СОДЕРЖАНИЕ

1 Аннотация

В данном документе приведено описание средств виртуализации ROSA Enterprise Linux Server. В основе виртуализации лежит библиотека libvirt. Управление виртуализацией доступно как из графического интерфейса ROSA Enterprise Linux Server, так и при помощи консольных команд. Руководство предназначено для пользователей, знакомых с базовыми возможностями Linux.

2 Используемые обозначения

Выделение важной информации

В документе для выделения информации, на которую стоит обратить внимание, используются примечания и иконки  ,  ,  .

Примечания выделяются подчеркиванием текста и содержат дополнительную информацию о конфигурировании RELS или о дополнительных возможностях команд.

Пример:

> Примечание: на nfs-клиенте просмотреть расшаренные папки можно командой
>
> `showmount -e 192.168.68.128`
>
> где `192.168.68.128` — адрес nfs-сервера.

Иконка  служит для выделения возможностей RELS, которые существенно упрощают работу с операционной системой, или сведений о готовой конфигурации, которую можно использовать при работе с RELS.

Пример:

>  Пример содержимого конфигурационного файла для кластера на двух хостах:
>
```
<cluster name="mycluster" config_version="2">
   <cman two_node="1" expected_votes="1"/>
   <clusternodes>
     <clusternode name="rosa2.int" nodeid="1">
         <fence>
         </fence>
     </clusternode>
     <clusternode name="rosa.int" nodeid="2">
```

```
            <fence>
            </fence>
          </clusternode>
        </clusternodes>
        <fencedevices>
        </fencedevices>
        <rm>
        </rm>
      </cluster>
```

Иконка  служит для того, чтобы обратить ваше внимание на те или иные особенности работы RELS. Эта информация не является критически необходимой, но мы рекомендуем строго следовать советам, приведенным с этой иконкой. Это поможет сэкономить время.

Пример:

 Перед запуском в промышленную эксплуатацию GFS мы рекомендуем проконсультироваться со специалистами технической поддержки РОСЫ, а также провести опытную эксплуатацию инфраструктуры с GFS в течении 2-3-х месяцев, чтобы понаблюдать за устойчивостью работы хранилища.

Иконка  служит для выделения критически важной информации. Внимательно читайте эту информацию и строго следуйте рекомендациям. В противном случае у вас могут возникнуть серьезные сбои или просто не запустятся важные сервисы RELS.

Пример:

 Примечание: иногда при соединении с хостом кластера через интерфейс luci возникает ошибка “`authentication to ricci agent failed`”. Данная проблема решается разрешением доступа к нужному порту в файерволе, а также установкой пароля ricci командой:

```
passwd ricci
```

на каждом хосте. Иногда также в этой ситуации стоит отключить selinux, если вы не уверены, что абсолютно правильно пользуетесь им.

2.1 Ссылки на другие разделы документа и рисунки

В документе используются ссылки на рисунки и другие разделы документа, для перехода по ссылке в PDF-версиях документа необходимо нажать клавишу CTRL и щелкнуть левой кнопкой мыши на ссылку.

Желаем приятного знакомства с возможностями RELS!

3 Введение в виртуализацию

Виртуализация — это процесс создания и предоставления набора виртуальных вычислительных ресурсов на базе реальных аппаратных ресурсов. В целом виртуализованными могут быть аппаратные платформы, операционные системы, сетевые ресурсы, хранилища данных. Обычно термин «виртуализация» используется применительно к нескольким операционным системам, которые исполняются на одной аппаратной платформе, однако выделенные им виртуальные ресурсы изолированы друг от друга. Существуют различные типы виртуализации:

1) Полная виртуализация
2) Паравиртуализация
3) Виртуализация уровня операционной системы

3.1 Полная виртуализация

Технология полной виртуализации использует виртуальную машину, которая осуществляет связь между гостевой операционной системой и непосредственно аппаратными средствами. Посредником между гостевой ОС и фактическим оборудованием является Монитор витруальных машин (гипервизор). Внутри гипервизора предусмотрены средства, выполняющие разделение и изоляцию аппаратных ресурсов потому что основные аппаратные средства не принадлежат операционной системе. При использовании данной технологии виртуализации могут выполняться только те операционные системы, которые поддерживают архитектуру физического оборудования.

Полная виртуализация имеет преимущества при консолидации серверов. В данном случае несколько серверов могут работать на одной физической вычислительной машине, при этом возможно резервирование виртуальных машин и их перенос на другой физический сервер, работающий по этой же технологии. Таким образом, может быть решена проблема недогруженности аппаратных ресурсов и балансировка нагрузки. Технология полной виртуализации является наиболее распространенной и применяется во многих программных продуктах: VMWare vSphere, VMWare Workstation, Xen, z/VM, Sun VirtualBox, Oracle VM, Microsoft Hyper-V (см.рис. 1).

Рисунок 1 — Полная виртуализация

3.2 Паравиртуализация

Технология паравиртуализации подразумевает модификацию ОС гостевой ВМ. Технология паравиртуализации разработана как альтернатива полной виртуализации, для того чтобы избежать проблем со старой архитектурой x86. При использовании данной технологии модифицируется только ядро ОС гостевой ВМ, а не библиотеки и приложения уровня пользователя. ОС гостевой ВМ общается с гипервизором на более высоком уровне. Гипервизор предоставляет ОС гостевой ВМ специальный программный интерфейс, с которым она и взаимодействует, вместо того, чтобы обращаться напрямую к аппаратным ресурсам. Продуктами, поддерживающими технологию паравиртуализации является то же программное обеспечение, которое обеспечивает реализацию технологий полной виртуализации (см. рис. 2).

