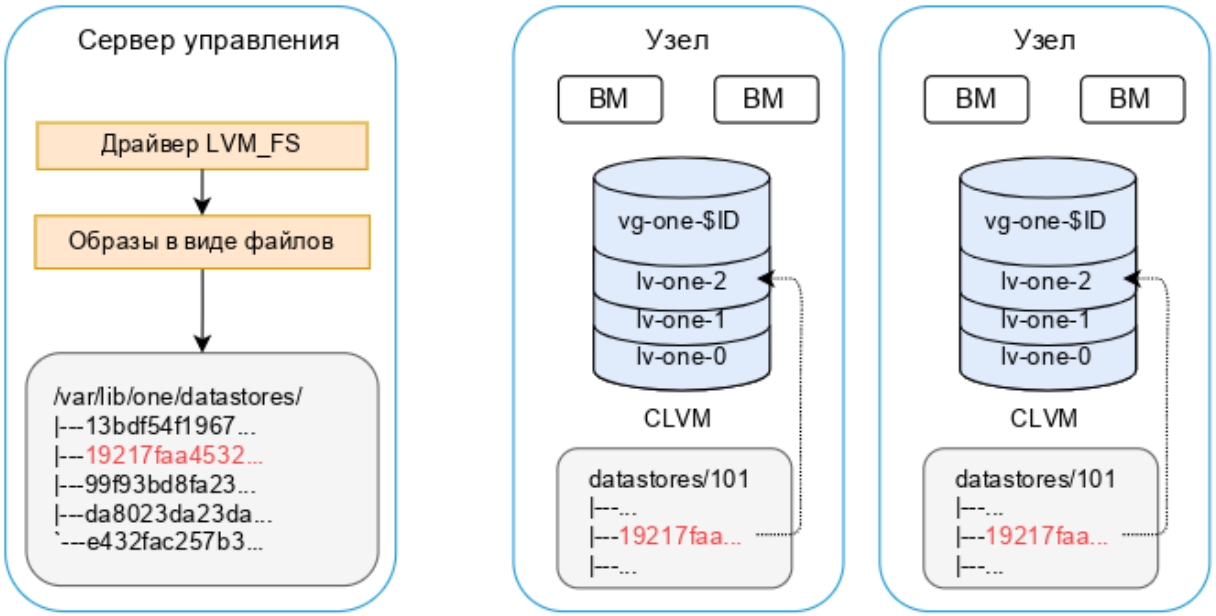
- использование разных файловых серверов для различных хранилищ данных, чтобы сбалансировать нагрузку ввода-вывода;
- применение локальных хранилищ для высоконагруженных ВМ;
- настройка кеширования и RAID на файловых серверах.

При создании ВМ её диски (файлы **`disk.i`**) копируются или связываются в соответствующем каталоге системного хранилища. Эти операции всегда выполняются на целевом узле.



Драйвер ***`lvm`*** предназначен для работы с SAN-сетями (iSCSI, Fibre Channel). Один и тот же LUN может быть экспортирован на все узлы кластера, при этом виртуальные машины работают непосредственно с SAN.

Образы в хранилище образов хранятся как обычные файлы (в **`/var/lib/one/datastores/<ID_хранилища>`**), однако при запуске ВМ они преобразуются в логические тома (LV) на узле. ВМ запускаются непосредственно на этих логических томах.



29.3. Хранилища по умолчанию

По умолчанию в OpenNebula создаются три хранилища:

- default (Images, ID=1) — хранилище образов;
- system (System, ID=0) — системное хранилище;
- files (Files, ID=2) — хранилище файлов.

По умолчанию они используют локальную файловую систему (каталоги **`/var/lib/one/datastores/<ID_хранилища>`**) и драйвер передачи ***`ssh`***.

Управление хранилищами осуществляется через:

- командную строку (**`onedatastore`**);
- веб-интерфейс Sunstone.

Примечание

Подробное описание команды **`onedatastore`** доступно в справочном руководстве:

```
$ man onedatastore
```

Примеры:

- получить список хранилищ данных:

```
$ onedatastore list
  ID NAME                    SIZE AVA CLUSTERS IMAGES TYPE DS
TM      STAT
   2 files                 195.8G 94% 0             0 fil  fs
ssh     on
```

```
   1 default                195.8G 94% 0             2 img  fs
ssh     on
   0 system                      - -   0             0 sys  -
ssh     on
```

- вывести информацию 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:         : 195.8G
FREE:          : 183.3G
USED:          : 2.5G
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
```

- вывести информацию 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 включает следующие разделы:

- ***INFORMATION*** — базовая информация (имя, путь, тип хранилища) и используемые драйверы (***DS_MAD*** и ***TM_MAD***);
- ***CAPACITY*** — показатели использования дискового пространства (общее, свободное и использованное);
- ***TEMPLATE*** — атрибуты хранилища;
- ***IMAGES*** — список образов, размещённых в хранилище.

В приведённом примере хранилище образов использует файловый драйвер (***DS_MAD="fs"***) и SSH для передачи (***TM_MAD="ssh"***). Для системного хранилища указывается только драйвер передачи (***TM_MAD***), а показатели использования (***CAPACITY***) не отображаются, поскольку драйвер ***ssh*** использует локальное хранилище каждого узла.

Примечание

Чтобы проверить доступное дисковое пространство на конкретном узле, используйте команду:

```
$ onehost show <ID_узла>
```

В зависимости от используемого драйвера и инфраструктуры для описания хранилища применяются различные атрибуты. Существует также набор общих атрибутов, применимых ко всем типам хранилищ. Эти атрибуты описаны в таблице Общие атрибуты хранилищ.

Таблица 29.1. Общие атрибуты хранилищ

Атрибут	Описание
Description	Произвольное описание
RESTRICTED_DIRS	Каталоги, запрещённые для размещения образов (через пробел)
SAFE_DIRS	Исключения из ***RESTRICTED_DIRS*** (через пробел)
NO_DECOMPRESS	Не распаковывать архивы при регистрации
LIMIT_TRANSFER_BW	Ограничение скорости загрузки (в байтах/с; допускаются суффиксы K, M, G)
DATASTORE_CAPACITY_CHECK	Проверять свободное место перед созданием образа
LIMIT_MB	Максимальный размер хранилища (в МБ)
BRIDGE_LIST	Список мостов, имеющих доступ к хранилищу (через пробел)
STAGING_DIR	Каталог для временных файлов (по умолчанию **/var/tmp**)
DRIVER	Драйвер работы с образами (переопределяет настройки образа)
COMPATIBLE_SYS_DS	Только для хранилищ образов. Список совместимых системных хранилищ (например, «0,100»)
CONTEXT_DISK_TYPE	Тип диска для контекстуализации: BLOCK или FILE (по умолчанию)

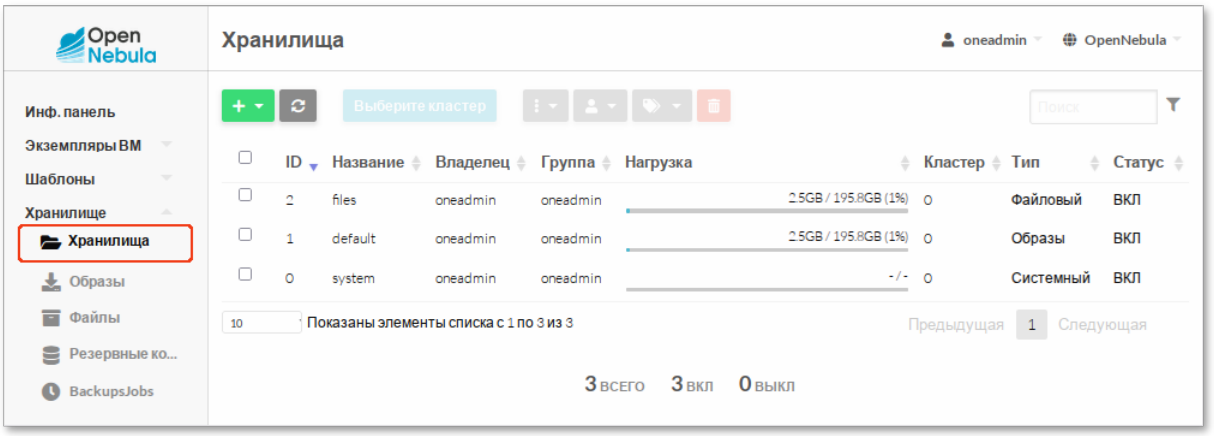
Примечание

Для работы с параметром ***BRIDGE_LIST*** на узлах должны быть установлены инструменты доступа к хранилищу и утилиты виртуализации (например, qemu-img).

Системные хранилища можно отключить, чтобы планировщик не размещал в них новые ВМ. При этом уже запущенные ВМ продолжат работать. Отключение хранилища:

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

Создание, включение, отключение, удаление и просмотр информации о хранилищах также доступны в веб-интерфейсе Sunstone:



29.4. Создание хранилищ

Для создания хранилища в OpenNebula необходимо:

1. Подготовить физическую инфраструктуру хранения (локальные диски, NFS, LVM и т. д.).
2. Зарегистрировать хранилище в OpenNebula, указав его тип, используемые драйверы и параметры.
3. Убедиться, что все узлы имеют доступ к данным в соответствии с выбранным режимом передачи.

29.4.1. Конфигурация с драйвером ssh («локальное хранилище»)

Этот режим использует локальные диски каждого узла виртуализации для хранения дисков работающих ВМ. Образы передаются с узла управления на вычислительные узлы по протоколу SSH.

29.4.1.1. Требования

На узле управления должно быть достаточно свободного пространства в каталоге **`/var/lib/one/datastores/`** для:

- хранилища образов (Images Datastore);
- системного хранилища (для временных дисков и файлов остановленных или ещё не развёрнутых ВМ).

На каждом узле виртуализации также должно быть достаточно свободного места в каталоге **`/var/lib/one/datastores/`** для хранения дисков ВМ, запущенных на данном узле.

Все узлы кластера (включая узел управления) должны:

- иметь корректно разрешаемые имена узлов (через DNS или файлы **`/etc/hosts`**);
- поддерживать беспарольный SSH-доступ между всеми узлами, включая доступ к localhost.

29.4.1.2. Регистрация хранилищ

Для работы в режиме ***ssh*** необходимо зарегистрировать два хранилища: системное и хранилище образов.

Для создания системного хранилища с использованием конфигурационного файла:

1. Создайте файл **`systemds.conf`**:

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

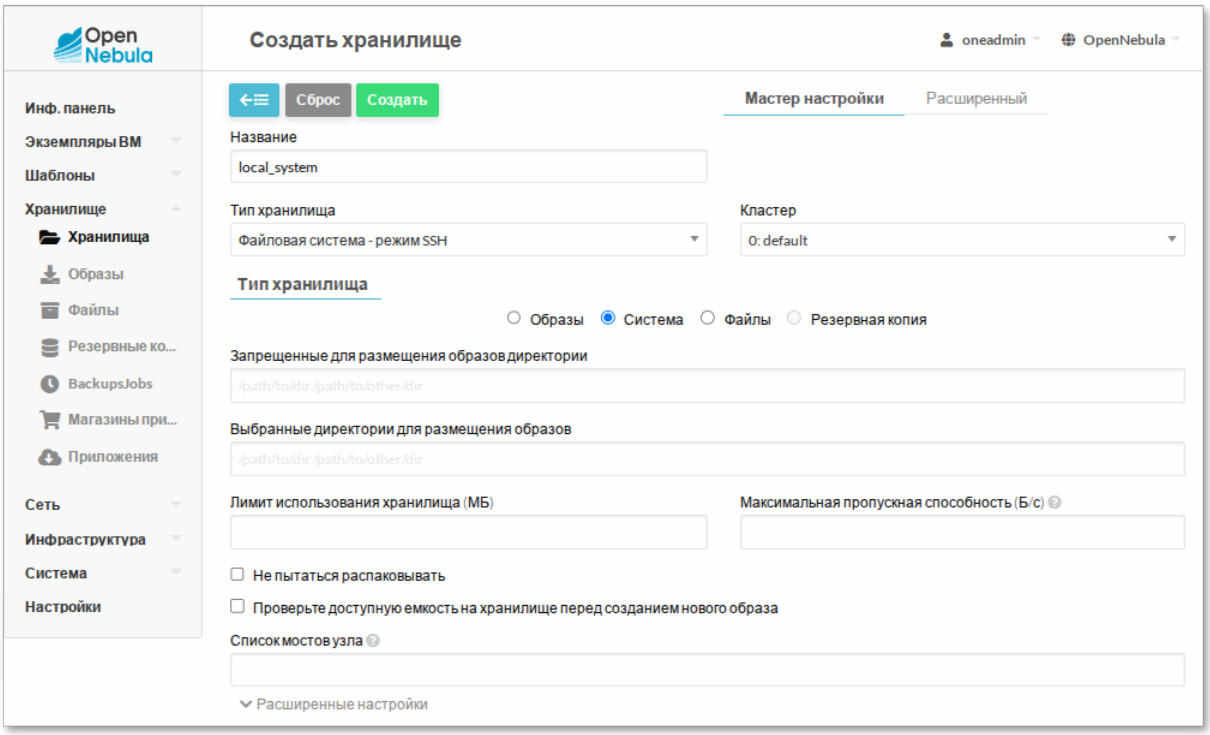
2. Зарегистрируйте хранилище:

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

Для создания системного хранилища необходимо указать следующие параметры:

- ***NAME*** — название хранилища;
- ***TYPE*** — SYSTEM_DS;
- ***TM_MAD*** — ssh (режим передачи данных).

Системное хранилище также можно зарегистрировать через веб-интерфейс Sunstone:



Для создания хранилища образов с использованием конфигурационного файла:

1. Создайте файл **`imageds.conf`**:

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

2. Зарегистрируйте хранилище:

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

Для создания хранилища образов необходимо указать следующие параметры:

- ***NAME*** — название хранилища;
- ***DS_MAD*** — fs (файловый драйвер хранилища данных);
- ***TYPE*** — IMAGE_DS;
- ***TM_MAD*** — ssh (режим передачи данных);
- ***CONVERT*** — yes (по умолчанию) или no; при значении yes формат образа будет преобразован в формат, поддерживаемый драйвером.

Примечание

Оба хранилища (системное и хранилище образов) должны использовать одинаковый драйвер передачи (***TM_MAD***), в данном случае — ***ssh***.

При использовании файлового драйвера и режима передачи ***ssh*** могут быть заданы следующие дополнительные параметры:

- ***DD_BLOCK_SIZE*** — размер блока для операций **dd** (по умолчанию: 64 КБ). Может быть задан в файле **/var/lib/one/remotes/etc/datastore/fs/fs.conf**;
- ***SUPPORTED_FS*** — список файловых систем, поддерживаемых для создания форматированных блоков данных (через запятую). Может быть установлен в файле **/var/lib/one/remotes/etc/datastore/datastore.conf**;
- ***FS_OPTS_<FS>*** — параметры создания файловой системы для форматированных блоков данных. Указывается отдельно для каждого типа файловой системы в файле **/var/lib/one/remotes/etc/datastore/datastore.conf**;
- ***SPARSE*** — если установлено в NO, файлы будут занимать всё выделенное пространство (неразрежённые файлы). Требует установки параметра ***QCOW2_STANDALONE="YES"*** в системном хранилище.

Примечание

Особенности режима ***TM_MAD=ssh***:

- для образов формата qcow2 используется утилита **qemu-img**;
- для образов формата raw используются утилиты **dd** или **tar**;
- параметр ***QCOW2_OPTIONS*** игнорируется.

29.4.2. Хранилище NFS/NAS

Конфигурация на базе NFS/NAS предполагает использование общего сетевого тома, смонтированного на всех узлах кластера (включая узел управления). Этот том используется для хранения:

- образов дисков (в Images Datastore);
- дисков работающих ВМ (в System Datastore).

Преимущества:

- быстрое развёртывание ВМ (с использованием hardlinks или copy-on-write);
- поддержка live-migration;
- централизованное управление хранилищем.

Ограничения:

- производительность зависит от пропускной способности сети и возможностей NFS-сервера;
- высокая I/O-нагрузка может стать узким местом инфраструктуры.

Для повышения масштабируемости можно использовать несколько NFS-серверов, что позволяет:

- балансировать I/O-нагрузку;
- применять разные политики SLA (резервное копирование, QoS);
- изолировать критические важные ВМ на отдельных томах.

Примечание

Формат образов qcow2 по умолчанию использует thin provisioning (динамическое выделение дискового пространства), что позволяет экономить место на хранилище.

29.4.2.1. Настройка управляющего узла

На управляющем узле необходимо смонтировать каталог хранилища образов с сервера NAS/SAN в каталог **`/var/lib/one/datastores/<ID_хранилища>`**. Если все хранилища данных относятся к одному типу, допускается монтирование всего каталога **`/var/lib/one/datastores/`**.

Примечание

На управляющем узле требуется монтировать только хранилища образов, а не системные хранилища данных.

Примечание

Рекомендуемые параметры монтирования NFS:

```
soft, intr, rsize=32768, wsize=32768
```

При использовании данной конфигурации libvirt/kvm доступ к файлам образов осуществляется от имени пользователя oneadmin.

Если файлы должны быть доступны от имени root, необходимо добавить параметр ***`no_root_squash`*** на NFS-сервере:

```
/export *(rw,sync,no_root_squash)
```

29.4.2.2. Настройка узлов виртализации

На каждом узле виртуализации необходимо смонтировать каталоги хранилищ данных в **`/var/lib/one/datastores/<ID_хранилища>`**.

29.4.2.3. Регистрация хранилищ в OpenNebula

После настройки хранилища на узле управления и вычислительных узлах необходимо зарегистрировать два хранилища: системное и хранилище образов.

Регистрация системного хранилища:

1. Создайте файл **`systemds.conf`**:

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

2. Зарегистрируйте хранилище:

```
$ onedatastore create systemds.conf
ID: 103
```

Для создания системного хранилища необходимо указать:

- ***NAME*** — название хранилища;
- ***TYPE*** — SYSTEM_DS;
- ***TM_MAD*** — shared (режим общей файловой системы);
- ***BRIDGE_LIST*** — список узлов с подключённым системным хранилищем (через пробел).

Регистрация хранилища образов:

1. Создайте файл **imageds.conf**:

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

2. Зарегистрируйте хранилище:

```
$ onedatastore create imageds.conf
ID: 104
```

Для создания хранилища образов необходимо указать:

- ***NAME*** — название хранилища;
- ***DS_MAD*** — fs (файловый драйвер);
- ***TYPE*** — IMAGE_DS;
- ***TM_MAD*** — shared (режим общей файловой системы);
- ***CONVERT*** — yes (по умолчанию) или no; при yes формат образа будет преобразован в формат, поддерживаемый драйвером.

Примечание

Если доступно несколько системных хранилищ данных, атрибут ***TM_MAD_SYSTEM*** будет установлен после выбора хранилища данных.

Примечание

Оба хранилища (системное и хранилище образов) должны использовать одинаковый драйвер передачи (***TM_MAD***).

После регистрации на узле управления в каталоге **/var/lib/one/datastores/** будут созданы каталоги **103** и **104**. На узлах виртуализации эти каталоги не создаются автоматически, поэтому их необходимо создать вручную:

```
$ mkdir /var/lib/one/datastores/103
$ mkdir /var/lib/one/datastores/104
```

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

```
# mount -t nfs 192.168.0.157:/export/storage /var/lib/one/datastores/104
```

Для автоматического монтирования при загрузке системы добавьте запись в файл **/etc/fstab**:

```
192.168.0.157:/export/storage /var/lib/one/datastores/104   nfs
intr,soft,nolock,_netdev,x-systemd.automount    0 0
```

Примечание

Для возможности монтирования NFS-хранилища на всех узлах должен быть запущен клиент NFS:

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

Список экспортируемых ресурсов NFS-сервера можно получить командой:

```
# showmount -e 192.168.0.157
```

Важно

После добавления записи в **/etc/fstab** и перезагрузки системы необходимо назначить владельца каталога хранилища:

```
# chown oneadmin: /var/lib/one/datastores/104
```

29.4.2.4. Дополнительная конфигурация

Дополнительно могут использоваться следующие параметры:

- ***QCOW2_OPTIONS*** — дополнительные параметры для **qemu-img create** (например, ***-o cluster_size=2M***). Задаются в файле **/etc/one/tmrc**;
- ***DD_BLOCK_SIZE*** — размер блока для операций **dd** (по умолчанию 64 КБ). Может быть задан в файле **/var/lib/one/remotes/etc/datastore/fs/fs.conf**;
- ***SUPPORTED_FS*** — список поддерживаемых файловых систем (через запятую). Может быть установлен в файле **/var/lib/one/remotes/etc/datastore/datastore.conf**;
- ***FS_OPTS_<FS>*** — параметры создания файловой системы для конкретного типа ФС. Указывается отдельно для каждого типа файловой системы в файле **/var/lib/one/remotes/etc/datastore/datastore.conf**;
- ***SPARSE*** — если установлено в NO, файлы будут занимать всё выделенное пространство (неразрежённые файлы). Требует установки параметра ***QCOW2_STANDALONE="YES"*** в системном хранилище;

- ***QCOW2_STANDALONE*** — если установлено в YES, создаётся автономный qcow2-диск без backing file.

Примечание

Перед добавлением новой файловой системы в список ***SUPPORTED_FS*** убедитесь, что команда **mkfs.<fs_name>** доступна на узле управления и гипервизорах.

29.4.3. Гибридная конфигурация: NFS + локальное хранилище

Для повышения производительности дисковых операций можно использовать гибридный подход:

- хранить образы в NFS-хранилище (***TM_MAD=shared***);
- запускать ВМ с локальных дисков (***TM_MAD=ssh***).

Настройка:

1. Зарегистрируйте два системных хранилища:
   - nfs_system (***TM_MAD=shared***);
   - local_system (***TM_MAD=ssh***).
2. Добавьте оба хранилища в один кластер.
3. В шаблоне ВМ укажите альтернативный режим развёртывания:

```
TM_MAD_SYSTEM="ssh"
```

Примечание

Такой подход увеличивает время развёртывания ВМ (из-за копирования по SSH), но существенно повышает производительность I/O во время их работы.

29.4.4. Хранилище SAN

Конфигурация на базе SAN предполагает использование блочных устройств (LUN), экспортированных по протоколам iSCSI или Fibre Channel. OpenNebula поддерживает два основных подхода:

- LVM-only — ВМ работают напрямую с логическими томами LVM;
- гибридный — образы хранятся в файловой системе поверх LVM, а ВМ используют блочные устройства.

Требования:

- минимум два LUN: один для хранилища образов, один для системного хранилища;

- все узлы кластера (включая сервер управления) должны иметь одинаковый доступ ко всем LUN.

Хранилище SAN может предоставлять доступ к образам двумя способами:

- режим NFS — файлы образов доступны непосредственно на узлах виртуализации через распределённую файловую систему (например, NFS или GlusterFS) с использованием драйвера ***fs_lvm***;
- режим SSH — файлы образов передаются на узлы виртуализации по SSH с использованием драйвера ***fs_lvm_ssh***.

29.4.4.1. Настройка сервера управления

Для сервера управления:

- в режиме LVM-only — сервер управления должен иметь доступ к LUN и быть включён в ***BRIDGE_LIST***;
- в гибридном режиме — необходимо смонтировать файловую систему (например, OCFS2) в каталог **/var/lib/one/datastores/<ID_хранилища>**.

Примечание

Если используется только LVM-only, сервер управления не требует монтирования файловых систем, но должен иметь доступ к LUN.

29.4.4.2. Настройка вычислительных узлов

Базовые требования:

- установлен пакет LVM2;
- служба lvmetad должна быть отключена. В файле **/etc/lvm/lvm.conf** необходимо установить:

```
use_lvmetad = 0
```

  и отключить службу, если она запущена:

```
# systemctl disable --now lvm2-lvmetad.service
```

- пользователь oneadmin добавлен в группу disk:

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

- все узлы должны иметь доступ к одним и тем же LUN;
- для каждого хранилища данных необходимо создать группу томов LVM (LVM VG) на общем LUN с именем vg-one-<system_ds_id>. Создание выполняется только на одном узле.

Примечание

После перезагрузки узла виртуализации логические тома необходимо активировать, чтобы они снова стали доступны гипервизору.

Если установлен пакет *opennebula-node-kvm*, активация выполняется автоматически. В противном случае, логические тома ВМ, работавших на узле до перезагрузки, необходимо активировать вручную командой:

```
# lvchange -ay <DEVICE>
```

Также можно использовать скрипт активации **`/var/tmp/one/tm/fs_lvm/activate`**, доступный в удалённых скриптах OpenNebula.

Конфигурация в режиме SSH

Диски ВМ представляют собой символические ссылки на блочные устройства. Дополнительные файлы ВМ (например, контрольные точки и файлы развёртывания) хранятся в каталоге **`/var/lib/one/datastores/<ID_хранилища>`**. Необходимо убедиться, что на локальном хранилище достаточно свободного пространства для этих файлов.

Настройка режима NFS

Каталоги хранилищ образов и системного хранилища данных должны быть доступны всем гипервизорам, например, с использованием NFS или аналогичных механизмов. Все узлы должны монтировать соответствующие каталоги и иметь доступ к образам и системным данным.

Примечание

Образы хранятся в общем файловом хранилище (NFS, GlusterFS и т.п.). Каталоги хранилищ данных и точки монтирования должны быть настроены аналогично обычному файловому хранилищу образов.

Рекомендуется сначала развернуть общее файловое хранилище, а затем заменить соответствующее системное хранилище данных на хранилище LVM, сохранив ту же точку монтирования.

29.4.4.3. Регистрация хранилищ в OpenNebula

После настройки хранилища на узле управления и вычислительных узлах необходимо зарегистрировать два хранилища: системное и хранилище образов.

Регистрация системного хранилища:

1. Создайте файл **`systemds.conf`**:

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

2. Зарегистрируйте хранилище:

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

Для создания системного хранилища необходимо указать:

- ***NAME*** — название хранилища;
- ***TYPE*** — SYSTEM_DS;
- ***TM_MAD*** — fs_lvm (режим NFS), fs_lvm_ssh (режим SSH);
- ***DISK_TYPE*** — BLOCK. Используется для временных (volatile) дисков;
- ***BRIDGE_LIST*** — список узлов, имеющих доступ к логическим томам (не требуется, если сервер управления имеет прямой доступ к LUN).

Регистрация хранилища образов:

1. Создайте файл **`imageds.conf`**:

```
NAME    = lvm-images
TM_MAD  = fs_lvm_ssh
TYPE    = IMAGE_DS
DISK_TYPE = "BLOCK"
DS_MAD  = fs
SAFE_DIRS = "/var/tmp /tmp"
```

2. Зарегистрируйте хранилище:

```
$ onedatastore create imageds.conf
ID: 106
```

Для создания хранилища образов необходимо указать:

- ***NAME*** — название хранилища;
- ***DS_MAD*** — fs (файловый драйвер);
- ***TYPE*** — IMAGE_DS;
- ***TM_MAD*** — fs_lvm (режим NFS), fs_lvm_ssh (режим SSH);
- ***DISK_TYPE*** — BLOCK;
- ***BRIDGE_LIST*** — список узлов с доступом к логическим томам (опционально).

Примечание

Системное хранилище и хранилище образов должны использовать одинаковый драйвер передачи (***TM_MAD***).

Атрибут ***BRIDGE_LIST*** можно опустить, если сервер управления имеет прямой доступ к LUN.

На сервере управления в каталоге **/var/lib/one/datastores/** будут автоматически созданы каталоги **105** и **106**. На вычислительных узлах эти каталоги необходимо создать вручную:

```
$ mkdir /var/lib/one/datastores/{105,106}
```

29.4.4.3.1. Настройка драйвера LVM

По умолчанию драйвер LVM обнуляет все создаваемые логические тома, чтобы предотвратить утечку данных между виртуальными машинами. Эта операция может занимать значительное время и увеличивать время развёртывания ВМ.

Поведение драйвера настраивается в файле **/var/lib/one/remotes/etc/fs_lvm/fs_lvm.conf**:

- ***ZERO_LVM_ON_CREATE*** — обнулять тома при создании (yes/no);
- ***ZERO_LVM_ON_DELETE*** — обнулять тома при удалении (рекомендуется включать);
- ***DD_BLOCK_SIZE*** — размер блока для операций **dd** (по умолчанию: 64 КБ).

Пример конфигурации:

```
ZERO_LVM_ON_CREATE=no
ZERO_LVM_ON_DELETE=yes
DD_BLOCK_SIZE=32M
```

Дополнительно для каждого типа хранилища можно задать:

- ***SUPPORTED_FS*** — список файловых систем, поддерживаемых для создания форматированных блочных устройств (через запятую). Может быть установлен в файле **/var/lib/one/remotes/etc/datastore/datastore.conf**;
- ***FS_OPTS_<FS>*** — параметры создания файловой системы для соответствующего типа ФС. Настраиваются в файле **/var/lib/one/remotes/etc/datastore/datastore.conf**.

29.4.4.4. Подключение СХД

29.4.4.4.1. Особенности подключения СХД по FC

Алгоритм подключения:

1. Подготовить СХД (создать LUN).

2. На сервере установить FC HBA-адаптеры и драйверы к ним.

3. Настроить сетевое подключение.

4. Подключить СХД к серверу.

5. Предоставить серверу доступ к СХД по WWPN.

Примечание

Для определения глобальных имён портов (WWPN) можно воспользоваться утилитой **`systool`** из пакета *sysfsutils*.

Пакет *sysfsutils* необходимо установить из репозитория:

```
# apt-get install sysfsutils
```

Чтобы получить WWPN, выполните команду:

```
# systool -c fc_host -A port_name
```

Пример вывода:

```
Class = "fc_host"
Class Device = "host1"
port_name = "0x10000090fa59a61a"
Device = "host1"
Class Device = "host16"
port_name = "0x10000090fa59a61b"
Device = "host16"
```

Просмотреть список подключённых устройств можно, например, с помощью команды:

```
# lsblk
```

Пример:

```
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 объёмом ~1 ТБ доступен по четырём путям.

29.4.4.4.2. Особенности подключения СХД по ISCSI

Все соединения iSCSI должны устанавливаться автоматически при загрузке узла. Для этого параметру ***node.startup*** необходимо задать значение *automatic*.

Значение параметра ***node.session.timeo.replacement_timeout*** по умолчанию составляет 120 секунд. Рекомендуется уменьшить его до 15 секунд.

Эти параметры задаются в файле **/etc/iscsi/iscsid.conf** (глобально). Если iSCSI-цель уже подключена, необходимо изменить настройки для конкретной цели в файле **/etc/iscsi/nodes/<TARGET>/<PORTAL>/default**.

На всех узлах необходимо:

1. Установить пакет *open-iscsi*, запустить и добавить сервис iscsid в автозагрузку:

   ```
   # apt-get install open-iscsi
   # systemctl enable --now iscsid
   ```

2. Указать параметры в файле **/etc/iscsi/iscsid.conf**:

   ```
   node.startup = automatic
   node.session.timeo.replacement_timeout = 15
   ```

3. Выполнить обнаружение iSCSI-целей и подключиться к ним:

   ```
   # 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-целей:

   - 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
     ```

5. После перезагрузки должны появиться подключенные устройства:

   ```
   # 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 объёмом ~1 ТБ доступен по четырём путям.

Примечание

Примеры использования команды **iscsiadm**:

- отключить хранилище (отключить все iSCSI-цели):

```
# iscsiadm -m node --logout
```

- отключить конкретную цель:

```
# iscsiadm -m node --targetname "iscsi-
target-1.test.alt:server.target1" --logout
```

- переопросить устройства iSCSI:

```
# iscsiadm -m node -R
```

- просмотреть активные iSCSI-сессии:

```
# iscsiadm -m session
```

29.4.4.5. Настройка Multipath

Многопутевой ввод-вывод (Multipath I/O) — технология, позволяющая использовать несколько маршрутов к одному устройству хранения. В случае отказа одного пути ОС автоматически использует альтернативный маршрут, что повышает отказоустойчивость и позволяет балансировать нагрузку.

Multipath объединяет несколько физических путей в одно логическое устройство, обеспечивая:

- автоматический выбор пути;
- переключение при отказе;
- увеличение пропускной способности за счёт балансировки нагрузки.

На всех узлах необходимо установить пакет *multipath-tools*:

```
# apt-get install multipath-tools
```

И запустить службу multipathd:

```
# systemctl enable --now multipathd && sleep 5; systemctl status multipathd
```

29.4.4.5.1. Конфигурация multipath

Примечание

Команда **multipath** используется для обнаружения и управления многопутевыми устройствами.

Основные 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 устройства из файла wwids;
- **-W** — сбросить файл wwids, чтобы включить только текущие многопутевые устройства;
- **-r** — принудительно перечитать конфигурацию.

После подключения СХД устройство должно автоматически определиться как multipath:

```
# 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 running
```

Вывод команды условно делится на три части:

- Информация об устройстве multipath:
  - mpatha (3600c0ff00014f56ee9f3cf6301000000) — алиас устройства
  - dm-0 — имя устройства dm
  - HP,P2000 G3 FC — производитель и модель
  - size=931G — размер
  - features='1 queue_if_no_path' — функции
  - hwhandler='1 alua' — аппаратный обработчик
  - wp=rw — режим доступа
- Информация о группе путей:
  - policy='service-time 0' — политика планирования
  - prio=50 — приоритет группы путей

- status=active — статус группы путей

- Информация о конкретных путях:
  - 16:0:1:1 — адрес устройства (host:channel:id:LUN)
  - sde — имя диска
  - 8:64 — 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 задаются в файле **`/etc/multipath.conf`**. Пример базовой конфигурации:

```
defaults {
    find_multipaths         yes
    user_friendly_names     yes
}
```

Если ***`user_friendly_names = no`***, устройство будет именоваться по WWID **`/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 ***`user_friendly_names = yes`***, то для имени multipath-устройства задаётся алиас (псевдоним) вида 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
```

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 предписывается внести в чёрный список любые найденные устройства хранения данных, за исключением нужного.

- Создать ещё одну секцию:

  ```
  multipaths {
    multipath {
          wwid "3600c0ff00014f56ee9f3cf6301000000"
          alias mpatha
    }
  }
  ```

  В этом случае устройство всегда будет доступно как **/dev/mapper/mpatha**. Вместо mpatha можно вписать любое желаемое имя устройства.

Примечание

Получить WWID устройства можно командой:

```
# /lib/udev/scsi_id -g -u -d /dev/sdb
3600c0ff00014f56ee9f3cf6301000000
```

Для устройств в одном multipath WWID совпадают.

В файл **/etc/multipath.conf** может также потребоваться внести рекомендованные производителем СХД параметры.

После изменения файла **/etc/multipath.conf** необходимо перезапустить службу multipathd:

```
# systemctl restart multipathd.service
```

Примечание

Проверить файл **/etc/multipath.conf** на наличие ошибок можно, выполнив команду:

```
# multipath -t
```

29.4.4.6. Разметка хранилища образов

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

Ниже приведён пример создания кластерной файловой системы OCFS2 на multipath-устройстве и подключения этого устройства в OpenNebula.

29.4.4.6.1. 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 идентичен 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 OCFS2-раздела с нового узла.

Создание конфигурации кластера возможно как с помощью команд, так и путём ручного редактирования файла **/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 узлы (команду необходимо выполнить для каждого узла):

```
#
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 совпадать со значением, указанным 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 выполните:

```
# /etc/init.d/o2cb load
checking debugfs...
Loading filesystem "ocfs2_dlmfs": OK
Creating directory '/dlm': OK
Mounting ocfs2_dlmfs filesystem at /dlm: OK
```

Затем переведите кластер в онлайн-состояние:

```
# /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:

1. Создать физический раздел **/dev/mapper/mpatha-part1** на устройстве **/dev/mapper/mpatha**:

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

2. Отформатировать созданный раздел:

   ```
   # 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
   ```

Таблица 29.2. Параметры команды mkfs.ocfs2

Параметр	Описание
-L метка_тома	Метка тома, позволяющая его однозначно идентифицировать его при подключении на разных узлах. Для изменения метки можно использовать утилиту **tunefs.ocfs2**
-C размер_кластера	Размер кластера — минимальная единица пространства, выделяемая файлу. Возможные значения: 4, 8, 16, 32, 64, 128, 256, 512 и 1024 КБ. Не может быть изменён после форматирования
-N количество_узлов_кластера	Максимальное количество узлов, которые могут одновременно монтировать том. Может быть изменено с помощью **tunefs.ocfs2**
-b размер_блока	Наименьшая единица пространства, адресуемая ФС. Возможные значения: 512 байт (не рекомендуется), 1 КБ, 2 КБ или 4 КБ (рекомендуется). Не может быть изменён после форматирования

Примечание

Для создания нового раздела может потребоваться предварительно удалить существующие данные на устройстве **/dev/mapper/mpathX** (использовать с осторожностью):

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

29.4.4.6.2. Использование OCFS2 в OpenNebula

На каждом узле OpenNebula необходимо добавить OCFS2 в список файловых систем, автоматически монтируемых при загрузке узла:

1. Определить UUID раздела:

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

2. Добавить запись 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
```

3. Проверить монтирование:

```
# mount -a
```

   Команда должна завершиться без вывода ошибок.

Пример итоговой 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** гарантирует, что раздел будет монтироваться только после запуска сетевой подсистемы.

Важно

При использовании файлового хранилища после добавления записи в **/etc/fstab** и перезагрузки ОС необходимо назначить владельцем каталога пользователя oneadmin. Например:

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

Примечание

Для просмотра OCFS2 файловых систем можно использовать команды:

```
# 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
DF49216AA83547C6B7C16962E9B7DCB6  DBF
```

29.4.4.7. Разметка системного хранилища

LUN для системного хранилища будет обслуживаться менеджером томов LVM.

Предварительные условия:

- служба lvmetad должна быть отключена. В файле **/etc/lvm/lvm.conf** необходимо установить:

  ```
  use_lvmetad = 0
  ```

  и отключить службу, если она запущена:

  ```
  # systemctl disable --now lvm2-lvmetad.service
  ```

- пользователь oneadmin добавлен 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-<ID_хранилища>.

Примечание

LUN должен быть доступен по пути **/dev/mapper/** и не содержать разметки.

При необходимости его можно очистить:

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

Настройка на узле управления:

1. Создать физический том (PV) на LUN:

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

2. Создать группу томов с именем vg-one-<ID_хранилища>

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

3. 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. Отображаемый объём должен соответствовать размеру LUN, выделенного на СХД.

После создания и запуска ВМ будет создан логический том:

```
# 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.

29.4.5. Хранилище файлов и ядер

Хранилище файлов и ядер предназначено для хранения обычных файлов, используемых в качестве ядер ВМ, виртуальных дисков или любых других файлов, которые необходимо передать ВМ в процессе контекстуализации.

Данный тип хранилища не предоставляет специальных механизмов хранения, но представляет собой простой и безопасный способ использования файлов в шаблонах ВМ.

Регистрация хранилища файлов и ядер:

1. Создайте файл **kernels_ds.conf**:

```
NAME    = kernels
DS_MAD  = fs
TM_MAD  = ssh
TYPE    = FILE_DS
SAFE_DIRS = /var/tmp/files
```

2. Зарегистрируйте хранилище:

```
$ onedatastore create kernels_ds.conf
ID: 107
```

Для создания хранилища файлов и ядер необходимо указать следующие параметры:

- ***NAME*** — название хранилища;
- ***TYPE*** — FILE_DS;
- ***DS_MAD*** — fs;
- ***TM_MAD*** — ssh.

Примечание

Приведённые выше значения ***DS_MAD*** и ***TM_MAD*** являются рекомендованными, однако при необходимости могут использоваться и другие поддерживаемые драйверы.

Рекомендуемый драйвер передачи ***ssh*** для хранилища файлов не требует специальной настройки на узлах виртуализации. Достаточно убедиться, что в каталоге размещения хранилищ данных (по умолчанию — **/var/lib/one/datastores**) имеется достаточно свободного пространства для хранения файлов ВМ.

29.5. Изменение параметров хранилища

Для изменения параметров существующего хранилища необходимо создать файл с требуемыми настройками и выполнить команду:

```
$ onedatastore update <ID_хранилища> <имя_файла>
```

Глава 30. Работа с образами в OpenNebula

OpenNebula позволяет пользователям создавать и управлять образами дисков и вспомогательных файлов, которые используются при развёртывании ВМ. Образы хранятся в хранилищах данных и могут совместно использоваться несколькими ВМ в зависимости от их типа и режима (постоянный/непостоянный).

Таблица 30.1. Типы образов

Тип	Описание	Хранилище
OS	Загрузочный диск с установленной операционной системой	Images Datastore
CDROM	ISO-образ (только для чтения). Можно подключить несколько образов, но загрузка возможна только с одного из них	Images Datastore
DATABLOCK	Блок данных (например, для размещения базы данных) или пустой диск для установки ОС	Images Datastore
KERNEL	Ядро Linux (vmlinuz)	Files & Kernels Datastore
RAMDISK	Образ initrd/initramfs	Files & Kernels Datastore
CONTEXT	Файл контекстуализации (для автоматической начальной настройки ВМ)	Files & Kernels Datastore

Примечание

Образы KERNEL и RAMDISK указываются в шаблоне ВМ через атрибуты ***`OS[kernel=...]`*** и ***`OS[initrd=...]`***, а не подключаются как отдельные диски.

Образы могут использоваться в двух режимах:

- Постоянные (persistent) — изменения, внесённые в образ во время работы ВМ, сохраняются после её завершения. Такой образ может одновременно использоваться только одной ВМ.
- Непостоянный (non-persistent) — изменения не сохраняются. При запуске ВМ создаётся временная копия образа, которая удаляется при остановке ВМ.

Образы в OpenNebula проходят через различные состояния, отражающие текущий этап их обработки.

Примечание

В выводе **`oneimage list`** отображается сокращённое имя STATE (например, rdy, disa).

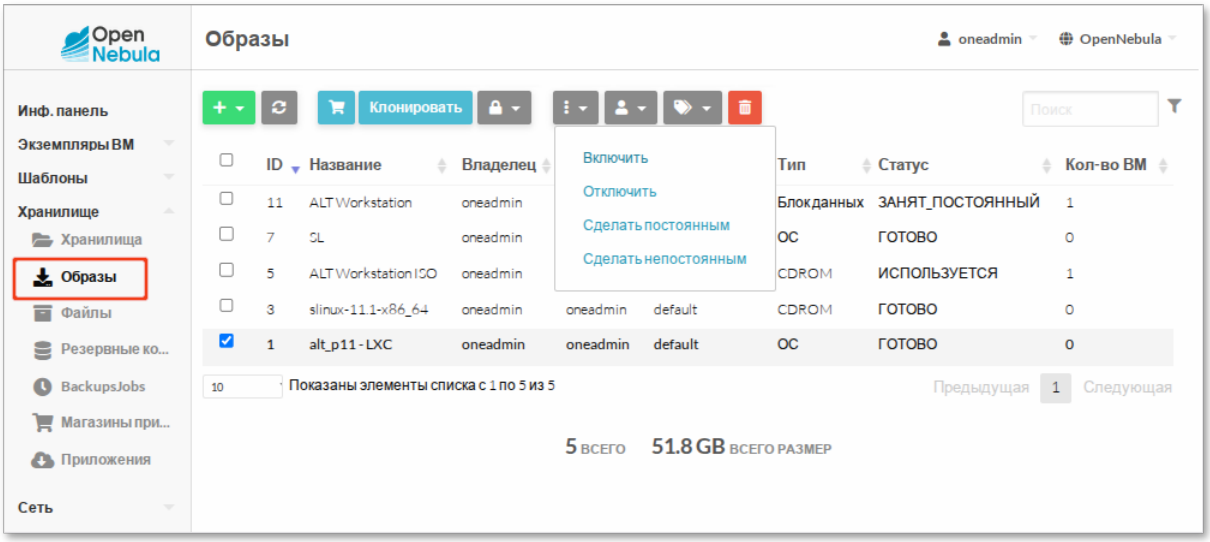
Таблица 30.2. Состояния образов

Состояние	Краткий псевдоним	Описание
INIT	init	Состояние инициализации
READY (ГОТОВО)	rdy	Образ готов к использованию
USED (ИСПОЛЬЗУЕТСЯ), USED_PERS (ЗАНЯТ_ПОСТОЯННЫЙ)	used	Образ используется как минимум одной ВМ. Образ нельзя удалить или изменить
DISABLED (Отключен)	disa	

Состояние	Краткий псевдоним	Описание
		Образ отключён владельцем. Новые ВМ не могут его использовать
LOCKED (ЗАБЛОКИРОВАН), LOCKED_USED, LOCKED_USED_PERS (ЗАБЛОКИРОВАН_ПОСТОЯННЫЙ)	lock	Образ копируется или создаётся. ВМ, использующие его, ожидают завершения операции
ERROR (Ошибка)	err	Операция завершилась неудачей (см. **`oneimage show`** для получения сообщения об ошибке)
CLONE (Клонирование)	clon	Образ клонируется
DELETE (Удаление)	dele	Образ удаляется из хранилища данных

Управление образами осуществляется с помощью:

- командной строки (**`oneimage`**);
- веб-интерфейса Sunstone.



Основные параметры команды **`oneimage`**:

- ***`--name <имя>`*** — имя нового образа;
- ***`--datastore <имя|ID>`*** — имя или идентификатор хранилища данных;
- ***`--description <описание>`*** — описание образа (необязательно);
- ***`--type <тип>`*** — тип образа: OS, CDROM, DATABLOCK, KERNEL, RAMDISK, CONTEXT;
- ***`--persistent`*** — создание постоянного образа;
- ***`--prefix <префикс>`*** — префикс устройства в гостевой ОС (hd, sd, vd);
- ***`--target <устройство>`*** — целевое устройство подключения диска;

- ***--path <путь/URL>*** — путь к файлу образа или URL-адрес;
- ***--source <источник>*** — источник данных для использования (применяется для нефайловых образов);
- ***--size <размер>*** — размер в МБ (используется для DATABLOCK или при изменении размера).

30.1. Создание образов дисков

При создании образов дисков для ВМ в OpenNebula доступны следующие подходы:

- ручная установка — установка ОС непосредственно в запущенной ВМ;
- приложения OpenNebula Marketplace — развёртывание готовых образов и шаблонов из официального каталога.

30.1.1. Ручная установка

При использовании гипервизора KVM возможно создание базовых образов операционных систем путём ручной установки ОС в ВМ, развёрнутой в OpenNebula. Этот метод обеспечивает максимальный контроль над процессом установки и конфигурацией системы.

Процесс ручного создания образа включает следующие шаги:

- **Шаг 1. Добавление установочного носителя**

  В хранилище образов необходимо загрузить установочный ISO-образ операционной системы и создать образ типа CDROM.

- **Шаг 2. Создание установочного диска**

  Диск, на который будет установлена операционная система, необходимо создать как образ типа DATABLOCK.

- **Шаг 3. Создание шаблона и экземпляра ВМ**

  Необходимо создать шаблон ВМ с подключёнными:

  - установочным CDROM;
  - пустым диском типа DATABLOCK.

  После этого создать экземпляр ВМ и выполнить установку гостевой ОС.

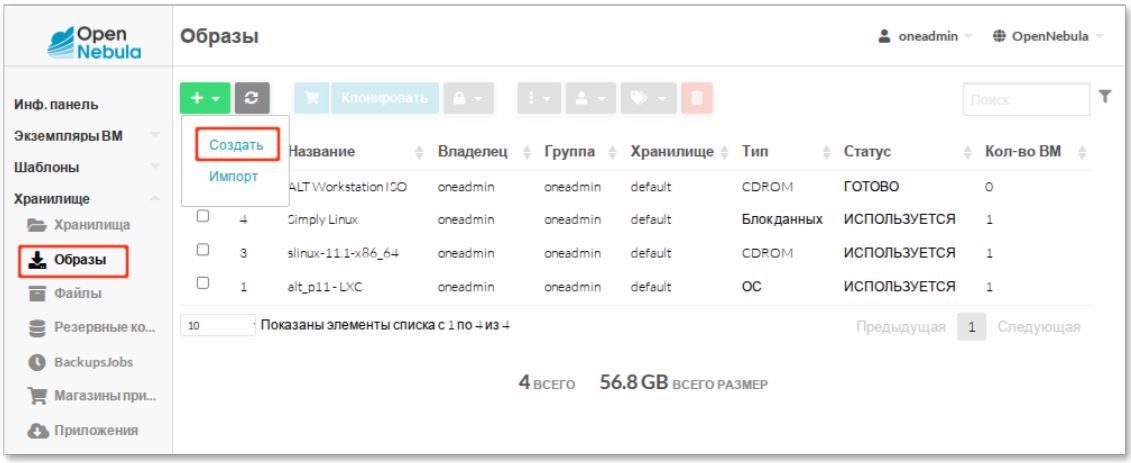
- **Шаг 4. Выключение ВМ и подготовка образа**

  После завершения установки и настройки операционной системы ВМ необходимо выключить. Постоянный диск с установленной ОС сохраняется в хранилище данных и может быть преобразован в образ типа OS для дальнейшего использования при массовом развёртывании ВМ.

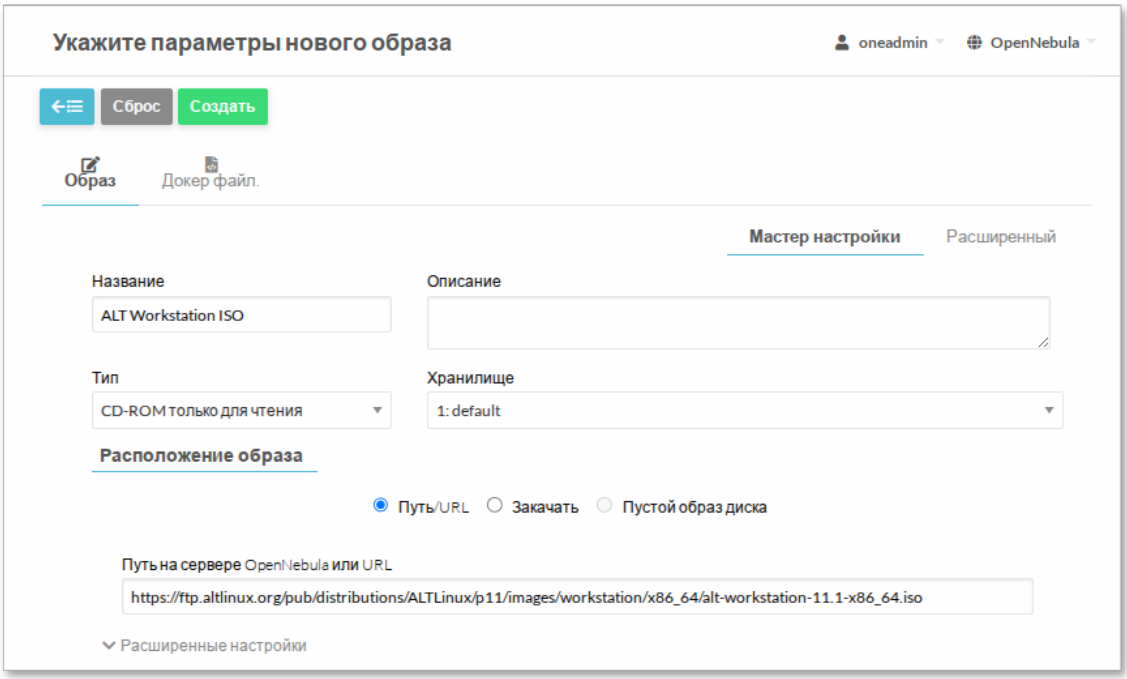
30.1.1.1. Создание образа типа CDROM

Создание образа типа CDROM (установочный ISO-образ):

1. Перейдите в раздел **Хранилище** → **Образы**.
2. На загруженной странице нажмите **+** → **Создать**:



3. В открывшемся окне укажите:
   - **Название**;
   - **Тип** — **CD-ROM только для чтения**;
   - **Хранилище**;
   - **Расположение образа** — **Путь/URL**;
   - **Путь на сервере OpenNebula или URL** — путь к файлу ISO-образа.



4. Нажмите кнопку **Создать**.

Примечание

Если указывается путь на сервере управления, ISO-образ должен находиться в каталоге, доступном пользователю oneadmin.

Пример создания образа типа CDROM в командной строке:

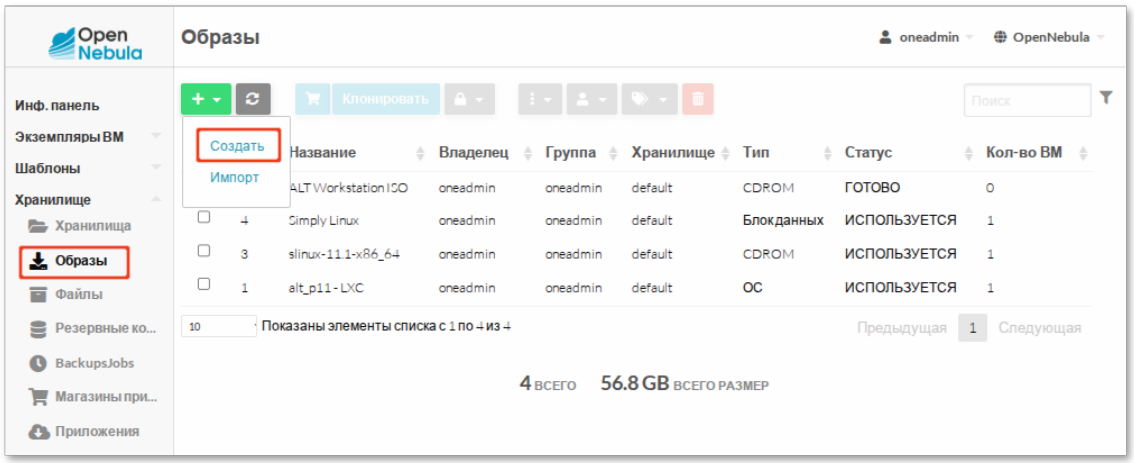
```
$ oneimage create -d 1 --name "ALT Workstation ISO" \
    --path /var/tmp/alt-workstation-11.1-x86_64.iso --type CDROM
ID: 31
```

где **`-d 1`** — хранилище данных по умолчанию (ID = 1).

30.1.1.2. Создание образа типа DATABLOCK

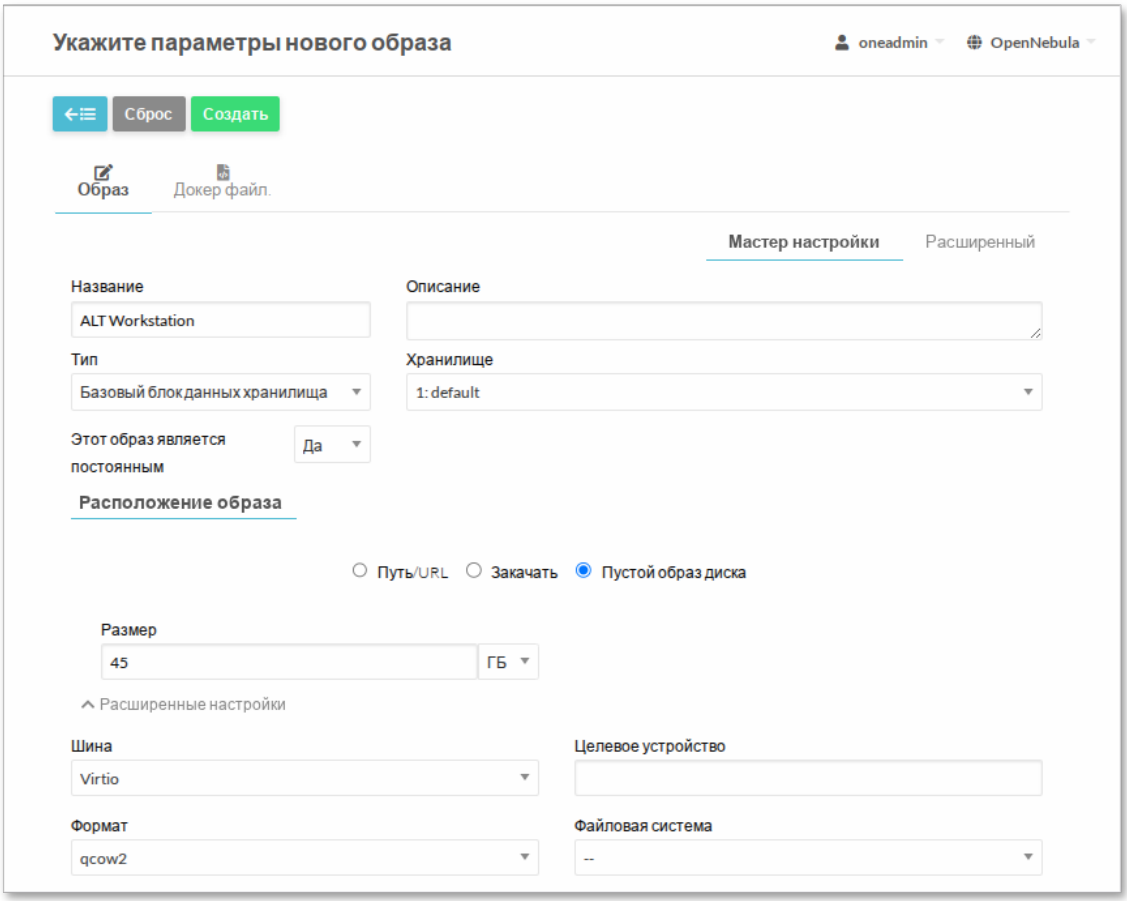
Создание пустого диска (DATABLOCK):

1. В разделе **Хранилище** → **Образы** нажмите **+** → **Создать**:



2. В открывшемся окне укажите:

   - **Название**;
   - **Тип** — **Базовый блок данных хранилища**;
   - **Хранилище**;
   - **Этот образ является постоянным** — **Да**;
   - **Расположение образа** — **Пустой образ диска**;
   - **Размер**;
   - **Формат** — **qcow2**.



3. Нажмите кнопку **Создать**.

Пример создания пустого образа диска в командной строке:

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

где:

- **-d 1** — хранилище данных по умолчанию (ID = 1);
- **--size 45G** — размер образа (45 ГБ);
- **--driver qcow2** — формат образа qcow2.

Примечание

Не задавайте слишком большой размер диска — его можно увеличить позднее при создании ВМ.

30.1.1.3. Создание шаблона ВМ

Шаблон определяет конфигурацию ВМ: количество CPU, объём памяти, диски, сетевые интерфейсы и другие параметры.

30.1.1.3.1. Создание шаблона в командной строке

Для создания шаблона в командной строке:

1. Создайте файл **`template`**:

```
NAME = "ALT Workstation"
CPU = "1"
MEMORY = "1024"
MEMORY_UNIT_COST = "MB"
HYPERVISOR = "kvm"

DISK = [
  IMAGE = "ALT Workstation ISO",
  IMAGE_UNAME = "oneadmin"
]

DISK = [
  DEV_PREFIX = "vd",
  IMAGE = "ALT Workstation",
  IMAGE_UNAME = "oneadmin"
]

NIC = [
  NETWORK = "VirtNetwork",
  NETWORK_UNAME = "oneadmin",
  SECURITY_GROUPS = "0"
]

INPUTS_ORDER = ""
LOGO = "images/logos/alt.png"

NIC_DEFAULT = [
  MODEL = "virtio" ]

OS = [
  BOOT = "disk1,disk0"  # Сначала основной диск (disk1), затем CDROM
(disk0)
]

GRAPHICS = [
  LISTEN = "0.0.0.0",
  TYPE = "SPICE"
]

CONTEXT = [
  NETWORK = "YES",
  SSH_PUBLIC_KEY = "$USER[SSH_PUBLIC_KEY]"
]

SCHED_REQUIREMENTS = "ID=\"0\""
```

2. Зарегистрируйте шаблон:

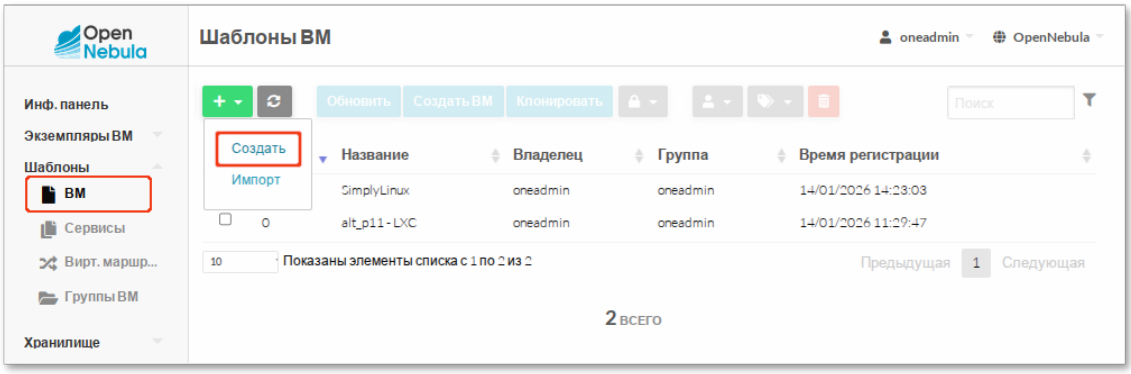
```
$ onetemplate create template
ID: 22
```

30.1.1.3.2. Создание шаблона в веб-интерфейсе Sunstone

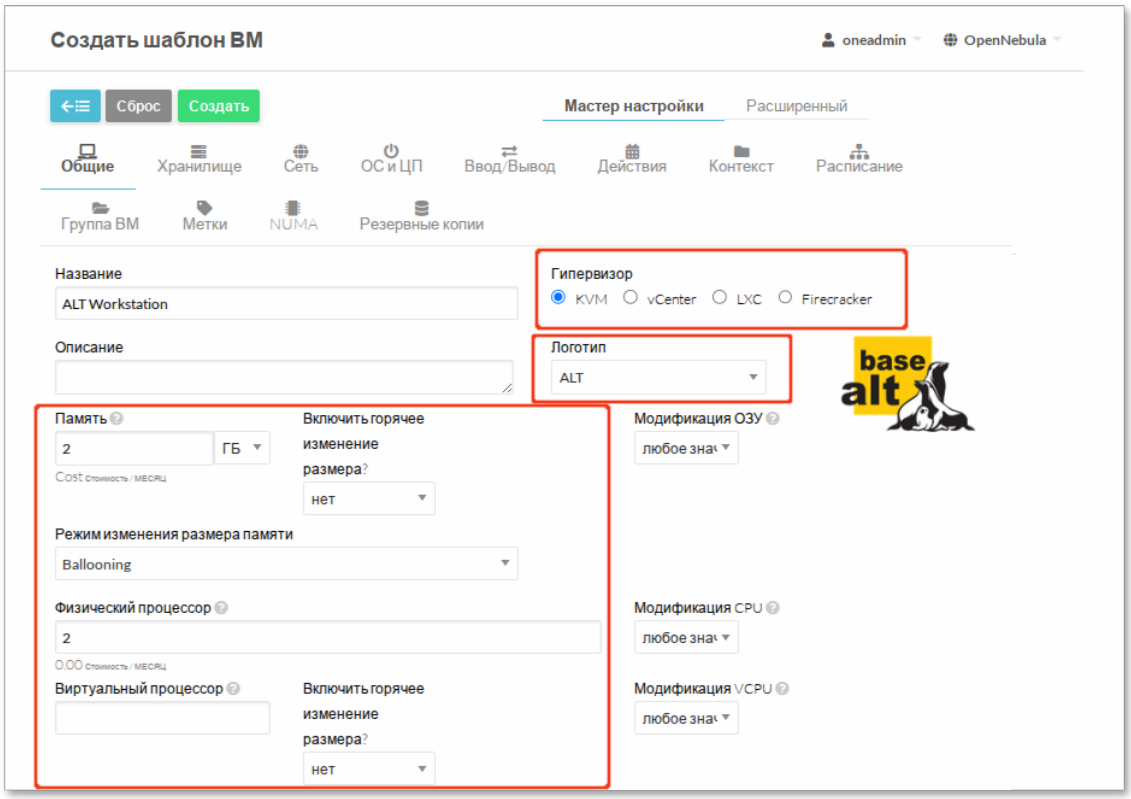
Для создания шаблона ВМ:

1. В левом меню выберите **Шаблоны** → **ВМ**.

2. На загруженной странице нажмите **+** → **Создать**:

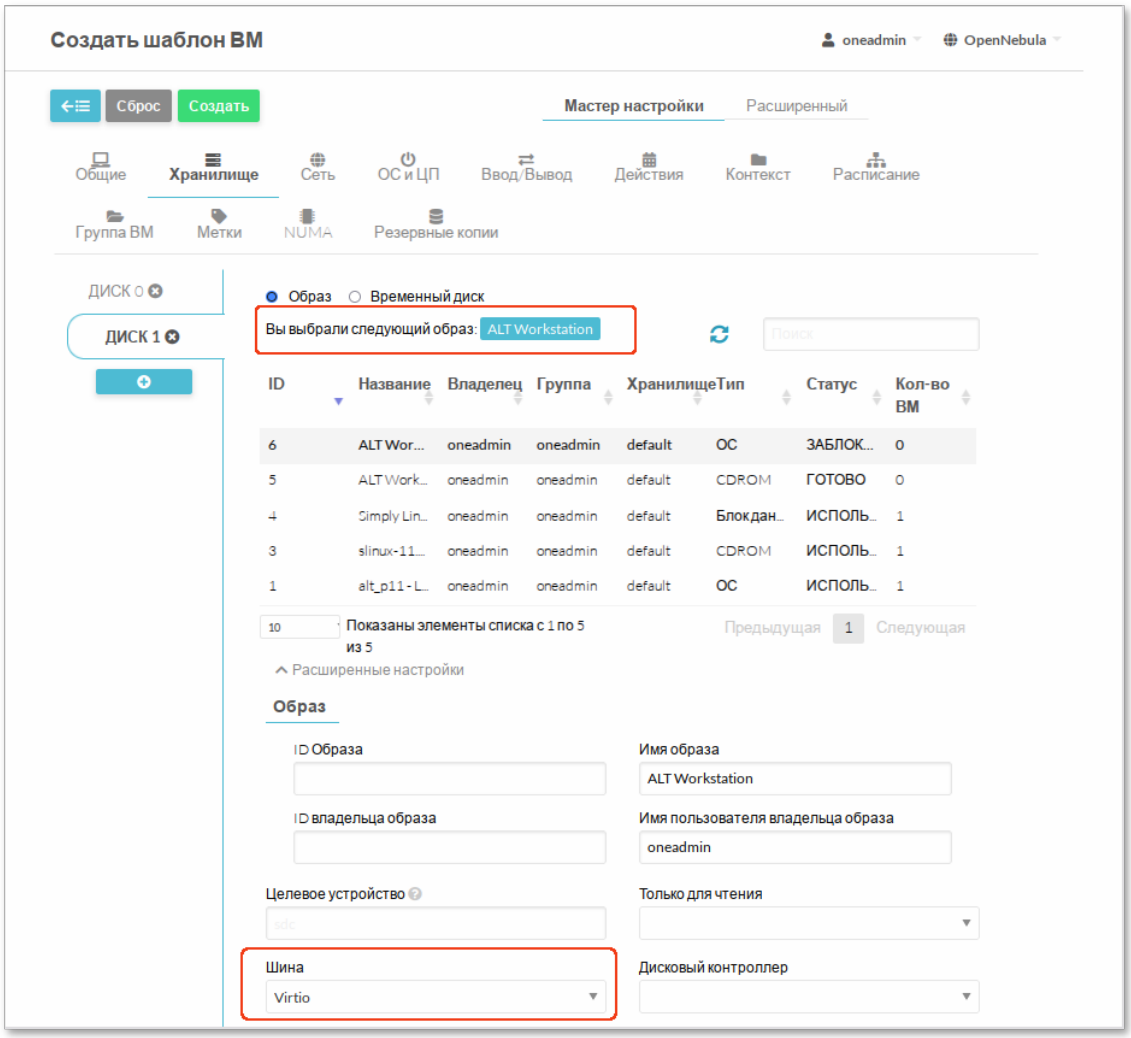


3. На вкладке **Общие** укажите параметры CPU, объём памяти и гипервизор:

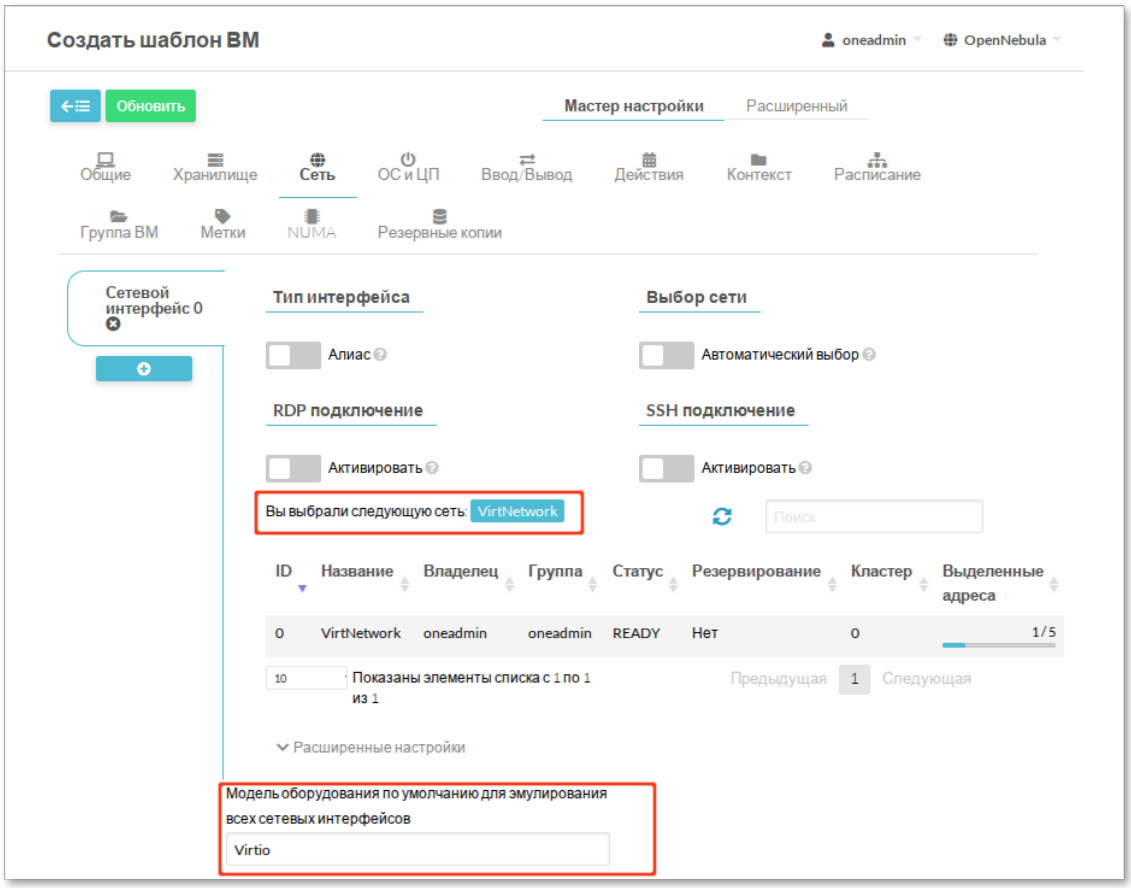


4. На вкладке **Хранилище**:

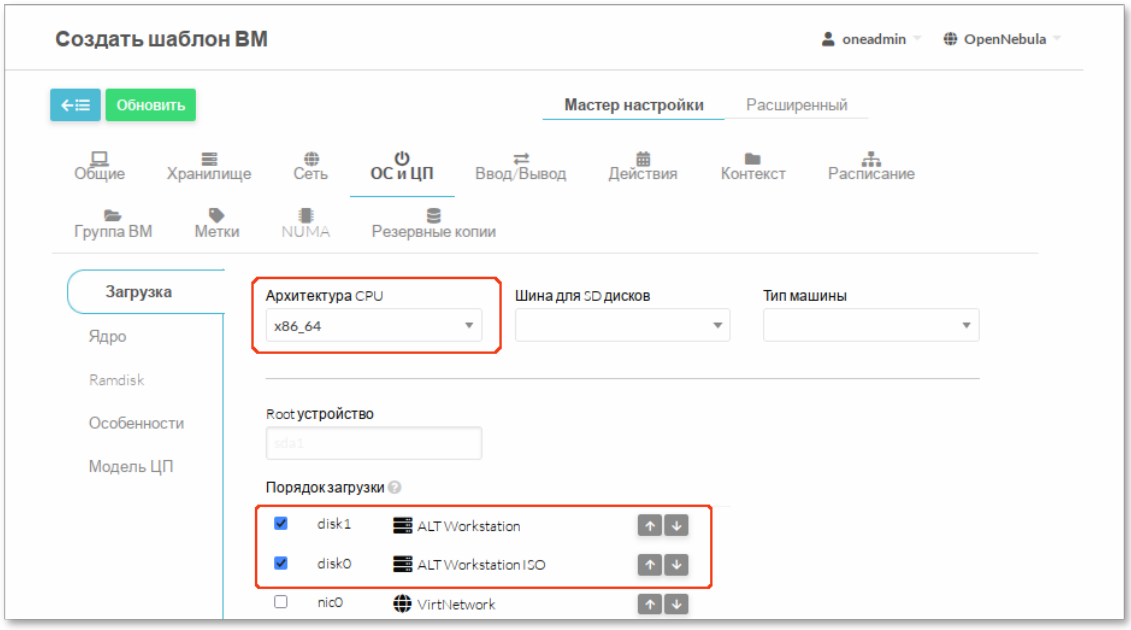
   - добавьте ISO-образ как первый диск;
   - добавьте пустой DATABLOCK как второй диск и укажите для него шину **`Virtio`**.



5. На вкладке **Сеть** укажите модель сетевого адаптера — **`Virtio`** и сеть, к которой будет подключена ВМ:



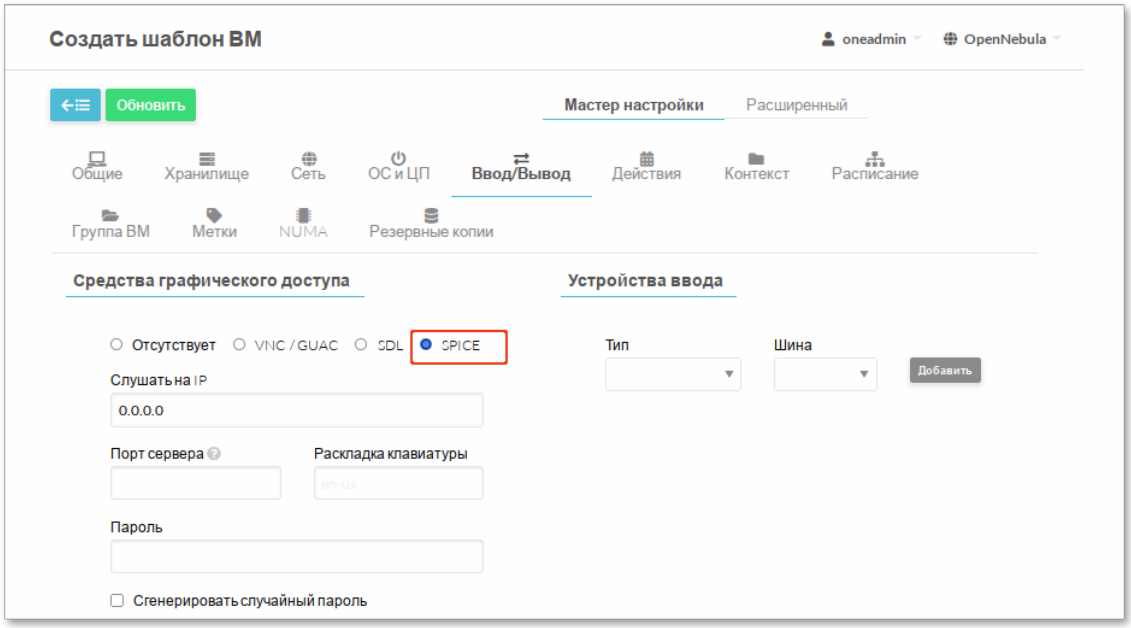
6. На вкладке **ОС и ЦПУ** укажите архитектуру устанавливаемой системы и задайте порядок загрузки — диск, затем CDROM:



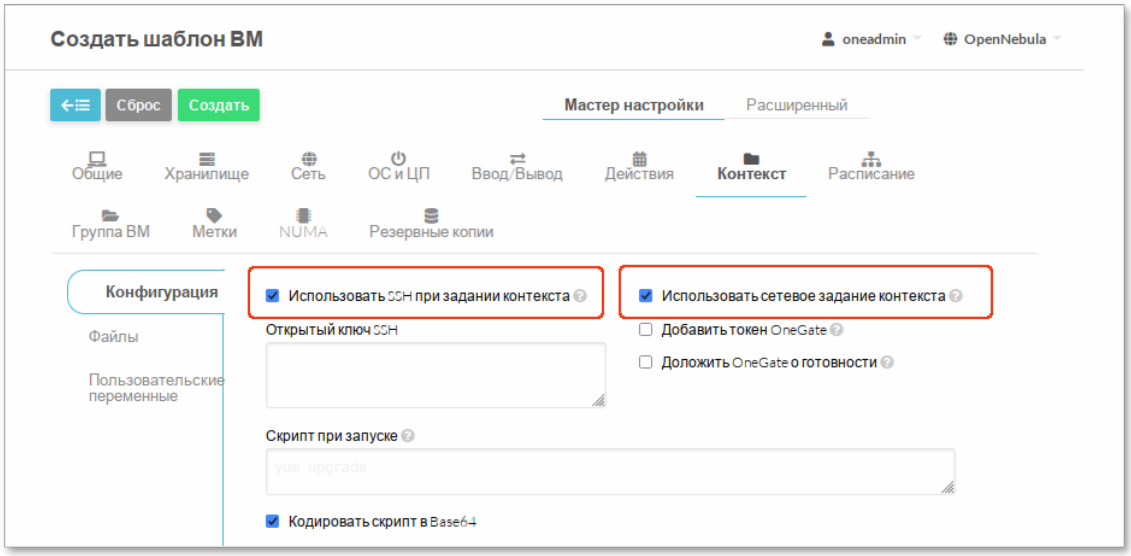
Примечание

При таком порядке загрузки, если диск пуст, загрузка произойдёт с CDROM. После установки операционной системы загрузка будет автоматически выполняться с диска.

7. На вкладке **Ввод/Вывод** выберите тип графической консоли, например SPICE:



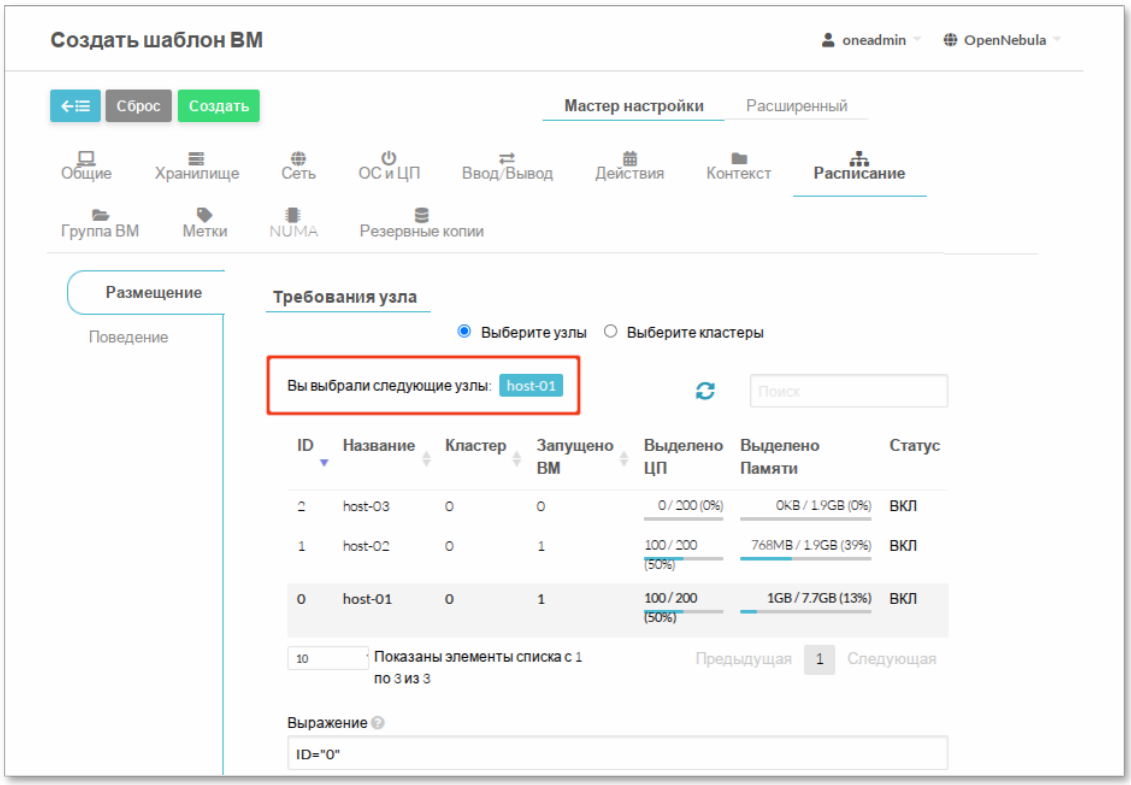
8. На вкладке **Контекст** включите сетевую контекстуализацию и укажите SSH-ключ:



Примечание

Если поле SSH-ключа оставить пустым, будет использована переменная **`$USER[SSH_PUBLIC_KEY]`**.

9. На вкладке **Расписание** при необходимости укажите кластер или конкретный узел, на котором должна быть размещена ВМ:

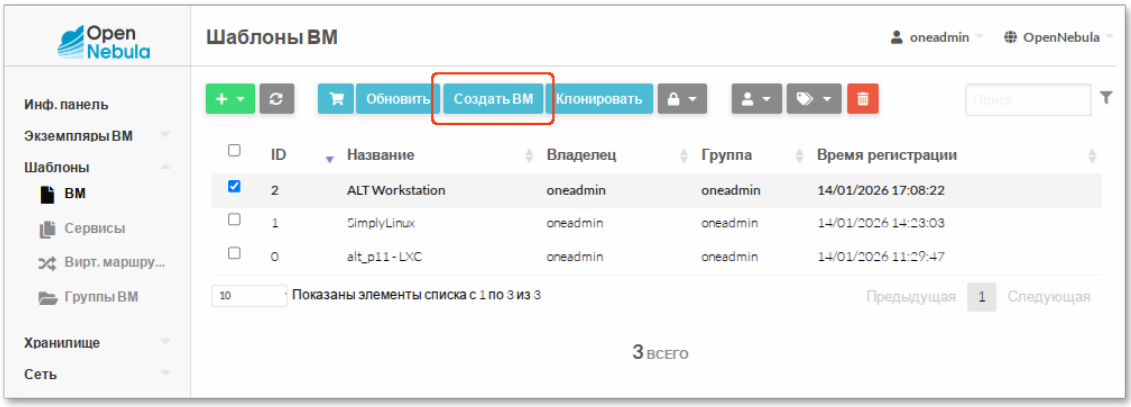


10. Нажмите кнопку **`Создать`**.

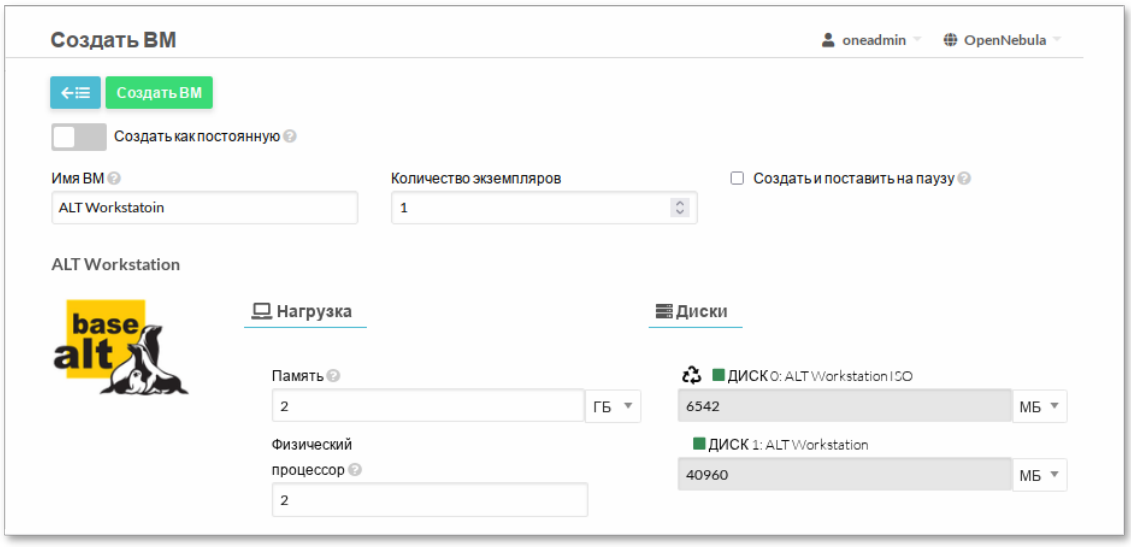
30.1.1.4. Создание ВМ

Создание ВМ в веб-интерфейсе Sunstone:

1. В левом меню выберите **Шаблоны → ВМ**.
2. Выберите нужный шаблон и нажмите **`Создать ВМ`**:



3. В открывшемся окне укажите имя ВМ и нажмите **`Создать ВМ`**:



Создание ВМ из шаблона в командной строке:

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

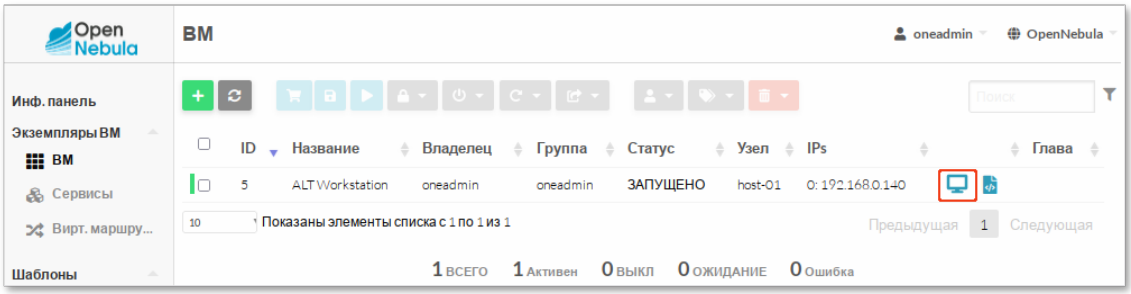
30.1.1.5. Подключение к ВМ и установка ОС

Примечание

Процесс создания ВМ может занять несколько минут. Необходимо дождаться, пока её статус изменится на **ЗАПУЩЕНО** (**RUNNING**).

Подключение через Sunstone:

1. Перейдите в раздел **Экземпляры ВМ** → **ВМ**.
2. Выберите ВМ и подключитесь к ней через SPICE:



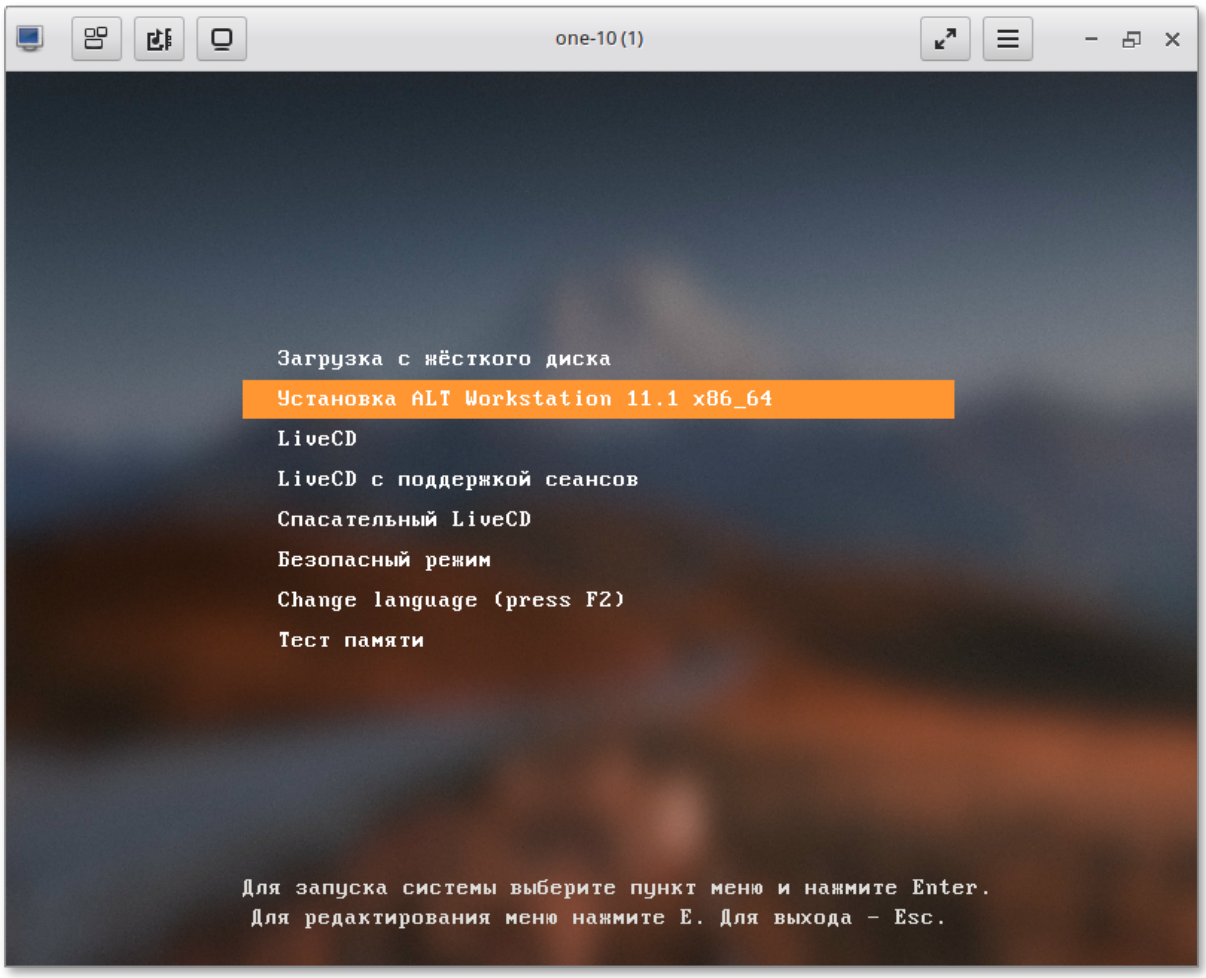
Для подключения через внешний клиент SPICE нужно указать:

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

где:

- 192.168.0.180 — IP-адрес узла с ВМ;
- 5 — идентификатор ВМ (номер порта равен 5900 + ID ВМ).

Выполните установку операционной системы в консоли ВМ:



30.1.1.6. Настройка контекстуализации

Контекстуализация позволяет автоматически настраивать ВМ при запуске: задавать имя узла, сетевые параметры, SSH-ключи и другие настройки.

Пример настройки контекстуализации на установленной ОС «Альт»:

1. Подключитесь к ВМ (через консоль или SSH).
2. Установите пакет контекстуализации (*opennebula-context*):

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

3. Настройте сеть через systemd-networkd:
   - установите пакет синхронизации времени (*systemd-timesyncd*):

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

   - создайте файл **/etc/systemd/network/lan.network** для автоматической настройки всех интерфейсов по DHCP:

```
[Match]
Name = *

[Network]
DHCP = ipv4
```

- отключите etcnet и NetworkManager, включите systemd-networkd и systemd-timesyncd:

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

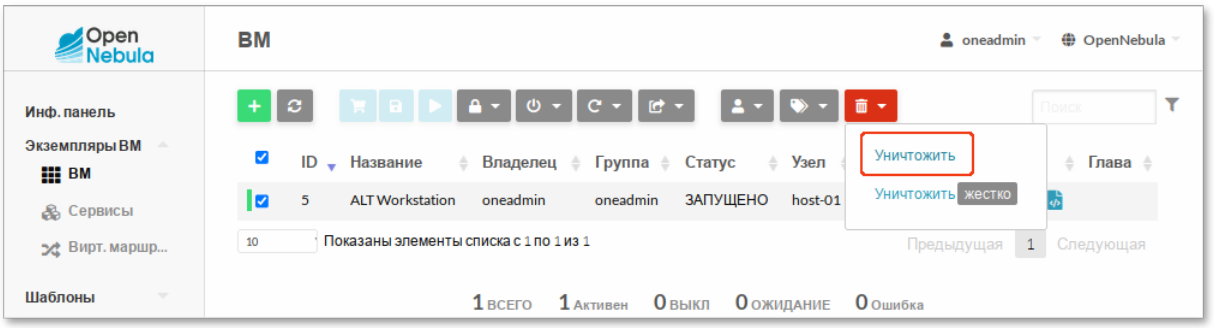
4. Перезагрузите ВМ.

После перезагрузки ВМ получит IP-адрес из пула OpenNebula (через IPAM) и будет доступна по SSH с использованием ключа из профиля пользователя.

30.1.1.7. Создание образа типа ОС

После завершения установки и настройки системы ВМ необходимо выключить и удалить.

Для удаления ВМ в Sunstone в разделе **Экземпляры ВМ** → **ВМ** выберите ВМ и нажмите кнопку **`Уничтожить`**:



Удаление ВМ в командной строке:

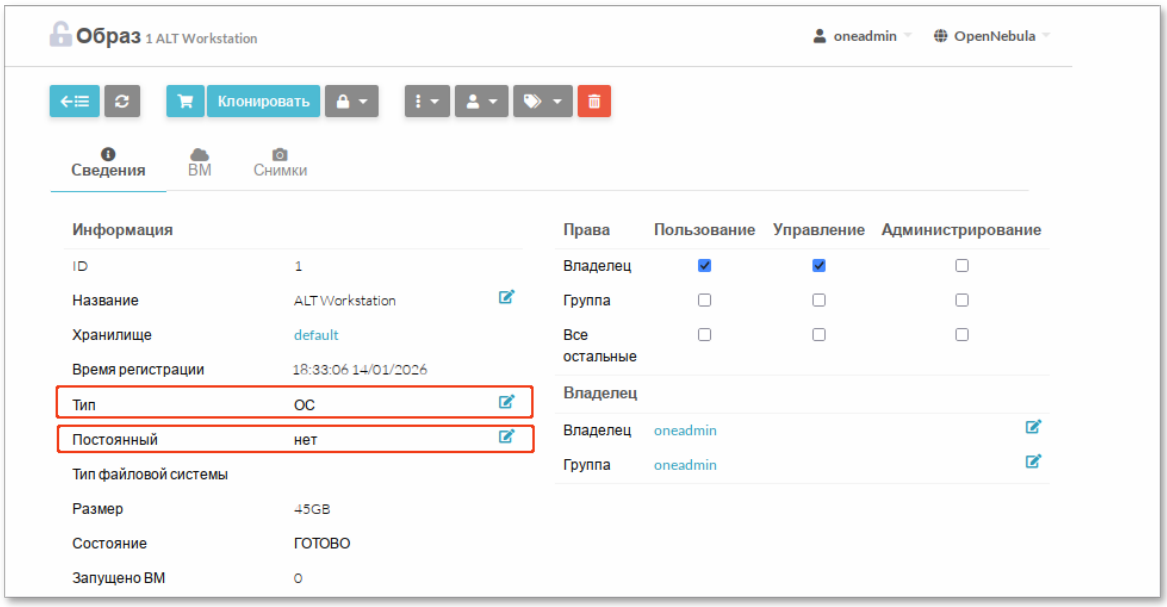
```
$ onevm terminate 5
```

Примечание

Поскольку диск был постоянным (persistent), он останется в хранилище и сохранит все внесённые изменения.

Преобразуйте образ в тип OS и сделайте его непостоянным:

- В Sunstone:
  - В разделе **Хранилище** → **Образы ВМ** выберите образ с установленной ОС.
  - Измените тип блочного устройства с **`Блок данных`** на **`ОС`**.
  - Установите состояние **`Не постоянный`**.



- в командной строке:

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

Примечание

Это позволит использовать образ для массового развёртывания ВМ, при котором каждая ВМ получит собственную копию диска.

Образ готов к использованию. Теперь можно:

- создавать новые ВМ на его основе;
- обновлять и использовать существующие шаблоны ВМ.

30.1.2. Использование магазинов приложений OpenNebula

Магазины приложений (Marketplace) в OpenNebula предоставляют удобный способ интеграции облачной инфраструктуры с популярными поставщиками готовых образов и приложений.

Магазин приложений может быть:

- публичным — доступным для всех установок OpenNebula (например, официальный OpenNebula Public);
- частным — размещённым локально внутри организации и может использоваться:
  - в одной зоне OpenNebula;
  - во всех зонах федерации (если настроен общий доступ).

Магазин хранит приложения Marketplace (Marketplace Appliance) — готовые к развёртыванию пакеты, включающие:

- один или несколько образов (дисков);
- метаданные: шаблоны виртуальных машин, определения сервисов из нескольких ВМ (Multi-VM Services), скрипты контекстуализации и т. д.

Примечание

Для доступа к публичным магазинам (включая OpenNebula Public) требуется подключение к Интернету.

30.1.2.1. Управление магазинами

По умолчанию включён только официальный публичный магазин OpenNebula. Остальные магазины изначально отключены.

Чтобы включить магазин приложений, используется команда:

```
$ onemarket enable <ID_магазина>
```

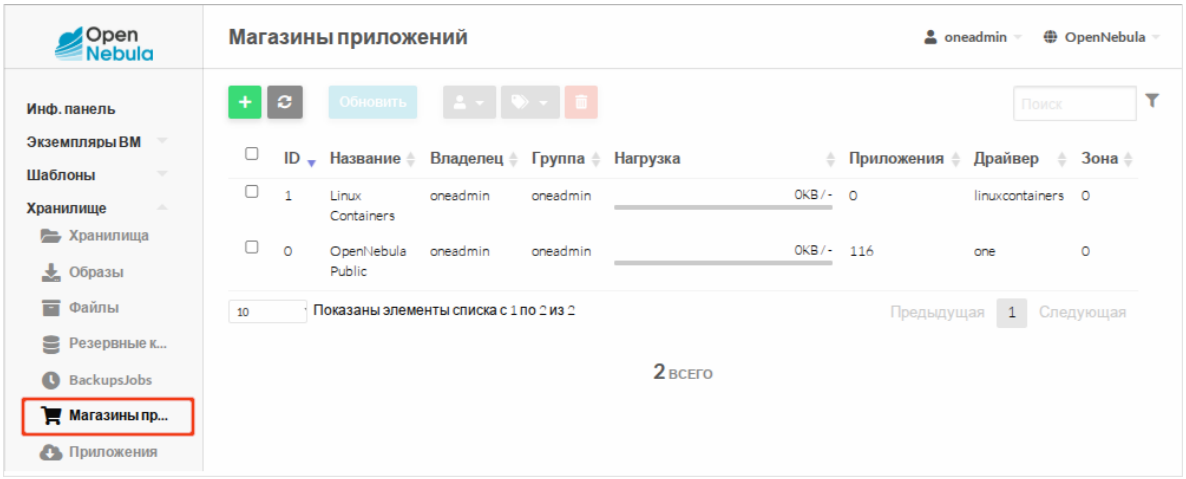
Отключить магазин можно с помощью команды:

```
$ onemarket disable <ID_магазина>
```

Отключение магазина не удаляет уже импортированные в облако приложения. Однако новые приложения из этого магазина перестанут отображаться, и синхронизация будет остановлена.

Список магазинов можно можно посмотреть:

- в веб-интерфейсе: **Хранилище** → **Магазины приложений**:



- в командной строке:

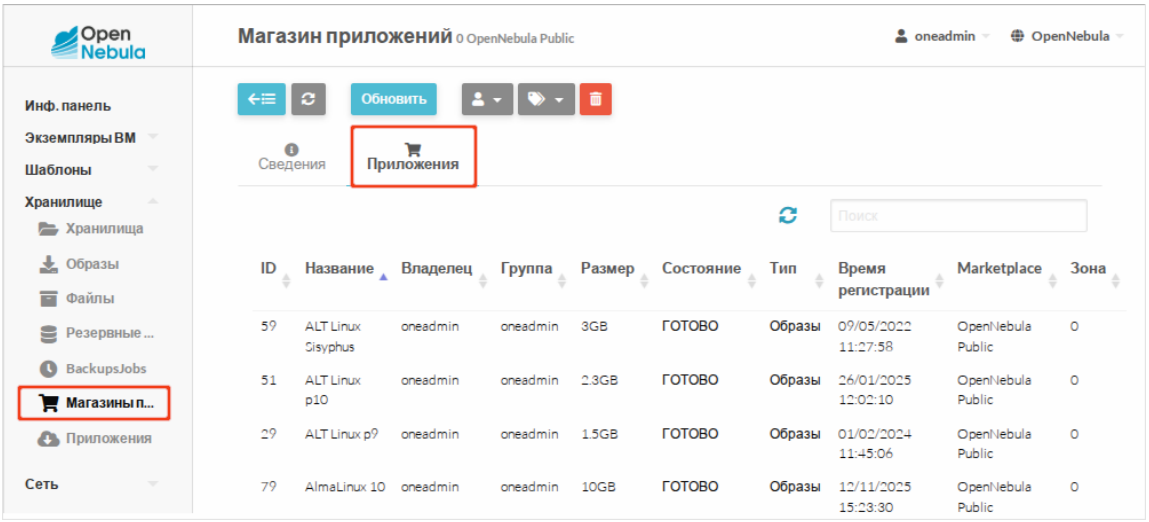
```
$ onemarket list
  ID NAME                                SIZE AVAIL        APPS MAD
ZONE STAT
   1 Linux Containers                      0M -               0 linuxco
0 off
   0 OpenNebula Public                     0M -             116 one
0 on
```

30.1.2.2. OpenNebula Public

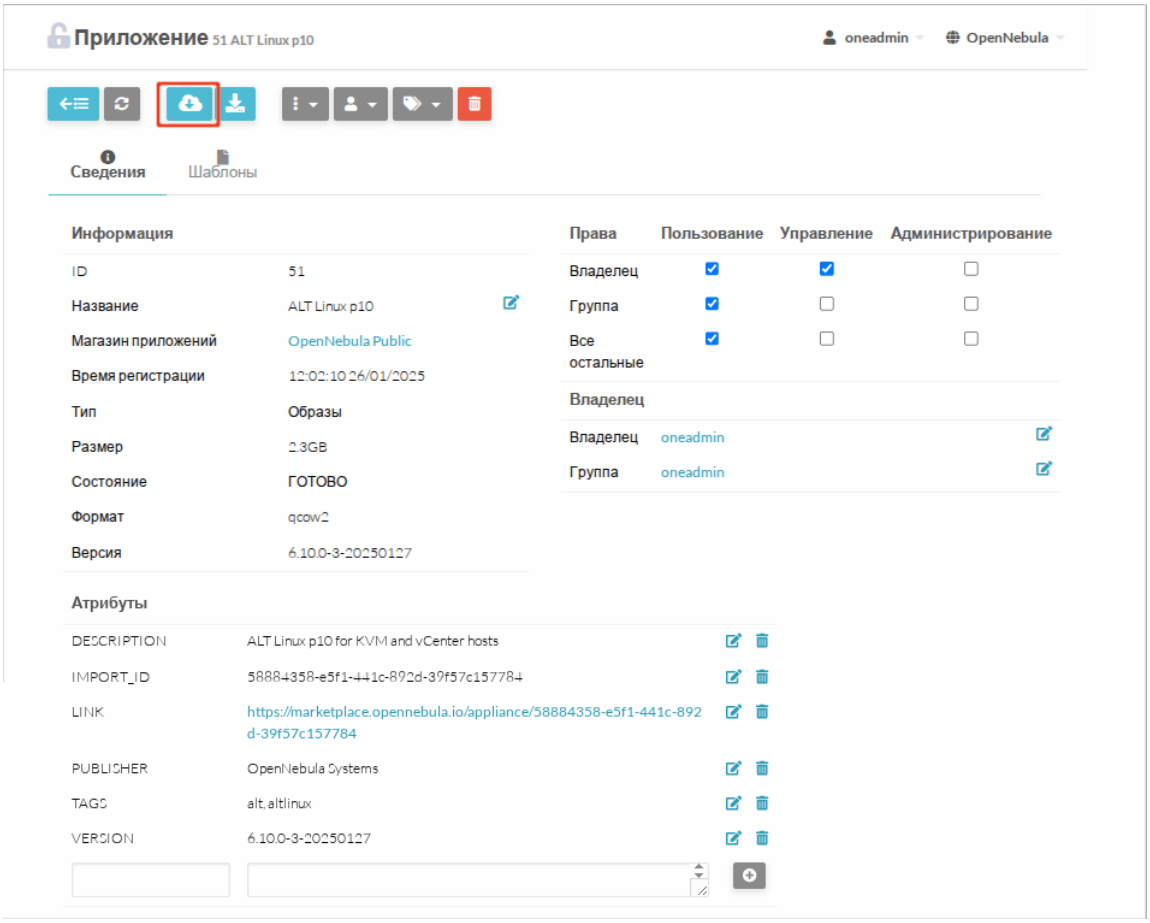
OpenNebula Public — это официальный каталог готовых к использованию виртуальных устройств и приложений.

Импорт приложения:

1. В веб-интерфейсе перейдите: **Хранилище** → **Магазины приложений** → **OpenNebula Public** → **Приложения**:

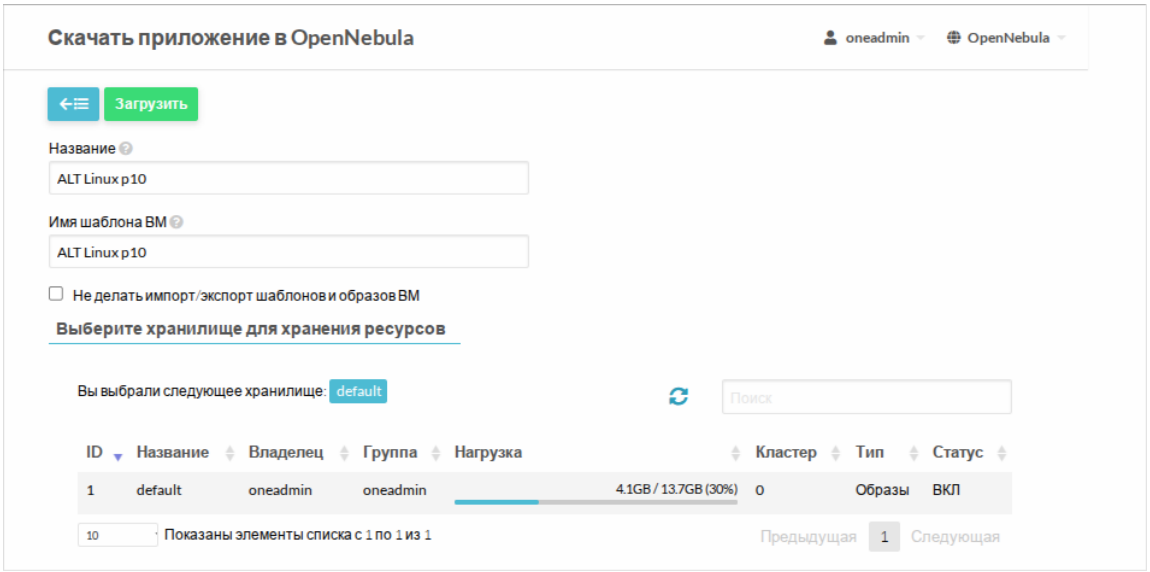


2. Выберите нужное приложение и нажмите кнопку **`Импорт в хранилище`**:



3. В открывшемся окне:

- укажите имя образа и имя шаблона;
- выберите целевое хранилище данных;
- нажмите кнопку **Загрузить**:



Каждое приложение содержит как минимум один образ и шаблон виртуальной машины.

Настройка образов, загруженных из магазина приложений:

1. Дождитесь, пока образ перейдёт в состояние ГОТОВО.
2. Измените тип образа на **`Постоянный`** (если планируете вносить изменения).
3. При необходимости отредактируйте шаблон (сеть, CPU, память и т. д.).
4. Создайте ВМ на основе шаблона.
5. Подключитесь к ВМ и выполните дополнительную настройку (установка ПО, обновление и т. д.).
6. После завершения настройки:
   - остановите и удалите ВМ;
   - измените тип образа обратно на **`Не постоянный`**, чтобы сэкономить место и обеспечить «чистое» состояние для будущих развёртываний.

Готовый образ можно использовать повторно или создавать на его основе новые шаблоны.

30.1.2.3. Linux Containers

Магазин Linux Containers предоставляет доступ к популярным образам контейнеров LXC (ALT, Ubuntu, CentOS и др.), адаптированным для работы в OpenNebula.

Примечание

Для импорта образа требуется:

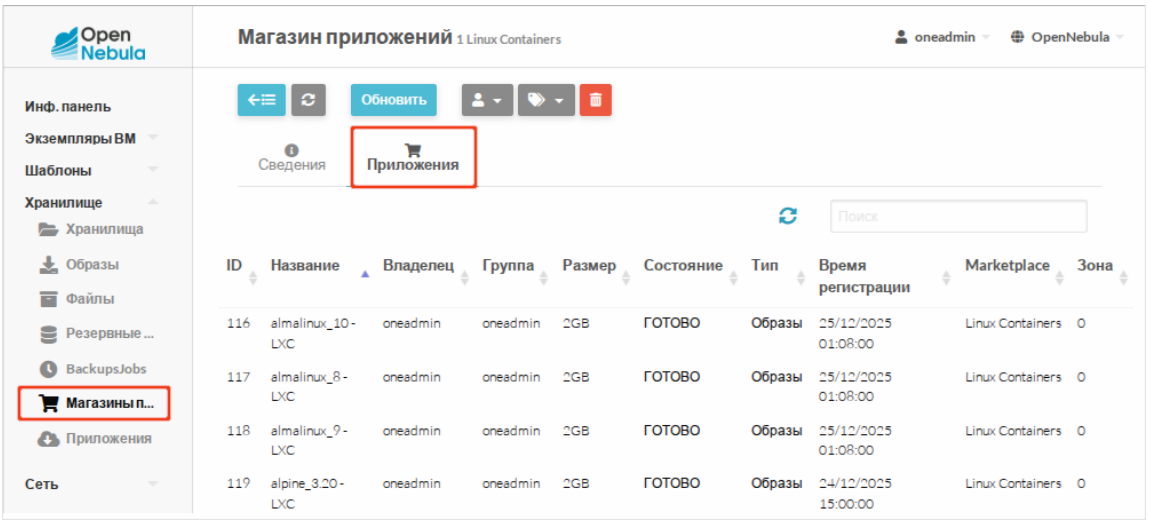
- около 6 ГБ свободного места в хранилище данных (плюс размер самого образа);
- настроенная поддержка LXC на узлах.

Примечание

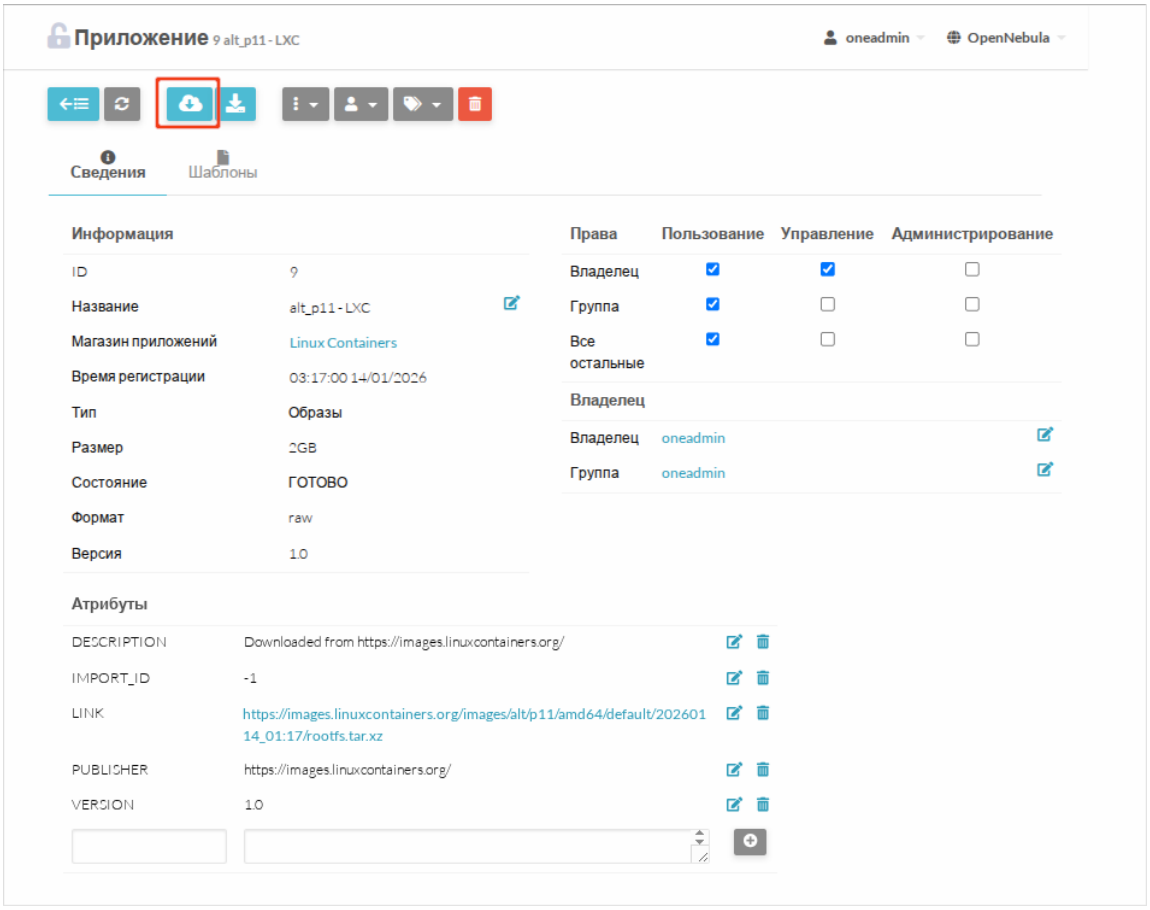
В процессе импорта в файловой системе контейнера создаётся файл **`/var/log/chroot.log`** содержащий журнал операций установки. В случае ошибок он может помочь в диагностике.

Импорт контейнера:

1. В веб-интерфейсе перейдите: **Хранилище** → **Магазины приложений** → **Linux Containers** → **Приложения**:

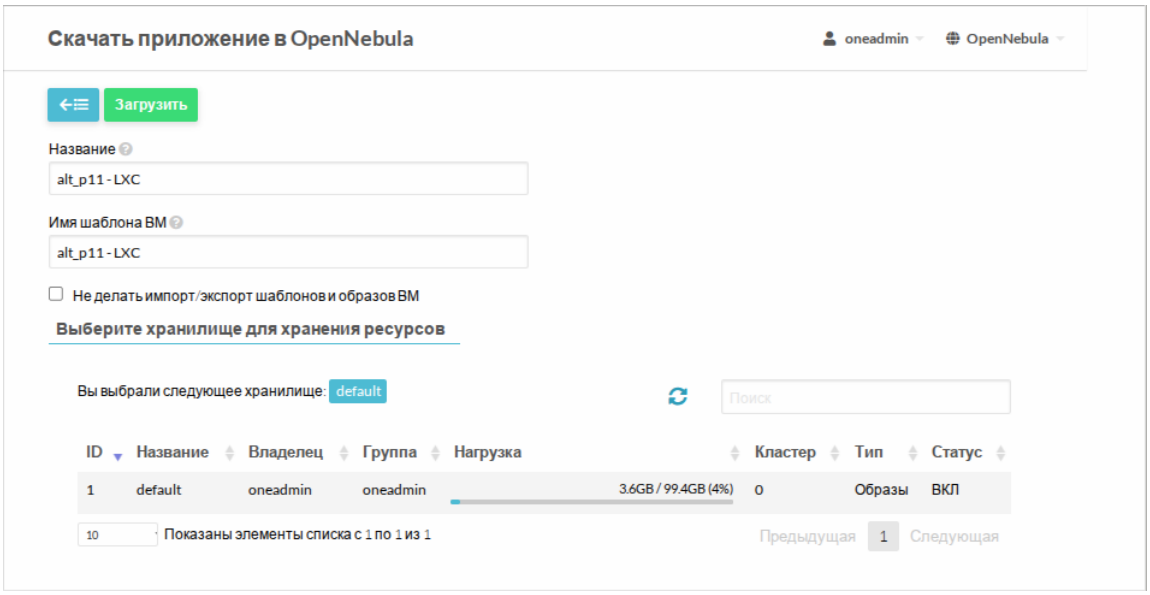


2. Выберите нужный LXC-образ и нажмите кнопку **Импорт в хранилище**:



3. В открывшемся окне:

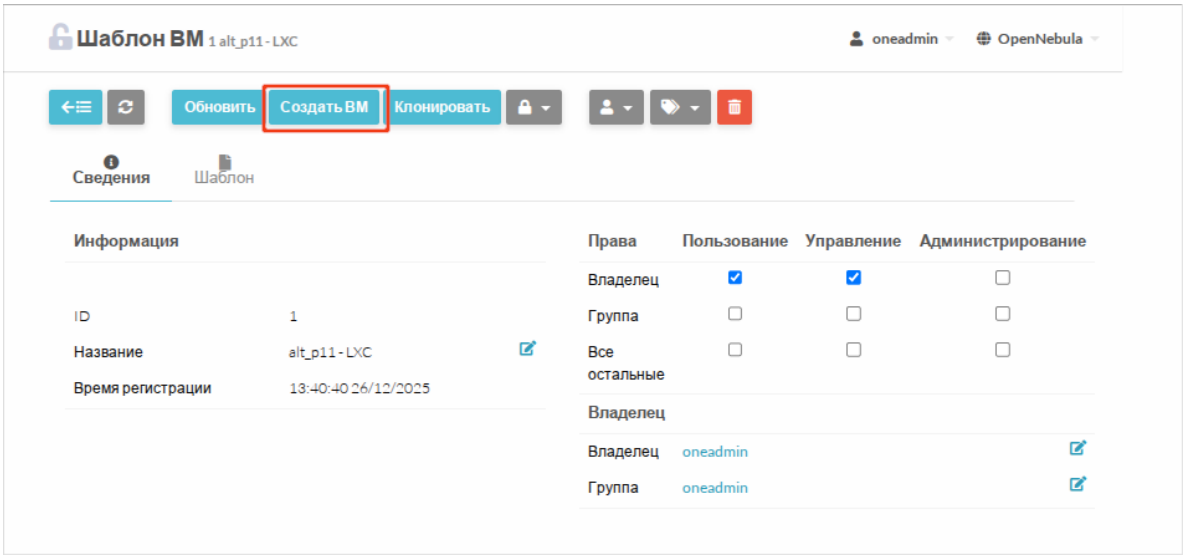
- укажите имя образа и шаблона;
- выберите целевое хранилище данных;
- нажмите кнопку **Загрузить**:



Каждый контейнер включает образ LXC и шаблон виртуальной машины, настроенный для запуска в режиме контейнера.

Развёртывание контейнера из шаблона выполняется точно так же, как и обычной ВМ:

- через интерфейс Sunstone:



- с помощью команды:

```
$ onevm create <шаблон>
```

Глава 31. Шаблоны ВМ

31.5. Управление шаблонами

В OpenNebula ВМ создаются на основе шаблонов ВМ — описаний конфигурации, которые можно многократно использовать для развёртывания экземпляров. Администраторы и пользователи могут регистрировать шаблоны в системе, делиться ими и настраивать под свои задачи.

Основные компоненты шаблона ВМ:

- объём оперативной памяти и количество CPU;
- сетевые интерфейсы, подключённые к виртуальным сетям;
- диски (образы ОС, данных, swap);
- дополнительные атрибуты: графика (VNC/SPICE), порядок загрузки, контекстуализация, запланированные действия и др.

Пример шаблона ВМ:

```
NAME   = "test-vm"
CPU    = 1
VCPU   = 1
MEMORY = 1024

DISK = [
IMAGE = "SL",
IMAGE_UNAME = "oneadmin"
]

DISK = [
TYPE = "swap",
SIZE = "1024"
]

NIC = [
NETWORK = "VirtNetwork",
NETWORK_UNAME = "oneadmin"
]

NIC = [
NETWORK = "Private",
NAME = "private_net"
]

GRAPHICS = [
TYPE = "vnc",
LISTEN = "0.0.0.0"
]

OS = [
BOOT = "disk0"
]
```

Примечание

Все строковые значения должны быть заключены в кавычки.

Простые шаблоны можно создать без использования файла, непосредственно из командной строки:

```
$ onetemplate create \
  --name test-vm \
  --cpu 1 \
  --vcpu 1 \
  --memory 1024 \
  --disk "SL" \
  --nic VirtNetwork
```

Полный список параметров для команды можно получить, выполнив:

```
$ onetemplate create -h
```

Рекомендации для шаблонов конечных пользователей:

- Образы, используемые в шаблонах для конечных пользователей, не должны быть постоянными (persistent).

  Постоянный образ может использоваться только одной ВМ одновременно, и следующий пользователь получит систему с изменениями предыдущего.

- Если пользователям требуется постоянное хранилище, рекомендуется использовать функцию создания постоянного экземпляра.

- Пользователи могут выбирать сетевые интерфейсы при запуске ВМ. Шаблоны могут содержать сетевые адаптеры по умолчанию, которые пользователь при необходимости может удалить или заменить.

- При добавлении сетевых интерфейсов рекомендуется указывать модель адаптера по умолчанию (например, virtio для KVM). Это можно сделать с помощью атрибута ***NIC_DEFAULT***, через мастер создания шаблонов или глобально — в конфигурации драйвера гипервизора.

31.1. Контекстуализация ВМ

Контекстуализация позволяет передавать данные ВМ при её запуске, включая:

- сетевые настройки (IP-адрес, шлюз, DNS);
- SSH-ключи;
- пользовательские переменные (например, пароли, названия сайтов);
- скрипты инициализации.

Данные указываются в разделе ***CONTEXT*** шаблона и обрабатываются агентом контекстуализации внутри гостевой ОС.

Пример контекстуализации:

```
CONTEXT = [
  NETWORK = "YES",
  SSH_PUBLIC_KEY = "$USER[SSH_PUBLIC_KEY]",
  BLOG_TITLE = "$BLOG_TITLE",
  MYSQL_PASSWORD = "$MYSQL_PASSWORD"
]

USER_INPUTS = [
  BLOG_TITLE = "M|text|Blog Title",
  MYSQL_PASSWORD = "M|password|MySQL Password"
]
```

Примечание

В гостевой ОС должен быть установлен пакет *opennebula-context*.

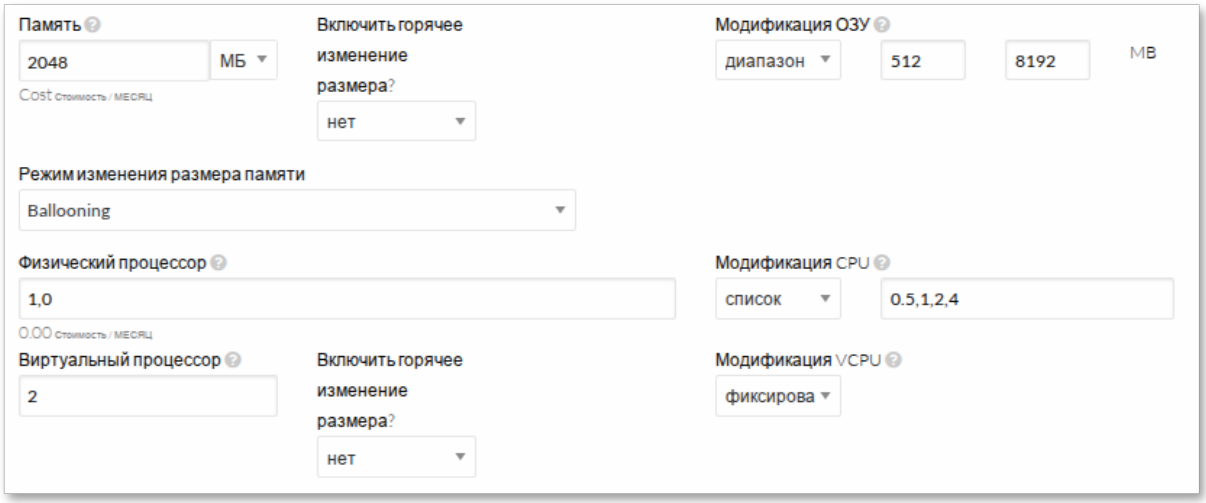
31.2. Подготовка шаблонов для конечных пользователей

Чтобы шаблон был удобен для совместного использования, его владелец может настроить параметризацию ресурсов и данных.

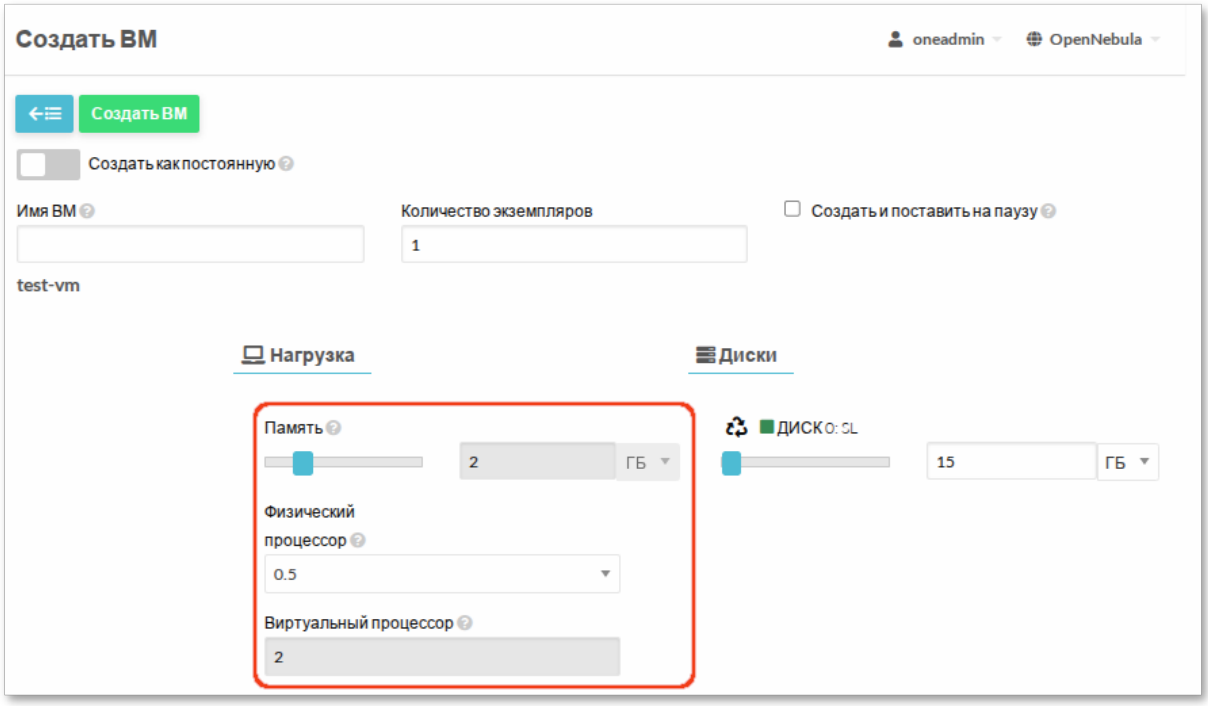
Атрибуты ***CPU***, ***MEMORY***, ***VCPU*** могут быть:

- фиксированными (fixed) — значение нельзя изменить;
- свободными (text) — произвольное числовое значение;
- ограниченными диапазоном (range) — значение из заданного диапазона (минимум–максимум);
- выбором из списка (list) — выбор одного значения из предопределённого списка.

Настройка параметризации ресурсов в Sunstone:



Выбор параметров при создании ВМ:



Если используется файл шаблона (а не Sunstone), параметризация определяется с помощью атрибутов ***USER_INPUTS***.

Отсутствие пользовательского ввода подразумевает возможность задания любого значения.

Пример синтаксиса ***USER_INPUTS***:

```
CPU    = "1"
MEMORY = "2048"
VCPU   = "2"

USER_INPUTS = [
  CPU    = "M|list|vCPU count|0.5,1,2,4|1",
  MEMORY = "M|range|Memory (MB)|512..8192|2048",
  VCPU   = "O|fixed|Virtual CPUs||2"
]
```

Формат поля ***USER_INPUTS***:

```
"<обязательность>|<тип>|<метка>|<значения>|<по умолчанию>"
```

где:

- M — обязательное поле, O — опциональное;
- тип: text, password, list, range, fixed.

Примечание

Для CPU допустимы дробные значения (например, 0.5, 1.0).

Для MEMORY и VCPU допускаются только целые числа.

Примечание

Для неинтерактивного использования пользовательских параметров можно применять опцию ***--user-inputs ui1,ui2,ui3***.

31.3. Запланированные действия

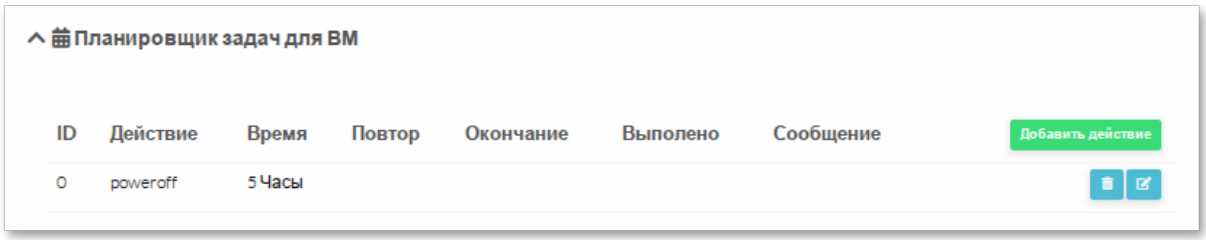
Для выполнения предопределённых операций над ВМ можно использовать запланированные действия.

Они позволяют автоматизировать управление ВМ, например, выключить ВМ через определённое время после запуска.

Пример запланированного действия

```
SCHED_ACTION = [
  ACTION = "poweroff",
  TIME   = "18000"   # через 5 часов (в секундах)
]
```

Запланированное действие также можно добавить при создании экземпляра ВМ через Sunstone:



31.4. Создание ВМ из шаблонов

В OpenNebula ВМ создаются на основе шаблонов с помощью команды:

```
$ onetemplate instantiate <ID_или_имя_шаблона>
```

Эта команда развёртывает один экземпляр ВМ. Для создания нескольких экземпляров используется опция ***--multiple***:

```
$ onetemplate instantiate my-template --multiple 3
VM IDs: 10, 11, 12
```

Примечание

Новые ВМ сначала находятся в состоянии pending (pend) до тех пор, пока планировщик не подберёт подходящий узел для размещения.

Пользователи могут частично изменять параметры шаблона при создании ВМ, если соответствующие атрибуты не ограничены в конфигурации OpenNebula.

Администратор может запретить изменение критических параметров, указав их в файле **`/etc/one/oned.conf`**, например:

```
VM_RESTRICTED_ATTR = "CPU"
VM_RESTRICTED_ATTR = "VPU"
VM_RESTRICTED_ATTR = "NIC"
```

Пример шаблона ВМ:

```
CPU = "1"
VCPU = "1"
MEMORY = "512"

DISK = [
IMAGE = "BaseOS" ]

NIC = [
NETWORK_ID = "0" ]
```

При указанных ограничениях пользователь может:

- изменить объём оперативной памяти (MEMORY);
- добавить новые диски.

Пользователь не может:

- изменить или удалить параметры CPU и VCPU;
- изменить или удалить существующие сетевые интерфейсы (NIC).

Пример корректного вызова:

```
$ onetemplate instantiate 0  \
  --memory 1G \
  --disk "SL"
```

Примечание

При переопределении дисков с помощью команды **`onetemplate instantiate`** все исходные диски шаблона заменяются. Чтобы сохранить оригинальный диск и добавить новый, необходимо указать все диски явно:

```
$ onetemplate instantiate 0  \
  --memory 1G \
  --disk "BaseOS, SL"
```

Планировщик OpenNebula автоматически развёртывает ВМ на одном из доступных узлов, если они соответствуют заданным требованиям.

Администратор может принудительно запустить развёртывание ВМ на конкретном узле с помощью команды:

```
$ onevm deploy <ID_ВМ> <ID_узла>
```

Задать владельца и группу ВМ можно при её создании (требуются права администратора):

```
$ onetemplate instantiate <Шаблон> \
  --as_uid <ID_пользователя> \
  --as_gid <ID_группы>
```

31.5. Управление шаблонами

Управление шаблонами ВМ осуществляется с помощью:

- командной строки (**onetemplate**);
- веб-интерфейса Sunstone (раздел **Шаблоны → ВМ**).

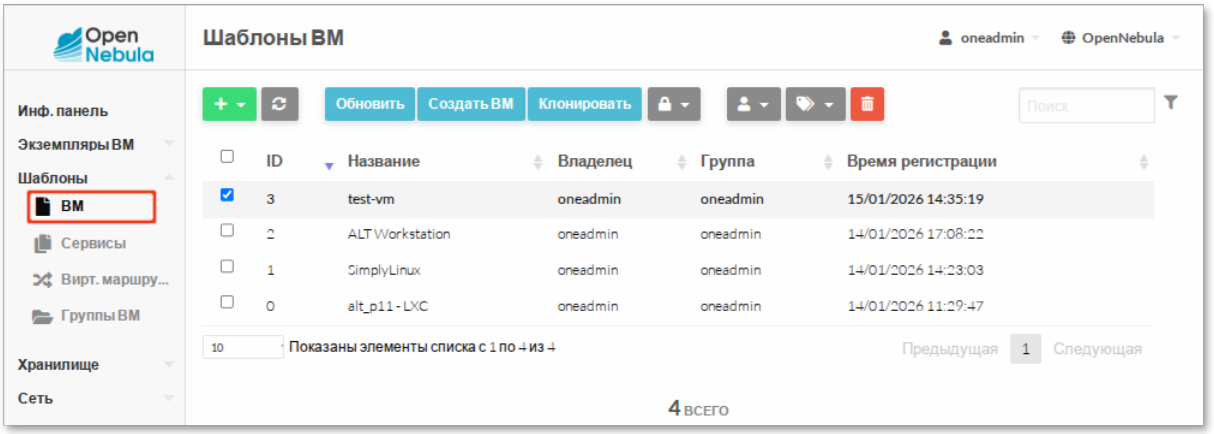


Таблица 31.1. Основные операции с шаблонами

Операция	Команда	Описание
Создать	**onetemplate create file.tmpl**	Регистрация нового шаблона
Клонировать	**onetemplate clone <ID_шаблона> new-name**	Создание копии шаблона
Обновить	**onetemplate update <ID_шаблона>**	Редактирование шаблона
Удалить	**onetemplate delete <ID_шаблона>**	Удаление (только владелец или админ)

Примечание

При клонировании с флагом ***--recursive*** OpenNebula создаёт копии всех образов, используемых в шаблоне, и делает их постоянными:

```
$ onetemplate clone 6 secure-copy --recursive
```

Атрибуты, запрещённые для переопределения при создании ВМ, также нельзя изменять при редактировании шаблона с помощью **`onetemplate update`**. Например, если в **`/etc/one/oned.conf`** указано:

```
VM_RESTRICTED_ATTR = "DISK/TOTAL_BYTES_SEC"
```

И шаблон содержит:

```
DISK = [
  IMAGE = "BaseOS",
  TOTAL_BYTES_SEC = "1048576"
]
DISK = [ IMAGE = "DataDisk" ]
```

Пользователь может удалить DataDisk, но не может удалить или изменить BaseOS, так как он содержит ограниченный атрибут.

Доступ к шаблонам регулируется системой разрешений OpenNebula (владелец, группа, другие).

Права на шаблоны управляются через восьмеричные коды (аналогично **`chmod`**).

Таблица 31.2. Управление правами доступа

Команда	Описание
`onetemplate chmod 0 640`	Группа может использовать шаблон
`onetemplate chmod 0 660`	Группа может использовать и управлять
`onetemplate chmod 0 664`	Группа — управление, остальные — использование

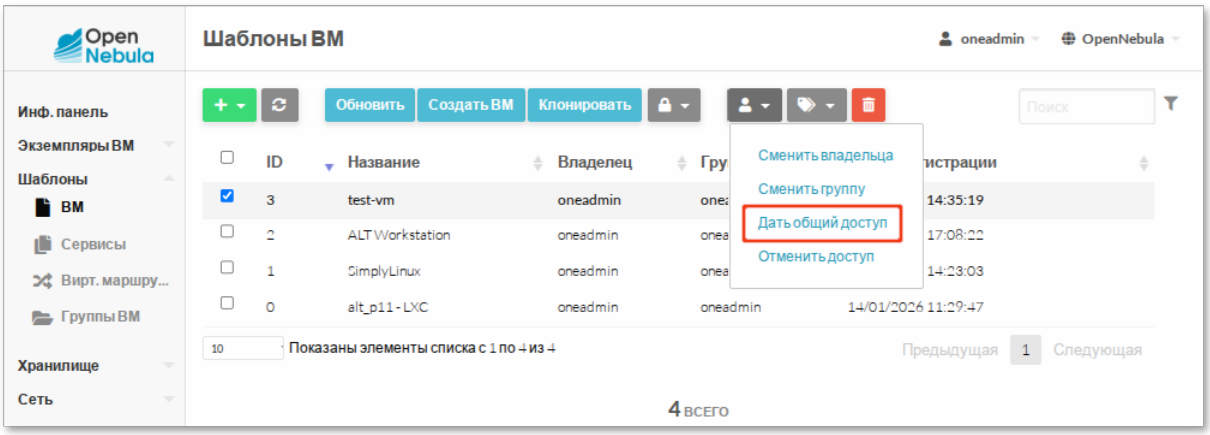
Пример:

```
$ onetemplate chmod 0 660
$ onetemplate show 0
...
PERMISSIONS
OWNER          : um-
GROUP          : um-
OTHER          : ---
```

Используйте опцию ***`--recursive`***, чтобы применить права также к всем образам, используемым в шаблоне:

```
$ onetemplate chmod 0 660 --recursive
```

В веб-интерфейсе Sunstone аналогом команды **`chmod --recursive 640`** является действие **`Дать общий доступ`**:



Глава 32. Виртуальные машины

32.1. Жизненный цикл ВМ

ВМ в OpenNebula проходит через последовательность состояний, отражающих текущий этап её жизненного цикла. Все состояния делятся на две категории:

- основное состояние (STATE) — например, ACTIVE, POWEROFF;

- состояние менеджера жизненного цикла (LCM_STATE) — например, PROLOG, RUNNING.

Примечание

В выводе **`onevm list`** отображается сокращённое имя состояния LCM_STATE (например, pend, runn).

Таблица 32.1. Состояния ВМ

Краткий псевдоним	Состояние	Описание
pend	Pending	ВМ ожидает развёртывания. Планировщик ищет подходящий узел
hold	Hold	Запуск ВМ приостановлен владельцем. ВМ может быть развёрнута вручную
clon	Cloning	ВМ ожидает завершения первоначального копирования одного или нескольких образов дисков в хранилище (образы находятся в заблокированном состоянии)
prol	Prolog	Копирование образов дисков на целевой узел
boot	Boot	Запуск гипервизора и загрузка ВМ
runn	Running	ВМ запущена и работает на узле
migr	Migrate	Выполняется миграция ВМ (live или cold)
hotp	Hotplug	Динамическое подключение или отключение дисков и сетевых интерфейсов
snap	Snapshot	Создается снимок ВМ
save	Save	Сохранение файлов ВМ при миграции, остановке или приостановке
epil	Epilog	Очистка ресурсов на узле после завершения работы ВМ
shut	Shutdown	Отправка ACPI-сигнала для корректного завершения работы гостевой ОС
stop	Stopped	ВМ остановлена, её состояние сохранено в системном хранилище
susp	Suspended	ВМ приостановлена; файлы остаются на узле для быстрого возобновления
poff	PowerOff	Аналогично состоянию Suspended, но файл контрольной точки не создаётся. Файлы ВМ остаются на узле. После корректного выключения гостевой ОС OpenNebula переводит ВМ в это состояние
unde	Undeployed	

Краткий псевдоним	Состояние	Описание
		ВМ выключена, её диски перенесены в системное хранилище. ВМ может быть возобновлена позже
drsz	Disk Resize	Выполняется изменение размера дисков ВМ
back	Backup	Выполняется резервное копирование ВМ
rest	Restore	Диски ВМ восстановлены из образа резервной копии
fail	Failed	Ошибка при развёртывании или выполнении операции
unkn	Unknown	ВМ недоступна и находится в неизвестном состоянии
clea	Cleanup-resubmit	ВМ ожидает завершения очистки узла драйверами после выполнения **`onevm recover --recreate`**
done	Done	ВМ завершена и удалена из списка активных, но сохранена в базе данных для учёта

32.2. Создание и вывод списка ВМ

Пример создания ВМ с именем my-vm из шаблона test-vm:

```
$ onetemplate list
  ID USER     GROUP    NAME
REGTIME
   3 oneadmin oneadmin test-vm                                    01/15
14:35:19
   2 oneadmin oneadmin ALT Workstation                            01/14
17:08:22
   1 oneadmin oneadmin SimplyLinux                                01/14
14:23:03
   0 oneadmin oneadmin alt_p11 - LXC                              01/14
11:29:47
$ onetemplate instantiate test-vm --name my-vm
VM ID: 10
```

Если в шаблоне определены параметры ***`USER_INPUTS`***, система запросит их значения:

```
$ onetemplate instantiate test-vm --name my-vm
There are some parameters that require user input. Use the string <<EDITOR>>
to launch an editor (e.g. for multi-line inputs)
  * (BLOG_TITLE) Blog Title
    April
  * (CPU)
    0  0.5
    1  1
    2  2
    3  4
```

```
    Please type the selection number: 1
  * (MEMORY)
    Press enter for default (2048). Integer in the range [512..8192]:
  * (MYSQL_PASSWORD) MySQL Password
    Password:
  * (VCPU)
    Fixed value of (2). Cannot be changed
VM ID: 11
```

Просмотр списка ВМ:

```
$ onevm list
  ID USER     GROUP    NAME          STAT  CPU     MEM HOST            TIME
  11 oneadmin oneadmin my-vm         prol    1      2G host-01         0d
00h00
```

Планировщик OpenNebula автоматически развернёт ВМ на одном из узлов с достаточным количеством доступных ресурсов.

Развёртывание может быть принудительно запущено пользователем oneadmin с помощью команды **`onevm deploy`**:

```
$ onehost list # найти подходящий узел
  ID NAME                CLUSTER    TVM      ALLOCATED_CPU      ALLOCATED_MEM
STAT
   2 host-03             default      0       0 / 200 (0%)     0K / 1.9G (0%)
on
   1 host-02             default      1       0 / 200 (0%)     0K / 1.9G (0%)
on
   0 host-01             default      1    100 / 200 (50%)    2G / 7.7G (25%)
on
$ onevm deploy 11 2 # развернуть ВМ 0 на узле 2
```

Получить подробную информацию о ВМ можно, выполнив команду:

```
$ onevm show 0
```

Вывод команды содержит:

- общее состояние ВМ;
- параметры ресурсов;
- шаблон ВМ;
- историю запусков и миграций.

32.3. Поиск ВМ

Команда **`onevm list --search`** позволяет фильтровать ВМ по атрибутам.

Синтаксис:

```
VM.<путь>=<значение>
```

- поддерживается логическое И с помощью символа &;
- поиск по всему шаблону: VM.TEMPLATE=<значение>;
- поиск по любому полю: *=<значение>.

Шаблоны поиска:

- % — любая последовательность символов (аналог *);
- _ — один произвольный символ;
- \%, _ — экранирование специальных символов.

Примеры:

- поиск по MAC-адресу:

```
$ onevm list --search 'VM.TEMPLATE.NIC[*].MAC=02:00:0c:00:3e:dd'
```

- поиск по имени и владельцу:

```
$ onevm list --search 'VM.NAME=test-vm&VM.UNAME=oneadmin'
```

- поиск по части имени:

```
$ onevm list --search 'VM.NAME=test%'
```

Примечание

Символ & нельзя экранировать — он всегда трактуется как логическое И.

Поиск выполняется по JSON-представлению ВМ, поэтому путь должен соответствовать структуре шаблона ВМ.

32.4. Основные операции с ВМ

Таблица 32.2. Завершение работы

Команда	Описание
onevm terminate <ID_ВМ>	Корректно завершает работу ВМ с помощью ACPI-сигнала. После остановки: - освобождаются ресурсы узла; - постоянные диски сохраняются; - снимки перемещаются в хранилище.

Команда	Описание
	Если ВМ не отвечает на ACPI в течение таймаута, она переводится в состояние POWEROFF
onevm terminate --hard <ID_ВМ>	Немедленно уничтожает ВМ без отправки ACPI-сигнала. Используется, если гостевая ОС не поддерживает корректное завершение работы или не отвечает

Примечание

Если пользователь выключает ВМ изнутри гостевой ОС (например, с помощью **shutdown -h now**), OpenNebula автоматически переводит её в состояние POWEROFF.

Таблица 32.3. Приостановка и выключение

Команда	Состояние	Описание
onevm suspend <ID_ВМ>	SUSPENDED	Сохраняет полное состояние ВМ (включая RAM) на узле. При возобновлении ВМ запускается на том же узле без перепланирования
onevm poweroff <ID_ВМ>	POWEROFF	Выключает ВМ через ACPI без сохранения состояния RAM. При запуске выполняется «холодная» загрузка на том же узле
onevm poweroff --hard <ID_ВМ>	POWEROFF	Принудительное выключение ВМ без отправки ACPI-сигнала

Таблица 32.4. Долгосрочная пауза (освобождение ресурсов узла)

Команда	Состояние	Описание
onevm undeploy <ID_ВМ>	UNDEPLOYED	Корректно выключает ВМ и переносит все её диски в системное хранилище. При возобновлении ВМ переходит в состояние PENDING, и планировщик выбирает новый узел
onevm undeploy --hard <ID_ВМ>	UNDEPLOYED	Принудительное выключение ВМ с немедленным переносом дисков в системное хранилище
onevm stop <ID_ВМ>	STOPPED	Аналогично undeploy, но дополнительно сохраняет состояние RAM для последующего восстановления

Возобновление работы:

```
$ onevm resume <ID_BM>
```

Поддерживаемые состояния: STOPPED, SUSPENDED, UNDEPLOYED, POWEROFF.

Таблица 32.5. Перезагрузка

Команда	Описание
onevm reboot <ID_BM>	Отправляет ACPI-сигнал перезагрузки гостевой ОС
onevm reboot --hard <ID_BM>	Принудительная перезагрузка (аналог жёсткого выключения и последующего запуска)

Таблица 32.6. Управление запуском

Команда	Описание
onevm instantiate --hold <шаблон> **onevm create --hold <файл>**	Создаёт ВМ в состоянии HOLD (планировщик игнорирует её до снятия удержания)
onevm hold <ID_BM>	Переводит существующую ВМ в состояние HOLD
onevm release <ID_BM>	Снимает удержание — ВМ переходит в состояние PENDING и передаётся под управление планировщика

32.5. Горячее подключение (Hotplug)

32.5.1. Диски

Подключение диска к работающей ВМ:

```
$ onevm disk-attach <BM> --image storage
```

Отключение диска:

```
$ onevm show <BM> # определить DISK_ID
$ onevm disk-detach <BM> <ID_диска>
```

Примечание

Требования:

- гостевая ОС должна поддерживать горячее подключение дисков (virtio-scsi, virtio-blk);
- образ должен быть непостоянным (non-persistent).

32.5.2. Сетевые интерфейсы

Подключение сетевого интерфейса:

```
$ onevm nic-attach <BM> --network <Сеть>
```

Отключение сетевого интерфейса:

```
$ onevm nic-detach <BM> <NIC_ID>
```

Поддерживается подключение (в том числе прямое) PCI- и SR-IOV-интерфейсов:

```
$ onevm nic-attach <BM> --network <Сеть> --pci '00:06.1'
```

Примечание

Ограничения:

- для предсказуемых PCI-адресов BM должна использовать шину PCIe (не PCI);
- SR-IOV требует поддержки со стороны оборудования и гипервизора.

32.5.3. Обновление параметров сети

Изменение параметров QoS в реальном времени:

```
$ cat update_nic.txt
NIC = [
INBOUND_AVG_BW = "512",
INBOUND_PEAK_BW = "1024"
]
$ onevm nic-update <BM> <NIC_ID> update_nic.txt
```

Примечание

Поддерживаемые параметры: INBOUND_AVG_BW, OUTBOUND_PEAK_BW, SCHEDULER, VIRT_MODE.

32.5.4. Группы безопасности (Security Groups)

Подключение группы безопасности:

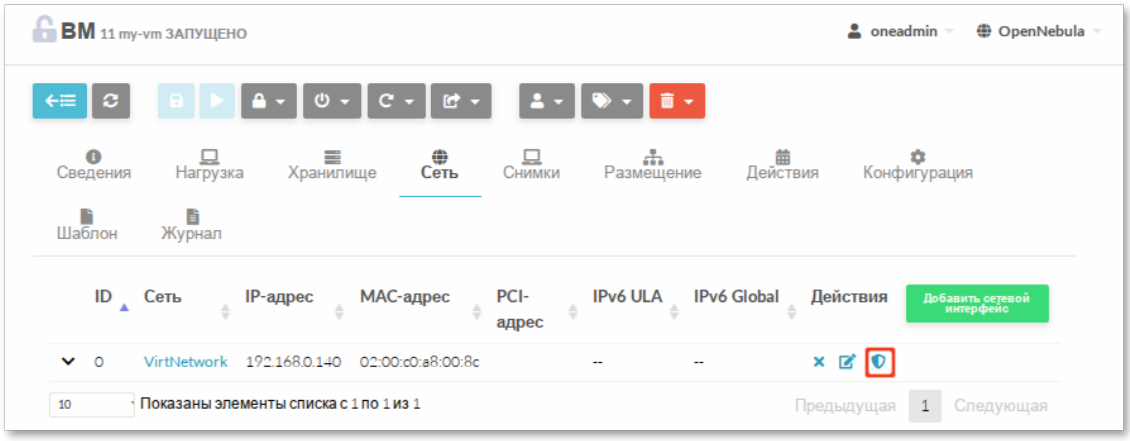
```
$ onevm sg-attach <BM> <NIC_ID> <SG_ID>
```

Отключение:

```
$ onevm sg-detach <BM> <NIC_ID> <SG_ID>
```

В Sunstone:

1. Откройте вкладку **Сеть** у выбранной ВМ.
2. Нажмите значок щита рядом с NIC для подключения группы безопасности:



3. Для отключения нажмите значок корзины.

Важно

Изменения применяются немедленно, в том числе для работающих ВМ.

32.5.5. Устройства PCI

Подключение PCI-устройств возможно только в состояниях POWEROFF или UNDEPLOYED:

```
$ onevm pci-attach <BM>  \
    --pci_class "0c03" \
    --pci_vendor "1912" \
    --pci_device "0015"
```

Отключение PCI-устройства:

```
$ onevm pci-detach <BM> <PCI_ID>
```

Поиск доступных PCI-устройств на узле:

```
$ onehost show <HOST_ID> | grep -A5 "PCI"
```

32.6. Снимки ВМ

Снимки позволяют сохранять состояние ВМ в определённый момент времени.

Снимок системы содержит текущее состояние дисков и, при live-snapshot, памяти ВМ. Можно создавать, удалять и восстанавливать моментальные снимки для запущенных ВМ:

```
$ onevm snapshot-create 17 "before domain"
$ onevm show 17
...
SNAPSHOTS
  ID         TIME NAME                                                 SNAPSHOT_ID
HYPERVISOR_ID
   0  01/19 18:49 before update                                              snap-0
   1  01/20 15:53 before domain

$ onevm snapshot-revert 17 0 --verbose
VM 17: reverting snapshot
```

Требования для snapshot-create:

- гипервизор: KVM;
- формат диска: qcow2;
- состояние ВМ: RUNNING (для live-snapshot) или POWEROFF/SUSPENDED (для offline-snapshot).

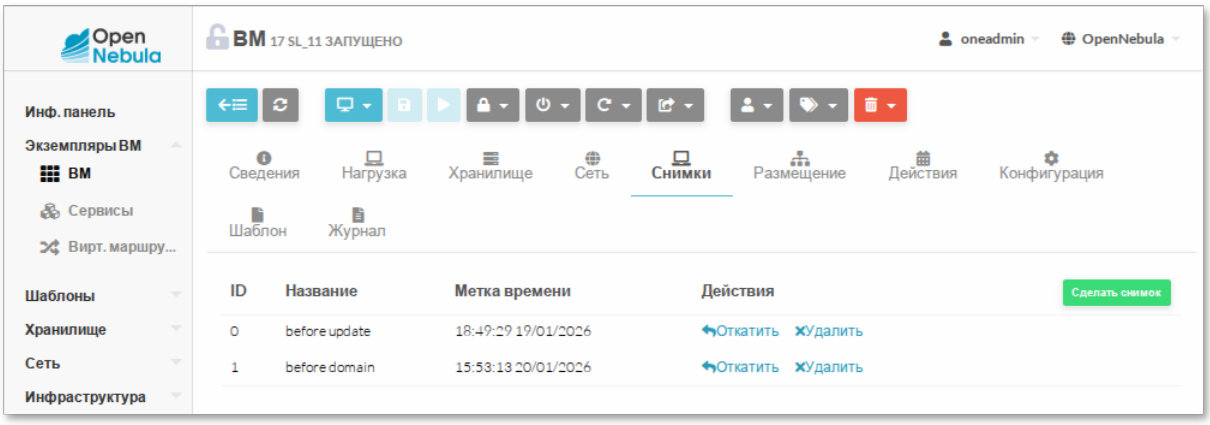
Примечание

Если для ВМ включены инкрементальные резервные копии (INCREMENTAL), то создание стандартных снимков (snapshots) для этой ВМ не поддерживается.

Удаление снимка выполняется командой:

```
$ onevm snapshot-delete 17
```

Управление снимками в веб-интерфейсе Sunstone доступно на вкладке **Снимки**:



32.7. Снимки дисков ВМ

OpenNebula поддерживает два типа операций со снимками дисков:

- **`onevm disk-snapshot-*`** — управление снимками: создание, восстановление, удаление, переименование;
- **`onevm disk-saveas`** — экспорт диска или снимка в новый образ.

Структура снимков

- **Древовидная (для qcow2)**

  Каждый снимок имеет родителя. Удалить можно только листовые снимки (без потомков).

- **Плоская (для Ceph)**

  Все снимки независимы и могут удаляться в произвольном порядке.

Команды управления:

- создать снимок диска:

```
$ onevm disk-snapshot-create <ID_ВМ> <ID_диска> <имя_снимка>
```

- восстановить снимок:

```
$ onevm disk-snapshot-revert <ID_ВМ> <ID_диска> <ID_снимка>
```

- удалить снимок:

```
$ onevm disk-snapshot-delete <ID_ВМ> <ID_диска> <ID_снимка>
```

Ограничения и требования:

- **`onevm disk-snapshot-revert`** требует состояния POWEROFF или SUSPENDED;
- Для live-snapshot (в состоянии RUNNING) необходимо:
  - KVM + qcow2 или хранилище с нативными снимками (Ceph);
  - включённый гостевой агент QEMU (***`guest_agent = "YES"`***) в шаблоне);
  - ***`DISK[CACHE] = "writethrough"`*** (рекомендуется для обеспечения согласованности данных).

Пример:

```
$ onevm disk-snapshot-create 17 0 newtest
$ onevm show 17
...
VM DISK SNAPSHOTS
AC  ID DISK PARENT            DATE SIZE         NAME
=>   0    0     -1  01/19 17:18:28 -/15G        test
     1    0     -1  01/19 29:19:26 -/15G        newtest
```

Примечание

Снимки постоянных образов сохраняются после удаления ВМ и могут быть использованы повторно или другими пользователями (при наличии прав доступа).

32.8. Изменение ресурсов ВМ

32.8.1. Изменение CPU и памяти

Поддерживаемые состояния: POWEROFF, UNDEPLOYED, STOPPED.

Для горячего изменения (в состоянии RUNNING) требуется:

- гипервизор: KVM или vCenter;
- атрибуты в шаблоне:

```
CPU_HOTPLUG = "YES"
MEMORY_HOTPLUG = "YES"
```

Рекомендуемая процедура увеличения ресурсов:

1. Корректно выключить ВМ.
2. Изменить параметры.
3. Возобновить работу ВМ.

Пример:

```
$ onevm poweroff web_vm
$ onevm resize web_vm --memory 2G --vcpu 2
$ onevm resume web_vm
```

Примечание

ВМ сохраняет все выделенные OpenNebula ресурсы (IP-адреса, сетевые настройки и т. д.).

Примечание

При горячем подключении vCPU новые ядра в гостевой ОС будут в состоянии offline. Их необходимо активировать вручную:

```
$ echo 1 > /sys/devices/system/cpu/cpu<ID>/online
```

32.8.2. Изменение размера дисков

Размер диска можно увеличить в любое время, включая состояние RUNNING.

Изменение размера диска при создании ВМ:

```
$ onetemplate instantiate <шаблон> --disk "alt:size=20G"
```

Размер образа alt будет изменен до 20 ГБ

Изменение размера диска существующей ВМ:

```
$ onevm disk-resize <ID_ВМ> <ID_диска> <новый_размер>
```

Требования:

- новый размер должен быть больше текущего;
- гостевая ОС должна поддерживать онлайн-расширение ФС (например, ext4, xfs);
- для корректной автоматической обработки требуется пакет *opennebula-context*.

32.9. Обновление конфигурации ВМ

Команда **`onevm updateconf`** позволяет изменить конфигурацию ВМ без её пересоздания.

Таблица 32.7. Поддерживаемые атрибуты

Атрибут	Податрибуты	Требуется перезагрузка
OS	ARCH, MACHINE, KERNEL, INITRD, BOOTLOADER, BOOT, KERNEL_CMD, ROOT, SD_DISK_BUS, UUID, FIRMWARE	Да
FEATURES	ACPI, PAE, APIC, LOCALTIME, HYPERV, GUEST_AGENT, VIRTIO_SCSI_QUEUES, VIRTIO_BLK_QUEUES, IOTHREADS	Нет
INPUT	TYPE, BUS	Нет
GRAPHICS	TYPE, LISTEN, PASSWD, KEYMAP, COMMAND	Нет
VIDEO	TYPE, IOMMU, ATS, VRAM, RESOLUTION	Нет
RAW	DATA, DATA_VMX, TYPE, VALIDATE	Да
CPU_MODEL	MODEL, FEATURES	Да
CONTEXT	Любые переменные (кроме ETH*)	Немедленно

Примечание

Особенности CONTEXT:

- изменения применяются немедленно, даже для работающих ВМ;
- при ошибке выполнения команду следует повторить вручную.

Пример обновления:

```
$ cat new_conf.txt
GRAPHICS = [ TYPE = "vnc", LISTEN = "0.0.0.0" ]
CONTEXT = [ BLOG_TITLE = "Updated Blog" ]

$ onevm updateconf <ID_BM> new_conf.txt
```

32.10. Клонирование ВМ

В OpenNebula можно создать постоянную копию ВМ двумя способами:

- запустить экземпляр из шаблона с флагом ***--persistent***;
- сохранить существующую ВМ с помощью команды **`onevm save`**.

В обоих случаях результатом является новый шаблон ВМ, содержащий копии всех дисков исходной виртуальной машины. Получившийся шаблон является приватным и доступен только владельцу (если права доступа не изменены вручную).

32.10.1. Создание постоянного экземпляра

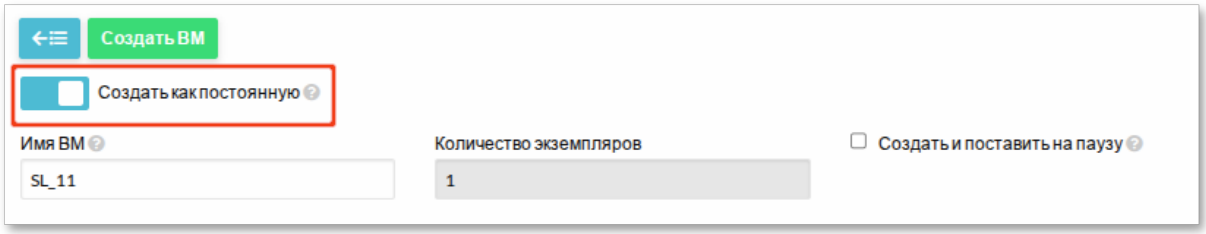
При использовании флага ***--persistent***:

- все диски клонируются как постоянные образы;
- создаётся новый шаблон ВМ на основе текущей конфигурации шаблона-источника.

Создание постоянного экземпляра:

```
$ onetemplate instantiate web_vm --persistent --name my_vm
VM ID: 30

$ onetemplate list # Появление нового шаблона
  ID USER            GROUP           NAME
REGTIME
   7 oneadmin        oneadmin        web_vm                       05/12
14:53:11
   8 oneadmin        oneadmin        my_vm                        05/12
14:53:38

$ onevm list # Созданные постоянные образы дисков
  ID USER       GROUP      NAME            DATASTORE     SIZE TYPE PER STAT
RVMS
   7 oneadmin   oneadmin   web-img         default       200M OS   Yes
used    1
   8 oneadmin   oneadmin   my_vm-disk-0    default       200M OS   Yes
used    1
```

При создании ВМ в веб-интерфейсе Sunstone активируйте переключатель **`Создать как постоянную`**:



Примечание

Ограничения и особенности:

- диски типа volatile (временные) не клонируются — в новом шаблоне они создаются как пустые;
- постоянные образы блокируются для одновременного использования другими ВМ.

32.10.2. Сохранение экземпляра ВМ

Команда **`onevm save`** сохраняет текущее состояние работающей или выключенной ВМ в новый шаблон.

Рекомендуемый порядок действий:

```
$ onevm poweroff web_vm          # сначала выключите ВМ
$ onevm save web_vm saved_web_vm --persistent
Template ID: 26
```

Что сохраняется:

- текущее содержимое дисков;
- фактические размеры ресурсов (CPU, MEMORY, VCPU);
- конфигурация сетевых интерфейсов.

Ограничения **`onevm save`**:

- volatile-диски не сохраняются;
- используется текущая версия исходного шаблона (если исходный шаблон был обновлён после запуска ВМ, изменения будут учтены);
- состояние RAM не сохраняется (ВМ будет загружаться заново).

32.11. Запланированные действия

Запланированные действия позволяют автоматизировать управление ВМ.

OpenNebula поддерживает следующие типы запланированных действий:

- точечные — выполняются один раз в заданное время;
- периодические — повторяются по расписанию;

- относительные — запускаются через N секунд после старта ВМ.

Большинство команд **onevm** поддерживают опцию ***--schedule***, позволяющую отложить выполнение действия до указанной даты и времени.

Примеры использования:

- выключить ВМ 11 в 00:00 20 сентября:

```
$ onevm suspend 11 --schedule "2026-09-20"
VM 11: suspend scheduled at 2026-09-20 00:00:00 +0200
```

- возобновить 23 сентября в 14:15:

```
$ onevm resume 11 --schedule "2026-09-23 14:15"
VM 11: resume scheduled at 2026-09-23 14:15:00 +0200
```

Просмотр запланированных действий:

```
$ onevm show 11
VIRTUAL MACHINE 11 INFORMATION
ID                  : 11
NAME                : my-vm

[...]

SCHEDULED ACTIONS
   ID  ACTION  ARGS   SCHEDULED REPEAT   END STATUS
    0 suspend     - 09/20 00:00            Next in 247.2 days
    1  resume     - 09/23 14:15            Next in 250.8 days
```

Управление расписанием:

- удалить действие:

```
$ onevm sched-delete <ID_BM> <ACTION_ID>
```

- обновить действие:

```
$ onevm sched-update <ID_BM> <ACTION_ID>
```

Атрибуты времени используют внутреннее Unix-время:

```
$ onevm sched-update 11 1
ID="1"
PARENT_ID="11"
TYPE="VM"
ACTION="resume"
TIME="1790165700"
REPEAT="-1"
END_TYPE="-1"
END_VALUE="-1"
DONE="-1"
WARNING="0"
```

Примечание

Атрибуты ID, PARENT_ID и TYPE являются системными атрибутами OpenNebula и не могут быть изменены вручную.

Таблица 32.8. Периодические действия

Опция	Описание	Пример
--hourly <N>	Каждые N часов	***--hourly 6***
--weekly <дни>	Дни недели (0=вс, 6=сб)	***--weekly "1,3,5"***
--monthly <дни>	Дни месяца (1-31)	***--monthly "1,15"***
--yearly <дни>	Дни года (0-365)	***--yearly "1,200"***
--end <N|дата>	Завершить после N повторов или к дате	***--end "2026-03-31"***

Еженедельный снимок по понедельникам и пятницам:

```
$ onevm snapshot-create 0 weekly_backup \
    --schedule "2025-09-22" \
    --weekly "1,5" \
    --end "2025-12-31"
```

Запланированные действия могут быть привязаны ко времени запуска ВМ. Они задаются в шаблоне и выполняются через указанное количество секунд после создания ВМ.

Пример шаблона ВМ с ***SCHED_ACTION***:

```
$ onetemplate update 0
SCHED_ACTION=[
   ACTION="terminate",
   ID="0",
   TIME="+3600" ]
```

В этом случае ВМ будет автоматически завершена через 1 час после запуска.

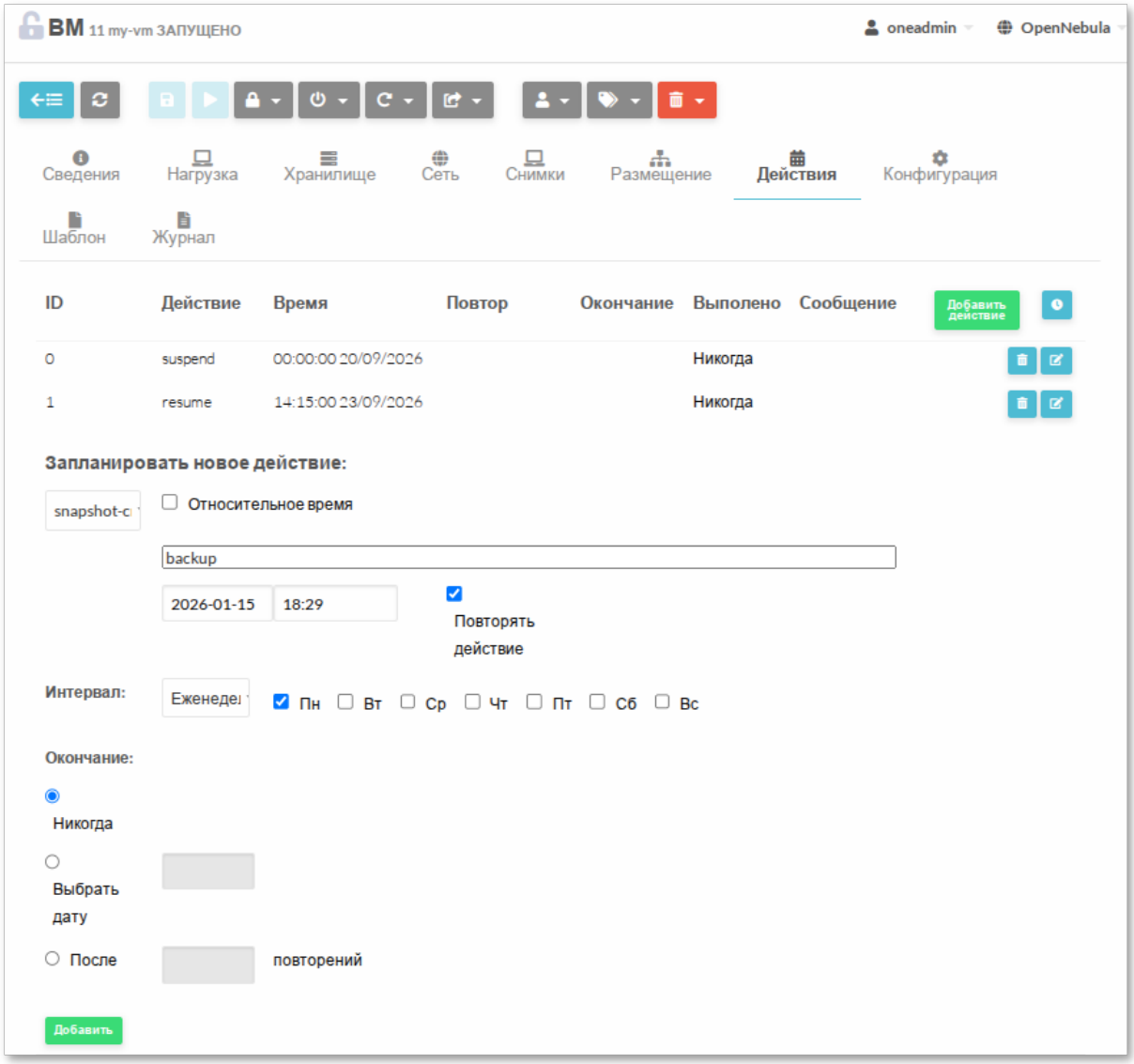
Таблица 32.9. Поддерживаемые запланированные действия

Действие	Аргументы	Пример
terminate	***[--hard]***	**terminate --hard**
undeploy	***[--hard]***	**undeploy**
hold / **release**	-	**hold**
stop / **suspend** / **resume**	-	**suspend**
reboot / **poweroff**	***[--hard]***	**reboot --hard**
snapshot-create	***<имя>***	**snapshot-create backup**
snapshot-revert	***<ID_снимка>***	**snapshot-revert 0**
disk-snapshot-create	***<ID_диска>,<имя>***	**disk-snapshot-create 0,daily**
backup	***<ID_хранилища>,[reset]***	**backup 100 reset**

Важно

Аргументы обязательны для действий, требующих параметров (например, **`snapshot-create`**).

Функциональность запланированных действий также доступна в Sunstone в мастере создания и обновления шаблонов ВМ на вкладке **Действия**. Запланированное действие также можно добавить при создании или редактировании ВМ через веб-интерфейс.



32.12. Пользовательские данные

Владелец ВМ может добавлять пользовательские метаданные с помощью атрибутов в разделе ***`USER TEMPLATE`***. Это удобно для хранения служебной информации, например: паролей, тегов, идентификаторов интеграций и других параметров, используемых внешними системами.

Пример:

```
$ onevm update 11
# В открывшемся редакторе добавьте:
ROOT_PASSWORD="s3cr3t"
```

Просмотр добавленных данных:

```
$ onevm show 0
...
USER TEMPLATE
BLOG_TITLE="April"
ROOT_PASSWORD="s3cr3t"
```

Примечание

Пользовательские данные не шифруются и доступны пользователям, имеющим права просмотра ВМ. Не рекомендуется хранить в них чувствительные данные без дополнительных мер защиты.

32.13. Мониторинг ВМ

OpenNebula собирает метрики использования ресурсов ВМ с помощью агентов мониторинга, работающих на узлах. Данные обновляются каждые 5–10 секунд и доступны через команду **`onevm show`**, а также в веб-интерфейсе Sunstone.

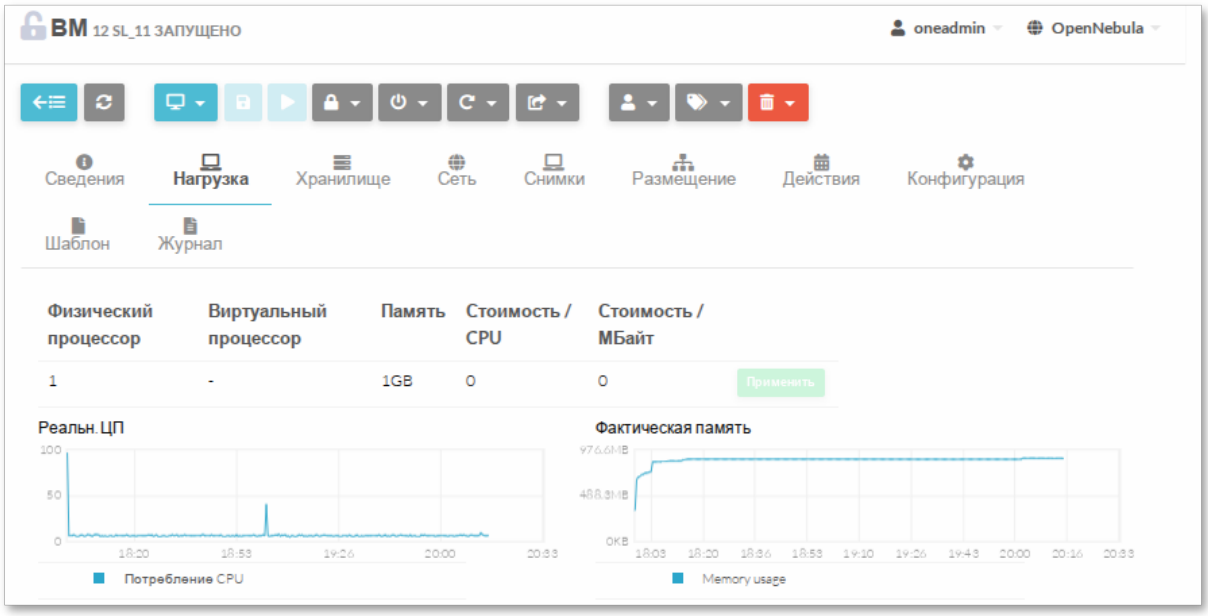


Таблица 32.10. Метрики мониторинга

Ключ	Описание	Единицы
CPU	Загрузка CPU	% (1 ядро = 1.0)
MEMORY	Используемая память	КБ
DISKRDBYTES	Байты, прочитанные с диска	байты
DISKRDIOPS	Количество операций чтения ввода-вывода	IOPS

Ключ	Описание	Единицы
DISKWRBYTES	Байты, записанные на диск	байты
DISKWRIOPS	Количество операций записи ввода-вывода	IOPS
NETRX	Полученные сетевые байты	байты
NETTX	Отправленные сетевые байты	байты

32.14. Разрешения ВМ

Каждая ВМ имеет набор прав доступа, управляемых владельцем. Права определяют, какие действия могут выполнять владелец, группа и остальные пользователи.

Таблица 32.11. Права доступа

Право	Обозначение	Действие
u	USE	Просмотр информации, подключение к консоли
m	MANAGE	Изменение состояния (остановка, перезагрузка)
a	ADMIN	Удаление ВМ, изменение владельца

Пример настройки прав:

```
$ onevm show 0
...
PERMISSIONS
OWNER          : um-
GROUP          : ---
OTHER          : ---
$ onevm chmod 0 640
$ onevm show 0
...
PERMISSIONS
OWNER          : um-
GROUP          : u--
OTHER          : ---
```

Администраторы могут изменять владельца и группу ВМ:

```
$ onevm chown <ID_BM> <пользователь>
$ onevm chgrp <ID_BM> <группа>
```

32.15. Административные операции

Таблица 32.12. Перепланирование и миграция

Команда	Описание
onevm resched <BM>	Перенос ВМ на оптимальный узел (в соответствии с политиками планировщика)

Команда	Описание
onevm migrate <ВМ> <Узел>	Холодная миграция (с остановкой ВМ)
onevm migrate --live <ВМ> <Узел>	Горячая миграция (без простоя)

Требования для live-migration:

- общее хранилище (NFS, Ceph);
- идентичные или совместимые CPU на узлах.

Если ВМ зависла в нестандартном состоянии или операция загрузки не завершается, можно использовать следующие параметры (**onevm recover**):

- ***--success*** — имитирует успешное выполнение действия при отсутствии драйвера;
- ***--failure*** — имитирует сбой действия при отсутствии драйвера;
- ***--retry*** — повторяет попытку выполнения текущего действия с драйвером (при необходимости можно использовать ***--interactive***, если проблема связана с Transfer Manager);
- ***--delete*** — удаляет ВМ, немедленно переводя её в состояние DONE;
- ***--recreate*** — удаляет ВМ и переводит её в состояние PENDING.

Команды migrate или resched могут использоваться для ВМ в состоянии UNKNOWN В этом случае ВМ может быть запущена на другом узле вручную или автоматически планировщиком. Такие операции допустимы только при использовании общего хранилища или если данные были перенесены администратором вручную. OpenNebula не выполняет никаких операций с хранилищем при такой миграции.

Примеры:

- принудительное удаление зависшей ВМ:

```
$ onevm recover <ID_BM> --delete
```

- повтор попытки развёртывания:

```
$ onevm recover <ID_BM> --retry
```

- пересоздание ВМ в состоянии PENDING:

```
$ onevm recover <ID_BM> --recreate
```

Важно

Используйте ***--delete*** только в том случае, если ВМ фактически отсутствует на узле.

32.16. Доступ к консоли ВМ

Sunstone поддерживает несколько методов доступа: VNC, SPICE, RDP и SSH.

Выбор метода зависит от типа ВМ, гостевой операционной системы и требований к производительности и безопасности.

32.16.1. VNC

VNC — универсальная графическая консоль, поддерживаемая большинством гипервизоров и клиентов.

32.16.1.1. Настройка ВМ

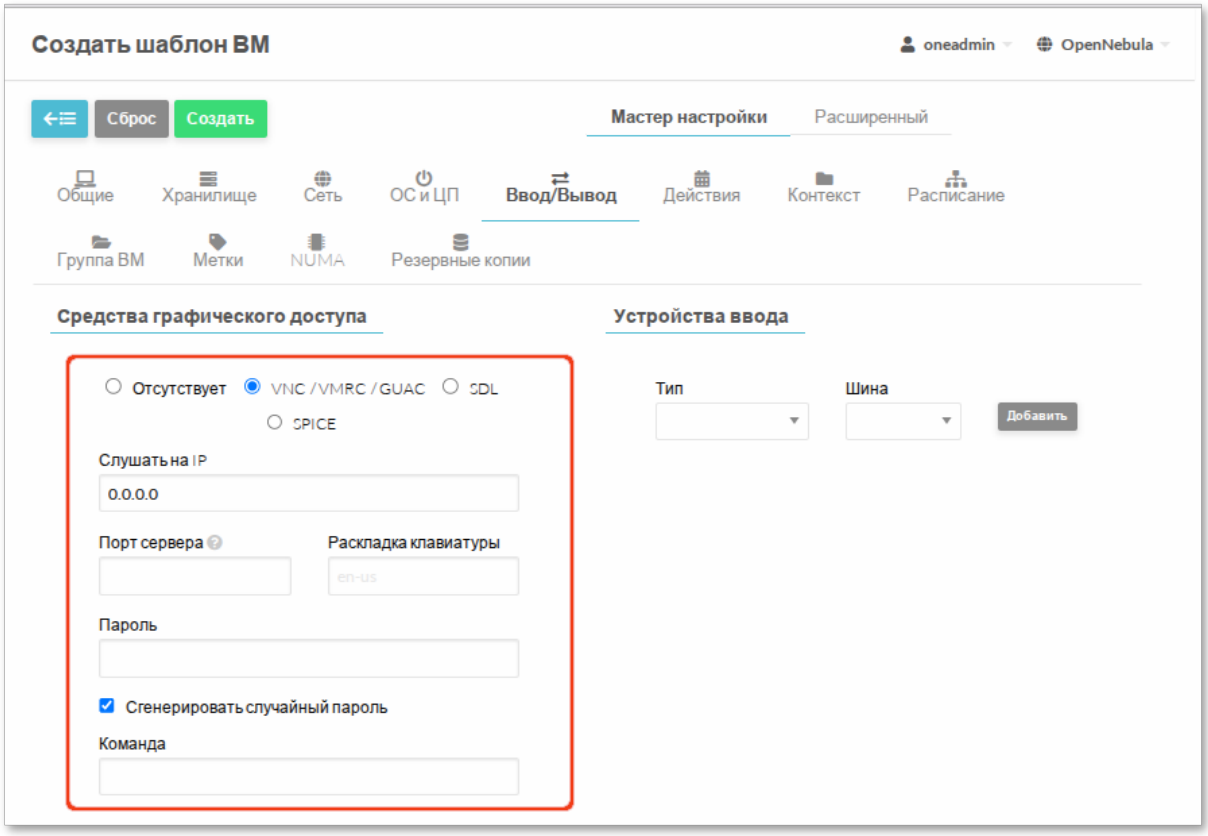
Для включения VNC необходимо добавить раздел ***GRAPHICS*** в шаблон ВМ:

```
GRAPHICS=[
LISTEN="0.0.0.0",  # Разрешить внешние подключения
TYPE="vnc"
]
```

Рекомендации по безопасности:

- всегда задавайте пароль (***PASSWD = "strong_password"***);
- ограничьте LISTEN IP-адресом узла управления, если внешний доступ не требуется.

В Sunstone настройка выполняется в шаблоне ВМ на вкладке **Ввод/вывод**:

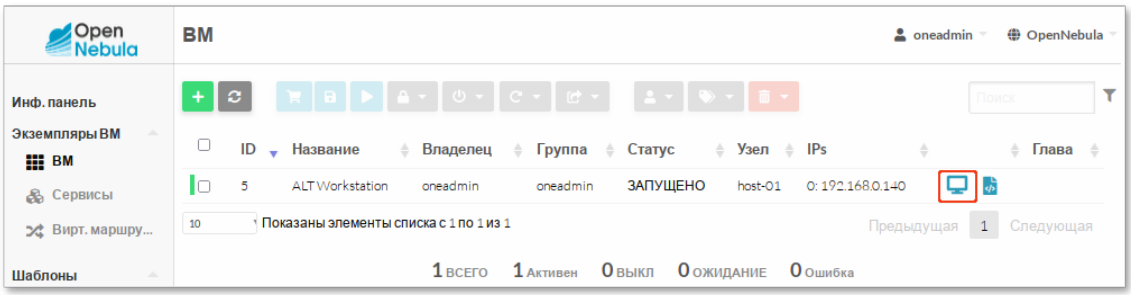


32.16.1.2. Подключение к ВМ

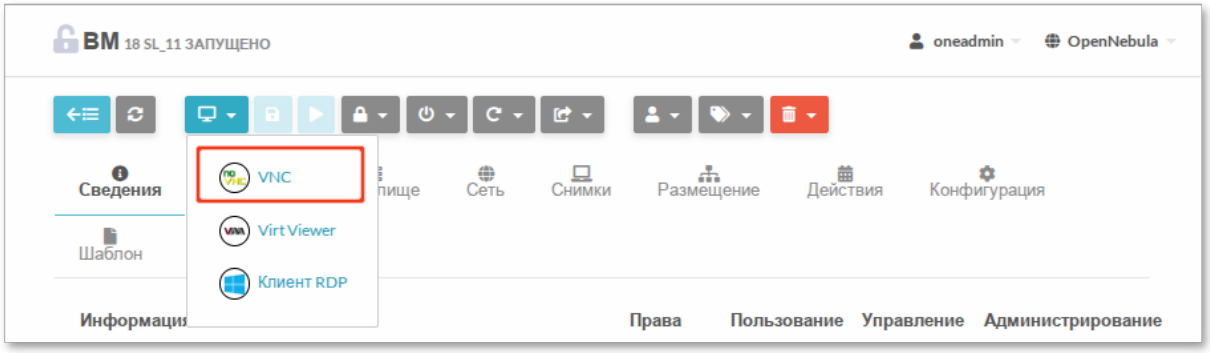
В веб-интерфейсе Sunstone кнопка VNC появляется автоматически при корректной настройке шаблона.

Подключение через Sunstone:

1. Перейдите в раздел **Экземпляры ВМ → ВМ**.
2. Выберите ВМ и подключитесь к ней через VNC:



Способ подключения также можно выбрать в окне настроек ВМ:



Для подключения через внешний клиент VNC

1. Нажмите кнопку **`VirtViewer`**, чтобы скачать файл **`.vv`**.
2. Откройте его в клиенте virt-viewer или Remote Viewer.

32.16.2. SPICE

SPICE — улучшенная графическая консоль, ориентированная на высокую производительность и удобную работу с графическим интерфейсом (буфер обмена, динамическое изменение разрешения, поддержка USB).

32.16.2.1. Настройка ВМ

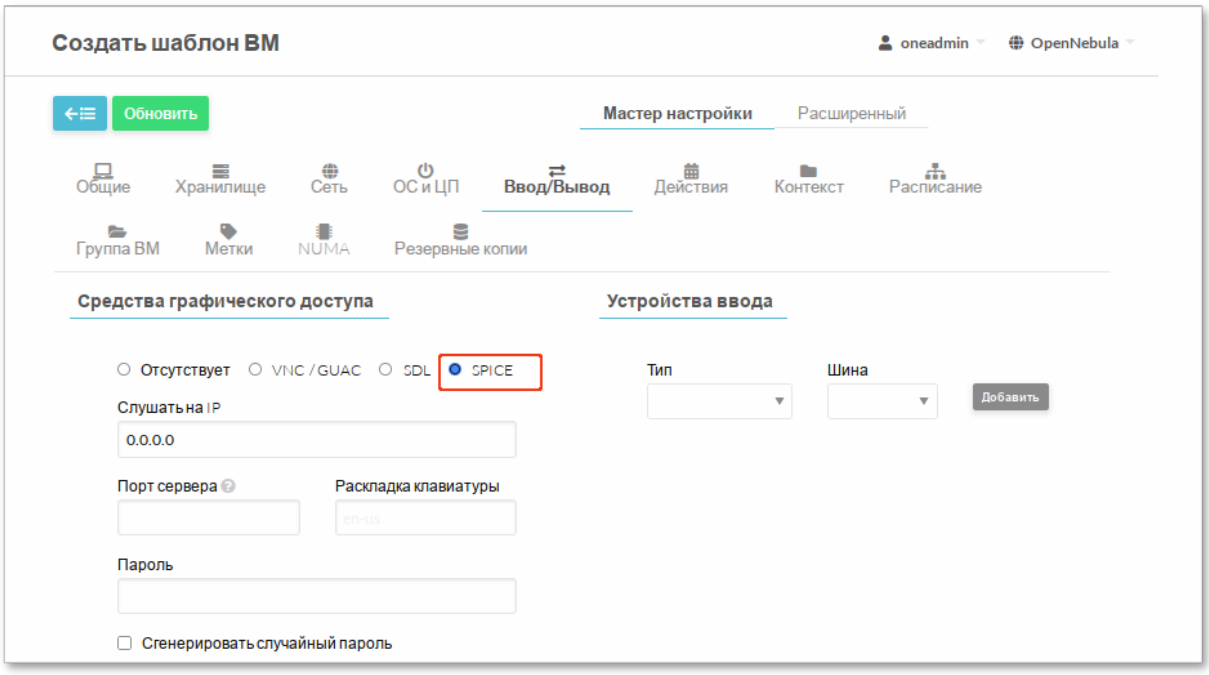
Для включения SPICE необходимо добавить раздел ***`GRAPHICS`*** в шаблон ВМ:

```
GRAPHICS=[
LISTEN="0.0.0.0",  # Разрешить внешние подключения
TYPE="spice"
]
```

Рекомендации по безопасности:

- всегда задавайте пароль (***`PASSWD = "strong_password"`***);
- ограничьте LISTEN IP-адресом узла управления, если внешний доступ не требуется.

В Sunstone настройка выполняется в шаблоне ВМ на вкладке **Ввод/вывод**:

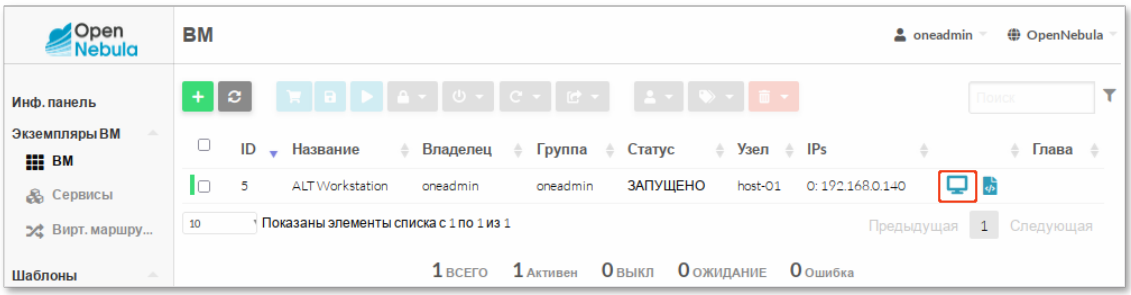


32.16.2.2. Подключение к ВМ

В веб-интерфейсе Sunstone кнопка SPICE появляется автоматически при корректной настройке шаблона.

Подключение через Sunstone:

1. Перейдите в раздел **Экземпляры ВМ → ВМ**.
2. Выберите ВМ и подключитесь к ней через SPICE:



Для подключения через внешний клиент SPICE нужно указать:

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

где:

- 192.168.0.180 — IP-адрес узла с ВМ;
- 5 — идентификатор ВМ (номер порта равен 5900 + ID ВМ).

32.16.3. Настройка ВМ для RDP

RDP (Remote Desktop Protocol) — протокол удалённого рабочего стола, используемый в Windows и некоторых Linux-системах.

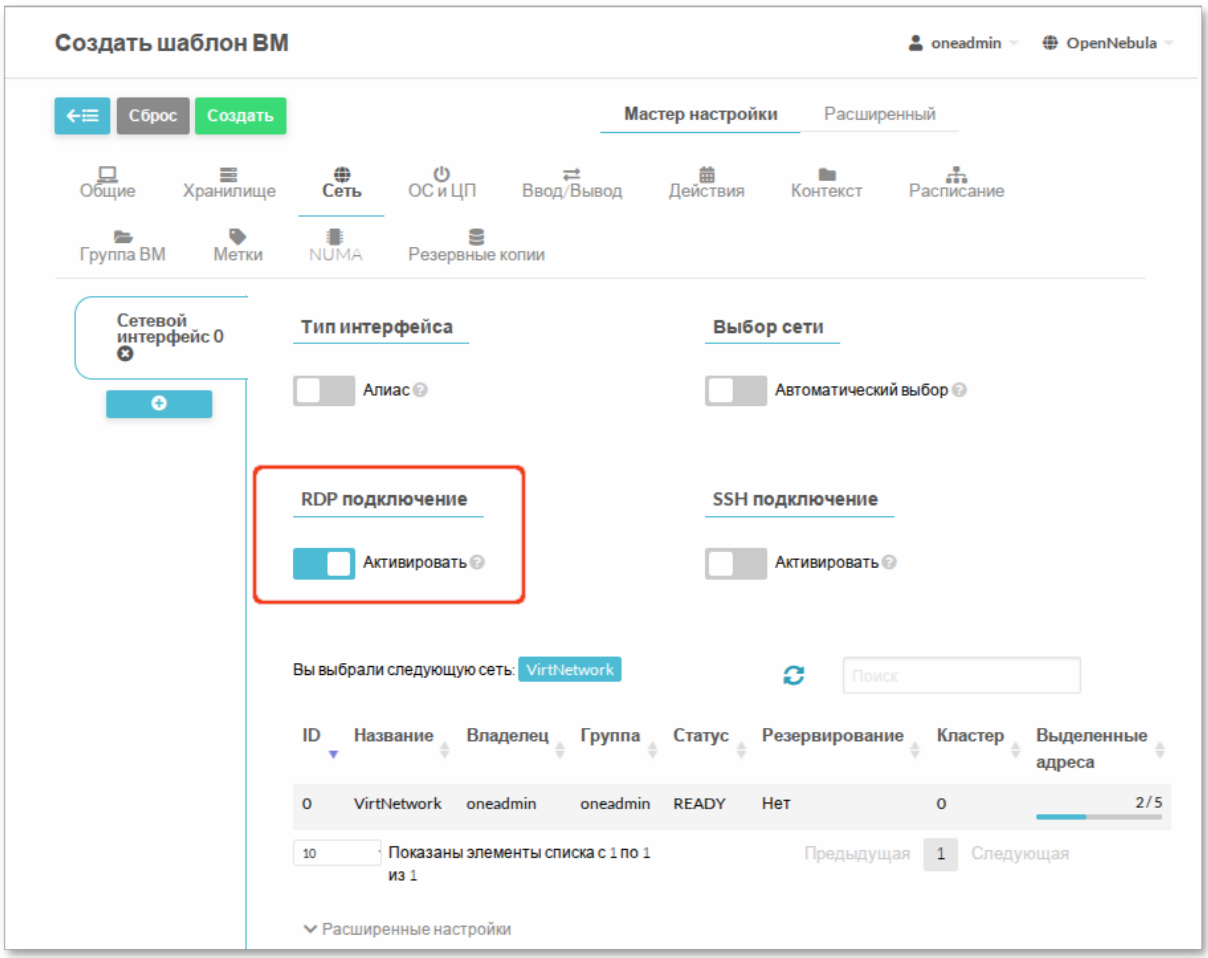
Важно

OpenNebula не управляет RDP-сервером внутри гостевой ОС. Он лишь предоставляет удобную ссылку на подключение.

Для отображения кнопки RDP в Sunstone в шаблоне ВМ или NIC необходимо указать:

```
NIC = [
  NETWORK = "VirtNetwork",
  RDP     = "YES"
]
```

Настройка также может быть выполнена в Sunstone:



Условия работы RDP:

- в гостевой ОС должен быть установлен и запущен RDP-сервер;
- ВМ должна иметь сетевую доступность с сервера управления Sunstone.

Особенности:

- Sunstone не проксирует RDP, а формирует ссылку подключения;
- для подключения используется внешний RDP-клиент (FreeRDP, Remmina и др.);

- протокол работает только после загрузки ОС и запуска сетевых служб.

Примечание

RDP-соединение может быть активировано только на одном сетевом адаптере. Соединение будет содержать только IP-адрес первого сетевого адаптера с включенным этим свойством.

Глава 33. Резервные копии

OpenNebula поддерживает два подхода к резервному копированию:

- резервное копирование отдельной ВМ — управление копиями для конкретной ВМ (подходит для небольшого числа ВМ);
- задания резервного копирования (Backup Jobs) — централизованное управление копиями для группы ВМ.

Поддерживаемые типы резервных копий:

- полная (Full) — каждая копия содержит полный образ всех дисков ВМ;
- инкрементальная (Incremental) — копируются только изменения с момента последней копии. Отслеживание изменений выполняется с использованием контрольных точек (dirty bitmap в QEMU).

Примечание

Для возможности создания инкрементальных резервных копий на узлах виртуализации необходимо установить пакет *libnbd*:

```
# apt-get install libnbd
```

Для дисков формата qcow2 инкрементальное копирование может работать в двух режимах, задаваемых параметром ***INCREMENT_MODE***:

- CBT (по умолчанию) — отслеживание изменений с помощью bitmap внутри того же файла диска;
- SNAPSHOT — создание отдельного снимка диска, в котором накапливаются изменения.

Особенности для Ceph (RBD):

- Полные копии: экспорт RBD → преобразование в qcow2 → хранение. Восстановление выполняется в обратном порядке: qcow2 → RAW → импорт в пул Ceph.
- Инкрементальные копии: данные хранятся в нативном формате Ceph (rbd export --export-format 2). Восстановление выполняется путём импорта базового образа с последовательным применением diff-файлов.

Копирование может выполняться:

- в горячем режиме (live-backup, для KVM);
- при выключенной ВМ.

Процесс резервного копирования состоит из трёх этапов:

1. Подготовка (Pre-backup):
   - файловые системы гостевой ОС замораживаются (через QEMU Guest Agent либо путём приостановки ВМ);
   - создаются временные структуры, позволяющие ВМ продолжить работу;
   - все диски копируются одновременно для обеспечения согласованности данных (crash-consistent).
2. Копирование (Backup):
   - полные образы или инкременты передаются в хранилище резервных копий;
   - используются драйверы целевого хранилища (например, restic).
3. Завершение (Post-backup):
   - временные файлы и вспомогательные структуры на узле гипервизора удаляются.

Примечание

Все диски формата RAW автоматически преобразуются в формат qcow2 для экономии дискового пространства.

Ограничения:

- инкрементальное копирование доступно только для гипервизора KVM и дисков формата qcow2 или RBD;
- горячее копирование поддерживается только для KVM;

- при подключении нового диска к ВМ с существующей инкрементальной цепочкой возникнет ошибка. Решение — использовать флаг ***--reset*** для создания новой цепочки резервных копий;
- инкрементальное копирование не поддерживается, если у ВМ существуют снимки дисков или системные снимки.

33.1. Хранилище резервных копий: Restic

Restic — инструмент резервного копирования с открытым исходным кодом, ориентированный на высокую скорость, безопасность и эффективность. В OpenNebula драйвер Restic использует протокол SFTP для передачи и хранения данных.

Ключевые возможности Restic:

- дедупликация — передаются только те блоки данных, которых ещё нет в репозитории;
- сжатие — включено по умолчанию;
- шифрование — все данные шифруются с использованием пароля репозитория.

Терминология:

- репозиторий — хранилище, в котором размещаются резервные копии. Restic создаёт собственную структуру каталогов и работает с ней через SFTP. Для каждой ВМ или задания резервного копирования создаётся отдельный репозиторий;
- снимок (Snapshot) — отдельная резервная копия, идентифицируемая уникальным хешем (например, bda78f31). Снимок включает все диски ВМ и её метаданные на момент создания;
- сервер резервного копирования — выделенный узел, на котором размещаются репозитории Restic и резервные копии ВМ.

33.1.1. Настройка сервера резервного копирования

Настройка сервера резервного копирования:

1. Подготовьте выделенный сервер с достаточным объёмом дискового пространства (например, отдельный iSCSI- или LVM-том).
2. Создайте учётную запись oneadmin:

```
# useradd -m -u 9869 oneadmin
# passwd oneadmin
```

3. Настройте беспарольный SSH-доступ:
   - скопируйте открытый SSH-ключ (**~oneadmin/.ssh/id_rsa.pub**) с узла управления OpenNebula и узлов виртуализации в файл **~oneadmin/.ssh/authorized_keys** на сервере резервного копирования:

```
$ ssh-copy-id -i /var/lib/one/.ssh/id_rsa.pub
<сервер_резервного_копирования>
```

- убедитесь, что подключение с узла управления OpenNebula и узлов виртуализации выполняется без запроса пароля:

```
$ ssh <сервер_резервного_копирования>
```

4. Создайте каталог для хранения репозиториев и назначьте владельца:

```
# mkdir -p /var/lib/one/datastores
# chown oneadmin:oneadmin /var/lib/one/datastores
```

5. Смонтируйте том хранения в каталог **/var/lib/one/datastores**, например:

```
# mount /dev/vgBackup/lvBackup /var/lib/one/datastores
```

Примечание

В примере используется LVM-том vgBackup/lvBackup, смонтированный в **/var/lib/one/datastores**.

6. Установите необходимые пакеты:

```
# apt-get install rsync qemu-img
```

33.1.2. Создание хранилища резервных копий в OpenNebula

Примечание

На узле управления должен быть установлен пакет *restic*:

```
# apt-get install restic
# cp /usr/bin/restic /var/lib/one/remotes/datastore/restic/restic
```

После установки необходимо синхронизировать файлы со всеми узлами кластера:

```
$ onehost sync -f
```

На узлах виртуализации должен появиться файл **/var/tmp/one/datastore/restic/restic**.

Для создания хранилища резервных копий с использованием конфигурационного файла:

1. Создайте шаблон хранилища **ds_restic.conf**:

```
NAME                = "RBackups"
TYPE                = "BACKUP_DS"
DS_MAD              = "restic"
TM_MAD              = "-"
RESTIC_SFTP_USER    = "oneadmin"
RESTIC_PASSWORD     = "opennebula"
RESTIC_SFTP_SERVER  = "192.168.0.197"
```

2. Зарегистрируйте хранилище:

```
$ onedatastore create ds_restic.conf
ID: 100
```

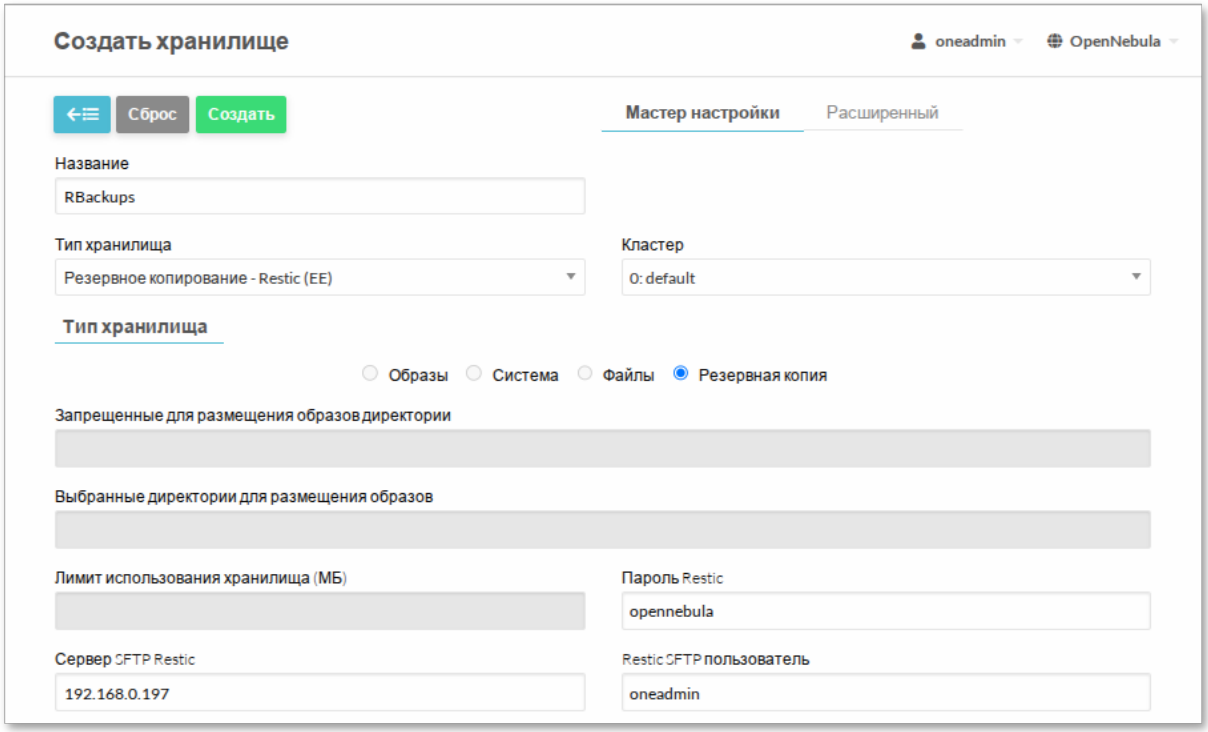
Для создания хранилища резервных копий Restic необходимо указать следующие параметры:

- ***NAME*** — название хранилища;
- ***TYPE*** — BACKUP_DS;
- ***DS_MAD*** — restic;
- ***RESTIC_SFTP_USER*** — пользователь на сервере резервного копирования;
- ***RESTIC_PASSWORD*** — пароль для шифрования репозитория Restic;
- ***RESTIC_SFTP_SERVER*** — IP-адрес или имя сервера резервного копирования.

Важно

Сервер, указанный в ***RESTIC_SFTP_SERVER***, должен быть доступен как с узла управления, так и со всех узлов гипервизора.

Хранилище резервных копий Restic также можно создать через веб-интерфейс Sunstone:



После успешной регистрации:

```
$ onedatastore list
  ID NAME                SIZE AVA CLUSTERS IMAGES TYPE DS      TM      STAT
 100 RBackups           134.7G 76% 0             0 bck  restic  -       on
```

Теперь можно создавать резервные копии ВМ.

33.1.3. Обслуживание и устранение неполадок

33.1.3.1. Очистка репозитория (prune)

Незадействованные данные автоматически удаляются при:

- удалении снимков вручную;
- применении политики хранения (retention policy).

Операция выполняется командой **`restic prune`**, которая запускается OpenNebula автоматически.

33.1.3.2. Разблокировка репозитория

Если репозиторий остался заблокированным (например, после сбоя), может появиться ошибка:

```
unable to create lock in backend: repository is already locked...
```

Чтобы разблокировать репозиторий:

1. Убедитесь, что нет активных операций резервного копирования.
2. Выполните на узле управления:

```
$ restic -r sftp:oneadmin@192.168.0.197:/var/lib/one/datastores/
<ID_хранилища>/<ID_BM> \
  --password-file <(echo "opennebula") unlock --remove-all
```

33.1.3.3. Ограничение нагрузки на I/O и CPU

Операции резервного копирования могут создавать значительную нагрузку. OpenNebula поддерживает следующие механизмы ограничения:

- приоритет процессов (через nice и ionice);
- ограничение ресурсов с помощью cgroups (через systemd-срезы).

Для использования cgroups необходимо делегировать контроллеры cpu и io пользователю oneadmin на всех узлах:

1. Создайте файл:

```
# /etc/systemd/system/user@9869.service.d/delegate.conf
[Service]
Delegate=cpu cpuset io
```

2. После перезагрузки узла проверьте:

```
$ cat /sys/fs/cgroup/user.slice/user-9869.slice/cgroup.controllers
cpuset cpu io memory pids
```

33.1.3.4. Изменение временного каталога резервного копирования

По умолчанию временные файлы резервной копии создаются в каталоге **/var/lib/one/datastores/<ID_хранилища>/<ID_BM>/backup**.

Чтобы изменить путь на всех узлах, отредактируйте параметр ***BACKUP_BASE_PATH*** в файле **/var/lib/one/remotes/etc/datastore/datastore.conf**.

33.2. Хранилище резервных копий: Rsync

Rsync — это утилита с открытым исходным кодом для передачи и синхронизации файлов, входящая в состав большинства дистрибутивов Linux. В OpenNebula реализована поддержка резервного копирования с использованием Rsync, позволяющая создавать полные и инкрементальные копии ВМ.

Инкрементальное резервное копирование в Rsync реализуется на файловом уровне с использованием жёстких ссылок (hard links). Это позволяет экономить дисковое пространство, однако не обеспечивает дедупликацию и отслеживание изменений на уровне блоков, как в Restic.

33.2.1. Настройка сервера резервного копирования

Настройка сервера резервного копирования:

1. Подготовьте выделенный сервер, доступный по сети со всех узлов гипервизора.
2. Создайте учётную запись (например, oneadmin) на сервере:

```
# useradd -m -u 9869 oneadmin
# passwd oneadmin
```

3. Настройте беспарольный SSH-доступ:
   - скопируйте открытый SSH-ключ (**~oneadmin/.ssh/id_rsa.pub**) с узла управления OpenNebula и узлов виртуализации в файл **~oneadmin/.ssh/authorized_keys** на сервере резервного копирования:

     ```
     $ ssh-copy-id -i /var/lib/one/.ssh/id_rsa.pub
     <сервер_резервного_копирования>
     ```

   - убедитесь, что подключение с узла управления OpenNebula и узлов виртуализации выполняется без запроса пароля:

     ```
     $ ssh <сервер_резервного_копирования>
     ```

4. Создайте каталог для хранения репозиториев и назначьте владельца:

```
# mkdir -p /var/lib/one/datastores
# chown oneadmin:oneadmin /var/lib/one/datastores
```

5. (Опционально) Смонтируйте том хранения в каталог **/var/lib/one/datastores**, например:

```
# mount /dev/vgBackup/lvBackup /var/lib/one/datastores
```

Примечание

В примере используется LVM-том vgBackup/lvBackup, смонтированный в **/var/lib/one/datastores**.

6. Установите необходимые пакеты:

```
# apt-get install rsync qemu-img
```

Примечание

Для передачи резервных копий рекомендуется использовать выделенную сеть, чтобы снизить нагрузку на основную инфраструктуру.

33.2.2. Создание хранилища резервных копий в OpenNebula

Для создания хранилища резервных копий с использованием конфигурационного файла:

1. Создайте шаблон хранилища **ds_rsync.conf**:

```
NAME        = "Rsync Backups"
TYPE        = "BACKUP_DS"
DS_MAD      = "rsync"
TM_MAD      = "-"
RSYNC_USER  = "oneadmin"
RSYNC_HOST  = "192.168.0.197"
```

2. Зарегистрируйте хранилище:

```
$ onedatastore create ds_rsync.conf
ID: 103
```

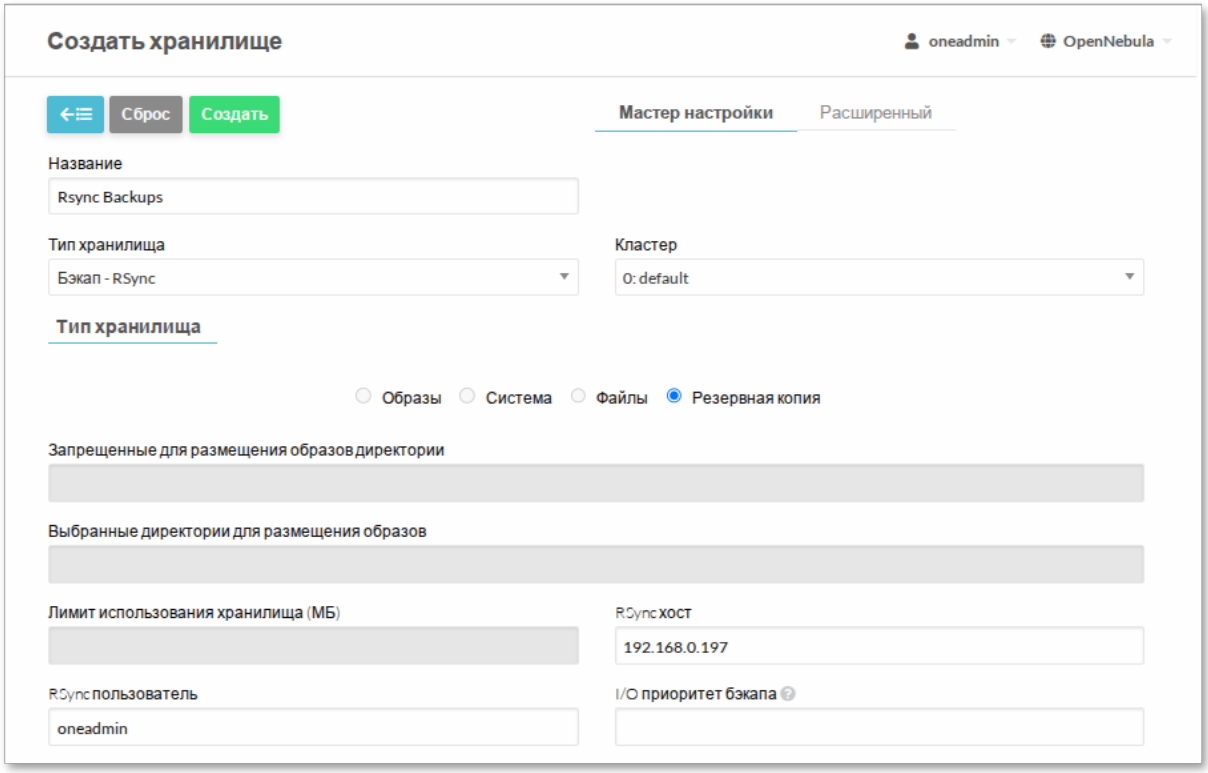
Для создания хранилища резервных копий Rsync необходимо указать следующие параметры:

- ***NAME*** — название хранилища;
- ***TYPE*** — BACKUP_DS;
- ***DS_MAD*** — rsync;
- ***RSYNC_USER*** — пользователь на сервере резервного копирования;
- ***RSYNC_HOST*** — IP-адрес или имя сервера резервного копирования.

Важно

Адрес ***RSYNC_HOST*** должен быть доступен как с узла управления, так и со всех узлов гипервизора.

Хранилище резервных копий Rsync также можно зарегистрировать через веб-интерфейс Sunstone:



После успешной регистрации:

```
$ onedatastore list
  ID NAME                SIZE AVA CLUSTERS IMAGES TYPE DS      TM      STAT
 103 Rsync Backups       134.7G 76% 0             0 bck  rsync   -       on
```

Теперь можно создавать резервные копии ВМ.

33.2.3. Обслуживание и устранение неполадок

33.2.3.1. Ограничение нагрузки на I/O и CPU

Операции резервного копирования могут создавать значительную нагрузку. OpenNebula поддерживает следующие механизмы ограничения:

- приоритет процессов (через nice и ionice);
- ограничение ресурсов с помощью cgroups (через systemd-срезы).

Для использования cgroups необходимо делегировать контроллеры cpu и io пользователю oneadmin на всех узлах:

1. Создайте файл:

```
# /etc/systemd/system/user@9869.service.d/delegate.conf
[Service]
Delegate=cpu cpuset io
```

2. После перезагрузки узла проверьте:

```
$ cat /sys/fs/cgroup/user.slice/user-9869.slice/cgroup.controllers
cpuset cpu io memory pids
```

33.2.3.2. Изменение временного каталога резервного копирования

По умолчанию временные файлы резервной копии создаются в каталоге **`/var/lib/one/datastores/<ID_хранилища>/<ID_BM>/backup`**.

Чтобы изменить путь на всех узлах, отредактируйте параметр ***`BACKUP_BASE_PATH`*** в файле **`/var/lib/one/remotes/etc/datastore/datastore.conf`**.

33.3. Настройка резервного копирования

Перед созданием резервных копий необходимо настроить параметры резервного копирования. Это можно сделать как в шаблоне ВМ, так и для уже существующей ВМ.

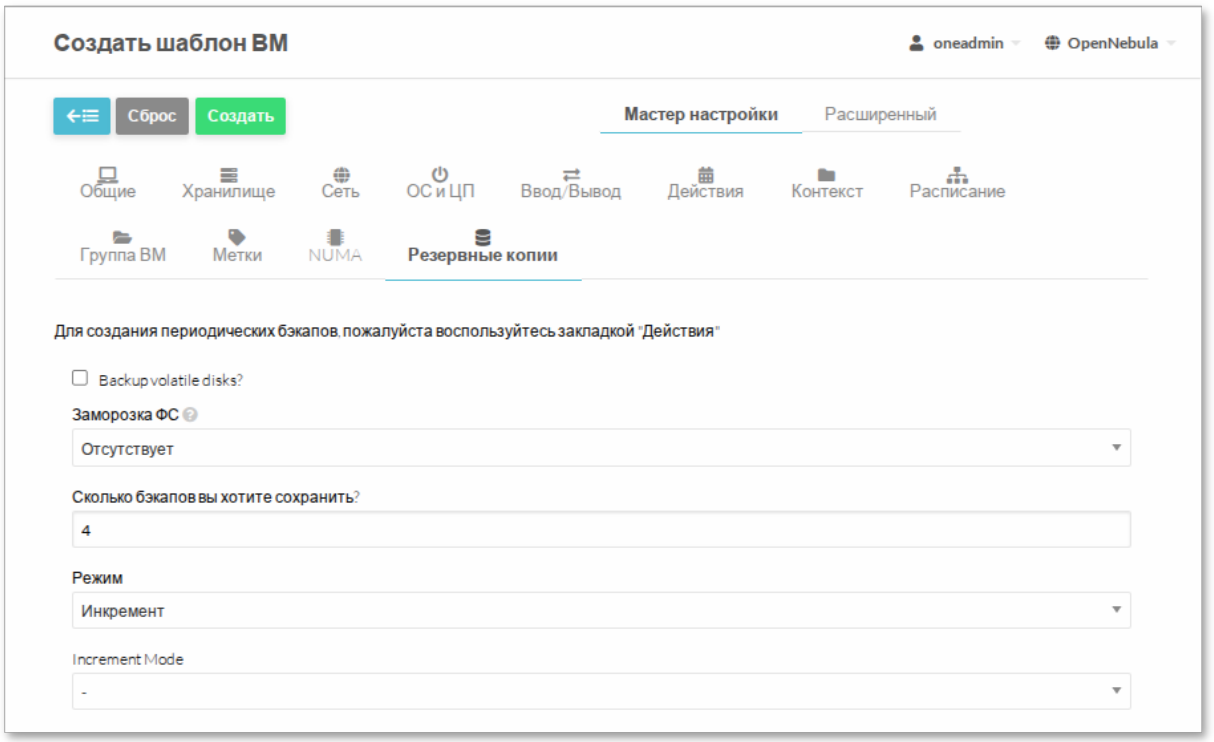
33.3.1. Через шаблон ВМ

Если параметры резервного копирования заданы в шаблоне, все ВМ, созданные на его основе, будут использовать эти настройки по умолчанию.

Пример шаблона с секцией ***`BACKUP_CONFIG`***:

```
NAME   = "Template - Backup"
CPU    = "1"
MEMORY = "2048"

DISK = [ IMAGE_ID = "1" ]

BACKUP_CONFIG = [
  FS_FREEZE = "NONE",
  KEEP_LAST = "4",
  MODE      = "INCREMENT"
]
```

При создании шаблона в веб-интерфейсе Sunstone эти параметры можно указать на вкладке **Резервные копии**:



33.3.2. Через ВМ

Для запущенной ВМ параметры резервного копирования можно задать или обновить с помощью вызова API **`updateconf`** или через командную строку.

Пример обновления конфигурации в командной строке:

```
$ onevm updateconf 13
```

Добавьте в редакторе:

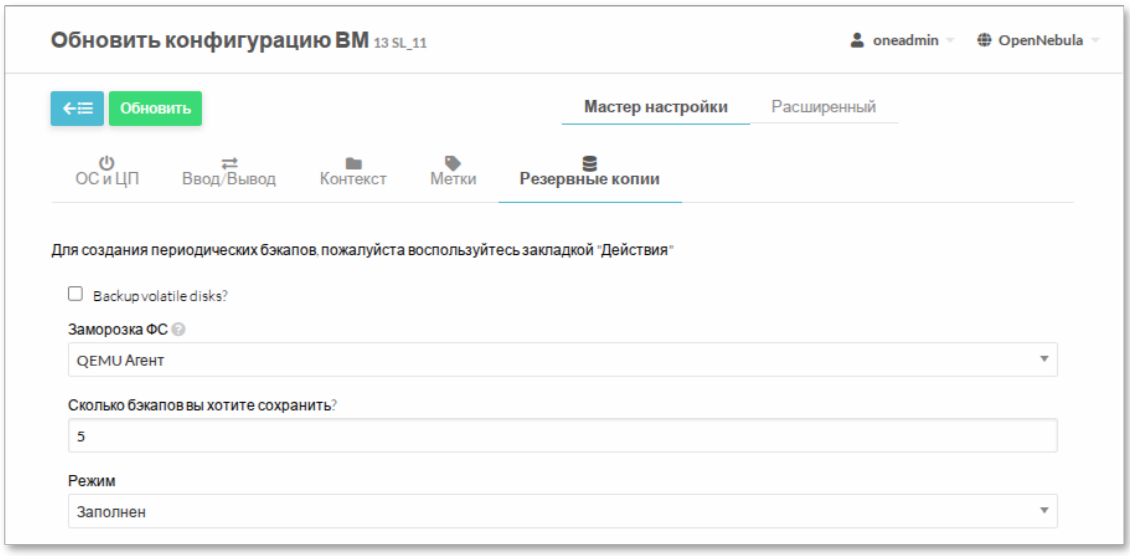
```
BACKUP_CONFIG = [
  FS_FREEZE = "AGENT",
  KEEP_LAST = "5",
  MODE      = "FULL"
]
```

Проверка конфигурации:

```
$ onevm show 13
VIRTUAL MACHINE 13 INFORMATION
ID                  : 13
NAME                : SL_11
…
BACKUP CONFIGURATION
BACKUP_VOLATILE="NO"
FS_FREEZE="SUSPEND"
INCREMENTAL_BACKUP_ID="-1"
KEEP_LAST="5"
LAST_INCREMENT_ID="-1"
MODE="FULL"
```

Для обновления конфигурации ВМ в веб-интерфейсе Sunstone:

1. Откройте ВМ.
2. На вкладке **Конфигурация** нажмите кнопку **`Обновить конфигурацию`**.
3. На вкладке **Резервное копирование** настроить параметры резервного копирования:



4. Нажмите кнопку **`Обновить`**.

Таблица 33.1. Атрибуты конфигурации резервного копирования

Атрибут	Описание	Значение по умолчанию
`BACKUP_VOLATILE`	Резервное копирование volatile-дисков	NO
`FS_FREEZE`	Метод заморозки ФС: NONE, AGENT, SUSPEND	NONE
`KEEP_LAST`	Сохранять последние N копий	не задано
`MODE`	Тип копии: FULL, INCREMENT	FULL
`INCREMENT_MODE`	Режим инкремента: CBT, SNAPSHOT	CBT
`INCREMENTAL_BACKUP_ID`	ID образа с цепочкой инкрементов (только для чтения)	-1
`LAST_INCREMENT_ID`	ID последнего инкремента (только для чтения)	-1

33.4. Выполнение резервного копирования

33.4.1. Разовое копирование (One-shot)

При создании разовой резервной копии используются параметры резервного копирования, настроенные для ВМ (например, ***`MODE`***), а также дополнительные аргументы команды:

- ***`ID_хранилища`*** — идентификатор хранилища резервных копий, в котором будет сохранена резервная копия;

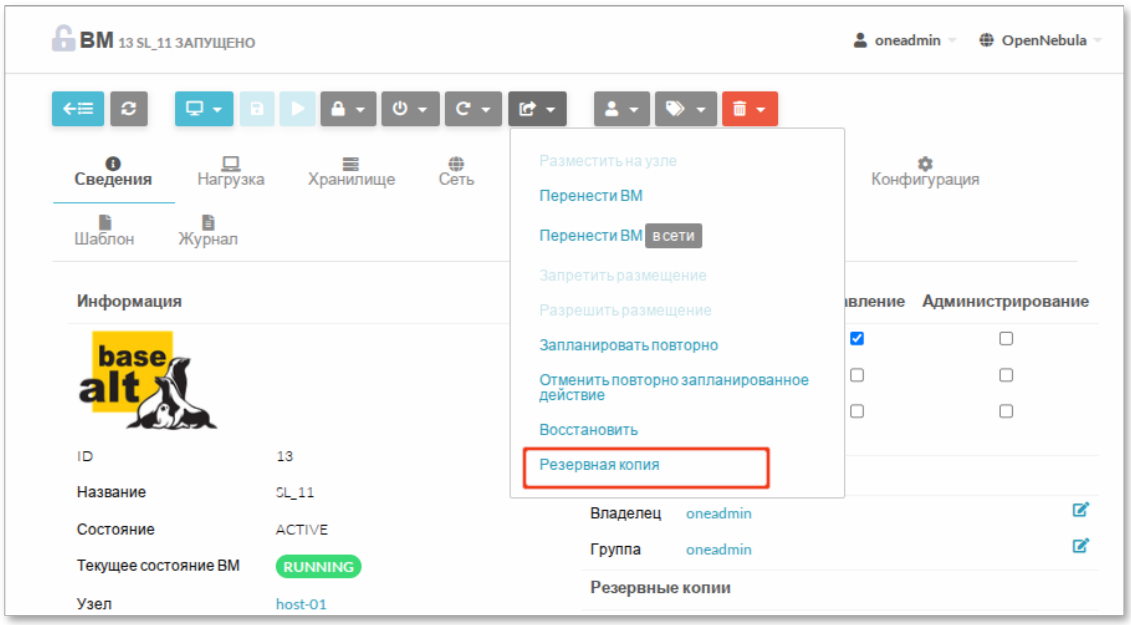
- ***-d*** (опционально) — при инкрементальном резервном копировании закрывает текущую активную цепочку инкрементов и начинает новую.

Запуск резервного копирования напрямую из командной строки доступен только пользователю oneadmin:

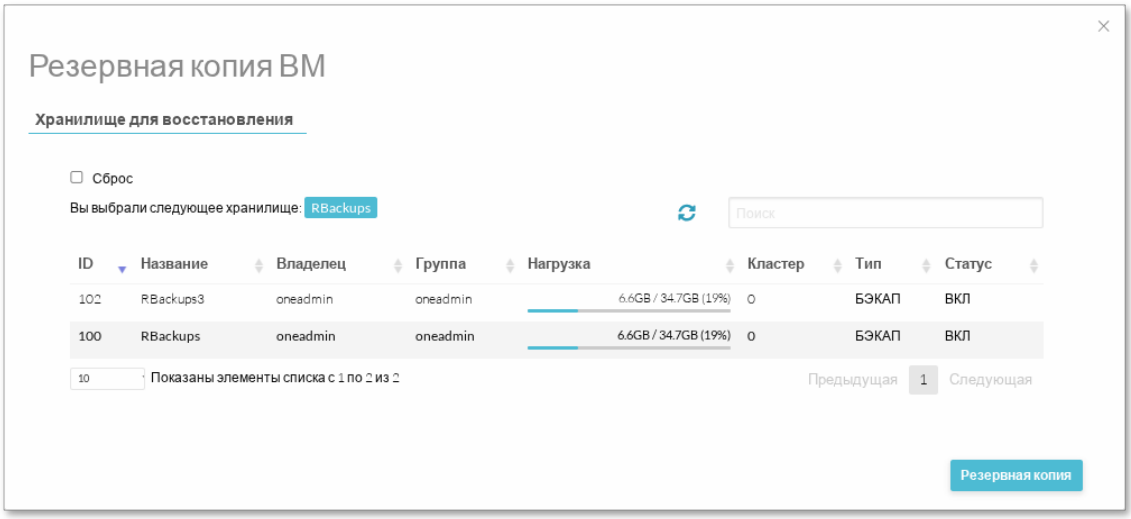
```
$ onevm backup -d 100 13
```

Запуск разового резервного копирования в веб-интерфейсе Sunstone:

1. Выберите ВМ.
2. В меню действий выберите **Резервная копия**:



3. В открывшемся окне выберите хранилище резервных копий:



4. Нажмите кнопку **Резервная копия**.

После завершения операции в информации о ВМ появятся сведения о созданной резервной копии и связанном с ней образе.

Пример:

```
$ onevm show 13
VIRTUAL MACHINE 13 INFORMATION
ID                  : 13
NAME                : SL_11

BACKUP CONFIGURATION
BACKUP_VOLATILE="NO"
FS_FREEZE="SUSPEND"
INCREMENTAL_BACKUP_ID="-1"
KEEP_LAST="2"
LAST_INCREMENT_ID="-1"
MODE="FULL"

VM BACKUPS
IMAGE IDS: 16
```

Информация об образе резервной копии:

```
$ oneimage show 16
IMAGE 16 INFORMATION
ID             : 16
NAME           : 13 16-Jan 13.42.11
USER           : oneadmin
GROUP          : oneadmin
LOCK           : None
DATASTORE      : RBackups
TYPE           : BACKUP
REGISTER TIME  : 01/16 13:42:11
PERSISTENT     : Yes
SOURCE         : 5ed6faf4
FORMAT         : raw
SIZE           : 12.2G
STATE          : rdy
RUNNING_VMS    : 1

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

IMAGE TEMPLATE

BACKUP INFORMATION
VM             : 13
TYPE           : FULL
```

В хранилище будет создан образ с меткой one-<VM_ID>.

33.4.2. Планируемое копирование

OpenNebula поддерживает периодическое резервное копирование с использованием механизма планирования действий.

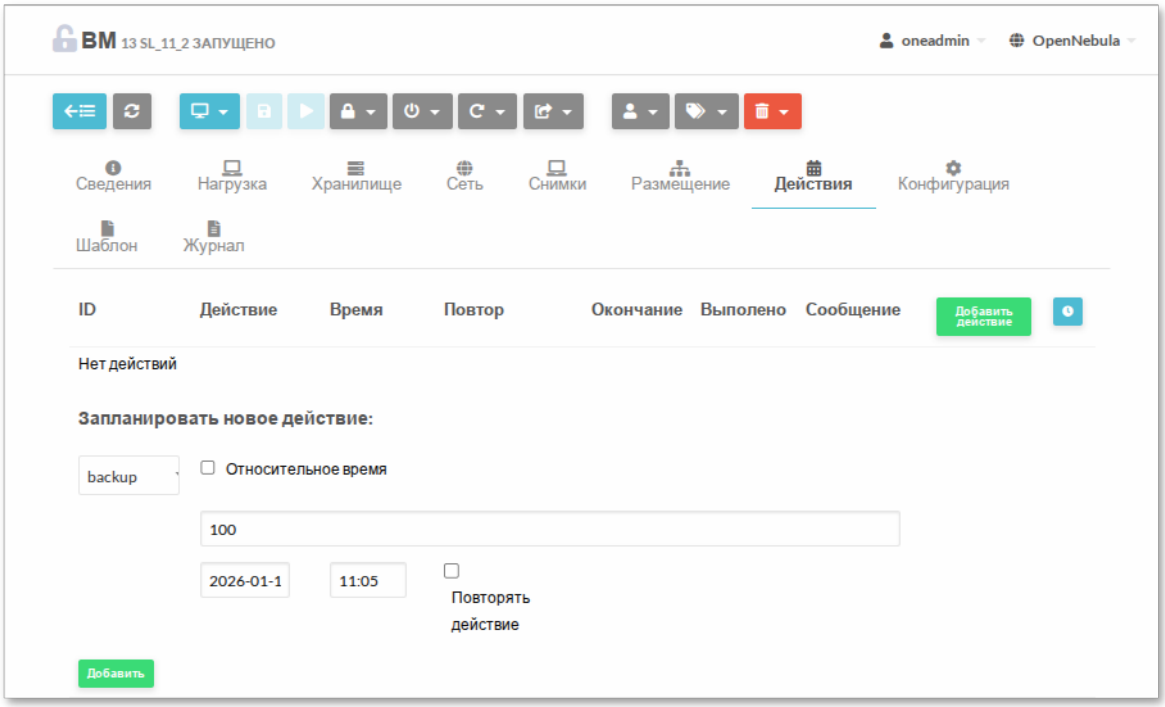
При создании задания необходимо указать идентификатор хранилища резервных копий в качестве аргумента операции.

Периодическое резервное копирование можно создать:

- с помощью параметра ***--schedule*** в командной строке:

```
$ onevm backup --schedule "0 2 * * *" -d 100 13 --weekly "1,3,5"
```

- в веб-интерфейсе Sunstone (вкладка **Действия**):



Примечание

За выполнение и планирование операций резервного копирования отвечает ядро OpenNebula. Ограничения на количество одновременно выполняемых операций можно задать в файле **/etc/one/oned.conf**:

- ***MAX_BACKUPS*** — максимальное количество активных операций резервного копирования во всей инфраструктуре. После достижения лимита новые задания запускаться не будут;
- ***MAX_BACKUPS_HOST*** — максимальное количество одновременно выполняемых операций резервного копирования на одном узле гипервизора.

33.4.3. Отмена операции резервного копирования

Если операция резервного копирования ещё не завершена, её можно отменить:

```
$ onevm backup-cancel 13
```

В этом случае:

- ВМ возвращается в исходное состояние (RUNNING или POWEROFF);

- временные файлы будут автоматически удалены при следующей операции резервного копирования.

Если ВМ остаётся в состоянии резервного копирования и не выходит из него (например, после сбоя):

```
$ onevm recover 13 --success
```

33.5. Просмотр резервных копий

Список всех резервных копий можно получить, выполнив команду:

```
$ oneimage list --backup
ID USER     GROUP    NAME                DATASTORE     SIZE TYPE PER STAT
RVMS
30 oneadmin oneadmin 18 20-Jan 12.18.34  rsync_loca   12.5G BK   Yes rdy
1
26 oneadmin oneadmin 15 17-Jan 20.44.03  Rbackups     12.4G BK   Yes rdy
1
17 oneadmin oneadmin 13 16-Jan 22.35.52  Rsync Back   12.2G BK   Yes rdy
1
16 oneadmin oneadmin 13 16-Jan 13.42.11  Rbackups     12.2G BK   Yes rdy
1
```

Для поиска резервных копий, относящихся к конкретной ВМ, можно отфильтровать вывод по идентификатору ВМ:

```
$ oneimage list --backup | grep " 13 "
```

Для получения подробной информации о конкретной резервной копии используется команда:

```
$ oneimage show <ID_образа>
```

Пример вывода для полной копии:

```
$ oneimage show 17
…
TYPE           : BACKUP
REGISTER TIME  : 01/16 22:35:52
PERSISTENT     : Yes
SOURCE         : e4e9a8
FORMAT         : raw
SIZE           : 12.2G
--
BACKUP INFORMATION
VM             : 13
TYPE           : FULL
```

В информации о резервной копии также указываются:

- тип резервной копии (FULL или INCREMENTAL);
- идентификатор ВМ, к которой относится резервная копия;

- формат диска;
- размер резервной копии;
- источник (хеш базового снимка).

Пример вывода для инкрементальной копии:

```
$ oneimage show 30 | grep -A5 BACKUP
YPE            : BACKUP
REGISTER TIME  : 01/20 12:18:34
PERSISTENT     : Yes
SOURCE         : 408fa3
FORMAT         : raw
SIZE           : 12.5G
--
BACKUP INFORMATION
VM             : 18
TYPE           : INCREMENTAL

BACKUP INCREMENTS
ID PID T SIZE                DATE SOURCE
0  -1 F 12.2G     01/20 12:18:34 408fa3
1   0 I 217M      01/20 12:55:21 f2acce
2   1 I 18M       01/20 13:18:17 f1c67d
```

Для инкрементальных резервных копий дополнительно выводится список инкрементов. Поля списка ***BACKUP INCREMENTS***:

- ***ID*** — идентификатор инкремента;
- ***PID*** — идентификатор родительского инкремента;
- ***T*** — тип резервной копии: F (полная), I (инкрементальная);
- ***SIZE*** — размер инкремента;
- ***DATE*** — дата и время создания;
- ***SOURCE*** — хеш снимка диска, на основе которого создан инкремент.

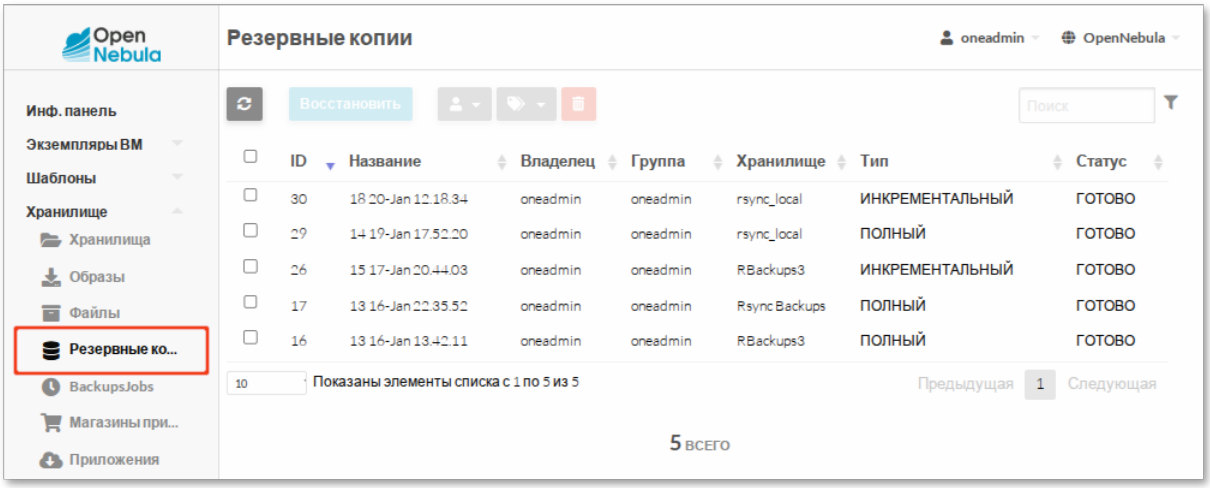
Атрибут ***SOURCE*** в образах резервных копий (и их инкрементах) представляет собой непрозрачный идентификатор, ссылающийся на объект резервной копии во внешней системе хранения (Restic, Rsync и т. п.), используемой данным хранилищем резервных копий.

Примечание

Для Restic снимок можно найти по ID ВМ:

```
$ export RESTIC_PASSWORD="opennebula"
$ restic -r sftp:oneadmin@192.168.0.197:/var/lib/one/datastores/100/13
snapshots --tag one-13
repository a8e6b00c opened (version 2, compression level auto)
created new cache in /var/lib/one/.cache/restic
ID        Time                 Host        Tags
Paths                                          Size
----------------------------------------------------------------------
----------------------------------------
5ed6faf4  2026-01-16 13:33:04  host-01     one-13      /var/lib/one/
datastores/0/13/backup/disk.0.0  12.239 GiB
                                                      /var/lib/one/
datastores/0/13/backup/vm.xml
----------------------------------------------------------------------
----------------------------------------
```

Список резервных копий также доступен в веб-интерфейсе Sunstone в разделе **Хранилище → Резервные копии**:



В данном разделе можно:

- просмотреть подробную информацию о резервной копии;
- удалить резервную копию;
- выполнить восстановление ВМ из резервной копии.

Для инкрементальных резервных копий на вкладке **добавляет** отображается цепочка снимков, входящих в инкрементальную резервную копию.

33.6. Восстановление резервных копий

OpenNebula поддерживает два метода восстановления резервных копий ВМ:

- восстановление на месте (In-place restore) — замена дисков существующей ВМ данными из резервной копии;
- полное восстановление (Full restore) — создание новых образов дисков и шаблона ВМ без привязки к исходной ВМ.

Если используется удалённый сервер резервного копирования без общей файловой системы, доступно только полное восстановление с помощью команды **`oneimage restore`**. При наличии общей файловой системы (например, NFS), доступной узлу управления и узлам виртуализации, доступен полный функционал восстановления, включая **`onevm restore`**.

33.6.1. Восстановление на месте (In-place restore)

Данный метод заменяет диски существующей ВМ данными из резервной копии.

Требования:

- ВМ должна быть выключена (состояние POWEROFF);
- возможно восстановление как всех дисков, так и только выбранных.

Примечание

Для использования восстановления в режиме In-place необходимо, чтобы хранилище типа BACKUP_DS было доступно через общую файловую систему (например, NFS) всем узлам облака — узлу управления и узлам виртуализации.

Команда восстановления:

```
$ onevm restore <ID_ВМ> <ID_образа>
```

Допустимые параметры:

- ***`--disk-id ID_диска`*** — восстановить только указанный диск;

- ***--increment ID_инкремента*** — использовать указанный идентификатор инкремента (если параметр не задан, используется последний).

Допустим, имеется:

- ВМ 15 в состоянии POWEROFF;
- резервная копия (образ 26) с тремя инкрементами:

```
$ oneimage show 26
IMAGE 26 INFORMATION
...
BACKUP INFORMATION
VM              : 15
TYPE            : INCREMENTAL

BACKUP INCREMENTS
 ID PID T SIZE                DATE SOURCE
  0  -1 F 12.2G     01/17 20:44:03 bb3b65eb
  1   0 I 35M       01/17 21:29:51 f5c79cfc
  2   1 I 139M      01/17 21:38:34 abb971bb
```

Чтобы восстановить ВМ из второго инкремента (ID=1), выполните:

```
$ onevm restore --increment 1 15 26
```

Важно

При восстановлении на месте все существующие снимки (snapshots) ВМ будут удалены.

33.6.2. Полное восстановление (Full restore)

Метод полного восстановления создаёт новые ресурсы на основе резервной копии:

- шаблон ВМ — с теми же параметрами (CPU, память, сеть), которые были у исходной ВМ на момент создания резервной копии;
- образы дисков — по одному на каждый диск, содержащийся в резервной копии.

Преимущества метода:

- не требуется существование исходной ВМ;
- возможна гибкая настройка сетевых параметров и других атрибутов перед запуском ВМ.

Доступные настройки при восстановлении:

- сброс сетевых параметров:
  - не сохранять IP/MAC-адреса;
  - полностью удалить определения сетевых интерфейсов.

- выбор инкремента (для инкрементальных копий):
  - по умолчанию используется последний инкремент;
  - можно указать конкретный инкремент.
- задание базового имени для создаваемых ресурсов.

Команда восстановления:

```
$ oneimage restore <ID_образа>
```

Допустимые параметры:

- ***-d хранилище*** — целевое хранилище;
- ***--disk-id ID_диска*** — восстановить только указанный диск;
- ***--increment ID_инкремента*** — использовать указанный идентификатор инкремента (если параметр не задан, используется последний);
- ***--name имя*** — имя нового образа;
- ***--no_ip*** — не сохранять адреса етевых адаптеров (MAC, IPv4 и IPv6);
- ***--no_nic*** — не сохранять определения сетевых интерфейсов.

Пример восстановления копии из образа 16 без сохранения IP-адресов:

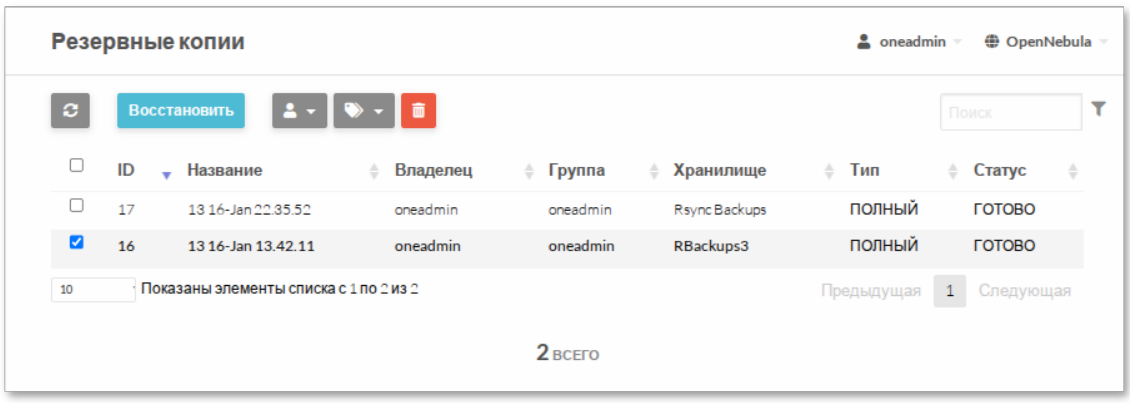
```
$ oneimage restore -d default --no_ip 16
VM Template: 5
Images: 18
```

Результат:

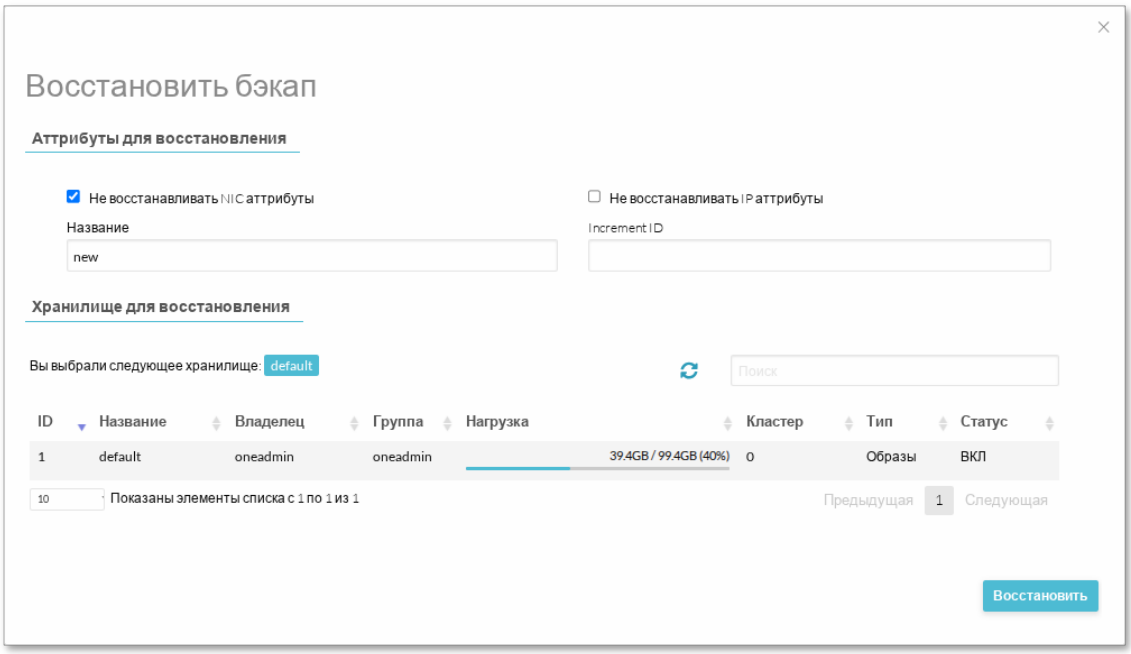
- создан шаблон 5 с именем вида <ID_ВМ>-<ID_снимка> (например, 13-5ed6faf4);
- создан образ 18 с именем вида <ID_ВМ>-<ID_снимка>-disk-<ID_диска> (например, 13-5ed6faf4-disk-0).

Пример восстановления в веб-интерфейсе Sunstone:

1. В разделе **Хранилище** → **Резервные копии** выберите резервную копию и нажмите кнопку **Восстановить**:



2. В открывшемся окне укажите имя нового шаблона и параметры восстановления:



Примечание

Будет создан образ с именем вида <имя>-disk-<ID_диска> (например, new-disk-0).

3. Нажмите кнопку **`Восстановить`**.

Примечание

После восстановления рекомендуется проверить созданный шаблон и при необходимости скорректировать параметры (сети, диски, контекстуализацию, пользовательские переменные).

33.7. Дополнительные возможности

33.7.1. Квоты и управление доступом

Хранилища резервных копий поддерживают те же механизмы управления ресурсами и доступом, что и обычные хранилища данных. что и обычные хранилища.

Квоты:

- ограничение общего объёма резервных копий для пользователя или группы;
- ограничение количества резервных копий (инкрементальная цепочка учитывается как одна копия).

Права доступа:

- управление правами через команду **`oneimage chmod`**;

- расширенная настройка с использованием ACL.

33.7.2. Многоуровневые политики резервного копирования

Для повышения надёжности хранения данных можно использовать несколько хранилищ резервных копий с разными расписаниями:

- еженедельное резервное копирование в локальное хранилище (in-house);
- ежемесячное резервное копирование в облачное хранилище (cloud-storage).

Пример:

```
# Еженедельно по пятницам
$ onevm backup --schedule "0 2 * * 5" -d 100 23

# Ежемесячно 1-го числа
$ onevm backup --schedule "0 3 1 * *" -d 200 23
```

Глава 34. Задания резервного копирования

Задания резервного копирования позволяют централизованно управлять операциями резервного копирования для группы ВМ. Это упрощает администрирование облачной инфраструктуры и обеспечивает:

- единую политику резервного копирования для нескольких ВМ (расписание, хранение, заморозка файловой системы);
- контроль нагрузки на систему (ограничение числа параллельных операций);
- мониторинг прогресса и оценку времени выполнения заданий.

34.1. Определение задания

Для создания задания резервного копирования необходимо указать следующие компоненты:

- список ВМ;
- параметры резервного копирования;
- приоритет — важность задания (опционально);
- расписание выполнения копирования.

Список ВМ указывается в порядке выполнения операций, через запятую:

```
BACKUP_VMS = "13,15,3"
```

Примечание

Порядок ВМ в списке определяет последовательность выполнения операций резервного копирования.

Все ВМ в задании используют одинаковые параметры, параметрам одиночного резервного копирования, например:

```
DATASTORE_ID = 101
FS_FREEZE    = "NONE"
KEEP_LAST    = "4"
MODE         = "INCREMENT"
INCREMENT_MODE = "SNAPSHOT"
```

Приоритет задания определяет порядок выполнения заданий резервного копирования:

- 0–49 — доступны обычным пользователям;
- 50–99 — зарезервированы для администраторов.

Также можно выбрать режим выполнения задания:

- SEQUENTIAL — последовательное копирование (обязательно для Restic);
- PARALLEL — параллельное копирование (с учётом системных лимитов).

Пример:

```
PRIORITY  = 7
EXECUTION = "SEQUENTIAL"
```

Примечание

Общее количество одновременно выполняемых операций резервного копирования регулируется параметром ***MAX_BACKUPS*** в файле **`/etc/one/oned.conf`**.

Расписание задаётся с помощью атрибута ***SCHED_ACTION*** (аналогично расписанию действий ВМ):

```
SCHED_ACTION = [
  REPEAT = "0",
  DAYS   = "1",     # Понедельник
  END_TYPE = "0",
  TIME   = "1768776420"
]
```

Примечание

Параметры ***ACTION*** и ***ARGS*** задаются автоматически и указывать их вручную не требуется.

Для одного задания можно определить несколько расписаний.

Пример полного задания (файл **my_backupjob.conf**):

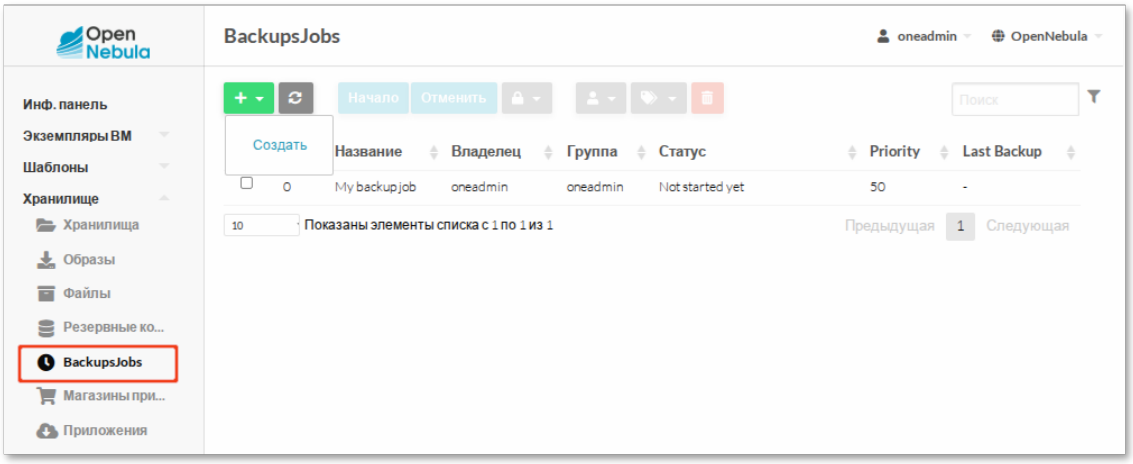
```
NAME = "My backup job"

BACKUP_VMS   = "12,13"
DATASTORE_ID = 100

FS_FREEZE = "NONE"
KEEP_LAST = "5"
MODE      = "INCREMENT"

PRIORITY  = 50
EXECUTION = "SEQUENTIAL"

SCHED_ACTION = [
  REPEAT="0", DAYS="1", END_TYPE="0", TIME="1768776420"
]
```

Регистрация задания:

```
$ onebackupjob create my_backupjob.conf
ID: 1
```

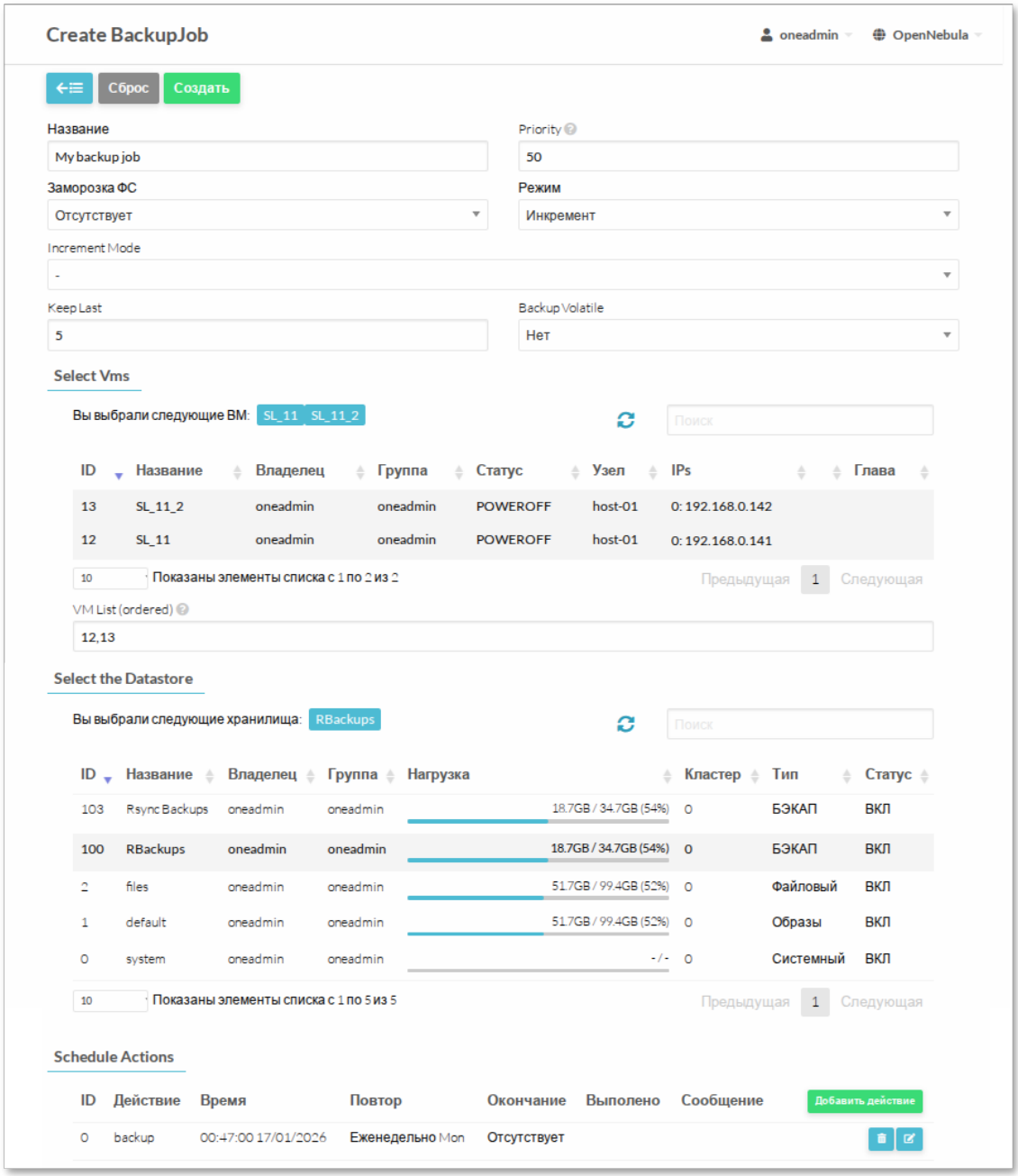
Создание задания в веб-интерфейсе Sunstone:

1. Перейдите в раздел **Хранилище** → **BackupsJobs**.
2. На загруженной странице нажмите **+** → **Создать**:



3. В открывшемся окне укажите:
   - имя задания;
   - параметры резервного копирования;

- список ВМ;
- хранилище для резервных копий;
- расписание выполнения.



4. Нажмите кнопку **`Создать`**.

Таблица 34.1. Атрибуты задания резервного копирования

Атрибут	Описание
`BACKUP_VMS`	Список идентификаторов ВМ, разделённых запятыми
`DATASTORE_ID`	Идентификатор хранилища резервных копий
`RESET`	

Атрибут	Описание
	Создать новую цепочку инкрементов (только для INCREMENT)
BACKUP_VOLATILE	Резервное копирование volatile-дисков (по умолчанию NO)
FS_FREEZE	Метод заморозки ФС: NONE (по умолчанию), AGENT, SUSPEND
KEEP_LAST	Сохранять последние N копий (по умолчанию не ограничено)
MODE	Тип копии: FULL (по умолчанию), INCREMENT
INCREMENT_MODE	Режим инкремента: CBT (по умолчанию), SNAPSHOT
INCREMENTAL_BACKUP_ID	ID образа с цепочкой инкрементов (только для чтения)
LAST_INCREMENT_ID	ID последнего инкремента (только для чтения)
PRIORITY	Приоритет задания от 0 (низкий) до 99 (высокий)
EXECUTION	Режим выполнения: SEQUENTIAL или PARALLEL
SCHED_ACTION	Расписание выполнения задания

34.2. Управление заданиями

Просмотр списка заданий:

```
$ onebackupjob list
  ID USER     GROUP    PRIO NAME              LAST              VMS
   0 oneadmin oneadmin 50   My backup job     01/17/26 12:34    12,13
```

Подробная информация о задании:

```
$ onebackupjob show 0
BACKUP JOB 0 INFORMATION
ID              : 0
NAME            : My backup job
USER            : oneadmin
GROUP           : oneadmin
LOCK            : None
PERMISSIONS
OWNER           : um-
GROUP           : ---
OTHER           : ---

LAST BACKUP JOB EXECUTION INFORMATION
TIME            : 01/17 12:34:10
DURATION        :   0d 00h00m27s

VIRTUAL MACHINE BACKUP STATUS
UPDATED         :
```

```
OUTDATED       :
ONGOING        :
ERROR          : 12,13

SCHEDULED ACTIONS
   ID ACTION  ARGS   SCHEDULED   REPEAT   END STATUS
    3 backup     - 01/19 00:47 Weekly 1  None Next in 1.51 days

TEMPLATE CONTENTS
BACKUP_VMS="12,13"
BACKUP_VOLATILE="NO"
DATASTORE_ID="100"
EXECUTION="SEQUENTIAL"
FS_FREEZE="NONE"
INCREMENT_MODE="CBT"
KEEP_LAST="5"
MODE="INCREMENT"
```

Немедленный запуск задания:

```
$ onebackupjob backup 1
```

При этом все ВМ, входящие в задание, помечаются как OUTDATED и ставятся в очередь на резервное копирование.

Состояние ВМ в задании:

- OUTDATED — требуется резервное копирование;
- ONGOING — резервное копирование выполняется;
- UPDATED — резервное копирование завершено успешно;
- ERROR — ошибка при выполнении резервного копирования.

34.3. Изменение конфигурации

Обновление параметров задания:

```
$ onebackupjob update 0
```

(в открывшемся редакторе измените нужные атрибуты)

Управление расписанием:

- добавить новое расписание:

  ```
  $ onebackupjob backup --schedule "0 2 * * *" 0
  ```

- обновить существующее расписание (по ID):

  ```
  $ onebackupjob sched-update 0 <ID_расписания>
  ```

- удалить расписание:

```
$ onebackupjob sched-delete 0 <ID_расписания>
```

Изменение приоритета:

```
$ onebackupjob priority 0 10
```

Задания резервного копирования поддерживают стандартные операции управления объектами OpenNebula:

- chmod — изменение прав доступа;
- chown/chgrp — смена владельца и группы;
- rename — переименование;
- delete — удаление;
- lock/unlock — блокировка и разблокировка.

Специфические операции:

- cancel — отмена текущих операций резервного копирования;
- retry — повторное копирование ВМ, завершившихся с ошибкой.

34.4. Восстановление из заданий

После завершения задания создаются образы резервных копий с именами вида:

```
<ID_ВМ> <ДЕНЬ>-<МЕСЯЦ> <ВРЕМЯ>
```

Например: 13 17-Jan 14.05.25.

Восстановление таких резервных копий выполняется аналогично обычным резервным копиям (см. раздел Восстановление резервных копий).

Глава 35. Управление пользователями

Доступ пользователей к ресурсам в OpenNebula регулируется системой разрешений. По умолчанию только владелец ресурса может управлять им или использовать его. Пользователи могут предоставлять другим пользователям (в своей группе или вне её) права на использование или управление своими ресурсами.

При создании группы может быть назначен администратор группы. Такой пользователь получает расширенные права на создание новых пользователей в группе и управление ресурсами, принадлежащими группе (через CLI или специальное представление в Sunstone). Группы могут быть назначены виртуальным центрам обработки данных (VDC).

По умолчанию аутентификация и авторизация выполняются внутренней службой OpenNebula. При необходимости эти функции можно делегировать внешним системам (например, LDAP). Подробнее см. в разделе Настройка аутентификации.

35.1. Пользователи

Пользователь в OpenNebula определяется именем пользователя и паролем. Каждый пользователь имеет уникальный идентификатор (ID) и принадлежит минимум одной группе.

При установке OpenNebula автоматически создаются:

- две административные учётные записи: oneadmin (ID 0) и serveradmin (ID 1);
- две группы по умолчанию: oneadmin и users.

35.1.1. Управление пользователями в консоли

Утилита **oneuser** предназначена для управления пользователями через командную строку.

Просмотр 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 <имя_пользователя> <пароль>
```

Примечание

В именах и паролях недопустимы следующие символы:

- Имя: пробел, :, табуляция, переводы строк;
- Пароль: пробел, табуляция, переводы строк.

Примечание

При использовании некоторых драйверов аутентификации (например, LDAP) пароль не требуется. В этом случае пользователя можно создать без него:

```
$ oneuser create --driver ldap <имя_пользователя>
```

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

```
$ oneuser chgrp <имя_пользователя> oneadmin
```

Удаление из группы выполняется путём перевода пользователя обратно в группу users.

Управление статусом пользователя:

```
$ oneuser disable <имя_пользователя> # временно отключить
$ oneuser enable <имя_пользователя> # включить
$ oneuser delete <имя_пользователя> # удалить
```

Примечание

Отключённые пользователи не могут входить в Sunstone и выполнять какие-либо действия в системе.

35.1.2. Аутентификация пользователей

35.1.2.1. Веб-интерфейс Sunstone

Для аутентификации в веб-интерфейсе Sunstone достаточно указать имя пользователя и пароль на странице входа.

35.1.2.2. CLI и API

Для работы с OpenNebula через командную строку (CLI) или любой API-клиент необходимо предоставить валидные учётные данные — пароль или токен аутентификации — для формирования строки сессии.

По умолчанию используется драйвер аутентификации core, реализующий простую проверку по связке пользователь–пароль.

Чтобы CLI-инструменты могли автоматически использовать учётные данные, их необходимо разместить в стандартном файле **`$HOME/.one/one_auth`**. Формат файла — одна строка вида:

```
<имя_пользователя>:<пароль>
```

После настройки пользователь сможет использовать CLI и API:

```
$ oneuser show
USER 8 INFORMATION
ID              : 8
NAME            : user
GROUP           : users
PASSWORD        :
a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3
```

Примечание

OpenNebula не хранит пароли в открытом виде. В базе данных сохраняется только хеш.

35.1.2.3. Токены аутентификации

Файл **$HOME/.one/one_auth** защищён только механизмами прав файловой системы. Для повышения безопасности рекомендуется использовать токены аутентификации вместо хранения паролей в открытом виде.

Токен — это временный идентификатор, который:

- генерируется OpenNebula;
- имеет ограниченное время жизни (по умолчанию — 10 часов);
- может быть привязан к конкретной группе, если пользователь состоит в нескольких группах.

При использовании группового токена:

- пользователь видит только ресурсы, принадлежащие этой группе;
- все новые ресурсы создаются в рамках указанной группы.

Любой пользователь может создать токен с помощью команды:

```
$ oneuser token-create
  * Do you want to overwrite the file /home/user/.one/one_auth? (Y|N):
Y
overwriting /home/user/.one/one_auth ...
Authentication Token is:
user:4b426939a0f030cc5edca60088c54fe38325e41fdd2fba8e51fc655978167e04
```

При создании токена он автоматически записывается в файл **~/.one/one_auth** и начинает использоваться CLI.

Дополнительные параметры:

- ***--time <секунды>*** — задать время жизни токена (значение -1 — бессрочный токен);
- ***--group <ID|имя>*** — привязать токен к указанной группе.

Список токенов отображается в выводе команды **oneuser show**:

```
TOKENS
     ID EGID  EGROUP     EXPIRATION
4b42693 *1    users      2025-12-25 20:14:18
985a500 1     users      2025-12-25 20:30:40
```

Примечание

Звёздочка (*) в колонке EGID означает, что токен не привязан к конкретной группе, а использует основную группу пользователя.

Чтобы использовать уже существующий токен (например, созданный администратором или полученный извне), его необходимо указать CLI:

```
$ oneuser token-set --token 985a500
export ONE_AUTH=/home/user/.one/aac6aa2e9bf92fc95d8ec0068c16b980.token;
export ONE_EGID=1
```

Команда выводит переменные окружения, которые можно экспортировать для текущей сессии.

Токен можно удалить, выполнив команду:

```
$ oneuser token-delete 985a500
Token removed.
Removing /home/user/.one/aac6aa2e9bf92fc95d8ec0068c16b980.token
```

После удаления токен становится недействительным и больше не может использоваться для аутентификации.

35.1.3. Шаблон пользователя

В разделе ***USER TEMPLATE*** можно хранить произвольные атрибуты, например:

```
DEPARTMENT=IT
EMAIL=user@test.alt
```

Эти данные могут использоваться в контекстуализации виртуальных машин, например:

```
ssh_key = "$USER[SSH_KEY]"
```

Шаблон также позволяет настраивать права операций над ВМ:

```
VM_USE_OPERATIONS
VM_MANAGE_OPERATIONS
VM_ADMIN_OPERATIONS
```

35.1.4. Управление собственной учётной записью

Пользователи могут:

- просматривать информацию о своей учётной записи;

- изменять пароль.

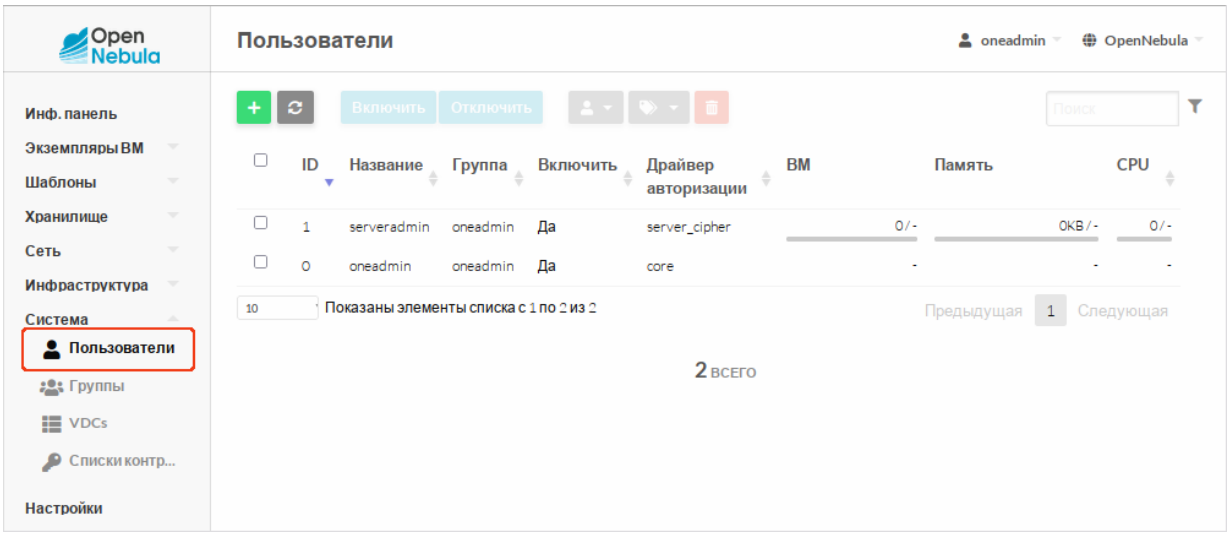
Пример для пользователя user:

```
$ oneuser list
ID NAME            ENAB GROUP      AUTH            VMS
MEMORY         CPU
8 user             yes  users      core       0 /   -      0M /       -
0.0 /   -

$ oneuser passwd 8 <новый_пароль>
```

В веб-интерфейсе Sunstone информация об учётной записи доступна в разделе **Настройки**.

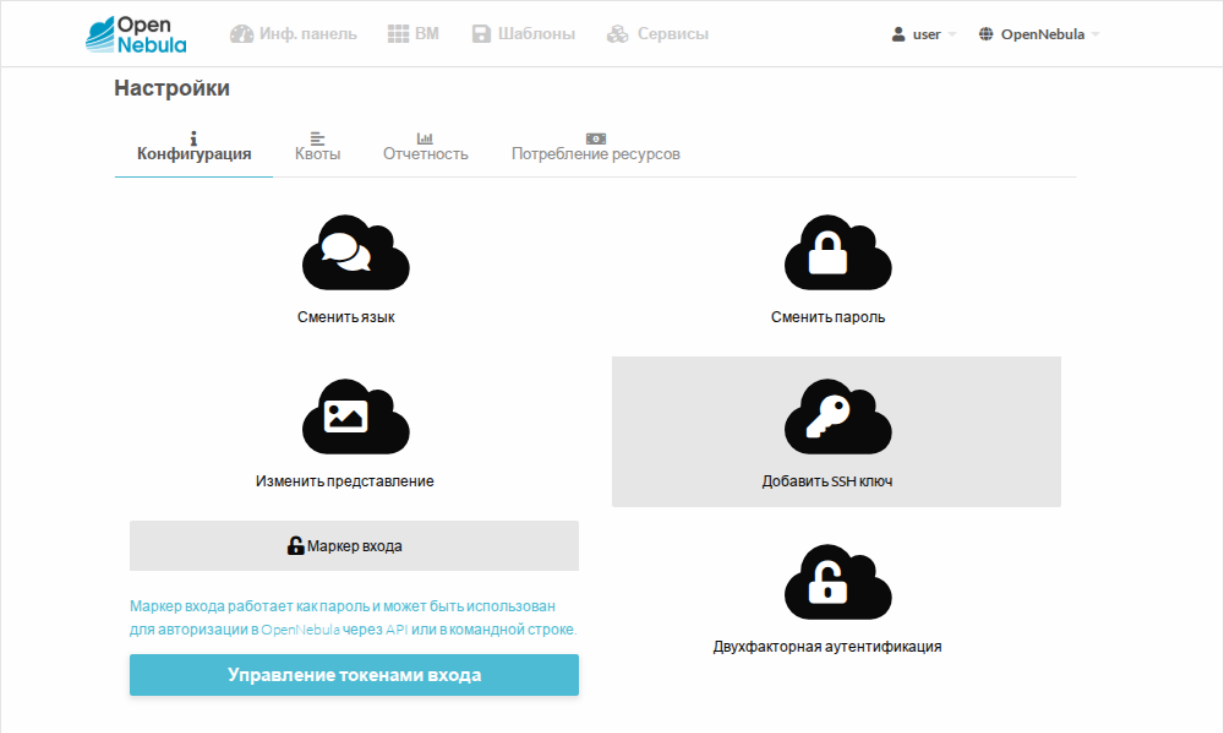
35.1.5. Управление пользователями в веб-интерфейсе

Все операции по управлению пользователями доступны в веб-интерфейсе Sunstone.



После входа пользователь может:

- изменить пароль;
- выбрать язык интерфейса;
- управлять токенами входа;
- добавить SSH-ключ для доступа к виртуальным машинам.



Примечание

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

35.1.6. Изменение учётных данных администраторов

Для изменения учётных данных oneadmin необходимо на сервере управления выполнить следующие действия:

```
$ oneuser passwd 0 <пароль>
$ echo 'oneadmin:<пароль>' > /var/lib/one/.one/one_auth
```

После изменения пароля следует перезапустить службу OpenNebula.

Примечание

Изменение пароля serveradmin не рекомендуется, так как он используется только для внутренней коммуникации между компонентами.

Для изменения учетных данных serveradmin необходимо на сервере управления выполнить следующие действия:

```
$ oneuser passwd 1 --sha256 0 <пароль>
$ echo 'serveradmin:<пароль>' > /var/lib/one/.one/oneflow_auth
$ echo 'serveradmin:<пароль>' > /var/lib/one/.one/onegate_auth
$ echo 'serveradmin:<пароль>' > /var/lib/one/.one/sunstone_auth
```

После изменения пароля следует перезапустить службу Sunstone.

35.2. Группы пользователей

В OpenNebula группы используются для логического разделения пользователей и ресурсов.

По умолчанию пользователь видит только ресурсы, принадлежащие его группе. Доступ к ресурсам других групп возможен только при явном предоставлении прав через ACL или при использовании виртуальных центров обработки данных (VDC).

Группа определяет границу видимости и управления ресурсами. Более гибкое разделение инфраструктуры можно реализовать с помощью VDC, где каждой группе назначаются отдельные ресурсы (хосты, хранилища, сети).

При установке OpenNebula автоматически создаются две группы:

- oneadmin — основная группа суперпользователя oneadmin (ID 0);
- users — группа по умолчанию, в которую попадают все новые пользователи.

35.2.1. Управление группами в консоли

Для управления группами используется утилита **onegroup**.

Просмотр списка групп:

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

Команда создания новой группы:

```
$ onegroup create <имя_группы>
```

Например:

```
$ onegroup create ALT
ID: 100
```

Созданной группе был присвоен идентификатор 100, что позволяет отличать пользовательские группы от предопределённых (служебных) групп OpenNebula.

При создании группы автоматически настраивается базовое правило ACL, разрешающее пользователям группы создавать стандартные ресурсы (ВМ, образы, шаблоны).

Для назначения пользователей группам используется команда **oneuser chgrp**:

```
$ oneuser chgrp -v oneuser ALT
```

Удаление из группы выполняется переводом пользователя обратно в группу users.

При создании группы можно указать специального администратора, который получит расширенные права только в рамках этой группы (в отличие от глобального администратора oneadmin).

Параметры команды **`onegroup create`**:

- ***`-n`***, ***`--name`*** — имя группы;
- ***`-u`***, ***`--admin_user`*** — создать администратора группы;
- ***`-p`***, ***`--admin_password`*** — пароль администратора;
- ***`-d`***, ***`--admin_driver`*** — драйвер аутентификации (например, ldap);
- ***`-r`***, ***`--resources`*** — типы ресурсов, которые могут создавать пользователи группы (по умолчанию VM+IMAGE+TEMPLATE).

Пример:

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

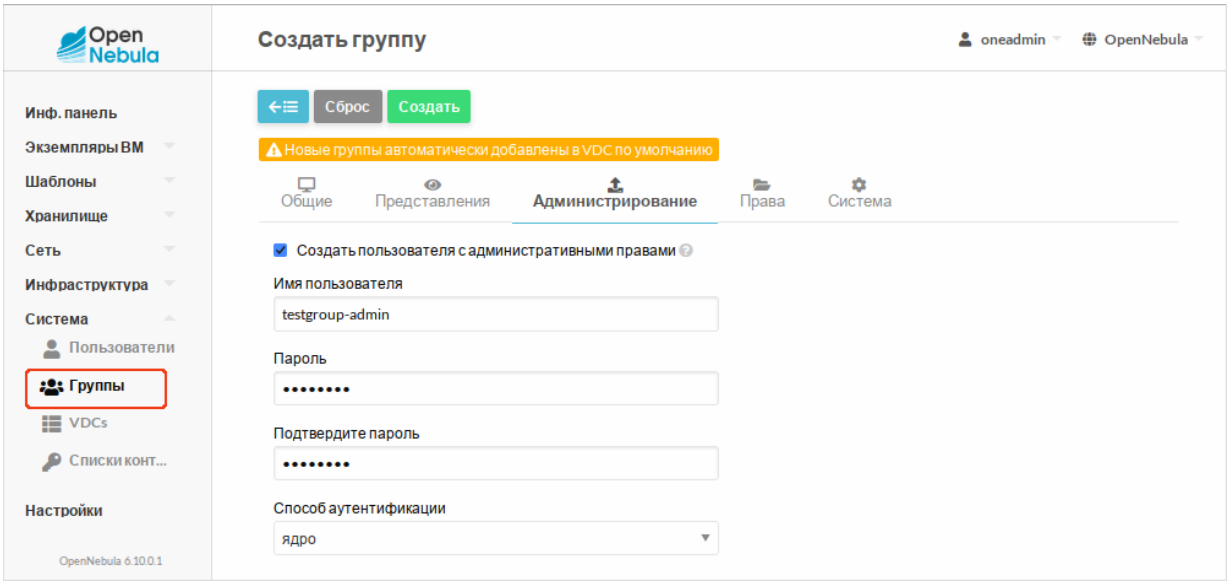
Эта команда создаст группу testgroup и пользователя testgroup-admin с правами администратора этой группы.

Любого существующего пользователя можно назначить администратором группы:

```
$ onegroup addadmin <ID_группы> <ID_пользователя>
```

35.2.2. Управление группами в веб-интерфейсе

Все операции по управлению группами доступны в веб-интерфейсе Sunstone:



35.3. Управление разрешениями

Большинство ресурсов OpenNebula (шаблоны, виртуальные машины, образы, виртуальные сети) имеют три уровня прав доступа:

- Владелец (OWNER)
- Группа (GROUP)

- Остальные (OTHER)

Для каждого уровня можно установить три типа прав:

- USE (u) — просмотр и использование ресурса (например, запуск ВМ, подключение к сети);
- MANAGEE (m) — изменение ресурса (остановка, изменение атрибутов, удаление);
- ADMIN (a) — административные действия (например, изменение владельца, обход ограничений).

Примечание

Пользователи, группы и узлы не поддерживают права USE/MANAGE/ADMIN.

35.3.1. Управление разрешениями в командной строке

Просмотр прав на ВМ с ID=8:

```
$ 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:

- владелец (oneadmin) имеет права USE и MANAGE;
- пользователи из той же группы и все остальные — не имеют доступа.

Права можно изменить с помощью команды **chmod** в восьмеричной нотации:

```
$ onevm chmod 8 607
```

Где:

- первая цифра — права владельца;
- вторая — группы;
- третья — остальных.

Каждая цифра — сумма:

- 4 = USE
- 2 = MANAGE
- 1 = ADMIN

В примере: 607 → владелец: 4+2=6 (um-), группа: 0 (---), остальные: 4+2+1=7 (uma).

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

При создании ресурса применяются права, заданные:

- глобально в **`/etc/one/oned.conf`** через **`DEFAULT_UMASK`**;
- индивидуально для каждого пользователя с помощью команды **`oneuser umask`**.

Также поведение зависит от параметра:

```
ENABLE_OTHER_PERMISSIONS = "YES"  # или "NO"
```

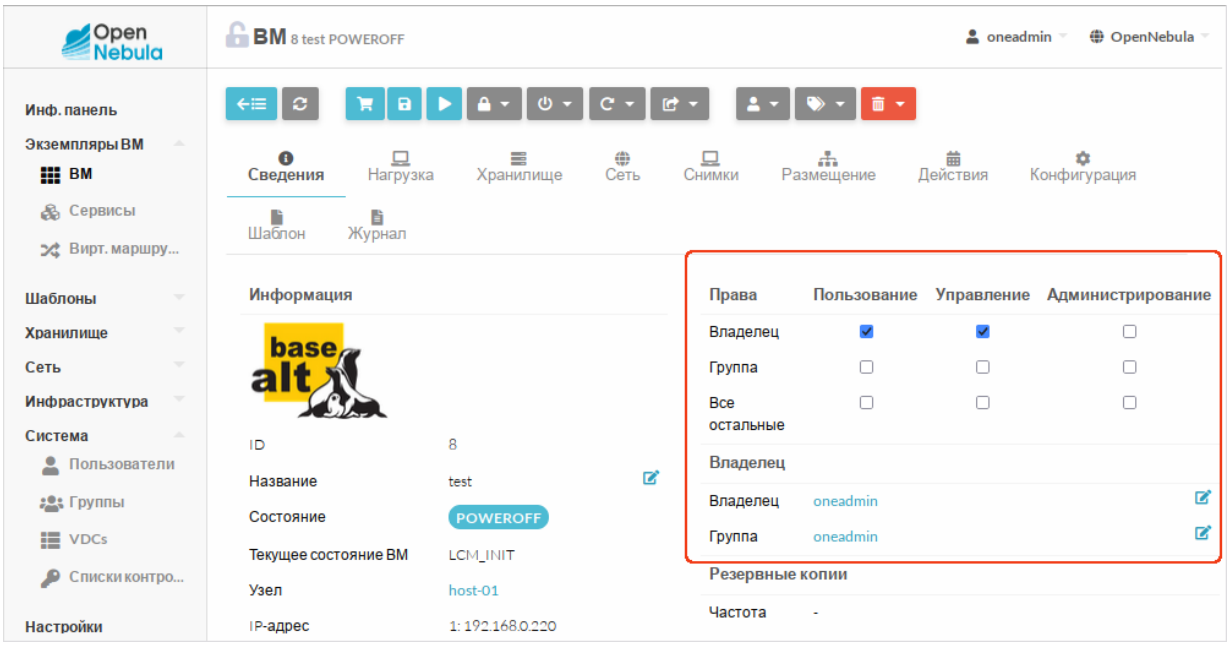
- **`YES`** — новые ресурсы получают права 666 (все могут использовать);
- **`NO`** — права 660 (только владелец и группа).

Примечание

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-- ---).

35.3.2. Управление разрешениями в веб-интерфейсе

В веб-интерфейсе Sunstone права можно настроить на вкладке **Сведения** при просмотре ресурса:



35.3.3. Блокировка ресурсов

OpenNebula позволяет заблокировать ресурс, чтобы предотвратить нежелательные операции (например, удаление).

Уровни блокировки:

- USE (или ALL) — запрещает все операции;
- MANAGE — запрещает изменение и администрирование;
- ADMIN — запрещает только административные действия.

Блокировать можно следующие ресурсы:

- VM
- NET
- IMAGE
- TEMPLATE
- DOCUMENT
- VROUTER
- MARKETPLACEAPP
- HOOK
- VMGROUP
- VNTEMPLATE

Пример:

```
$ oneimage lock 2
$ oneimage delete 2
[one.image.delete] User [4] : Not authorized to perform MANAGE IMAGE [2].
$ oneimage unlock 2
```

Примечание

Разблокировать ресурс может только владелец или пользователь oneadmin.

35.4. Управление правилами ACL

Система ACL (Access Control List) позволяет гибко настраивать права для любого пользователя или группы на любые операции с ресурсами.

Каждое правило имеет формат:

```
<пользователь> <ресурс> <операции> [<зона>]
```

Синтаксис:

- пользователь:
  - #<ID> — конкретный пользователь;
  - @<ID> — группа пользователей;
  - * — все пользователи;
- ресурс:
  - типы ресурсов: VM, HOST, NET, IMAGE, TEMPLATE, и т. д.;
  - идентификатор: #<ID> (объект), @<ID> (группа), %<ID> (кластер), * (все);
  - * — все;
- операции:
  - USE, MANAGE, ADMIN, CREATE.

Пример правила:

```
#5 IMAGE+TEMPLATE/@103 USE+MANAGE
```

Пользователь #5 может использовать и управлять всеми образами и шаблонами, принадлежащими группе @103.

Примечание

Разница между *NET/#47USE" и *NET/@47USE:

- #47 — сеть с ID 47;
- @47 — все сети, принадлежащие группе 47.

Команды управления правилами:

```
$ oneacl list # просмотр списка правил
$ oneacl create "#3 IMAGE/#30 USE" # создание правила
$ oneacl delete <ID> # удаление правила
```

Просмотр списка правил:

```
$ oneacl list
  ID     USER RES_VHNIUTGDCOZSvRMAPt   RID OPE_UMAC  ZONE
   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---    #0
```

Эти правила соответствуют следующим описаниям:

```
@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 создаются автоматически при инициализации OpenNebula. Последнее правило было добавлено вручную командой:

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

Расшифровка столбцов в выводе **`oneacl list`**:

- ID — идентификатор правила.
- USER — субъект, к которому применяется правило:
  - #<ID> — конкретный пользователь;
  - @<ID> — группа пользователей;
  - * — все пользователи.

- Resources — типы ресурсов, на которые распространяется правило. Каждая позиция в строке соответствует определённому типу ресурса:
  - V — ВМ;
  - H — узел (хост);
  - N — виртуальная сеть;
  - I — образ;
  - U — пользователь;
  - T — шаблон;
  - G — группа;
  - D — хранилище данных;
  - C — кластер;
  - O — документ;
  - Z — зона;
  - S — группа безопасности;
  - v — виртуальный ЦОД (VDC);
  - R — виртуальный маршрутизатор;
  - M — магазин приложений;
  - A — приложение из магазина;
  - P — группа ВМ;
  - t — шаблон виртуальной сети;
  - B — задание резервного копирования.

  Наличие символа (V, H, N и т. д.) означает, что правило применяется к данному типу ресурсов. Прочерк (-) — отсутствие ограничений для этого типа.
- RID — идентификатор ресурса:
  - #<ID> — конкретный ресурс;
  - @<ID> — ресурсы, принадлежащие группе;
  - %<ID> — ресурсы в кластере;
  - * — все ресурсы указанного типа.
- Operations — разрешённые операции:
  - U — USE (использование);
  - M — MANAGE (управление);

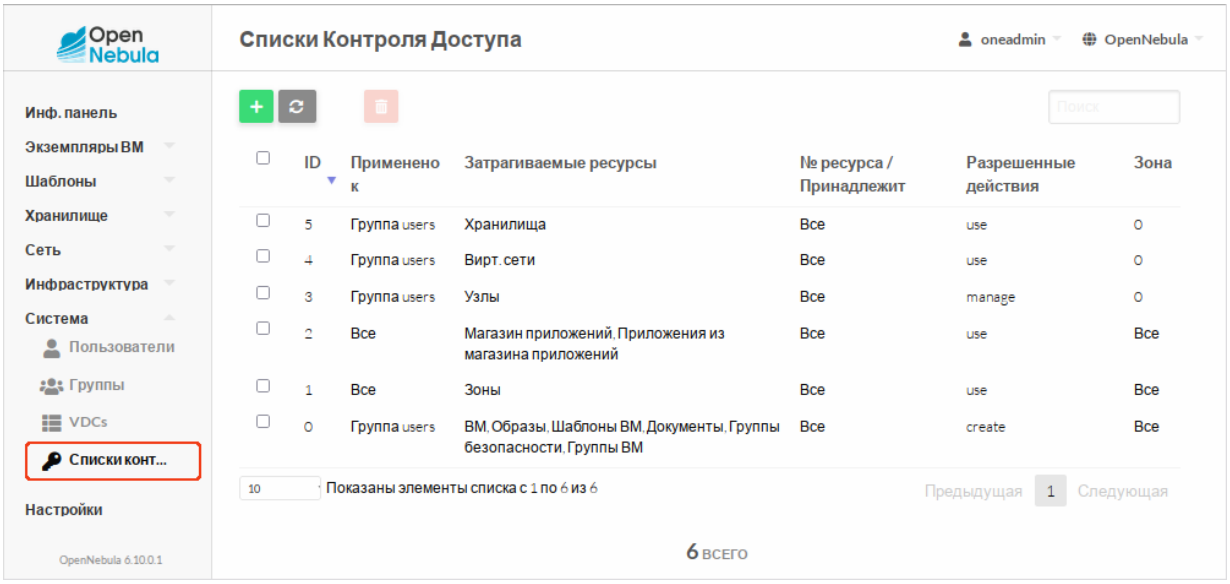
- A — ADMIN (администрирование);
- C — CREATE (создание).

- Zone — зона, в которой действует правило:
  - #<ID> — конкретная зона;
  - * — все зоны.

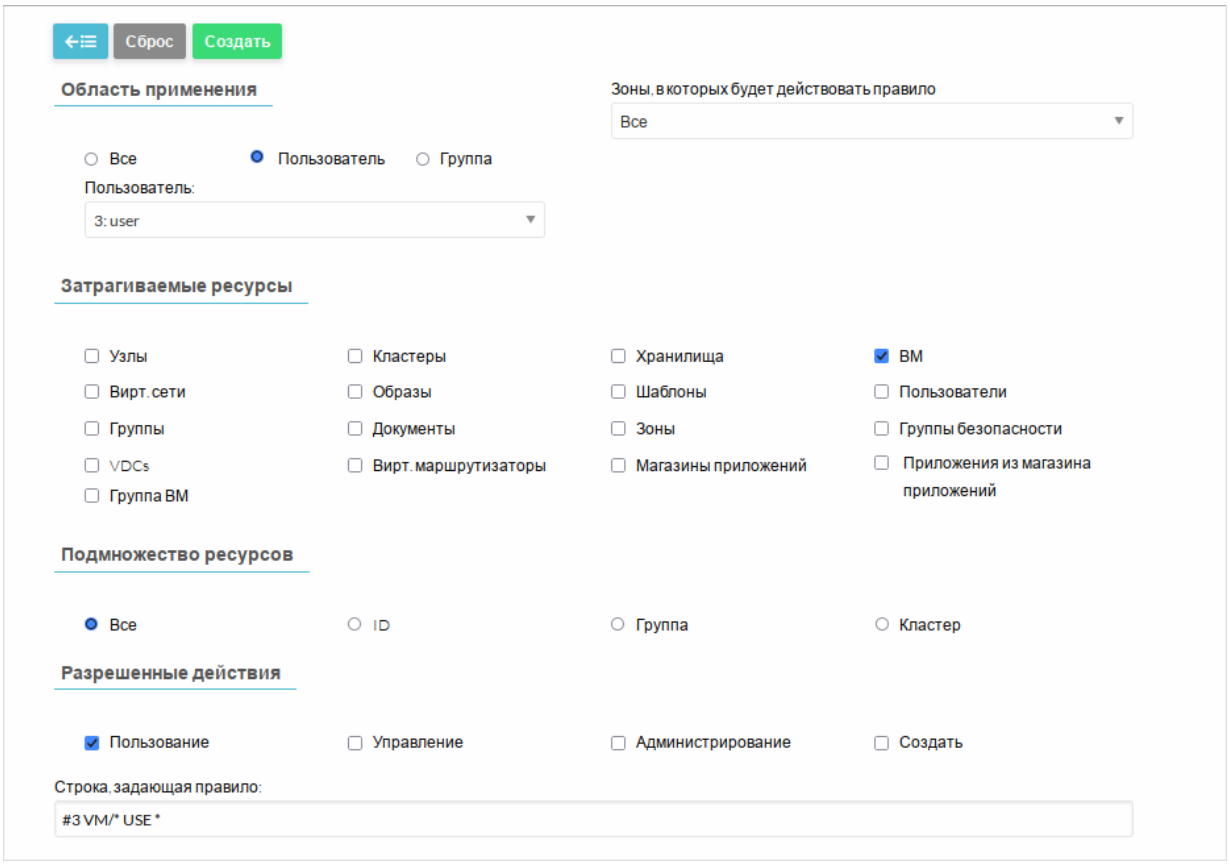
Важные особенности ACL:

- Правила только расширяют права — если хотя бы одно правило разрешает операцию, она будет выполнена.
- ACL не может ограничить права, предоставленные через стандартные разрешения (chmod).