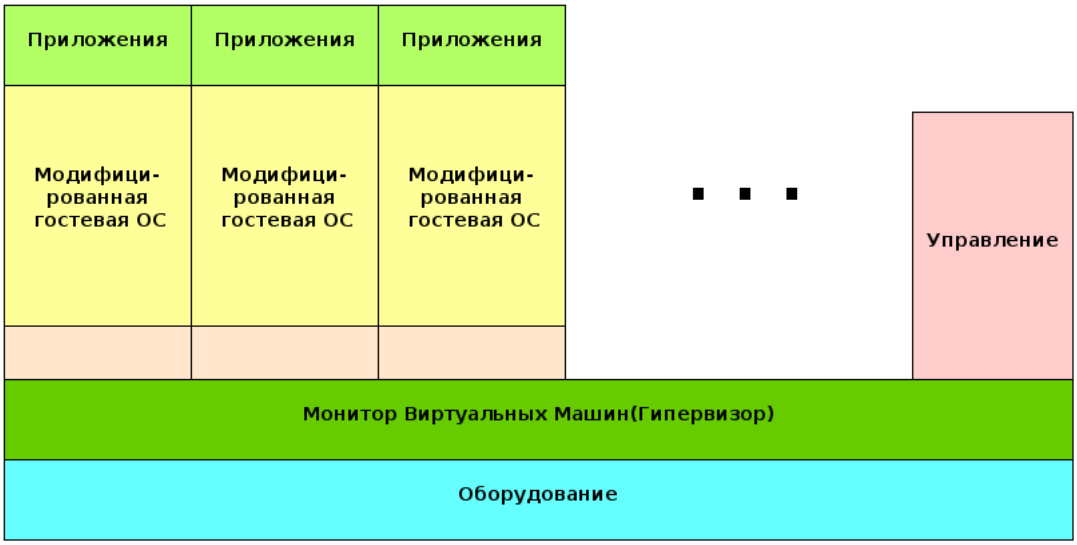

Рисунок 2 — Паравиртуализация

3.3 Виртуализация уровня операционной системы

Виртуализация уровня ОС, также имеющая название виртуализации с использованием виртуальных контейнеров, виртуализирует сервера на уровне операционной системы (ядра). Этот метод поддерживает единственную операционную систему и просто изолирует независимые серверы друг от друга. При этом создаются изолированные контейнеры на одном физическом сервере и экземпляре ОС. Для гостевого ПО создается только собственное сетевое и аппаратное окружение.

Технология виртуализации уровня операционной системы создает существенно меньшую (почти нулевую) дополнительную нагрузку на компьютер по сравнению с гипервизорами. Однако основным ее недостатком является поддержка только однородных сред, то есть в контейнере поддерживается только то программное обеспечение, которое может функционировать на основной ОС.

Технология виртуализации уровня операционной системы в основном реализована в Parallels Virtuozzo Containers, продукте компании Parallels (см. рис. 3).

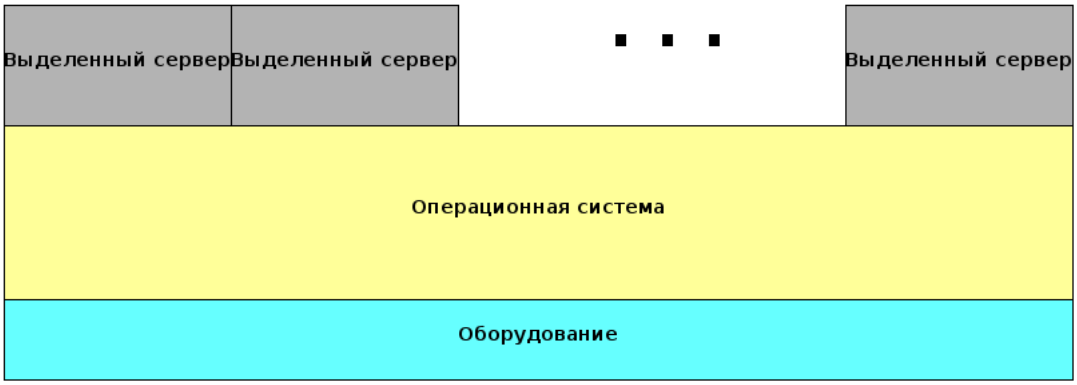

Рисунок 3 — Виртуализация уровня операционной системы

3.4 Введение в миграцию

Миграция — это процесс переноса гостевой виртуальной машина между физическими хостами. Миграция возможна благодаря виртуализации аппаратных ресурсов, которая обеспечивает выполнение ВМ в виртуальном окружении без прямой привязки к конкретным реальным аппаратным ресурсам. Миграция может быть двух типов: оффлайн-миграция и «живая» миграция.

3.4.1. Offline-миграция

При данном типе миграции выполнение виртуальной машины, подлежащей переносу, останавливается, ВМ переносится на другой хост и ее выполнение возобновляется на новых свободных вычислительных ресурсах.

3.4.2. Живая миграция

Живая миграция обеспечивает перенос активной (без прекращения выполнения) виртуальной машины между хостами. В процессе миграции и переноса памяти виртуальной машины происходит одновременный перенос дискового пространства (хранилища данных), принадлежащей ВМ. Технологии живой миграции хранилищ данных обеспечивают миграцию виртуальных машин без временного простоя.

3.4.3. Преимущества миграции

Балансировка нагрузки

Возможность перенести несколько виртуальных машин с перегруженного физического хоста на менее загруженный.

Обновления аппаратуры

Возможность безопасно перенести ВМ между хостами, а в это время заменить аппаратное обеспечение исходного хоста, при этом ВМ не будет остановлена.

Энергосбережение

Возможность перераспределить ВМ между хостами таким образом, чтобы сделать некоторые аппаратные хосты свободными от ВМ. Эти хосты можно выключить до того момента, пока не потребуются дополнительные аппаратные ресурсы.

Географическое распределение хостов

Возможность переносить ВМ на виртуальные хосты, которые расположены в другой географической точке. Это позволит уменьшить время задержки передачи пакетов пользователям, обеспечить дополнительную защиту информации или другие преимущества. Для сохранения данных ВМ необходимо будет использовать распределенные сетевые хранилища. Для управления ими используется утилита `libvirt`.

3.5 Несколько фактов о виртуализации

3.5.1. Затраты при виртуализации

На наш взгляд, в целом использование виртуализованных систем позволяет существенно снизить TCO бизнеса в среднесрочной и долгосрочной перспективе. Помните, что хотя переход к виртуализованной платформе требует затрат, однако дальнейшее использование виртуализации хостов обеспечит экономию средств на персонале, энергопотреблении и вынужденных простоях ИТ-сервисов. Переход к виртуализованной платформе требует, как и любой проект, детального планирования. Мы рекомендуем использовать заблаговременное планирование перехода и инструменты анализа рентабельности инвестиций (ROI analyses) для подбора оптимального варианта виртаулизованной платформы

для предприятия.

Почему виртуализация снижает издержки?

Меньше затрат энергии – возможность консолидировать ВМ на минимально необходимом числе физических хостов позволяет временно отключить остальные, а, значит, и сократить энергопотребление.

Увеличенный срок работы программного обеспечения. Иногда устаревшее программное обеспечение (которому еще не подготовлена адекватная замена) уже не запускается на новых аппаратных платформах. Для запуска такого ПО можно использовать виртуальное окружение.

Уменьшение необходимых аппаратных ресурсов — возможность консолидировать ВМ на нескольких хостах высвобождает аппаратные ресурсы, которые можно использовать для других нужд организации.

3.5.2. Обучение и виртуализация

В целом, изучение виртуализации не сложнее обучения любому другому ИТ-процессу. Значительная часть навыков работы с реальной аппаратурой вполне применима для работы с виртуальным окружением, все остальные необходимые навыки могут быть легко адаптированы

3.5.3. Быстродействие

Ранее виртуализованные платформы существенно уступали в быстродействии реальному аппаратному обеспечению. Однако в последние годы разница в быстродействии зачастую не превышает 10% или вообще отсутствует.

3.5.4. Переносимость

Виртуализация обеспечивает широкие возможности переносимости: можно копировать или переносить виртуальные машины, в которых необходимо протестировать или внести изменения в ПО. Эти действия не будут оказывать негативного эффекта на всю ИТ-систему, поскольку каждая ВМ выполняется изолированно.

3.5.5. Восстановление

Виртуализация существенно упрощает процесс восстановление данных и ИТ-сервисов. В случае серьезного сбоя при использовании физической платформы может понадобиться полная переустановка операционной системы, что повлечет за собой большие трудозатраты на восстановление потерянных данных. При виртуализации возможно использовать живую миграцию ВМ и перезапускать машину на другом физическом хосте.

3.5.6. Безопасность

Механизмы SELinux используются для повышения безопасности системы. SELinux (Security-Enhanced Linux — «Linux с улучшенной безопасностью») реализует системы мандатного управления доступом (MAC), которые могут работать параллельно с классической дискреционной системой контроля доступа.

SELinux позволяет задавать явные правила, в соответствии с которыми субъекты (пользователи и программы) смогут обращаться к объектам (файлам и устройствам). Основу SELinux составляют три технологии:

- мандатный контроль доступа;
- ролевой доступ RBAC;
- система типов (доменов).

Использование SELinux позволяет существенно увеличить безопасность используемых хостов и виртуальных машин.

3.6 KVM и libvirt

KVM (Kernel-based Virtual Machine) — это программное решение, обеспечивающее виртуализацию на Linux-платформах. Поддерживается виртуализация на платформе x86 (x86_64). В гостевых ВМ могут быть запущены операционные Windows и Linux.

Гипервизор KVM поддерживает технологию **overcommitting**, которая позволяет выделять виртуальным машинам в совокупности больше аппаратных ресурсов (процессоров и оперативной памяти), чем есть на реальном хосте. Например, при наличие 50 ГБ оперативной памяти на хосте, можно выделить, скажем, двум виртуальным машинам по 40 ГБ каждой. Однако это не означает, что данные ресурсы будут выделены ВМ одновременно. Дело в том, что каждой ВМ в разное время необходимо различное количество ресурсов. В нашем примере это будет означать, что, например, ВМ с СУБД, основная нагрузка на которую приходится на рабочее время, будет потреблять 40 ГБ днем, а сервер (ВМ) бэкапирования, основная нагрузка на который приходится на внерабочее время, будет потреблять 40 ГБ ночью. Таким образом, overcommitting позволит существенно более эффективно распоряжаться доступными аппаратными ресурсами.

 Будьте осторожны! Некорректное использование overcommitting может привести к нестабильной работе всего физического хоста.

KVM также поддерживает технологию экономного распределения дискового пространства (thin provisioning). Которая позволяет выделять гостевым реальным машинам суммарно больше дискового пространства, чем реально есть в физическом хранилище. Естествен-

но, что выделенное пространство не будет одновременно доступно каждой ВМ. Однако на практике многие ВМ будут использовать не более 40% выделенного им пространства, таким образом, суммарно дискового пространства будет достаточно. Плюсом данной технологии является также относительно малая фрагментация дискового пространства, что гарантирует отсутствие снижения скорости работы с дисковым пространством при использовании *thin provisioning*.

 Будьте осторожны! Использование *thin provisioning* может привести к нестабильной работе всего физического хоста.

KVM поддерживает также технологию **Kernel SamePage Merging (KSM)**, которая позволяет создавать общие для гостевых виртуальных машин страницы оперативной памяти. Эти страницы могут быть использованы для общих ВМ библиотек или других, часто используемых различными ВМ, данных.

В составе RELS используется свободная библиотека **libvirt**, которая предоставляет интерфейс к различным технологиям виртуализации, в том числе KVM.

Libvirt обеспечивает:

– Интерфейс управления локальными и сетевыми ВМ;

– API для выполнения всех стандартных операций над ВМ, включая: создание, запуск, приостановку, выключение, удаление, миграцию и др.

 Libvirt предоставляет только те операции над ВМ, которые поддерживаются гипервизором.

RELS поддерживает основные средства управления функциями Libvirt: `virsh`, `virt-manager`:

– **virhs** — это консольное средство управления функциями Libvirt. Это основное средство для администрирования виртуализации с использованием скриптов.

– **virt-manager** — это графическая утилита управления. В графическом интерфейсе поддерживаются основные операции над ВМ, просмотр статистики их работы, просмотр информации о доступных устройствах.

4 Менеджер виртуальных машин

Менеджер виртуальных машин (virt-manager) предоставляет графический интерфейс для управления виртуальными машинами.

Virt-manager не входит в стандартный вариант устанавливаемого RELS. Для работы с virt-manager необходимо установить его, библиотеку libvirt, гипервизор kvm и соответ-

ствующие пакеты из зависимостей. Для установки пакетов воспользуйтесь консолью или специальной утилитой «Установка и удаление программ» (см. Руководство пользователя ROSA Enterprise Linux Server).

В процессе установки RELS (см. Руководство пользователя ROSA Enterprise Linux Server) возможно установить все необходимые для виртуализации компоненты сервера, в том числе и менеджер виртуальных машин. Для этого на шаге выбора репозитория (см. Руководство пользователя ROSA Enterprise Linux Server) выберите «Настроить сейчас», в появившемся окне (рис. 4) отметьте все компоненты в группе «Виртуализация» и продолжите установку. В этом случае вы получите возможность работы с менеджером виртуальных машин и иными возможностями виртуализации сразу после установки RELS.