В веб-интерфейсе можно создавать, редактировать и удалять ACL-правила:



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:



35.5. Настройка аутентификации

По умолчанию OpenNebula использует встроенную систему аутентификации на основе пары «пользователь/пароль». Учётные данные пользователей хранятся в базе данных OpenNebula.

Для интеграции с внешними системами (например, LDAP) могут быть подключены специализированные драйверы аутентификации. При этом допускается одновременное использование нескольких методов аутентификации, а для каждого пользователя можно указать свой способ входа.

Следует учитывать, что различные методы аутентификации поддерживают разные интерфейсы доступа (CLI, API, Sunstone).

Таблица 35.1. Интерфейсы и методы аутентификации

Метод аутентификации	Поддерживаемые интерфейсы
Встроенная (пользователь/пароль, токены)	API, CLI, Sunstone
LDAP	API, CLI, Sunstone
SSH	API, CLI
X.509	CLI, Sunstone
Аутентификация через Sunstone (remote)	Только Sunstone

Примечание

Аутентификация через Sunstone (remote-аутентификация) используется преимущественно при интеграции с внешними системами идентификации и требует дополнительной настройки веб-сервера (например, Apache или Nginx), через который осуществляется доступ к Sunstone.

35.5.1. Аутентификация по SSH-ключам

Аутентификация по SSH-ключам позволяет пользователям взаимодействовать с OpenNebula через интерфейс командной строки (CLI) и API, используя стандартные SSH-ключи вместо паролей. При этом аутентификационный токен шифруется закрытым SSH-ключом пользователя, что повышает уровень безопасности и устраняет необходимость хранения паролей в открытом виде.

Важно

Данный метод аутентификации работает только для CLI и API и не применяется к веб-интерфейсу.

Примечание

Поддерживаются только RSA-ключи (формат id_rsa).

Механизм аутентификации по SSH включён в OpenNebula по умолчанию (см. раздел AUTH_MAD в файле **`/etc/one/oned.conf`**):

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

35.5.1.1. Генерация SSH-ключа пользователем

Пользователь генерирует пару SSH-ключей:

```
$ ssh-keygen -t rsa
```

Рекомендации:

- не следует изменять путь по умолчанию (**`$HOME/.ssh/id_rsa`**), так как OpenNebula ищет ключ именно там;
- закрытый ключ рекомендуется защитить парольной фразой (passphrase).

35.5.1.2. Создание нового пользователя

Создание нового пользователя:

1. Пользователь генерирует публичную часть ключа для передачи администратору:

   ```
   $ oneuser key
   Enter old passphrase:
   MIIBCAKCAQEApUO+JISjSf02rFVtDr1yar/
   34EoUoVETx0n+RqWNav+5wi+gHiPp3e03AfEkXzjDYi8F
   ...
   ```

   Полученную строку необходимо передать администратору облака.

2. Администратор:

   - сохраняет полученную строку в файл (например, **`/tmp/pub_key`**);
   - создаёт пользователя, указав публичный SSH-ключ:

     ```
     $ oneuser create newuser --ssh --read-file /tmp/pub_key
     ```

     альтернативный вариант (не рекомендуется с точки зрения безопасности) — если у администратора есть доступ к закрытому ключу пользователя:

     ```
     $ oneuser create newuser --ssh --key /home/newuser/.ssh/id_rsa
     ```

35.5.1.3. Перевод существующего пользователя на SSH-аутентификацию

Для существующего пользователя можно изменить метод аутентификации и задать публичный ключ:

```
$ oneuser chauth newuser ssh
$ oneuser passwd newuser --ssh --read-file /tmp/pub_key
```

35.5.1.4. Генерация аутентификационного токена

Перед использованием CLI пользователь должен сгенерировать аутентификационный токен:

```
$ oneuser login newuser --ssh
Enter old passphrase:
* Do you want to overwrite the file /home/newuser/.one/one_auth? (Y|N):
y
overwriting /home/user/.one/one_auth ...
Authentication Token is:
newuser:062ca55c97e0e327a50d1027f59cdedc68495cb836ddeebc68d1ffd4eb293cab
```

Команда **`oneuser login`**:

- использует закрытый ключ **`$HOME/.ssh/id_rsa`** (если не указан иной путь);
- сохраняет токен в файл, заданный переменной окружения ONE_AUTH (по умолчанию — **`$HOME/.one/one_auth`**);
- устанавливает время жизни токена — 10 часов.

Чтобы указать другой SSH-ключ или изменить срок действия токена, можно использовать параметры:

```
$ oneuser login newuser --ssh --key /path/to/id_rsa --time 3600
```

Примечание

Если токен утерян или скомпрометирован, достаточно дождаться окончания срока его действия либо сменить SSH-ключ и сгенерировать новый токен.

35.5.2. LDAP аутентификация

Аутентификация LDAP позволяет пользователям использовать те же учётные данные, что и в LDAP, обеспечивая централизованное управление доступом. Любой пользователь, успешно прошедший аутентификацию в LDAP, сможет использовать OpenNebula (при условии наличия соответствующих прав).

35.5.2.1. Включение LDAP

Драйвер LDAP включён по умолчанию в конфигурации OpenNebula (см. раздел AUTH_MAD в файле **/etc/one/oned.conf**):

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

Для автоматического создания пользователей при первом входе в LDAP необходимо:

1. Добавить метод **default** в атрибут **AUTHN**:

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

2. Создать символическую ссылку:

   ```
   # ln -s /var/lib/one/remotes/auth/ldap /var/lib/one/remotes/auth/default
   ```

Важно

В файле **/etc/one/sunstone-server.conf** должно быть указано:

```
:auth: opennebula
```

35.5.2.2. Конфигурация LDAP

Основной файл настройки — **`/etc/one/auth/ldap_auth.conf`**. Формат: YAML-хеш, где:

- каждый ключ (кроме **`:order`**) описывает один LDAP-сервер;
- **`:order`** — массив с порядком опроса серверов.

Примечание

Элементы в **`:order`** обрабатываются по порядку, пока пользователь не будет успешно аутентифицирован или не будет достигнут конец списка. Серверы, не указанные в **`:order`**, использоваться не будут.

Таблица 35.2. Параметры файла /etc/one/auth/ldap_auth.conf

Параметр	Описание
`:user`	Имя пользователя, имеющего право выполнять запросы к LDAP. Не требуется при анонимном доступе
`:password`	Пароль для пользователя, определенного в ***`:user`***. Не требуется при анонимном доступе
`:auth_method`	Поддерживается только **`:simple`**
`:encryption`	Можно установить значение **`:simple_tls`**, если требуется SSL-соединение
`:host`	LDAP-сервер
`:port`	Порт LDAP-сервера (обычно 389 или 636)
`:timeout`	Тайм-аут подключения и аутентификации (в секундах)
`:base`	Базовый DN для поиска пользователей
`:group_base`	Альтернативный DN для поиска групп
`:group`	Если задано, только пользователи из этой группы могут входить
`:user_field`	Атрибут LDAP, содержащий логин (например, uid, sAMAccountName)
`:group_field`	Атрибут членства в группе (по умолчанию member). Для вложенных групп в AD можно использовать: member:1.2.840.113556.1.4.1941
`:mapping_generate`	Автоматическая генерация файла сопоставления групп
`:mapping_timeout`	Интервал обновления файла сопоставления (в секундах)
`:mapping_filename`	Имя файла сопоставления (должно быть уникальным для каждого сервера)
`:mapping_key`	Ключ в шаблоне группы, содержащий DN LDAP-группы
`:mapping_default`	Группа OpenNebula по умолчанию для пользователей без сопоставления. Установите false, чтобы запретить вход таким пользователям
`:rfc2307bis:`	true для Active Directory, false для OpenLDAP
`:group_admin_group_dn:`	

Параметр	Описание
	DN группы, члены которой становятся администраторами всех сопоставленных групп в OpenNebula

Примечание

Автоматическое назначение администраторов групп можно отключить, установив ***DRIVER_MANAGED_GROUP_ADMIN = "NO"*** в разделе ***ldap*** файла **oned.conf**.

35.5.2.3. Управление пользователями

При включённой автоматической регистрации создание пользователей вручную не требуется.

Пользователи могут хранить учётные данные в файле, указанном в переменной ONE_AUTH (по умолчанию — **$HOME/.one/one_auth**), в формате:

```
<user_dn>:ldap_password
```

где:

- <user_dn> — DN пользователя в LDAP;
- ldap_password — пароль пользователя.

Если DN или пароль содержат пробелы или специальные символы, используйте команду экранирования: **oneuser encode <user> [<password>]**. Например:

```
$ oneuser encode 'cn=First Name,dc=institution,dc=country' 'pass word'
cn=First%20Name,dc=institution,dc=country:pass%20word
```

Результат этой команды следует поместить в файл $ONE_AUTH.

Для повышения безопасности следует использовать токены аутентификации (см. Токены аутентификации), чтобы не хранить пароль в открытом виде.

35.5.2.4. Обновление существующих пользователей до LDAP

Перевести существующего пользователя на аутентификацию через LDAP можно, выполнив команду:

```
$ oneuser chauth <id|name> ldap
```

35.5.2.5. Примеры настройки

Для подключения к Active Directory нужны:

- сервер AD с поддержкой простой аутентификации по имени пользователя и паролю;
- пользователь с правами на чтение в дереве пользователей Active Directory.

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

- **:user** — пользователь AD с правами на чтение в дереве пользователей (пользователь указывается в формате opennebula@test.alt);
- **:password** — пароль пользователя;
- **:host** — IP-адрес или имя сервера AD (имя должно разрешаться через DNS или **/etc/hosts**);
- **: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** для настройки аутентификации в домене AD (домен test.alt):

```
 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

Если доступ к серверу LDAP осуществляется по протоколу ldap без шифрования, на контроллере домена SambaDC необходимо отключить ldaps в **/etc/samba/smb.conf** в секции [global]:

```
ldap server require strong auth = no
```

и перезапустить samba:

```
# systemctl restart samba
```

Примечание

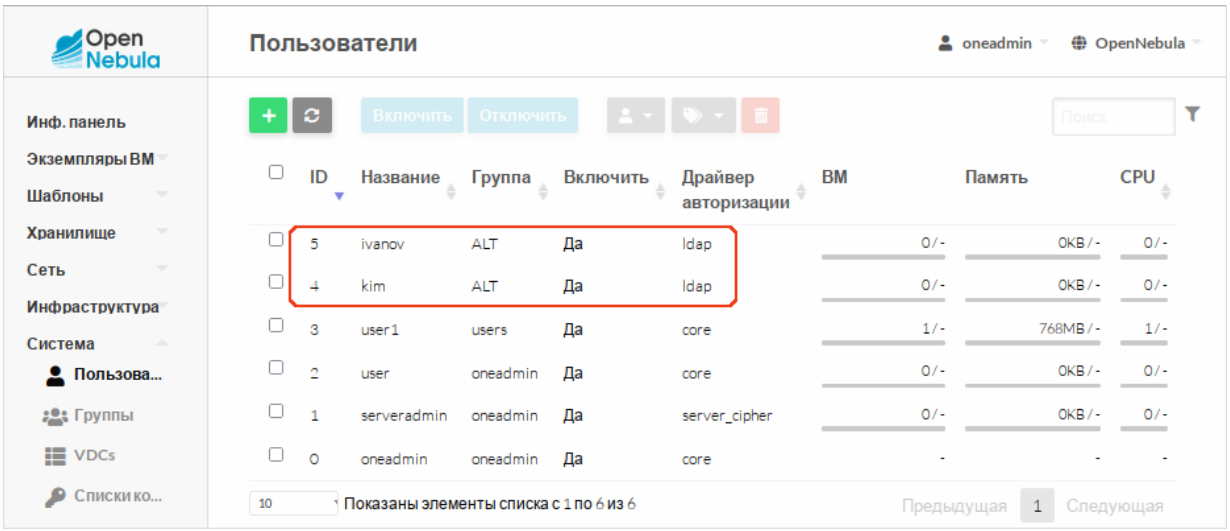
Группа, указанная в параметре ***:mapping_default*** должна существовать в OpenNebula.

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 успешной аутентификации пользователь AD отображается в интерфейсе Sunstone:



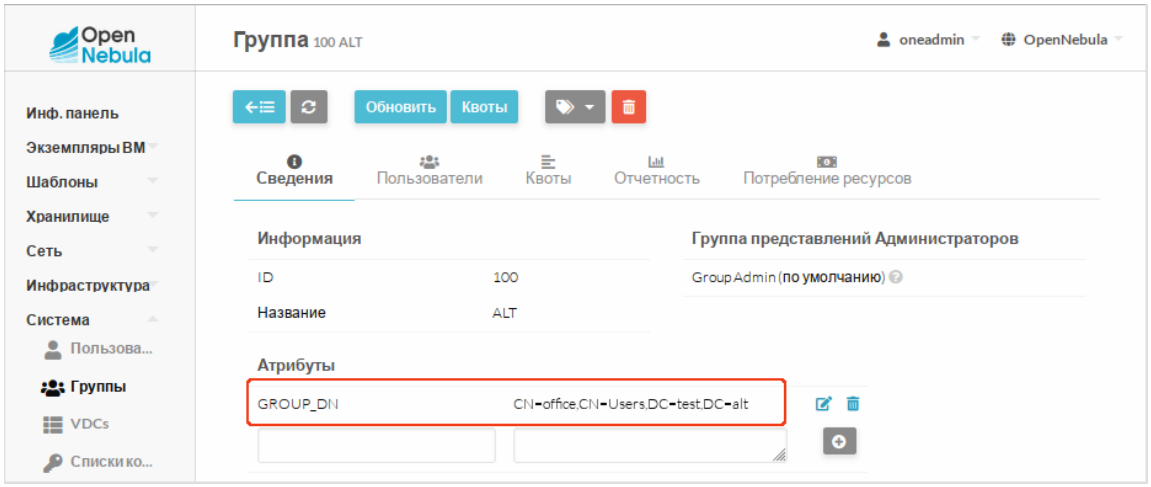
35.5.2.6. Сопоставление групп

Пользователи LDAP могут автоматически включаться в группы OpenNebula на основе их членства в LDAP-группах.

Автоматическая генерация файла сопоставления:

1. В шаблоне группы OpenNebula добавить строку:

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



2. В файле **`/etc/one/auth/ldap_auth.conf`** указать:

```
:mapping_generate: true
:mapping_key: GROUP_DN
```

   Поиск DN сопоставляемой группы будет осуществляться в этом параметре шаблона.

3. При следующей аутентификации будет создан файл **`/var/lib/one/server1.yaml`**:

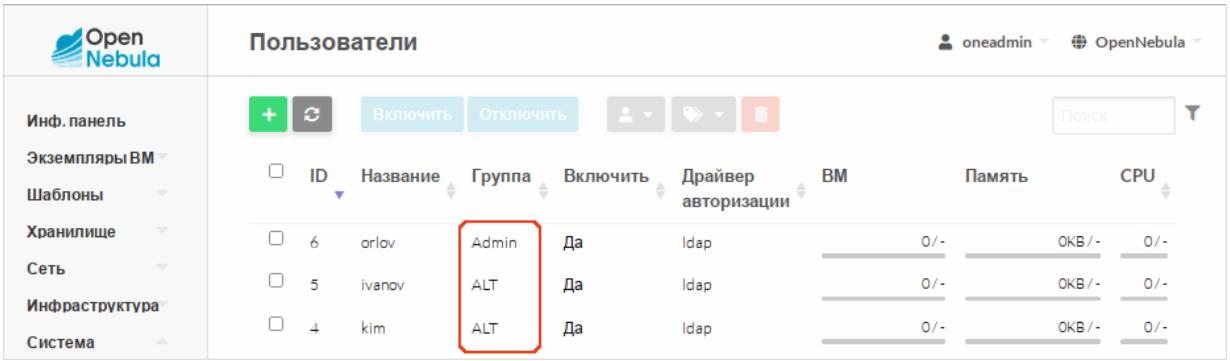
```
---
CN=office,CN=Users,DC=test,DC=alt: '100'
```

   Пользователи группы AD office будут сопоставлены с группой ALT (ID=100).

Можно отключить автоматическую генерацию файла сопоставления, установив значение **`:mapping_generate: false`** и создав файл **`/var/lib/one/server1.yaml`**:

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

Пользователи из office попадут в группу с ID 100, из Domain Admins — в ID 101.



Примечание

При использовании нескольких LDAP-серверов следует использовать уникальные ***:mapping_key*** и ***:mapping_file***.

Группы пользователя обновляются при каждой аутентификации. Если используется токен, обновление произойдёт только после его истечения Максимальный срок действия токена настраивается в **oned.conf**.

Чтобы отключить автоматическое управление группами, необходимо установить в **oned.conf**:

```
AUTH_MAD_CONF = [
  ldap = [
    DRIVER_MANAGED_GROUPS = "NO"
  ]
]
```

35.5.3. Аутентификация в Sunstone

По умолчанию веб-интерфейс Sunstone использует стандартный механизм аутентификации OpenNebula — логин и пароль пользователя, хранящиеся в базе данных OpenNebula.

Sunstone поддерживает все методы аутентификации, доступные в OpenNebula. Однако все пользователи, входящие через Sunstone, должны существовать в базе данных OpenNebula — даже при использовании внешних систем (LDAP, Kerberos и др.).

Процесс аутентификации в Sunstone зависит от значения параметра ***:auth:*** в файле **/etc/one/sunstone-server.conf** и может работать в двух режимах:

- **opennebula** — локальная или делегированная аутентификация через ядро OpenNebula;
- **remote** — внешняя аутентификация, выполняемая веб-сервером (Apache, Nginx и др.).

35.5.3.1. Локальная аутентификация (opennebula)

В этом режиме (используется по умолчанию) пользователь вводит логин и пароль на странице входа Sunstone. Эти данные передаются в ядро OpenNebula, которое проверяет их в соответствии с настроенным драйвером аутентификации (встроенный, LDAP, SSH и др.).

После успешного входа:

- Sunstone создаёт сессию на основе JWT (JSON Web Token);
- все последующие запросы к API аутентифицируются и авторизуются с использованием этой сессии.

Этот режим соответствует настройке:

```
:auth: opennebula
```

и используется по умолчанию.

Примечание

Этот режим используется для всех типов аутентификации через ядро OpenNebula, включая LDAP и SSH.

35.5.3.2. Внешняя аутентификация (remote)

В режиме удалённой аутентификации Sunstone не запрашивает у пользователя логин и пароль. Вместо этого:

1. Веб-сервер (Apache, Nginx и др.) выполняет аутентификацию пользователя (например, через Kerberos, OAuth, клиентские сертификаты или базовую HTTP-аутентификацию).
2. Веб-сервер передаёт имя аутентифицированного пользователя Sunstone через HTTP-заголовок (обычно HTTP_REMOTE_USER).
3. Sunstone ищет пользователя с таким именем в базе OpenNebula и авторизует его без проверки пароля.

При этом значение пароля в базе данных OpenNebula не используется как секрет, а служит лишь идентификатором пользователя.

Чтобы перевести существующего пользователя на удалённую аутентификацию, необходимо изменить драйвер аутентификации пользователя на public и обновить пароль, указав внешний идентификатор пользователя:

```
$ oneuser chauth <логин> public "<логин>"
```

В качестве идентификатора, как правило, используется имя пользователя.

Создание нового пользователя с remote-аутентификацией:

```
$ oneuser create <логин> "<логин>" --driver public
```

Для активации этого метода необходимо изменить параметр ***:auth:*** в файле конфигурации (**/etc/one/sunstone-server.conf**):

```
:auth: remote
```

После изменения конфигурации требуется перезапустить сервис Sunstone:

```
# systemctl restart opennebula-sunstone
```

После этого:

- форма входа в Sunstone не будет отображать поля имени пользователя и пароля;
- вся информация о пользователе будет получаться из внешнего механизма аутентификации.

Примечание

Вся ответственность за проверку подлинности пользователя лежит на внешнем компоненте (веб-сервере). Sunstone доверяет полученному имени пользователя без дополнительной проверки.

35.5.3.3. Аутентификация LDAP/AD

Аутентификация через LDAP или Active Directory настраивается на уровне ядра OpenNebula (см. раздел LDAP аутентификация).

В Sunstone не требуется специальной настройки: параметр ***:auth:*** должен оставаться равным **opennebula**:

```
:auth: opennebula
```

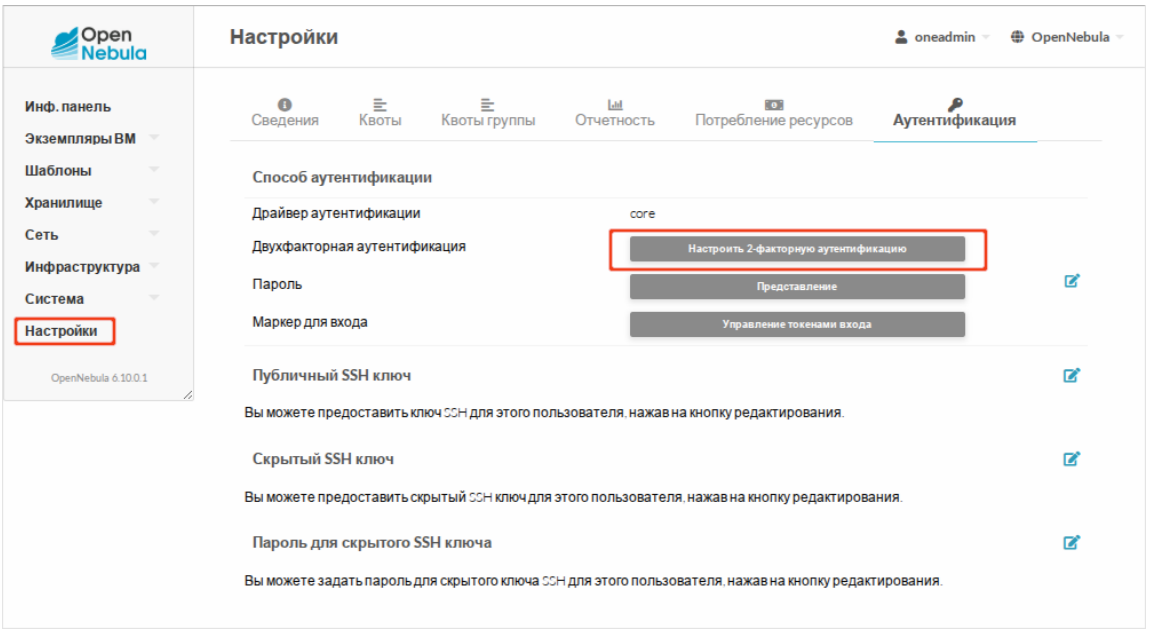
Пользователь вводит логин и пароль в стандартной форме входа, а проверка выполняется через LDAP-драйвер OpenNebula.

35.5.3.4. Двухфакторная аутентификация

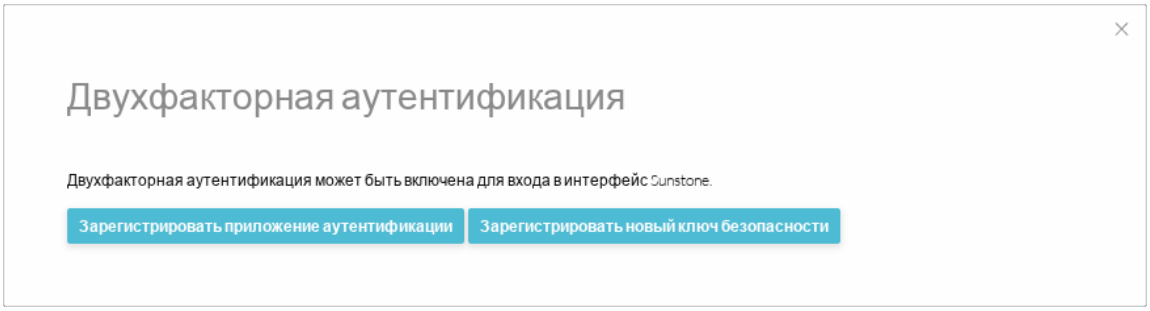
Двухфакторная аутентификация (2FA) повышает безопасность входа в систему, требуя не только логин и пароль, но и динамический одноразовый код, генерируемый приложением-аутентификатором (например, Google Authenticator, FreeOTP, Microsoft Authenticator и др.).

Включение двухфакторной аутентификации:

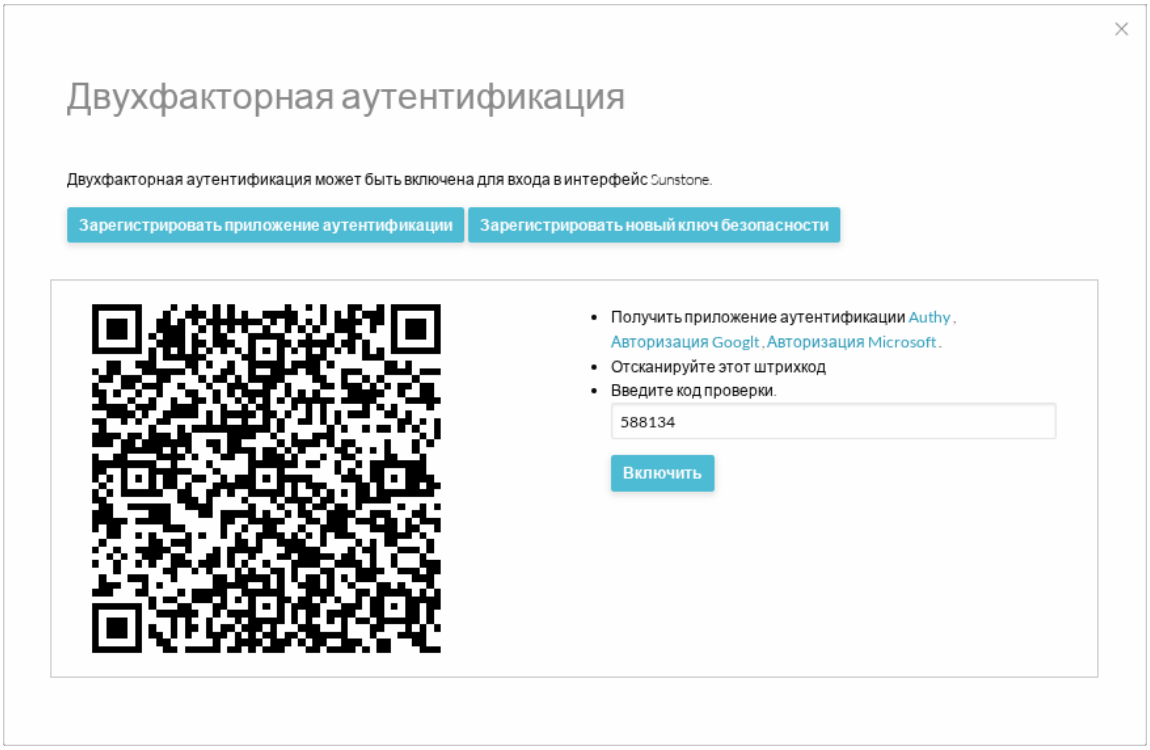
1. Войти в веб-интерфейс Sunstone.
2. Перейти в раздел **Настройки** → **Аутентификация** и нажать кнопку **Настроить 2-факторную аутентификацию**:



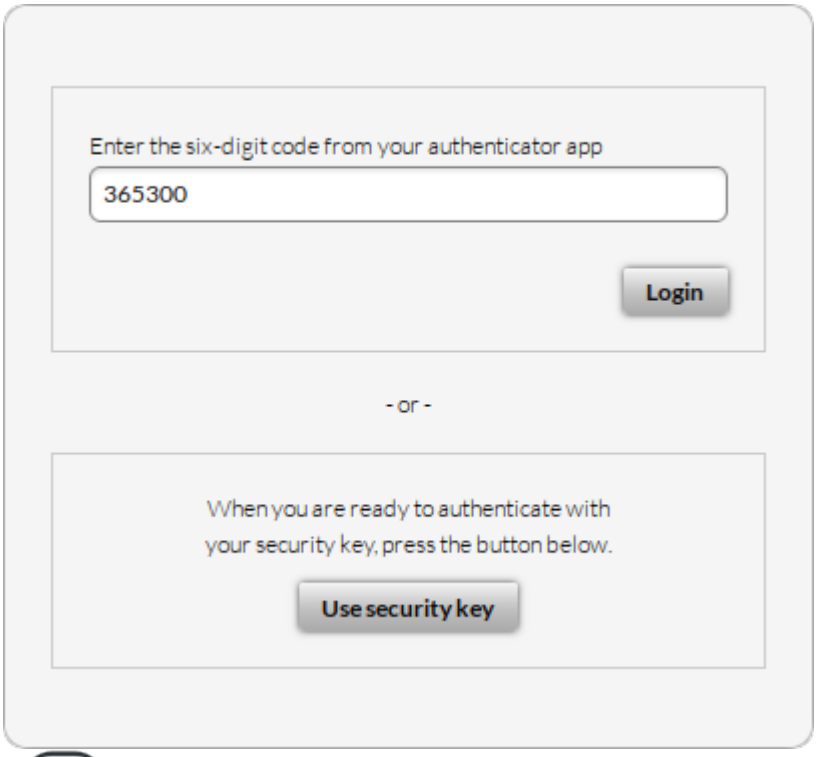
3. В открывшемся окне нажать **`Зарегистрировать приложение аутентификации`**:



4. Отсканировать QR-код с помощью приложения-аутентификатора и ввести шестизначный код подтверждения:

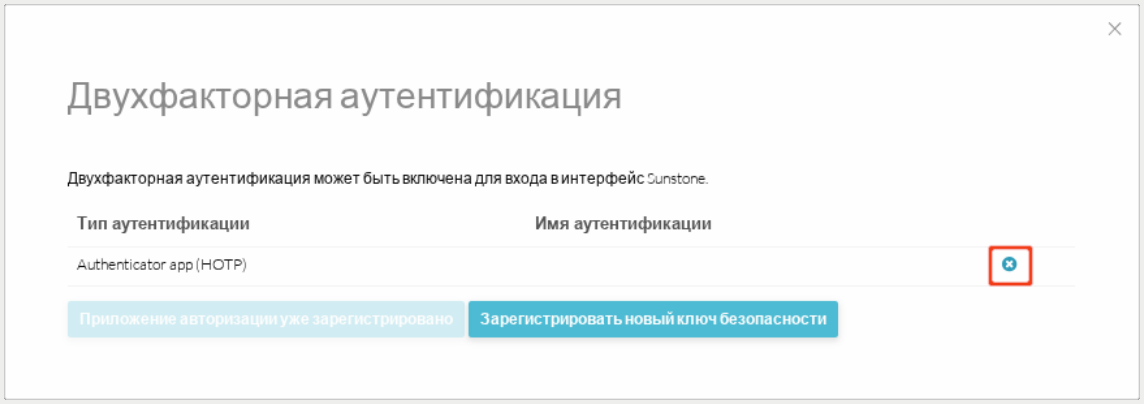


При входе в систему после ввода логина и пароля на странице входа будет запрошен шестизначный код из приложения-аутентификатора:



Примечание

Чтобы отключить двухфакторную аутентификацию, необходимо перейти в раздел **Настройки** → **Аутентификация**, нажать кнопку **Настроить 2-факторную аутентификацию** и в открывшемся окне нажать
:



Глава 36. Настройка отказоустойчивого кластера (High Availability)

OpenNebula поддерживает отказоустойчивость (HA) для ключевых компонентов:

- oned — ядро системы;
- mm_sched — планировщик.

Для обеспечения согласованности состояния между узлами используется распределённый протокол Raft.

Состояние системы (пользователи, виртуальные машины, ACL и другие объекты) хранится в базе данных. Все изменения состояния сначала записываются в журнал (log) в виде последовательности SQL-операций.

В кластере OpenNebula один сервер выбирается лидером (Leader). Все операции записи выполняются только через него. Остальные серверы работают в режиме ведомых (Follower).

Примечание

Принцип работы:

- Leader принимает изменения состояния системы;
- изменения записываются в журнал и реплицируются на большинство ведомых серверов;
- после подтверждения большинством запись фиксируется в базе данных;
- Leader регулярно отправляет Follower служебные сообщения (heartbeats) для поддержания своего статуса.

Если Leader становится недоступен, ведомые серверы автоматически инициируют выборы нового Leader. После этого кластер продолжает работу без потери согласованности данных.

Примечание

Операции чтения могут выполняться на любом узле. Операции записи, отправленные на Follower, автоматически перенаправляются на Leader.

Для настройки HA требуется:

- нечётное количество серверов (рекомендуемый размер развёртывания — 3 или 5 серверов, что обеспечивает отказоустойчивость при отказе 1 или 2 серверов соответственно);
- рекомендуется идентичная аппаратная конфигурация серверов;

- идентичная программная конфигурация серверов (за исключением ***SERVER_ID*** в **/etc/one/oned.conf**);
- рекомендуется использовать единый тип СУБД (например, MySQL) на всех узлах;
- серверы должны иметь беспарольный SSH-доступ для связи друг с другом;
- плавающий IP-адрес, который будет автоматически назначаться Leader;
- общая файловая система;
- общие хранилища данных должны быть смонтированы на всех узлах.

Добавлять дополнительные серверы или удалять старые можно после запуска кластера.

В данном примере показана настройка HA-кластера из трёх узлов:

- 192.168.0.185 — opennebula (будущий Leader)
- 192.168.0.184 — one-02
- 192.168.0.183 — one-03
- 192.168.0.200 — плавающий IP

36.1. Настройка первого узла (Leader)

Настройка первого узла:

1. Запустите OpenNebula и добавьте локальный сервер в 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://192.168.0.185:2633/RPC2

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


ZONE SERVERS
ID NAME            ENDPOINT
 0 opennebula      http://192.168.0.185:2633/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"
```

2. Остановите службу и настройте ***SERVER_ID*** в файле **/etc/one/oned.conf**:

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

3. Включите 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"
]
```

Примечание

Имя сетевого интерфейса должно соответствовать фактическому интерфейсу на узле.

4. Запустите OpenNebula и проверьте зону:

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

ZONE SERVERS
ID NAME            ENDPOINT
 0 opennebula      http://192.168.0.185:2633/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/RPC2"
```

Сервер opennebula стал Leader-сервером.

5. Убедитесь, что плавающий IP назначен:

```
$ ip -o a sh enp0s3
2: enp0s3    inet 192.168.0.185/24 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:fe6f:f3cb/64 scope link \
valid_lft forever preferred_lft forever
```

Примечание

Плавающий IP-адрес (192.168.0.200) следует использовать в:

- ENDPOINT зоны;
- параметре ***MONITOR_ADDRESS*** в **/etc/one/monitord.conf**.

36.2. Добавление дополнительных серверов

Предупреждение

Процедура полностью заменяет базу данных OpenNebula на добавляемом узле. Убедитесь, что на этом узле нет уникальных данных (например, локальных ВМ).

Примечание

Рекомендуется добавлять узлы по одному, чтобы избежать конфликтов и обеспечить стабильность кластера.

Пошаговая процедура:

1. Подготовка на Leader-узле:

   a. Создайте резервную копию базы данных:

   ```
   $ mysqldump -u oneadmin -p'oneadmin' \
     -h localhost --add-drop-database --databases --add-drop-table
   opennebula  \
     > /var/lib/one/mysql_localhost_opennebula.sql
   ```

   b. Скопируйте резервную копию базы и файлы из каталога **/var/lib/one/.one/** на новый узел:

   ```
   $ scp /var/lib/one/mysql_localhost_opennebula.sql 192.168.0.184:/
   tmp

   $ ssh 192.168.0.184 rm -rf /var/lib/one/.one
   $ scp -r /var/lib/one/.one/ 192.168.0.184:/var/lib/one/
   ```

2. На Follower-узле:

    a. Остановите OpenNebula:

    ```
    # systemctl stop opennebula
    ```

    b. Восстановите базу:

    ```
    $ mysql -u oneadmin -p'oneadmin' -h localhost opennebula < /tmp/
    mysql_localhost_opennebula.sql
    ```

3. На Leader-узле:

    a. Добавьте в зону новый узел:

    ```
    $ onezone server-add 0  \
      --name one-02 \
      --rpc http://192.168.0.184:2633/RPC2
    ```

    b. Проверьте статус зоны:

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


    ZONE SERVERS
    ID NAME            ENDPOINT
     0 opennebula      http://192.168.0.185:2633/RPC2
     1 one-02          http://192.168.0.184:2633/RPC2

    HA & FEDERATION SYNC STATUS
    ID NAME            STATE      TERM       INDEX      COMMIT
    VOTE  FED_INDEX
     0 opennebula      leader     1          59         59
    0     -1
     1 one-02          error      -          -          -
    -     -

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

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

4. На Follower-узле:

    a. Обновите конфигурацию ***SERVER_ID*** в файле **/etc/one/oned.conf**:

    ```
    FEDERATION = [
        MODE        = "STANDALONE",
        ZONE_ID     = 0,
        SERVER_ID   = 1,   # ← ID из вывода onezone show
        MASTER_ONED = ""
    ```

```
]

# Скрипты управления плавающим IP (как в настройке Leader)
RAFT_LEADER_HOOK = [ ... ]
RAFT_FOLLOWER_HOOK = [ ... ]
```

b. Запустите OpenNebula:

```
# systemctl start opennebula
```

5. На Leader-узле проверьте состояние зоны:

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


ZONE SERVERS
ID NAME            ENDPOINT
 0 opennebula      http://192.168.0.185:2633/RPC2
 1 one-02          http://192.168.0.184:2633/RPC2

HA & FEDERATION SYNC STATUS
ID NAME            STATE      TERM       INDEX      COMMIT     VOTE
FED_INDEX
 0 opennebula      leader     1          59         59         0     -1
 1 one-02          follower   1          39         39         0     -1

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

Статус нового сервера изменился на follower.

Повторите 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 один сервер находится в состоянии ошибки, исправьте его, прежде чем продолжить.

Проверка работоспособности кластера:

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


ZONE SERVERS
ID NAME            ENDPOINT
 0 opennebula      http://192.168.0.185:2633/RPC2
```

```
 1 one-02          http://192.168.0.184:2633/RPC2
 2 one-03          http://192.168.0.183:2633/RPC2

HA & FEDERATION SYNC STATUS
ID NAME            STATE      TERM       INDEX      COMMIT     VOTE
FED_INDEX
 0 opennebula      leader     1          59         59         0     -1
 1 one-02          follower   1          39         39         0     -1
 2 one-03          follower   1          39         39         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.

36.3. Управление серверами кластера

Команда добавления сервера:

```
$ onezone server-add <ID_зоны> --name name --rpc rpc
```

Возможные параметры:

- **-n, --name** — имя сервера зоны;
- **-r, --rpc** — адрес RPC-интерфейса (частный IP);
- **-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 <ID_зоны> <ID_узла>
```

Возможные 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.

36.4. Восстановление отказавшего сервера

Если Follower-сервер недоступен в течение некоторого времени, он может выйти за пределы окна восстановления Raft, т.е. журнал может не содержать всех записей, необходимых для его обновления. Чтобы восстановить этот сервер, необходимо:

1. На Leader создать новую резервную копию БД и скопировать её на отказавший сервер (повторно использовать предыдущую резервную копию нельзя).
2. На Follower:
    a. Остановите OpenNebula.
    b. Восстановите БД из новой копии.
    c. Запустите OpenNebula.
3. На Leader сбросьте статус отказавшего узла:

```
$ onezone server-reset <ID_зоны> <ID_узла>
```

36.5. Включение/отключение зоны

Во время технического обслуживания можно временно отключить зону:

```
$ onezone disable <ID_зоны>
```

Примечание

Отключение зоны не останавливает уже запущенные ВМ, а лишь блокирует операции управления.

Поведение отключённой зоны:

- разрешены только операции чтения (просмотр ВМ, хостов, шаблонов);
- запрещены операции записи (создание/удаление ВМ, изменение конфигурации).

После выполнения команды некоторые ВМ могут оставаться во временных состояниях (например, prolog, epilog). Дождитесь завершения всех операций перед началом обслуживания.

Возобновление работы зоны:

```
$ onezone enable <ID_зоны>
```

36.6. Sunstone

Существует два подхода к развёртыванию веб-интерфейса:

- Базовый вариант:
  - Sunstone запущен на каждом узле управления;
  - клиенты обращаются только к Leader через плавающий IP-адрес.
- Расширенный вариант:
  - используется балансировщик нагрузки (например, HAProxy) на узлах управления;
  - балансировщик распределяет запросы Sunstone / XML-RPC между всеми узлами;
  - операции записи автоматически перенаправляются на Leader.

36.7. Настройка параметров Raft

Параметры алгоритма Raft задаются в **`/etc/one/oned.conf`**.

Таблица 36.1. Параметры Raft

Атрибут	Описание
`LIMIT_PURGE`	Количество записей журнала, удаляемых за одну операцию очистки
`LOG_RETENTION`	Минимальное количество записей в журнале; определяет окно синхронизации между узлами
`LOG_PURGE_TIMEOUT`	Интервал очистки журнала (в секундах)
`ELECTION_TIMEOUT_MS`	Тайм-аут запуска выборов при отсутствии heartbeats от Leader
`BROADCAST_TIMEOUT_MS`	Частота отправки heartbeats
`XMLRPC_TIMEOUT_MS`	Тайм-аут Raft-операций через API (0 = бесконечно)

Примечание

Любое изменение этих параметров может привести к неожиданному поведению во время переключения на резервный сервер и к сбою всего кластера. Всегда тестируйте изменения в изолированной среде!

36.8. Синхронизация конфигурации между узлами

Команда **`onezone serversync`** автоматизирует синхронизацию конфигурации в HA-кластере.

Данная команда:

1. сравнивает файлы в **`/etc/one/`** между узлами;
2. заменяет локальные файлы на версии с удалённого сервера;

3. исключение: **`/etc/one/oned.conf`** — сохраняется блок FEDERATION, остальное перезаписывается;
4. создаёт резервные копии в **`/etc/one/`**;
5. перезапускает только затронутые службы.

Важно

Используйте эту опцию только в HA-кластерах, не применяйте к федеративным развёртываниям.

Примечание

Файлы, отсутствующие на удалённом сервере (например, резервные копии), не будут удалены.

Важно

Команда выполняется от имени пользователя root (поскольку она изменяет файлы конфигурации) и требует беспарольный SSH-доступ к root@<узел> и oneadmin@<узел>.

Синтаксис:

```
# onezone serversync <IP_Leader> # Синхронизация конфигурации
# onezone serversync <IP_Leader> --db # Синхронизация + база данных
```

Глава 37. Обеспечение высокой доступности ВМ

Цель данного раздела — предоставить информацию о подготовке к сбоям ВМ и узлов, а также об их восстановлении. Сбои делятся на две категории:

- сбои физической инфраструктуры — отказы узлов (аппаратные сбои, проблемы с сетью);
- сбои виртуальной инфраструктуры — аварийное завершение работы ВМ (крах ОС, зависание приложений).

В обоих случаях OpenNebula предоставляет механизм автоматического переключения (failover), позволяющий минимизировать время простоя сервисов.

37.1. Обработка сбоев узлов

Когда OpenNebula обнаруживает, что узел перешёл в состояние ERROR, может быть автоматически запущен специальный хук для восстановления сервисов.

По умолчанию в поставке OpenNebula доступен скрипт **`ft/host_error.rb`**, который позволяет:

- мигрировать или перезапускать ВМ на других узлах;
- сократить простой сервисов при аппаратных сбоях.

Примечание

Автоматическая обработка состояния ERROR не включена по умолчанию и требует явной настройки хука.

Настройка хука для состояния ERROR:

1. Создайте хук на основе шаблона **`/usr/share/one/examples/host_hooks/error_hook`**:

```
ARGUMENTS       = "$TEMPLATE -m -p 5"
ARGUMENTS_STDIN = "yes"
COMMAND         = "ft/host_error.rb"
NAME            = "host_error"
STATE           = "ERROR"
REMOTE          = "no"
RESOURCE        = HOST
TYPE            = state
```

   В приведённом примере:

   - используется миграция ВМ (***`-m`***);
   - переключение не выполняется, если узел восстановится в течение 5 циклов мониторинга (***`-p 5`***).

2. Зарегистрируйте хук:

```
$ onehook create /usr/share/one/examples/host_hooks/error_hook
```

Таблица 37.1. Параметры скрипта ft/host_error.rb

Параметр	Описание
`$TEMPLATE`	Шаблон узла, перешедшего в состояние ERROR. Передаётся в формате XML, закодированном в base64
`-m`	Миграция ВМ на другой узел (доступно только при использовании shared-хранилищ: NFS, Ceph)
`-r`	Удаление и пересоздание ВМ (состояние ВМ теряется)
`-d`	Полное удаление ВМ
`-f`	

Параметр	Описание
	Принудительный перезапуск ВМ, находящихся в состояниях SUSPEND или POWEROFF
-p <n>	Не выполнять переключение, если узел восстановится в течение <n> циклов мониторинга
--no-fencing	Отключить изоляцию (fencing) неисправного узла

Примечание

При сетевых сбоях возможна ситуация, когда одна и та же ВМ запускается одновременно на двух узлах. Это может привести к повреждению данных, особенно при использовании shared-хранилищ.

Для безопасной работы HA-механизма необходимо обязательно настроить fencing (изоляцию неисправного узла). Использование HA без fencing считается небезопасным.

37.2. Настройка чувствительности HA

Скорость реакции на сбой определяется параметрами мониторинга, задаваемыми в файле **`/etc/one/monitord.conf`**:

- ***`BEACON_HOST`*** — интервал отправки beacon-сообщений для проверки доступности узла;
- ***`MONITORING_INTERVAL_HOST`*** — интервал мониторинга состояния узла.

Если SSH-подключение к узлу «зависает», механизм HA может не сработать. Чтобы избежать этого, рекомендуется настроить таймауты SSH (для пользователя oneadmin) в файле **`/var/lib/one/.ssh/config`**:

```
Host *
  ConnectTimeout 15
  ServerAliveInterval 10
  ControlMaster no
  ControlPersist 70s
  StrictHostKeyChecking no
  UserKnownHostsFile /dev/null
```

Примечание

Слишком короткие интервалы могут вызывать ложные срабатывания при временных сетевых проблемах. Необходимо найти баланс между скоростью реакции и стабильностью.

37.3. Настройка Fencing (изоляции узла)

Для безопасного автоматического переключения требуется изоляция неисправного узла (fencing) с использованием IPMI, iLO, DRAC и аналогичных механизмов.

Шаг 1. Создание скрипта fencing

В файле **/var/lib/one/remotes/hooks/ft/fence_host.sh** удалите строку:

```
echo "Fence host not configured, please edit ft/fence_host.sh" && exit 1
```

и укажите учётные данные устройства управления питанием:

```
USERNAME="admin"
PASSWORD="secret"
```

Шаг 2. Указание IP-адреса fencing-устройства

Для каждого узла задайте атрибут ***FENCE_IP***:

```
$ onehost update <ID_узла>
```

В открывшемся редакторе добавьте:

```
FENCE_IP="192.168.0.100"
```

Шаг 3. Настройка команды fencing

Пример для оборудования HPE iLO5:

```
while [ "$RETRIES" -gt 0 ]; do
    fence_ilo5 -P \
        --ip="$FENCE_IP" \
        --password="$PASSWORD" \
        --username="$USERNAME" \
        --action="$ACTION" && exit 0
    RETRIES=$((RETRIES-1))
    sleep $SLEEP_TIME
done
```

Параметр ***$ACTION*** обычно принимает значения off, reboot или on.

Примечание

Рекомендуется использовать утилиты из пакета *fence-agents* (например, fence_ipmilan, fence_drac), соответствующие используемому оборудованию.

Часть V. Установка пакетов для опытных пользователей

Содержание

Введение

Примечание

В установочный комплект Альт Виртуализация редакция ONE включено наиболее употребительное программное обеспечение. Для установки дополнительных программных пакетов можно использовать репозиторий продукта (p10).

В современных системах на базе Linux существует огромное число общих ресурсов: разделяемых библиотек, содержащих стандартные функции, исполняемые файлы, сценарии и стандартные утилиты и т.д. Этими общими ресурсами пользуются сразу несколько программ. Удаление или изменение версии одного из составляющих систему компонентов может повлечь неработоспособность других, связанных с ним компонентов, или может привести к выводу из строя всей системы. В контексте системного администрирования проблемы такого рода называют нарушением *целостности системы*. Задача администратора — обеспечить наличие в системе согласованных версий всех необходимых программных компонентов (обеспечение целостности системы).

Для установки, удаления и обновления программ, а также поддержания целостности системы в Linux в первую очередь стали использоваться программы *менеджеры пакетов* (например, такие, как **rpm**). С точки зрения менеджера пакетов программное обеспечение представляет собой набор компонентов — программных *пакетов*. Пакеты содержат в себе набор исполняемых программ и вспомогательных файлов, необходимых для корректной работы программного обеспечения. Менеджеры пакетов облегчают установку программ: они позволяют проверить наличие необходимого для работы устанавливаемой программы компонента подходящей версии непосредственно в момент установки. Менеджеры пакетов производят необходимые процедуры для регистрации программы во всех операционных средах пользователя: сразу после установки программа становится доступна пользователю из командной строки и появляется, если это было предусмотрено, в меню приложений всех графических оболочек.

Часто компоненты, используемые различными программами, выделяют в отдельные пакеты и помечают, что для работы ПО, предоставляемого пакетом A, необходимо установить пакет B. В таком случае говорят, что пакет A *зависит* от пакета B или между пакетами A и B существует *зависимость*.

Отслеживание зависимостей между такими пакетами представляет собой важную задачу для любого дистрибутива. Некоторые компоненты пакетов могут быть взаимозаменяемыми, т.е. может обнаружиться несколько пакетов, предлагающих затребованный ресурс.

Ещё более сложной является задача контроля целостности и непротиворечивости установленного в системе ПО. Представим, что некие программы A и B требуют наличия в системе компонентов C версии 1.0. Обновление версии пакета A, требующее обновления компонентов C до новой версии (например, до версии 2.0, использующей новый интерфейс доступа), влечёт за собой обязательное обновление и программы B.

На практике менеджеры пакетов оказались неспособны эффективно устранить нарушения целостности системы и предотвратить все коллизии при установке или удалении программ. Особенно остро этот недостаток сказался на обновлении систем из централизованного репозитория, в котором пакеты непрерывно обновляются, дробятся на более мелкие и т.п. Именно этот недостаток стимулировал создание систем управления программными пакетами и поддержания целостности ОС.

Для автоматизации и контроля описанных выше процессов стала применяться Усовершенствованная система управления программными пакетами **APT** (от англ. Advanced Packaging Tool). Автоматизация и контроль достигаются путём создания одного или нескольких внешних репозиториев. В них хранятся доступные для установки пакеты программ.

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, разрешает конфликты, находит пути их корректного устранения, руководствуясь сведениями из внешних репозиториев.

Система **APT** состоит из нескольких утилит. Чаще всего используется утилита управления пакетами **`apt-get`**. Она автоматически определяет зависимости между пакетами и строго следит за её соблюдением при выполнении любой из следующих операций: установка, удаление или обновление пакетов.

Глава 38. Источники программ (репозитории)

Отличие репозиториев, с которыми работает **APT**, от простого набора пакетов — наличие метаинформации. В ней содержится индекс находящихся в репозитории пакетов и сведения о них. Поэтому, чтобы получить всю информацию о репозитории, **APT** достаточно получить его индексы.

APT может пользоваться 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. Для **APT** не имеет значения расположение пакета в том или ином репозитории.

Важно

Для одновременного подключения нескольких репозиториев необходимо отслеживать их совместимость друг с другом, т.е. их пакетная база должна отражать один определённый этап разработки. Совместное использование репозиториев, относящихся к разным дистрибутивам, или смешивание стабильного репозитория с нестабильной веткой разработки (Sisyphus) может привести к различным неожиданностям и трудностям при обновлении пакетов.

APT осуществляет взаимодействие с репозиториями при помощи различных протоколов доступа. Наиболее популярные — HTTP и FTP.

Для того чтобы **APT** мог использовать тот или иной репозиторий, информацию о нём 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
```

Здесь:

- 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, rsh, ssh, cdrom, copy;
- путь — путь к репозиторию в терминах выбранного метода;
- база — относительный путь к базе данных репозитория;
- название — название репозитория.