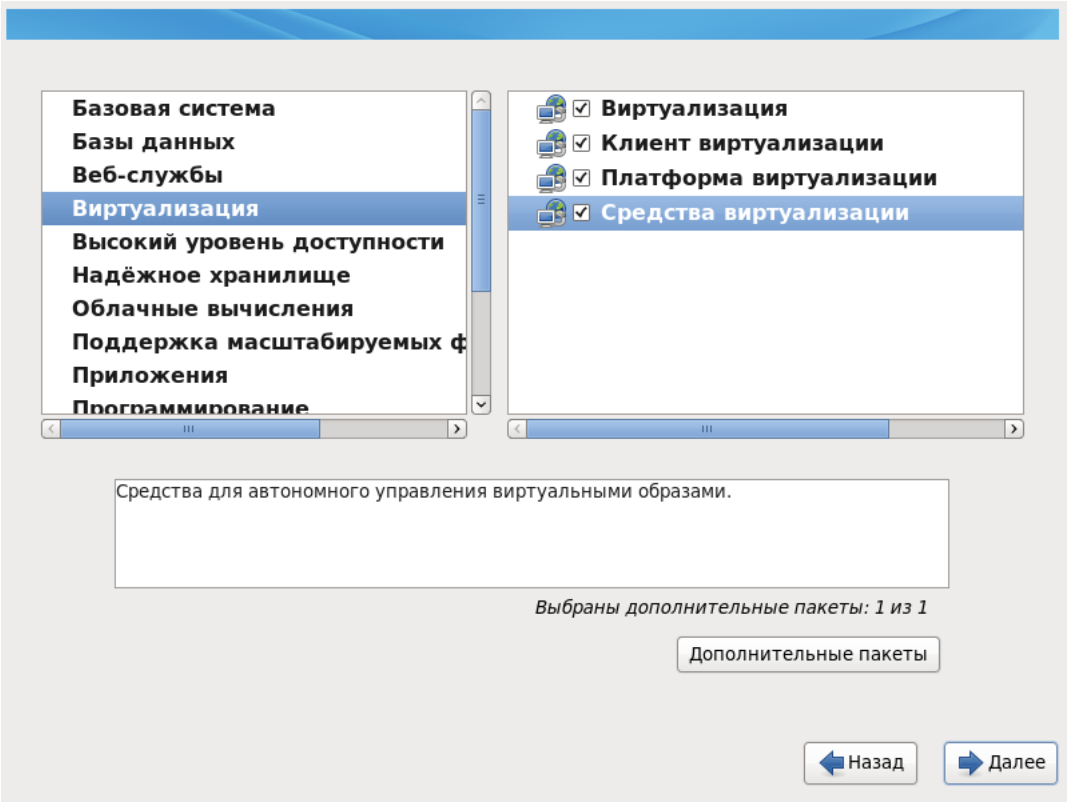

Рисунок 4 — Выбор пакетов виртуализации

Менеджер виртуальных машин обеспечивает графическое представление состояния гипервизора и гостевых виртуальных машин, а также позволяет выполнять следующие операции:

1) создание гостевых ВМ;
2) выделение ВМ оперативной памяти;
3) выделение ВМ ресурсов процессора;
4) мониторинг выполнения ВМ;
5) сохранение, восстановление, приостановка, возобновление, выключение и запуск

ВМ;

6) работа с текстовой и графической консолями;

7) выполнение клонирования и миграции (поддерживается оба варианта: offline, «живая»).

4.1 Запуск менеджера виртуальных машин

Для запуска Менеджера виртуальных машин нужно в главном меню RELS выбрать пункты **Системные —> Менеджер виртуальных машин** (рис. 5).

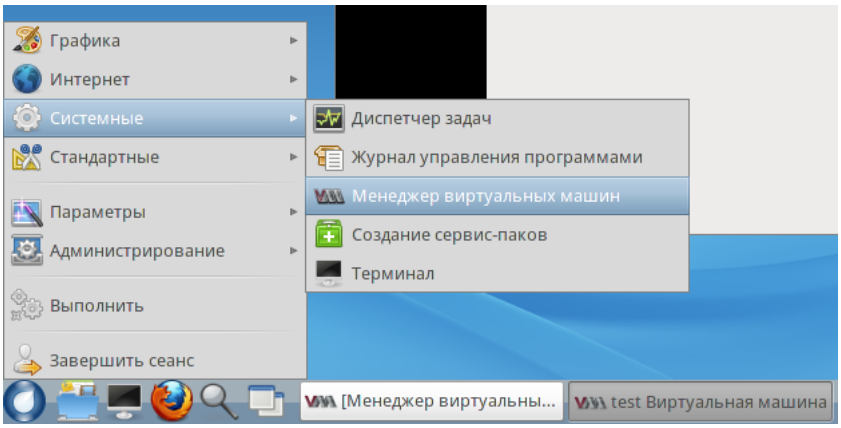

Рисунок 5 — Запуск менеджера виртуальных машин

После этого откроется главное окно Менеджера виртуальных машин (рис. 6).

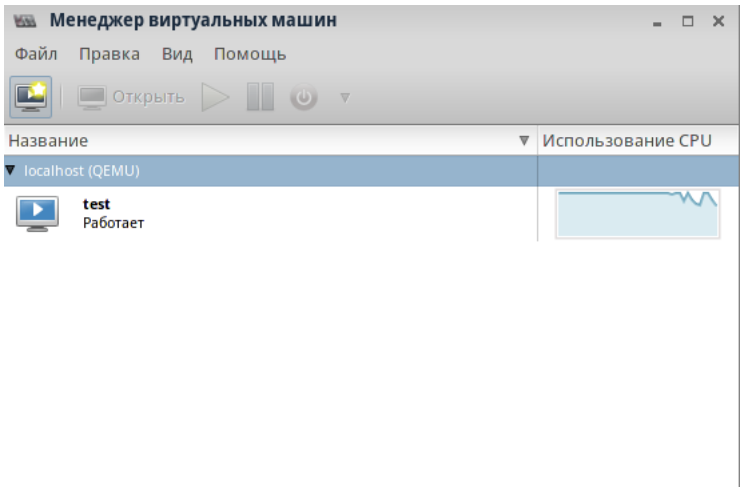

Рисунок 6 — Менеджер виртуальных машин

Примечание: запустить Менеджер виртуальных машин можно удаленно, используя ssh. Для этого необходимо выполнить следующие команды:

```
ssh -X <host's address>
[remotehost]# virt-manager
```

4.2 Главное окно Менеджера виртуальных машин

В главном окне виртуальных машин (рис. 6) отображается список созданных гостевых виртуальных машин, их состояние, а также дополнительная информация, например, о загрузке процессоров виртуальных машин. Также доступны кнопки с основными операциями с ВМ:

– Создать ВМ ( );

– Открыть ВМ (Открыть);

– Запустить ВМ ( );

– Приостановить ВМ ( );

– Выключить ВМ ( ).

Для того, чтобы открыть гостевую ВМ, также можно дважды нажать правой кнопкой мыши на новую ВМ. ВМ открывается в новом окне RELS, (рис. 7).

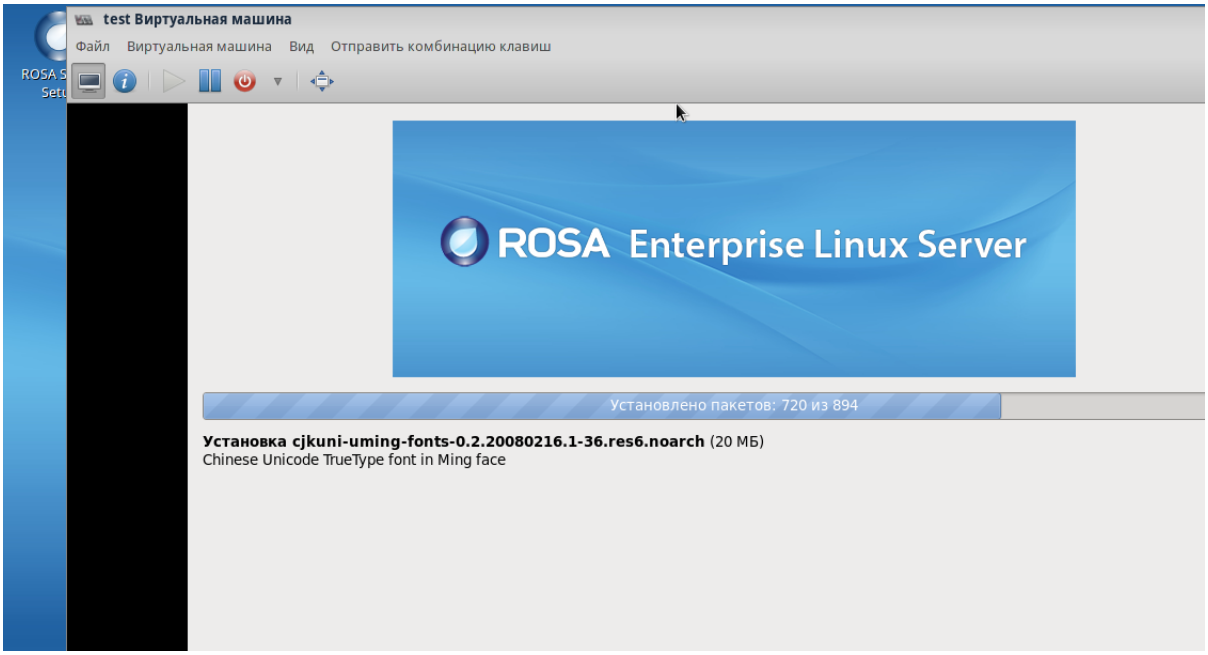

Рисунок 7 — Активная виртуальная машина

4.3 Параметры виртуального оборудования

Чтобы просмотреть параметры выделенного ВМ машине оборудования, в окне ВМ нажмите на кнопку (для возврата к ВМ нужно нажать на кнопку), после этого откроется страница с перечнем виртуального оборудования ВМ и его параметрами (рис. 8).

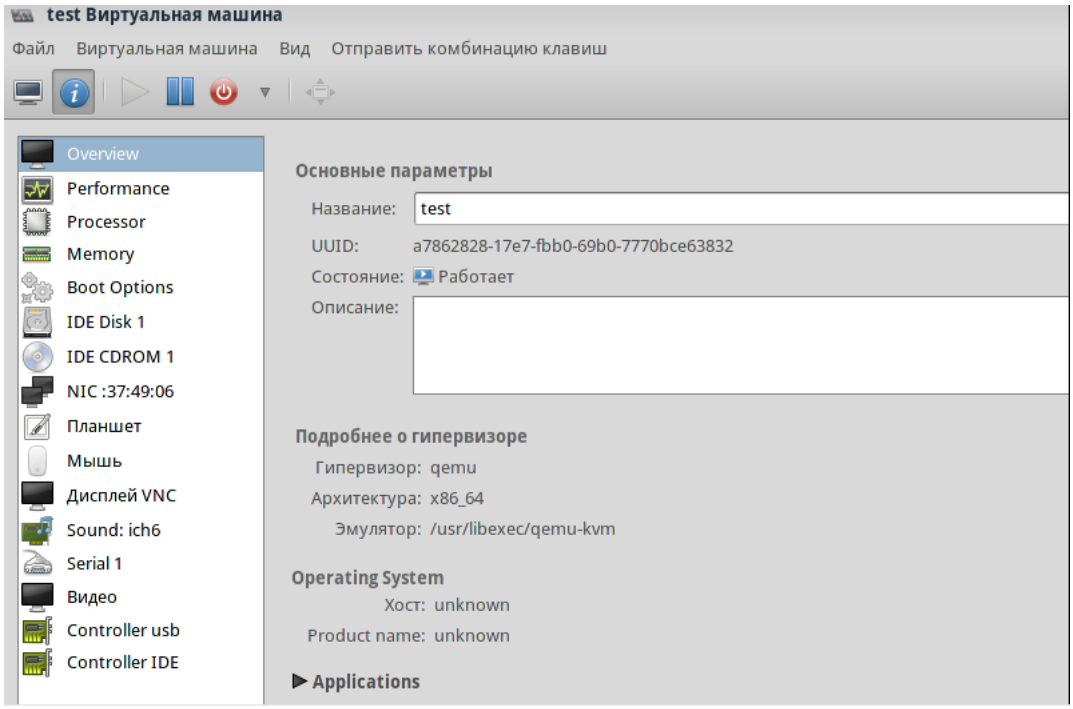

Рисунок 8 — Параметры оборудования ВМ

4.4 Графическая консоль ВМ

В графической консоли ВМ (вызывается двойным нажатием по ВМ из главного окна Менеджера виртуальных машин) можно работать в обычном режиме с ОС ВМ в ее графическом интерфейсе (см. рис. 9).