Непосредственно после установки дистрибутива Альт Виртуализация редакция ONE в файлах **`/etc/apt/sources.list.d/*.list`** обычно указывается интернет-репозиторий, совместимый с установленным дистрибутивом.

После редактирования списка репозиториев в **`sources.list`**, необходимо обновить локальную базу данных **APT** о доступных пакетах. Это делается командой **`apt-get update`**.

Если в **`sources.list`** присутствует репозиторий, содержимое которого может изменяться (например, постоянно разрабатываемый репозиторий или репозиторий обновлений по безопасности), то прежде чем работать с **APT**, необходимо синхронизировать локальную базу данных с удалённым сервером командой **`apt-get update`**. Локальная база данных создаётся заново при каждом изменении в репозитории: добавлении, удалении или переименовании пакета.

При установке определённого пакета **APT** производит поиск самой новой версии этого пакета во всех известных ему репозиториях вне зависимости от способа доступа к ним. Так, если в репозитории, доступном в сети Интернет, обнаружена более новая в сравнении с компакт-диском версия программы, то **APT** начнёт загружать соответствующий пакет из сети Интернет. Поэтому, если подключение к сети Интернет отсутствует или ограничено низкой пропускной способностью канала или высокой стоимостью, то следует закомментировать строчки (добавить в начало строки символ #) в **`/etc/apt/sources.list`**, относящиеся к ресурсам в сети Интернет.

38.1. Редактирование репозиториев

38.1.1. Утилита apt-repo для работы с репозиториями

Для редактирования репозиториев можно воспользоваться скриптом **apt-repo**:

- просмотреть список активных репозиториев:

```
apt-repo
```

- добавить репозиторий в список активных репозиториев:

```
apt-repo add репозиторий
```

- удалить или выключить репозиторий:

```
apt-repo rm репозиторий
```

- обновить информацию о репозиториях:

```
apt-repo update
```

- справка о команде **apt-repo**:

```
man apt-repo
```

или

```
apt-repo --help
```

Примечание

Для выполнения большинства команд необходимы права администратора.

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
```

38.1.2. Добавление репозитория на сменном носителе

Для добавления в **sources.list** репозитория на сменном диске в **APT** предусмотрена специальная утилита — **apt-cdrom**.

Чтобы добавить запись о репозитории на сменном диске 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 подключённом носителе:

```
rpm cdrom:[ALT Server-V 10.2 x86_64 build 2024-05-13]/ ALTLinux main
```

38.1.3. Добавление репозиториев вручную

Для изменения списка репозиториев можно отредактировать в любом текстовом редакторе файлы из каталога **/etc/apt/sources.list.d/**.

Примечание

Для изменения этих файлов необходимы права администратора.

В файле ***alt*.list** может содержаться такая информация:

```
# ftp.altlinux.org (ALT Linux, Moscow)

# ALT Platform 10
#rpm [p10] ftp://ftp.altlinux.org/pub/distributions/ALTLinux p10/branch/
x86_64 classic
#rpm [p10] ftp://ftp.altlinux.org/pub/distributions/ALTLinux p10/branch/
x86_64-i586 classic
#rpm [p10] ftp://ftp.altlinux.org/pub/distributions/ALTLinux p10/branch/
noarch classic

rpm [p10] http://ftp.altlinux.org/pub/distributions/ALTLinux p10/branch/
x86_64 classic
rpm [p10] http://ftp.altlinux.org/pub/distributions/ALTLinux p10/branch/
x86_64-i586 classic
rpm [p10] http://ftp.altlinux.org/pub/distributions/ALTLinux p10/branch/
noarch classic
```

По сути, каждая строчка соответствует некому репозиторию. Не активные репозитории — строки, начинающиеся со знака #. Для добавления нового репозитория, достаточно дописать его в этот или другой файл.

После обновления списка репозиториев следует обновить информацию о них (выполнить команду **`apt-get update`** или **`apt-repo update`**).

Глава 39. Поиск пакетов

Если точное название пакета неизвестно, то для его поиска можно воспользоваться утилитой **`apt-cache`**. Данная утилита позволяет искать пакет не только по имени, но и по его описанию.

Команда **`apt-cache search`** ***`подстрока`*** позволяет найти все пакеты, в именах или описании которых присутствует указанная подстрока. Например:

```
$ apt-cache search telegraf
ceph-mgr-telegraf - Telegraf module for Ceph Manager Daemon
telegraf - The plugin-driven server agent for collecting and reporting
metrics
```

Для того чтобы подробнее узнать информацию о найденном пакете и получить его подробное описание, воспользуйтесь командой **`apt-cache show`**:

```
$ apt-cache show telegraf
Package: telegraf
Section: Development/Other
Installed Size: 154119764
Maintainer: Alexey Shabalin (ALT Team) <shaba@altlinux.org>
Version: 1.24.2-alt1:p10+326352.200.3.1@1691242931
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 (= 1.24.2-alt1:p10+326352.200.3.1)
Architecture: x86_64
Size: 29082799
MD5Sum: d9daf730a225fb47b2901735aa01ed17
Filename: telegraf-1.24.2-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. К сожалению, описание на русском языке в настоящее время есть не у всех пакетов, но наиболее актуальные описания переведены.

Глава 40. Установка или обновление пакета

Важно

Для установки пакетов требуются привилегии администратора.

Установка пакета с помощью APT выполняется командой **apt-get install *имя_пакета***.

Важно

Перед установкой и обновлением пакетов необходимо выполнить команду обновления индексов пакетов:

```
# apt-get update
```

apt-get позволяет устанавливать в систему пакеты, требующие для работы наличие других, пока ещё не установленных пакетов. В этом случае он определяет, какие пакеты необходимо установить. **apt-get** устанавливает их, пользуясь всеми доступными репозиториями.

Установка пакета *telegraf* командой **apt-get install telegraf** приведёт к следующему диалогу с **APT**:

```
# 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 также и для обновления уже установленного пакета или группы пакетов. В этом случае **apt-get** дополнительно проверяет, есть ли обновлённая, в сравнении с установленной в системе, версия пакета в репозитории.

При помощи **APT** можно установить и отдельный rpm-пакет, не входящий в состав репозиториев (например, полученный из сети Интернет). Для этого достаточно выполнить команду

```
# apt-get install /путь/к/файлу.rpm
```

При этом **APT** проведёт стандартную процедуру проверки зависимостей и конфликтов с уже установленными пакетами.

Иногда в результате операций с пакетами без использования **APT** целостность системы нарушается, и **apt-get** отказывается выполнять операции установки, удаления или обновления. В этом случае необходимо повторить операцию, задав опцию **-f**, заставляющую **apt-get** исправить нарушенные зависимости, удалить или заменить конфликтующие пакеты. В этом случае необходимо внимательно следить за сообщениями, выводимыми **apt-get**. Любые действия в этом режиме обязательно требуют подтверждения со стороны пользователя.

Глава 41. Удаление установленного пакета

Для удаления пакета используется команда **apt-get remove *имя_пакета***. Для того чтобы не нарушать целостность системы, будут удалены и все пакеты, зависящие от удаляемого. В случае удаления пакета, который относится к базовым компонентам системы, **apt-get** потребует дополнительное подтверждение с целью предотвращения возможной случайной ошибки.

Важно

Для удаления пакетов требуются привилегии администратора.

При попытке с помощью **apt-get** удалить базовый компонент системы, вы увидите следующий запрос на подтверждение операции:

```
# apt-get remove filesystem
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
Следующие пакеты будут УДАЛЕНЫ:
  ...
ВНИМАНИЕ: Будут удалены важные для работы системы пакеты
Обычно этого делать не следует. Вы должны точно понимать возможные
последствия!
  ...
0 будет обновлено, 0 новых установлено, 2648 пакетов будет удалено и 0 не
будет обновлено.
Необходимо получить 0B архивов.
После распаковки будет освобождено 8994MB дискового пространства.
Вы делаете нечто потенциально опасное!
Введите фразу 'Yes, do as I say!' чтобы продолжить.
```

Предупреждение

Каждую ситуацию, в которой **APT** выдаёт такой запрос, необходимо рассматривать отдельно. Вероятность того, что после выполнения этой команды система окажется неработоспособной, очень велика.

Глава 42. Обновление системы

42.1. Обновление всех установленных пакетов

Для обновления всех установленных пакетов необходимо выполнить команды:

```
# 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.

Примечание

Несмотря на то, что команда **apt-get upgrade** существует, использовать её следует осторожно, либо не использовать вовсе.

Она позволяет обновить только те установленные пакеты, для которых в репозиториях, перечисленных в **/etc/apt/sources.list**, имеются новые версии.

Никакие другие пакеты при этой операции из системы удалены не будут. Этот способ полезен при работе со стабильными пакетами приложений, относительно которых известно, что они при смене версии изменяются несущественно.

Иногда, однако, происходит изменение в наименовании пакетов или изменение их зависимостей. Такие ситуации не обрабатываются командой **apt-get upgrade**, в результате чего происходит нарушение целостности системы: появляются неудовлетворённые зависимости. Для разрешения этой проблемы существует режим обновления в масштабе дистрибутива — **apt-get dist-upgrade**.

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 отчёте **apt-get**, которым **APT** предварит само обновление.

Примечание

Команда **apt-get dist-upgrade** обновит систему, но ядро ОС не будет обновлено.

42.2. 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, которую рекомендуется выполнить немедленно.

Если с новым ядром что-то пойдёт не так, вы сможете вернуться к предыдущему варианту, выбрав его в начальном меню загрузчика.

После успешной загрузки на обновленном ядре можно удалить старое, выполнив команду:

```
# remove-old-kernels
```

Часть VI. Основы администрирования Linux

Содержание

Глава 43. Общие принципы работы ОС

43.1. Процессы и файлы

ОС Альт Виртуализация редакция ONE 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. Командный интерпретатор представляет собой 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.

43.1.1. 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 продолжается только после переведения его в «нормальный» режим работы.

43.1.2. 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 дистрибутиве монтируется по умолчанию в каталог **/media/cdrom** (путь в дистрибутиве обозначается с использованием **/**, а не ****, как в DOS/Windows).

Текущий каталог обозначается **./**.

43.1.3. Структура каталогов

Корневой каталог **/**:

- **/bin** — командные оболочки (shell), основные утилиты;
- **/boot** — содержит ядро системы;
- **/dev** — псевдофайлы устройств, позволяющие работать с устройствами напрямую. Файлы в **/dev** создаются сервисом **udev**
- **/etc** — общесистемные конфигурационные файлы для большинства программ в системе;
- **/etc/rc?.d**, **/etc/init.d**, **/etc/rc.boot**, **/etc/rc.d** — каталоги, где расположены командные файлы, выполняемые при запуске системы или при смене её режима работы;
- **/etc/passwd** — база данных пользователей, в которой содержится информация об имени пользователя, его настоящем имени, личном каталоге, его зашифрованный пароль и другие данные;
- **/etc/shadow** — теневая база данных пользователей. При этом информация из файла **/etc/passwd** перемещается в **/etc/shadow**, который недоступен для чтения всем, кроме пользователя root. В случае использования альтернативной схемы управления теневыми паролями (TCB), все теневые пароли для каждого пользователя располагаются в каталоге **/etc/tcb/*имя пользователя*/shadow**;
- **/home** — домашние каталоги пользователей;
- **/lib** — содержит файлы динамических библиотек, необходимых для работы большей части приложений, и подгружаемые модули ядра;
- **/lost+found** — восстановленные файлы;
- **/media** — подключаемые носители (каталоги для монтирования файловых систем сменных устройств);
- **/mnt** — точки временного монтирования;
- **/opt** — вспомогательные пакеты;
- **/proc** — виртуальная файловая система, хранящаяся в памяти компьютера при загруженной ОС. В данном каталоге расположены самые свежие сведения обо всех процессах, запущенных на компьютере.
- **/root** — домашний каталог администратора системы;
- **/run** — файлы состояния приложений;

- **/sbin** — набор программ для административной работы с системой (системные утилиты);
- **/selinux** — виртуальная файловая система SELinux;
- **/srv** — виртуальные данные сервисных служб;
- **/sys** — файловая система, содержащая информацию о текущем состоянии системы;
- **/tmp** — временные файлы.
- **/usr** — пользовательские двоичные файлы и данные, используемые только для чтения (программы и библиотеки);
- **/var** — файлы для хранения изменяющихся данных (рабочие файлы программ, очереди, журналы).

Каталог **/usr**:

- **/usr/bin** — дополнительные программы для всех учетных записей;
- **/usr/sbin** — команды, используемые при администрировании системы и не предназначенные для размещения в файловой системе root;
- **/usr/local** — место, где 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, находящихся в очереди на обработку для того или иного процесса (очереди печати, непрочитанные или не отправленные письма, задачи cron т.д.).

43.1.4. 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, представляющая собой путь в файловой системе к данному файлу, где каждое следующее имя отделяется от предыдущего наклонной чертой (слешем). Если название маршрута начинается со слеша, 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;
- символ подчеркивания (_);
- точка (**.**).

Для удобства работы точку можно использовать для отделения имени файла от расширения файла. Данная возможность может быть необходима пользователям или некоторым программам, но не имеет значение для shell.

43.1.5. Имена дисков и разделов

Все физические устройства вашего компьютера отображаются в каталог **/dev** файловой системы дистрибутива (об этом — ниже). Диски (в том числе IDE/SATA/SCSI/SAS жёсткие диски, USB-диски) имеют имена:

- **/dev/sda** — первый диск;
- **/dev/sdb** — второй диск;
- и т.д.

Диски обозначаются **/dev/sd*X***, где *X* — a, b, c, d, e, … в зависимости от порядкового номера диска на шине.

Раздел диска обозначается числом после его имени. Например, **/dev/sdb4** — четвертый раздел второго диска.

43.1.6. Разделы, необходимые для работы ОС

Для работы ОС на жестком диске (дисках) должны быть созданы, 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 на диске много свободного места, то можно создать отдельные разделы для каталогов **/usr**, **/home**, **/var**.

43.2. Работа с наиболее часто используемыми компонентами

43.2.1. Виртуальная консоль

Система Альт Виртуализация редакция ONE предоставляет доступ к виртуальным консолям, с которых можно осуществлять одновременно несколько сеансов работы в системе (login session).

Только что установленная система Альт Виртуализация редакция ONE, возможно, предоставляет доступ только к первым шести виртуальным консолям, к которым можно обращаться, нажимая комбинации клавиш **Alt**+**F1** — **Alt**+**F6** (**Ctrl**+**Alt**+**F1** — **Ctrl**+**Alt**+**F6**).

43.2.2. Командные оболочки (интерпретаторы)

Для управления ОС используются командные интерпретаторы (shell).

Зайдя в систему, Вы увидите приглашение — строку, содержащую символ «**$**» (далее этот символ будет обозначать командную строку). Программа ожидает ваших команд. Роль командного интерпретатора — передавать ваши команды операционной системе. По своим функциям он соответствует **command.com** в DOS, но несравненно мощнее. При помощи командных интерпретаторов можно писать небольшие программы — сценарии (скрипты). В Linux доступны следующие командные оболочки:

- **bash** — самая распространенная оболочка под linux. Она ведет историю команд и предоставляет возможность их редактирования;
- **pdksh** — клон korn shell, хорошо известной оболочки в UNIX™ системах.

Проверить, какая оболочка используется в данный момент можно, выполнив команду:

```
$ echo $SHELL
```

Оболочкой по умолчанию является Bash (Bourne Again Shell) — самая распространённая оболочка под Linux, которая ведет историю команд и предоставляет возможность их редактирования. В дальнейшем описании работы с Альт Виртуализация редакция ONE будут использоваться примеры с использованием этой оболочки.

43.2.3. Командная оболочка Bash

В Bash имеется несколько приемов для работы со строкой команд. Например, можно использовать следующие сочетания:

- **Ctrl**+**A** — перейти на начало строки;
- **Ctrl**+**U** — вырезать/удалить все символы слева от курсора до начала строки в буфер обмена;
- **Ctrl**+**C** — остановить текущую задачу.

Для ввода нескольких команд одной строкой можно использовать разделитель «**;**». По истории команд можно перемещаться с помощью клавиш ↑ («вверх») и ↓ («вниз»).

Чтобы найти конкретную команду в списке набранных, не пролистывая всю историю, можно нажать **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 клавишу **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
```

43.2.4. Команда

Простейшая команда состоит из одного «слова», например, команда **cal**, выводящая календарь на текущий месяц.

```
$ cal
      Март 2025
Пн Вт Ср Чт Пт Сб Вс
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

43.2.5. Команда и параметры

```
$ cal 1 2026
     Январь 2026
Пн Вт Ср Чт Пт Сб Вс
          1  2  3  4
```

```
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
```

Команда **`cal 1 2026`** состоит из двух частей — собственно команды **`cal`** и «остального». То, что следует за командой, называется *параметрами* (или аргументами), и они вводятся для изменения поведения команды. В большинстве случаев, первое слово считается именем команды, а остальные — её параметрами.

43.2.6. Команда и ключи

Для решения разных задач одни и те же действия необходимо выполнять по-разному. Например, для синхронизации работ в разных точках земного шара лучше использовать единое для всех время (по Гринвичу), а для организации собственного рабочего дня — местное время (с учётом сдвига по часовому поясу и разницы зимнего и летнего времени). И то, и другое время показывает команда **`date`**, только для работы по Гринвичу ей нужен дополнительный параметр ***`-u`*** (он же ***`--universal`***).

```
$ date
Чт 13 мар 2025 12:10:52 EET
$ date -u
Чт 13 мар 2025 10:11:05 UTC
```

Такого рода параметры называются *ключами* или *модификаторами выполнения*. Ключ принадлежит данной конкретной команде и сам по себе смысла не имеет. Этим он отличается от других параметров (например, имён файлов, чисел), имеющих собственный смысл, не зависящий ни от какой команды. Каждая команда может распознавать некоторый набор ключей и соответственно изменять своё поведение. Один и тот же ключ может определять для разных команд совершенно разные значения.

Для формата ключей нет жёсткого стандарта, однако существуют договорённости:

- Если ключ начинается на -, то это *однобуквенный ключ*. За -, как правило, следует один символ, чаще всего буква, обозначающая действие или свойство, которое этот ключ придаёт команде. Так проще отличать ключи от других параметров.
- Если ключ начинается на --, то он называется *полнословным ключом*. Полнословный формат ключа начинается на два знака --, за которыми следует полное имя обозначаемого этим ключом содержания.

Некоторые ключи имеют и однобуквенный, и полнословный формат, а некоторые — только полнословный.

Информацию о ресурсах каждой команды можно получить, используя ключ ***`--help`***. К примеру, получить подсказку о том, что делает команда **`rm`**, можно, набрав в терминале **`rm --help`**.

43.2.7. Обзор основных команд системы

Все команды, приведенные ниже, могут быть запущены в режиме консоли. Для получения более подробной информации используйте 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`**

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 права администратора. Чтобы вернуться к правам пользователя, необходимо ввести команду:

```
exit
```

Более подробную информацию о режиме суперпользователя вы можете прочитать в главе Режим суперпользователя.

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 [ОПЦИИ...] [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.

Основные операции с файлами и каталогами

Команда `ls`

Команда **ls** (list) печатает в стандартный вывод содержимое каталогов.

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.

Предупреждение

Удалив файл, вы не сможете его восстановить!

Синтаксис:

```
rm [ОПЦИИ...] <ФАЙЛ>
```

Основные опции:

- **-f** — никогда не запрашивать подтверждения;
- **-i** — всегда запрашивать подтверждение;
- **-r, -R** — рекурсивно удалять содержимое указанных каталогов.

Пример. Удалить все файлы **html** в каталоге **~/html**:

```
rm -i ~/html/*.html
```

Команда mkdir

mkdir — команда для создания новых каталогов.

Синтаксис:

```
mkdir [-p] [-m права] <КАТАЛОГ...>
```

Команда rmdir

Команда **rmdir** удаляет каталоги из файловой системы. Каталог должен быть пуст перед удалением.

Синтаксис:

```
rmdir [ОПЦИИ...] <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 K 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 ФАЙЛ
```

Команда grep

Команда **grep** имеет много опций и предоставляет возможности поиска символьной строки в файле.

Синтаксис:

```
grep [шаблон_поиска] <ФАЙЛ>
```

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] [-Oуровень] [-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

whereis сообщает путь к исполняемому файлу программы, ее исходным файлам (если есть) и соответствующим страницам справочного руководства.

Синтаксис:

```
whereis [ОПЦИИ...] <ФАЙЛ>
```

Опции:

- **-b** — вывод информации только об исполняемых файлах;
- **-m** — вывод информации только о страницах справочного руководства;
- **-s** — вывод информации только об исходных файлах.

Мониторинг и управление процессами

Команда ps

Команда **ps** отображает список текущих процессов.

Синтаксис:

```
ps [ОПЦИИ...]
```

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.

Синтаксис:

```
df [ОПЦИИ] [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] <ФАЙЛ...>
```

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.

Использование многозадачности

Альт Виртуализация редакция ONE — это многозадачная система.

Для того чтобы запустить программу в фоновом режиме, необходимо набрать «**&**» после имени программы. После этого оболочка даст 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. Этот метод также позволяет вам работать на другой консоли, если консоль, которую вы использовали до этого, не отвечает или вам необходимо остановить зависшую программу.

Команда bg

Команда **bg** позволяет 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 «%».

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`**.

43.3. Стыкование команд в системе Linux

43.3.1. 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).

43.3.2. 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.

43.3.3. 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
```

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.

43.3.4. Недеструктивное перенаправление вывода

Эффект от использования символа > для перенаправления вывода файла является деструктивным; т.е, 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.

Глава 44. Средства управления дискреционными правами доступа

44.1. Команда chmod

Команда **chmod** предназначена для изменения прав доступа файлов и каталогов.

Синтаксис:

```
chmod [ОПЦИИ] РЕЖИМ[,РЕЖИМ]... <ФАЙЛ>
```

```
chmod [ОПЦИИ] --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***. Режим может быть представлен как в символьном виде, так и в числовом (восьмеричном) виде, представляющем 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, разделённых запятыми. Каждая такая команда начинается с нуля или более букв «ugoa», которые указывают, чьи права доступа к файлу будут изменены:

- u — владелец файла;
- g — пользователи, входящие в группу, к которой принадлежит файл (за исключением владельца);
- o — все остальные пользователи, не являющиеся ни владельцем, ни членами группы;
- a — все категории пользователей (u, g и o).

Если не задана ни одна буква, то автоматически будет использована буква «a», но биты, установленные в umask, не будут затронуты.

Примечание

Права доступа образуют иерархию, исключающую вышестоящие категории. Права группы не включают владельца, а права остальных пользователей не включают ни владельца, ни членов группы.

Операторы:

- + — добавляет выбранные права доступа;
- - — удаляет выбранные права доступа;
- = — задаёт только указанные права, заменяя существующие.

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 файл с установленным битом выполнения;
- s — установка битов setuid (для владельца) или setgid (для группы);
- t — sticky-бит, предотвращающий удаление файла другими пользователями.

Числовой режим состоит не более чем из четырёх восьмеричных цифр (от нуля до семи), которые складываются из битовых масок с разрядами «4», «2» и «1». Пропущенные разряды дополняются ведущими нулями:

- первый разряд — установка битов setuid (4), setgid (2) и sticky (1);
- второй разряд — права доступа для владельца: чтение (4), запись (2) и выполнение (1);
- третий разряд — права доступа для группы: чтение (4), запись (2) и выполнение (1);
- четвертый разряд — права доступа для остальных пользователей (не входящих в данную группу): чтение (4), запись (2) и выполнение (1).

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
```

- разрешить выполнение файла **f2** всем пользователям:

```
$ chmod +x f2
```

- запретить удаление файла **f3**:

```
$ chmod +t f3
```

- дать всем права на чтение, запись и выполнение, а также установить бит setgid для файла **f4**:

```
$ chmod =rwx,g+s f4
$ chmod 2777 f4
```

44.2. Команда chown

Команда **chown** изменяет владельца и/или группу для каждого заданного файла.

Синтаксис:

```
chown [КЛЮЧ]…[ВЛАДЕЛЕЦ][:[ГРУППА]] <ФАЙЛ>
```

```
chown [ОПЦИИ] --reference=ИФАЙЛ <ФАЙЛ>
```

Основные опции:

- **-R** — рекурсивно изменять файлы и каталоги;
- **--reference=ИФАЙЛ** — использовать владельца и группу файла ИФАЙЛ.

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 каталога **/u** на пользователя test:

```
chown test /u
```

- поменять владельца и группу каталога **/u**:

```
chown test:staff /u
```

- поменять владельца каталога **/u** и вложенных файлов на test:

```
chown -hR test /u
```

44.3. Команда chgrp

Команда **chgrp** изменяет группу для каждого заданного файла.

Синтаксис:

```
chgrp [ОПЦИИ] ГРУППА <ФАЙЛ>
```

```
chgrp [ОПЦИИ] --reference=ИФАЙЛ <ФАЙЛ>
```

Основные опции:

- **-R** — рекурсивно изменять файлы и каталоги;
- **--reference=ИФАЙЛ** — использовать группу файла ИФАЙЛ.

44.4. 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, выводится её 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.

44.5. Команда chattr

Команда **chattr** управляет расширенными атрибутами файловых систем ext3 и ext4.

Синтаксис:

```
chattr [ -RVf ] [+-=aAcCdDeFijmPsStTux] [ -v версия ] <ФАЙЛЫ> …
```

Опции:

- **-R** — рекурсивно изменять атрибуты каталогов и их содержимого. Символические ссылки игнорируются;
- **-V** — выводит расширенную информацию и версию программы;
- **-f** — подавлять сообщения 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`**).

44.6. Команда lsattr

Команда **`lsattr`** показывает расширенные атрибуты файлов.

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.

44.7. Команда getfacl

Команда **`getfacl`** показывает списки контроля доступа (ACL) — расширенные права, дополняющие традиционные.

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:other:---
```

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 остальных. Строки 11 — 15 показывают ACL по умолчанию, ассоциированный с данным каталогом.

44.8. Команда setfacl

Команда **`setfacl`** изменяет ACL файлов и каталогов. 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 **`-R`**);
- **`-P`** — не переходить по символическим ссылкам на каталоги (имеет смысл только в сочетании с **`-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
```

Глава 45. Режим суперпользователя

45.1. 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.

45.2. 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.

Важно

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.

45.3. 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**.

Глава 46. Управление пользователями

46.1. 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, введите 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, будьте предельно внимательны и осторожны в переназначении групповых прав на системные каталоги.

46.2. Команда useradd

Команда useradd регистрирует нового пользователя или изменяет информацию по умолчанию о новых пользователях.

Синтаксис:

```
useradd [ОПЦИИ...] <ИМЯ ПОЛЬЗОВАТЕЛЯ>
useradd -D [ОПЦИИ...]
```

Некоторые 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
```

46.3. 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.

46.4. 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 Альт Виртуализация редакция ONE для проверки паролей на слабость используется модуль PAM *passwdqc*.

46.5. 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.

46.6. 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 на день (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**.

46.7. 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.

Предупреждение

В данном случае системный каталог станет доступным для записи пользователям группы pw_users (создайте эту группу и включите туда пользователей).

Предупреждение

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
```

46.8. 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 используйте утилиту **chpasswd**. На стандартный вход ей следует подавать список, каждая строка которого будет выглядеть как ***имя:пароль***.

46.9. Удаление пользователей

Для удаления пользователей используйте **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.

Глава 47. Система инициализации systemd и sysvinit

47.1. Запуск операционной системы

47.1.1. Запуск системы

Алгоритм запуска компьютера приблизительно такой:

1. BIOS компьютера.

2. Загрузчик системы (например, LILO, GRUB или другой). В загрузчике вы можете задать параметры запуска системы или выбрать систему для запуска.

3. Загружается ядро Linux.

4. Запускается на выполнение первый процесс в системе — init.

Ядром запускается самая первая программа в системе **init**. Её задачей является запуск новых процессов и повторный запуск завершившихся. Вы можете посмотреть, где расположился **init** в иерархии процессов вашей системы, введя команду **`pstree`**.

От конфигурации **init** зависит, какая система инициализации будет использована.

47.1.2. Система инициализации

Система инициализации — это набор скриптов, которые будут выполнены при старте системы.

Существуют разные системы инициализации, наиболее популярной системой являются **sysvinit** и ее модификации. **systemd** разрабатывается как замена для **sysVinit**.

В Альт Виртуализация редакция ONE используется **systemd**.

47.2. Системы инициализации systemd и sysvinit

47.2.1. sysvinit

System V — классическая схема инициализации, на которой базируются многие дистрибутивы. Привычна и довольно проста для понимания: **init** описывает весь процесс загрузки в своем конфигурационном файле **`/etc/inittab`**, откуда вызываются другие программы и скрипты на определенном этапе запуска.

47.2.2. systemd

systemd является альтернативной системой инициализации Linux, вобравшей в себя достоинства классического **System V init** и более современных **launchd** (OS X), **SMF** (Solaris) и **Upstart** (Ubuntu, Fedora), но при этом лишенной многих их недостатков. Он разрабатывался для обеспечения лучшего выражения зависимостей между службами, что позволяет делать одновременно больше работы при загрузке системы, и уменьшить время загрузки системы.

systemd (system daemon) реализует принципиально новый подход к инициализации и контролю работы системы. Одним из ключевых новшеств этого подхода является высокая степень параллелизации запуска служб при инициализации системы, что в перспективе позволяет добиться гораздо более высокой скорости, чем традиционный подход с последовательным запуском взаимозависимых служб. Другим важным моментом является контроль над точками монтирования (не-жизненно-важные файловые системы можно монтировать только при первом обращении к ним, не тратя на это время при инициализации системы) и устройствами (можно запускать и останавливать определенные службы и при появлении или удалении заданных устройств). Для отслеживания групп процессов используется механизм cgroups, который также может быть использован для ограничения потребляемых ими системных ресурсов.

Удобство **systemd** особенно заметно на компьютерах для домашнего пользования — когда пользователи включают и перезагружают компьютер ежедневно. В отличие от **sysvinit**, подвисание при запуске одного сервиса не приведет к остановке всего процесса загрузки.

47.3. Примеры команд управления службами, журнал в systemd

Обратите внимание, что команды **`service`** и **`chkconfig`** продолжат работать в мире **systemd** практически без изменений. Тем не менее, в этой таблице показано как выполнить те же действия с помощью встроенных утилит **systemctl**.

Таблица 47.1. Команды управления службами

Команды Sysvinit	Команды Systemd	Примечания
service frobozz start	systemctl start frobozz.service	Используется для запуска службы (не перезагружает постоянные)
service frobozz stop	systemctl stop frobozz.service	Используется для остановки службы (не перезагружает постоянные)
service frobozz restart	systemctl restart frobozz.service	Используется для остановки и последующего запуска службы
service frobozz reload	systemctl reload frobozz.service	Если поддерживается, перезагружает файлы конфигурации без прерывания незаконченных операций
service frobozz condrestart	systemctl condrestart frobozz.service	Перезапускает службу, если она уже работает
service frobozz status	systemctl status frobozz.service	Сообщает, запущена ли уже служба
ls /etc/rc.d/init.d/	systemctl list-unit-files --type=service (preferred) ls /lib/systemd/system/ *.service /etc/systemd/system/ *.service	Используется для отображения списка служб, которые можно запустить или остановить. Используется для отображения списка всех служб.
chkconfig frobozz on	systemctl enable frobozz.service	Включает службу во время следующей перезагрузки, или любой другой триггер
chkconfig frobozz off	systemctl disable frobozz.service	Выключает службу во время следующей перезагрузки, или любой другой триггер
chkconfig frobozz	systemctl is-enabled frobozz.service	

Команды Sysvinit	Команды Systemd	Примечания
		Используется для проверки, сконфигурирована ли служба для запуска в текущем окружении
chkconfig --list	systemctl list-unit-files --type=service(preferred) ls /etc/systemd/system/*.wants/	Выводит таблицу служб. В ней видно, на каких уровнях загрузки они (не)запускаются
chkconfig frobozz --list	ls /etc/systemd/system/*.wants/frobozz.service	Используется, для отображения на каких уровнях служба (не)запускается
chkconfig frobozz --add	systemctl daemon-reload	Используется, когда вы создаете новую службу или модифицируете любую конфигурацию

Глава 48. Что происходит в системе

Человеку, отвечающему за работоспособность системы, очень важно всегда отчётливо представлять происходящие в ней события. Теоретически, никакое происшествие не должно ускользнуть от его внимания. Однако компьютерные системы настолько сложны, что отслеживать *все* события в них — выше человеческих возможностей. Для того чтобы довести поток служебной информации до разумного объёма, её надо *просеять* (выкинуть незначащие данные), *классифицировать* (разделить на несколько групп сообразно тематике) и *журнализировать* (сохранить в доступном виде для дальнейшего анализа).

В ОС Альт Виртуализация редакция ONE функция записи информации о системных событиях и событиях безопасности обеспечивается с помощью системной службы **systemd-journald**. Она создает и поддерживает структурированные, индексированные журналы, на основе регистрируемой информации, полученной от ядра, от пользовательских процессов через вызов Libc syslog, от потоков STDOUT/STDERR системных служб через собственный API. Журналы данного инструмента хранятся в бинарном виде в **`/var/log/journal`**, что исключает возможность просмотра содержимого данных файлов стандартными утилитами обработки текстовых данных. Для просмотра логов используется утилита **journalctl**.

Можно запускать **`journalctl`** с разными ключами:

- **`journalctl -b`** — покажет сообщения только с текущей загрузки;
- **`journalctl -f`** — покажет только последние сообщения и продолжит печатать новые записи при добавлении их в журнал.

Можно посмотреть сообщения определенного процесса:

- **`journalctl _PID`**=1 — покажет сообщения первого процесса (init).

Для ознакомления с прочими возможностями читайте руководство по **journalctl**. Для этого используйте команду **`man journalctl`**.

Стоит заметить, что некоторые службы (например, веб-сервер **apache**) *самостоятельно* ведут журнализацию своих событий, поэтому информацию о количестве и местоположении их журналов можно почерпнуть из их файлов настроек (обычно журналы хранятся в **/var/log/**).

Файл настройки **journald** находится в **/etc/systemd/journald.conf**. Справку по этому файлу можно получить, выполнив команду **man journald.conf**.

Новые рапорты, поступающие в системный журнал, наиболее актуальны, а предыдущие, по мере их устаревания, эту актуальность утрачивают. Если самые старые данные в журнале не удалять, файловая система рано или поздно окажется переполненной. Узнать объем имеющихся на текущий момент логов можно с помощью команды:

```
# journalctl --disk-usage
```

Ротация журналов:

- для удаления старых файлов журналов с помощью указания размера используется опция **--vacuum-size**. Например, удалить журналы, оставив только последние 200 Мб:

  ```
  # journalctl --vacuum-size=200M
  ```

- для удаления старых записей по времени используется опция **--vacuum-time**. Например, удалить журналы, оставив только журналы за последний месяц:

  ```
  # journalctl --vacuum-time=1months
  ```

Примечание

Следует обратить внимание, что запуск **--vacuum-size=** оказывает лишь косвенное влияние на вывод, показанный **--disk-usage**, поскольку последний включает активные файлы журнала, в то время как операция очистки работает только с архивными файлами журнала. Аналогично, **--vacuum-time=** может фактически не уменьшить количество файлов журнала ниже указанного числа, так как не удалит активные файлы журнала.

Настройки ротации файлов журнала можно прописать в конфигурационном файле **/etc/systemd/journald.conf**, например:

- установить ограничения на размер хранимых файлов журнала:

  ```
  SystemMaxUse=1G
  ```

- ограничить размер отдельных файлов журнала:

  ```
  SystemMaxFileSize=200M
  ```

- установить максимальное время хранения записей в одном файле журнала перед ротацией в следующий:

  ```
  MaxFileSec=20day
  ```

- ограничить время хранения файлов журнала (удалять файлы журнала, содержащие записи старше указанного промежутка времени):

```
MaxRetentionSec=1months
```

Примечание

После внесения изменений в файл **/etc/systemd/journald.conf** необходимо перезапустить службу **systemd-journald**:

```
# systemctl restart systemd-journald
```

Некоторые файлы в **/var/log/** — не текстовые, они являются неполноценными журналами и представляют собой «свалку событий» для служб авторизации и учёта. Текстовую информацию о входе пользователей в систему и выходе оттуда можно получить по команде **last**, а узнать о тех, кто в данный момент пользуется системой, помогут команды **w** и **who**.

Множество важной информации может дать анализ *загруженности* системы — сведения о процессорном времени и потреблении оперативной памяти (**ps**, **top**, **vmstat**), сведения об использовании дискового пространства (**du**, **df**) и сведения о работе сетевых устройств (**netstat**).

Глава 49. Документация

Каждый объект системы Linux обязательно сопровождается документацией, описывающей их назначение и способы использования. От пользователя системы не требуется заучивать все возможные варианты взаимодействия с ней. Достаточно понимать основные принципы её устройства и уметь находить справочную информацию.

Не пренебрегайте чтением документации: она поможет вам избежать многих сложностей, сэкономить массу времени и усилий при установке, настройке и администрировании системы, поможет найти нужное для работы приложение и быстро разобраться в нём.

49.1. Экранная документация

Почти все системы семейства UNIX, включая систему Linux, имеют экранную документацию. Её тексты содержат документацию по системным командам, ресурсам, конфигурационным файлам и т. д., а также могут быть выведены на экран в процессе работы.

49.1.1. man

Для доступа к экранной документации используется команда **man** (сокращение от manual). Каждая страница руководства посвящена одному объекту системы. Для того чтобы прочесть страницу руководства по программе, необходимо набрать **man** *название_программы*. К примеру, если вы хотите узнать, какие опции есть у команды **date**, вы можете ввести команду:

```
$ man date
```

Большинство экранной документации написано для пользователей, имеющих некоторое представление о том, что делает данная команда. Поэтому большинство текстов экранной документации содержит исключительно технические детали команды без особых пояснений. Тем не менее, экранная документация оказывается очень ценной в том случае, если вы помните название команды, но её синтаксис просто выпал у вас из памяти.

Поиск по описаниям **man** осуществляется командой **apropos**. Если вы точно не знаете, как называется необходимая вам программа, то поиск осуществляется по ключевому слову, к примеру, **apropos** *date* или при помощи ввода слова, обозначающего нужное действие, после команды **man -k** (например, **man -k** *copy*). Слово, характеризующее желаемое для вас действие, можно вводить и на русском языке. При наличии русского перевода страниц руководства **man** результаты поиска будут выведены на запрашиваемом языке.

«Страница руководства» занимает, как правило, больше одной страницы экрана. Для того чтобы читать было удобнее, **man** запускает программу постраничного просмотра текстов. Страницы перелистывают пробелом, для выхода из режима чтения описания команд **man** необходимо нажать на клавиатуре **q**. Команда **man man** выдаёт справку по пользованию самой командой **man**.

Документация в подавляющем большинстве случаев пишется на простом английском языке. Необходимость писать на языке, который будет более или менее понятен большинству пользователей, объясняется постоянным развитием Linux. Дело не в том, что страницу руководства нельзя перевести, а в том, что её придётся переводить всякий раз, когда изменится описываемый ею объект! Например, выход новой версии программного продукта сопровождается изменением его возможностей и особенностей работы, а следовательно, и новой версией документации.

Тем не менее, некоторые наиболее актуальные руководства существуют в переводе на русский язык. Свежие версии таких переводов на русский язык собраны в пакете *man-pages-ru*. Установив этот пакет, вы добавите в систему руководства, для которых есть перевод, и **man** по умолчанию будет отображать их на русском языке.

49.1.2. info

Другой источник информации о Linux и составляющих его программах — справочная подсистема *info*. Страница руководства, несмотря на обилие ссылок различного типа, остаётся «линейным» текстом, структурированным только логически. Документ *info* — это настоящий гипертекст, в котором множество небольших страниц объединены в дерево. В каждом разделе документа *info* всегда есть оглавление, из которого можно перейти к нужному подразделу, а затем вернуться обратно (ссылки для перемещения по разделам текста помечены *). Для получения вспомогательной информации о перемещении по тексту используйте клавишу **h**. Полное руководство *info* вызывается командой **info info**. Команда **info**, введённая без параметров, предлагает пользователю список всех документов *info*, установленных в системе.

49.2. Документация по пакетам

Дополнительным источником информации об интересующей вас программе, в основном на английском языке, является каталог **/usr/share/doc** — место хранения разнообразной документации.

Каждый пакет также содержит поставляемую вместе с включённым в него ПО документацию, располагающуюся обычно в каталоге **/usr/share/doc/имя_пакета**. Например, документация к пакету *file-5.44* находится в **/usr/share/doc/file-5.44**. Для получения полного списка файлов документации, относящихся к пакету, воспользуйтесь командой **rpm -qd** *имя_установленного_пакета*.

В документации к каждому пакету вы можете найти такие файлы как **README**, **FAQ**, **TODO**, **ChangeLog** и другие. В файле **README** содержится основная информация о программе — имя и контактные данные авторов, назначение, полезные советы и пр. **FAQ** содержит ответы на часто задаваемые вопросы; этот файл стоит прочитать в первую очередь, если у вас возникли проблемы или вопросы по использованию программы, поскольку большинство проблем и сложностей типичны, вполне вероятно, что в **FAQ** вы тут же найдёте готовое решение. В файле **TODO** записаны планы разработчиков на реализацию той или иной функциональности. В файле **ChangeLog** записана история изменений в программе от версии к версии.

Для поиска внешней информации о программе, например, адреса сайта программы в сети Интернет можно использовать команду **rpm -qi** *имя_установленного_пакета*. В информационном заголовке соответствующего пакета, среди прочей информации, будет выведена искомая ссылка.

Часть VII. Техническая поддержка продуктов «Базальт СПО»

Содержание

Глава 50. Покупателям нашей продукции

«Базальт СПО» предоставляет следующие виды технической поддержки:

- Поддержка продукта входит в стоимость лицензии и включает регулярный выпуск обновлений, исправление ошибок, устранение уязвимостей в течение всего срока жизни дистрибутива.
- Поддержка пользователей обеспечивает качественную эксплуатацию продукта. Техническая поддержка эксплуатации продуктов «Базальт СПО» оказывается в объеме SLA. Доступны три уровня SLA («Базовый», «Стандартный» и «Расширенный»).

Право на получение консультационной и технической поддержки вы приобретаете при покупке большинства продуктов торговой марки Альт. Сроки и объём помощи указаны в сертификате технической поддержки.

Условия технической поддержки можно найти на странице сайта «Базальт СПО»: http://www.basealt.ru/support.

Глава 51. Пользователям нашей продукции

Вне зависимости от того, скачали вы или же приобрели наш дистрибутив, задавать вопросы или обсуждать их с сообществом пользователей дистрибутивов Альт вы можете на форуме или в списках рассылки.

Помощь сообщества:

- Документация сообщества: https://altlinux.org
- Форум: https://forum.altlinux.org
- Списки рассылки: https://lists.altlinux.org
- Сообщить об ошибке: https://bugs.altlinux.org
- Репозиторий: https://packages.altlinux.org
- Сборочная среда: https://git.altlinux.org
- Telegram-канал сообщества: https://telegram.me/alt_linux

Ресурсы компании «Базальт СПО»:

- Сайт компании: https://www.basealt.ru
- Контакты: https://basealt.ru/contacts
- Новости обновлений безопасности: https://cve.basealt.ru

Форум и списки рассылки читают опытные пользователи, профессиональные системные администраторы и разработчики «Базальт СПО». Сообщество пользователей и специалистов окажет содействие в поиске ответа на ваш вопрос или посоветует выход из сложной ситуации. При обращении к данному виду помощи у вас нет гарантии на полноту и своевременность ответа, но мы стараемся не оставлять без ответа вопросы, задаваемые в списках.