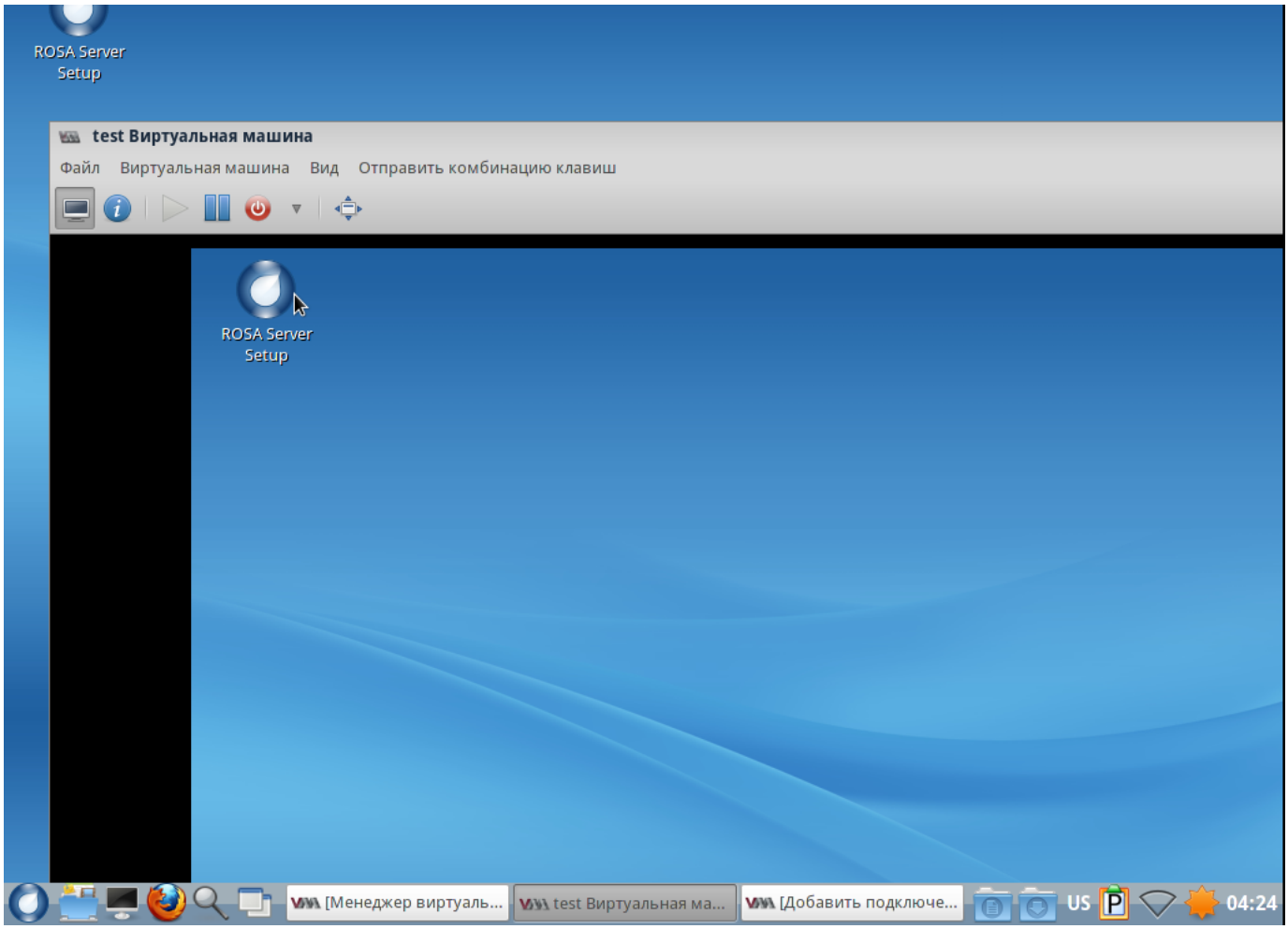

Рисунок 9 — Запущенная ВМ

Графическая консоль (VNC) считается небезопасной, однако в RELS предусмотрены дополнительные механизмы обеспечения безопасности ВМ. Кроме того, виртуальным машинам доступен только локальный адрес хоста (127.0.0.1). При удаленном подключении возможно использовать протокол TLS для обеспечения большей безопасности управления гостевыми ВМ.

4.5 Создание удаленного подключения к ВМ

Чтобы установить соединение с удаленной ВМ в Менеджере виртуальных машин следует:

1) Нажать на кнопку **Файл**, после чего нажать на кнопку **Создать новое подключение** (рис. 10).

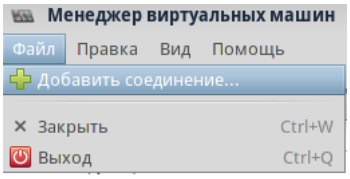

Рисунок 10 — Создать новое подключение

2) В появившемся окне (рис. 11), выберите гипервизор (QEMU/KVM используется в RELS). Если подключение будет осуществляться к локальному хосту, то отмечать пункт *Connect to remote host* не нужно. В противном случае (если необходимо подключиться к удаленному хосту), установите галочку. Выберите тип подключения (доступные типы: SSH (рекомендуемый), SSL/TLS, TCP), введите имя пользователя, под которым будет производиться подключение к хосту, а также введите имя хоста. После этого необходимо нажать на кнопку **Подключиться**. Далее откроется окно подтверждения подключения, в котором необходимо ввести “yes”, после чего ввести пароль пользователя, под которым осуществляется подключение. Подключение к удаленной ВМ будет завершено, новая ВМ отобразится в главном окне менеджера виртуальных машин.

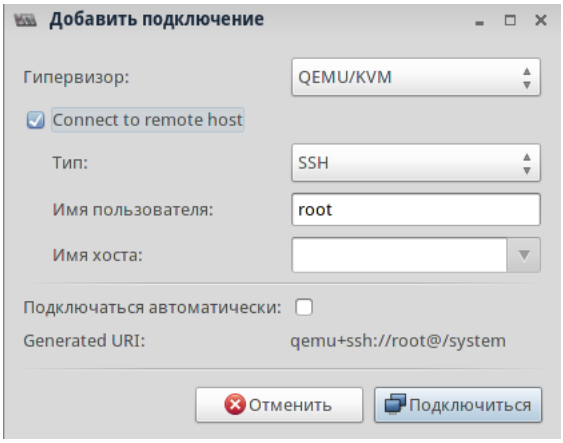

Рисунок 11 — Добавить подключение

4.6 Просмотр параметров ВМ

Для просмотра параметров ВМ в главном окне Менеджера виртуальных машин выделите нужную ВМ, выберите **Правка —> Подробнее о виртуальной машине**, в окне виртуальной машины нажмите на кнопку  . В Менеджере виртуальных машин откроется окно с параметрами выбранной виртуальной машины (рис. 12).

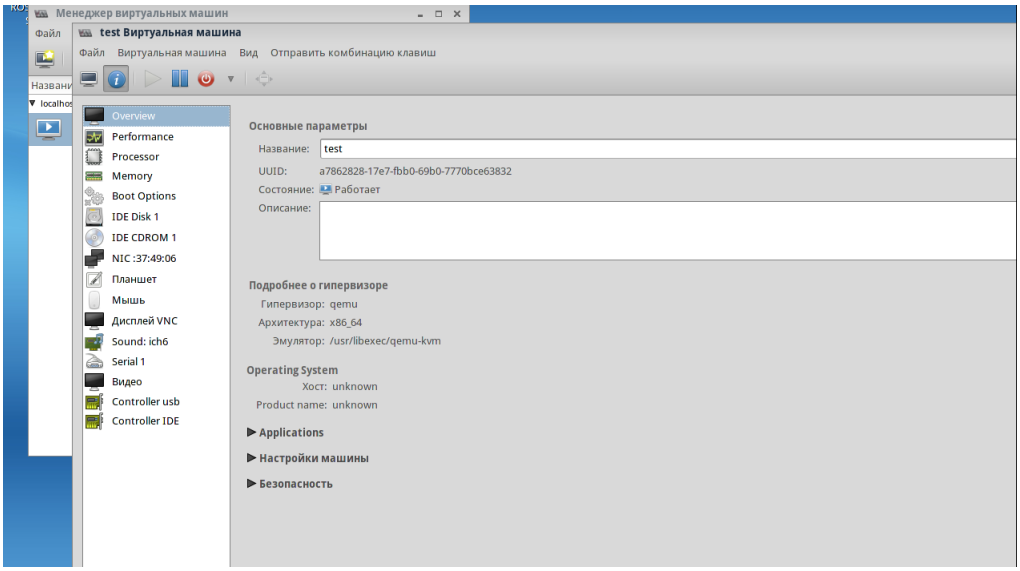

Рисунок 12 — Параметры ВМ

Выберите пункт **Performance** в левой боковой панели. Во вкладке **Performance** можно просмотреть сведения об использовании процессора, оперативной памяти, дискового и сетевого ввода/вывода (рис. 13).

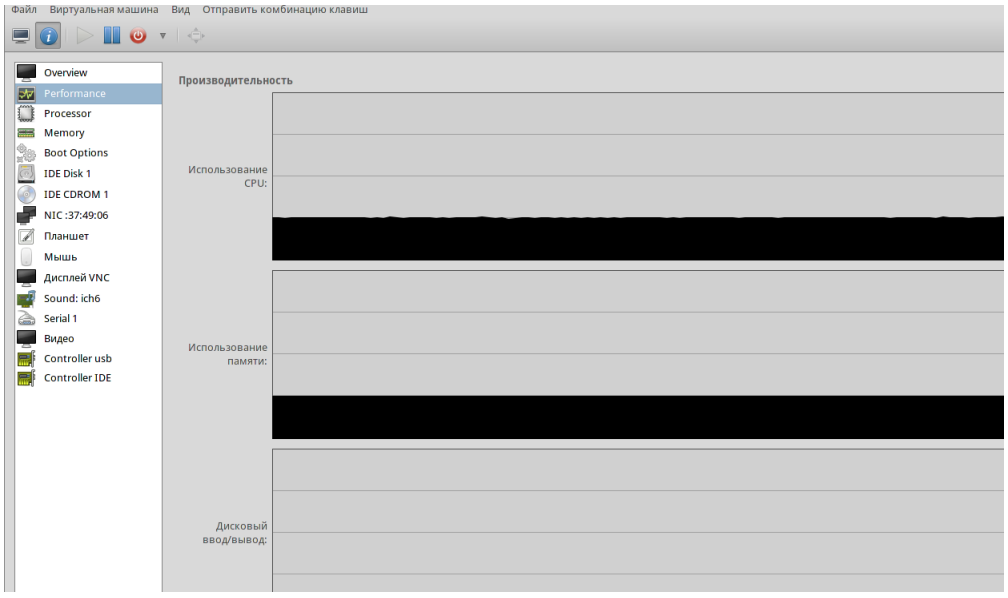

Рисунок 13 — Performance

Выберите пункт **Processor** в левой боковой панели. В этой вкладке можно просмотреть и настроить параметры выделения процессоров ВМ, такие как количество процессоров и ядер, модель выделяемых процессоров, их топологию и др. (рис. 14).

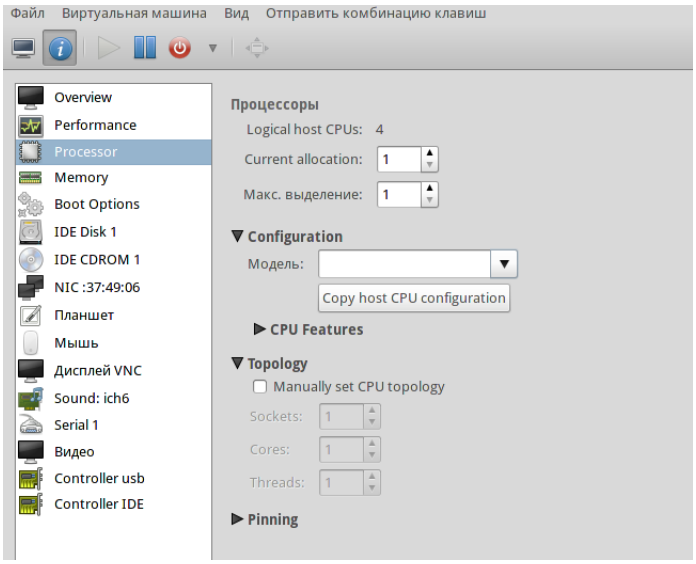

Рисунок 14 — Processor

Выберите пункт Memory в левой боковой панели. В этой вкладке можно просмотреть текущее и настроить новое распределение оперативной памяти для ВМ (рис. 15).

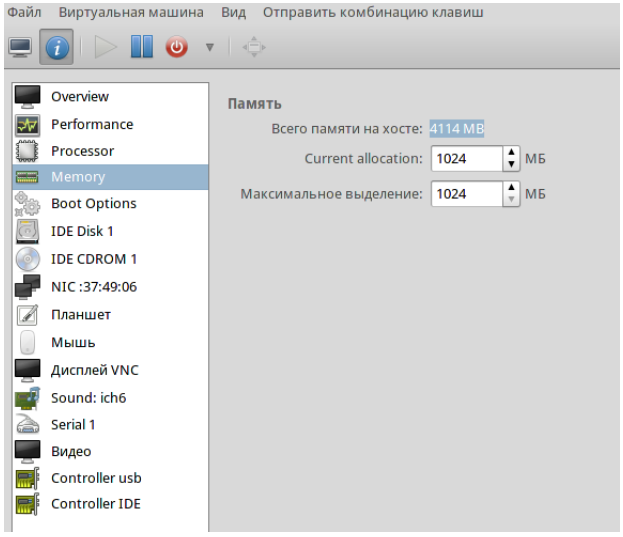

Рисунок 15 — Memory

Каждый виртуальный диск, выделенный ВМ, отображается в левой боковой панели. В нашем примере к ВМ подключено два виртуальных диска: **IDE Disk 1** и **IDE CDROM 1**. Кликните по нужному диску, чтобы просмотреть его параметры, изменить их или удалить диск (рис. 16).

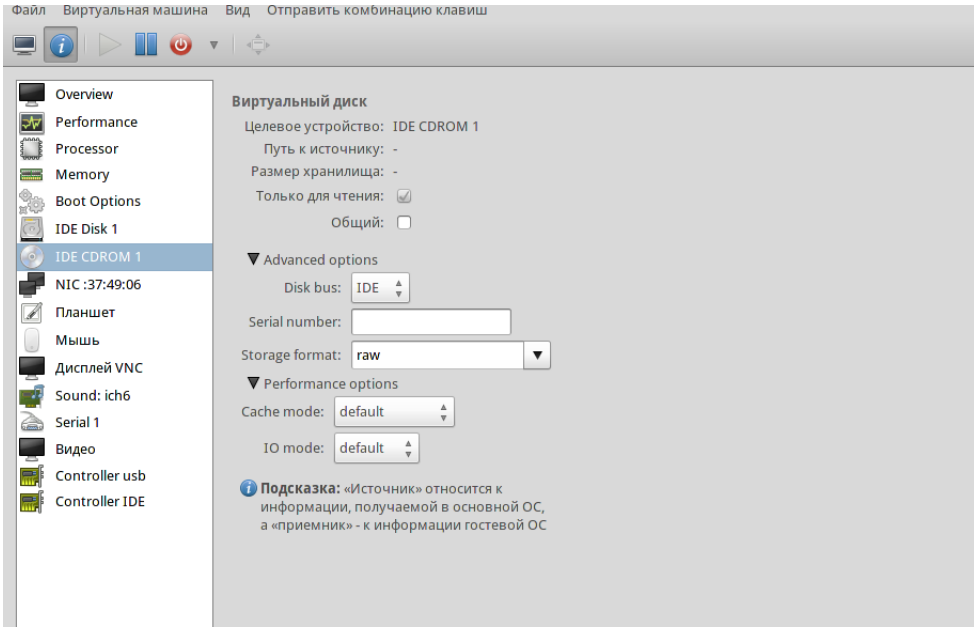

Рисунок 16 — Виртуальный диск

Каждая виртуальная, доступная ВМ, отображается в левой боковой панели параметров ВМ. В нашем примере это сеть **NIC:37:49:06**. Здесь можно просмотреть или изменить параметры сети, а также удалить сеть, нажав на нужный пункт левой кнопкой мыши (рис. 17).

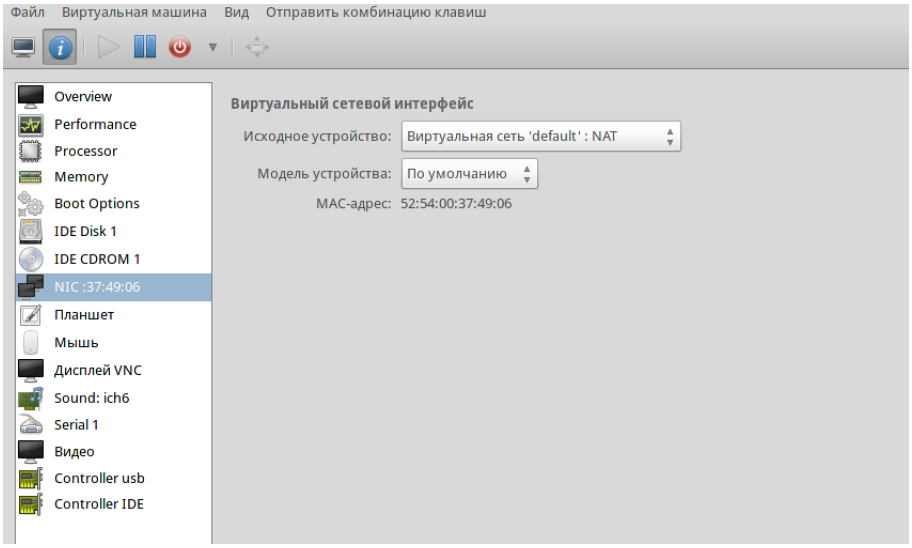

Рисунок 17 — Виртуальный сетевой интерфейс

4.7 Настройка статистики ВМ

Чтобы настроить отображение статистики выполнения ВМ, в главном окне Менеджера виртуальных машин нажмите **Правка —> Параметры** (рис. 18).

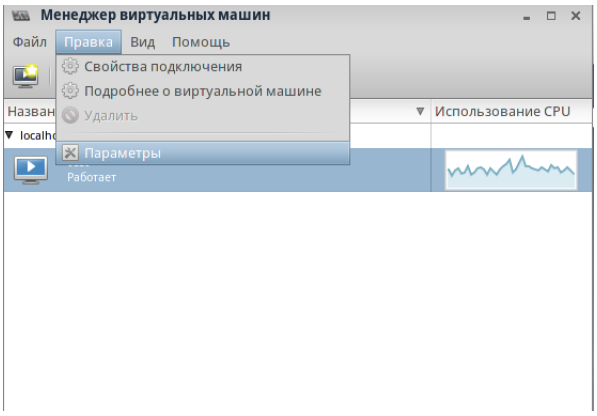

Рисунок 18 — Запуск отображения статистики

В появившейся оконной форме «Настройки» выберите вкладку «Статистика» (рис. 19).

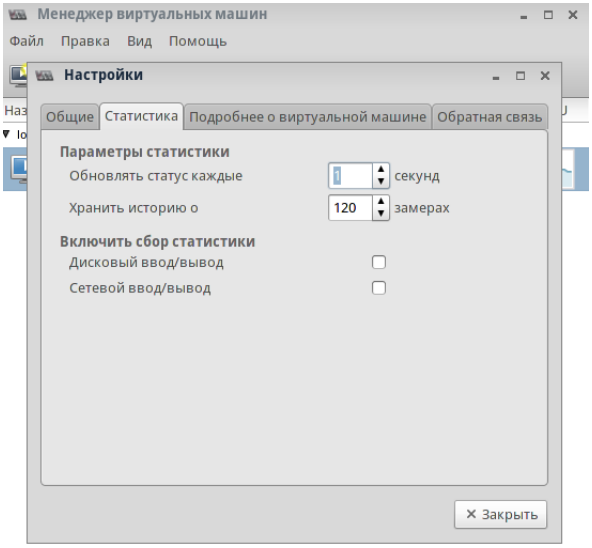

Рисунок 19 — Настройка статистики

В оконной форме настройки статистики можно настроить периодичность обновления статистики, продолжительность ее хранения, а также включить сбор статистики о дисковом и/или сетевом вводе-выводе, отметив нужные пункты.

4.8 Просмотр статистики ВМ

Статистику выполнения ВМ можно просмотреть в главном окне Менеджера виртуальных машин. Для настройки параметров отображения статистики воспользуйтесь вкладкой «Вид» в главном окне. Здесь можо выбрать, какие именно составляющие статистики нужно отображать на главной странице Менеджера. Доступны следующие статистические данные (рис. 20).

- **Guest CPU Usage** — использование процессора гостевыми ВМ;
- **Host CPU Usage** — использования процессора на хосте;
- **Дисковый ввод/вывод** — использование дисковой системы;

– **Сетевой ввод/вывод** — использование сети;

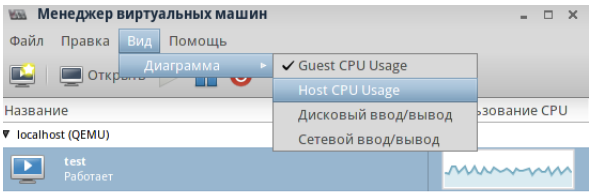

Рисунок 20 — Настройка отображения статистики

Чтобы иметь возможность просматривать ту или иную составляющую статистики, необходимо ее выбрать, нажав на нужный пункт мышью. После настройки отображения появится возможность просмотра статистики в главном окне Менеджера виртуальных машин (рис. 21).

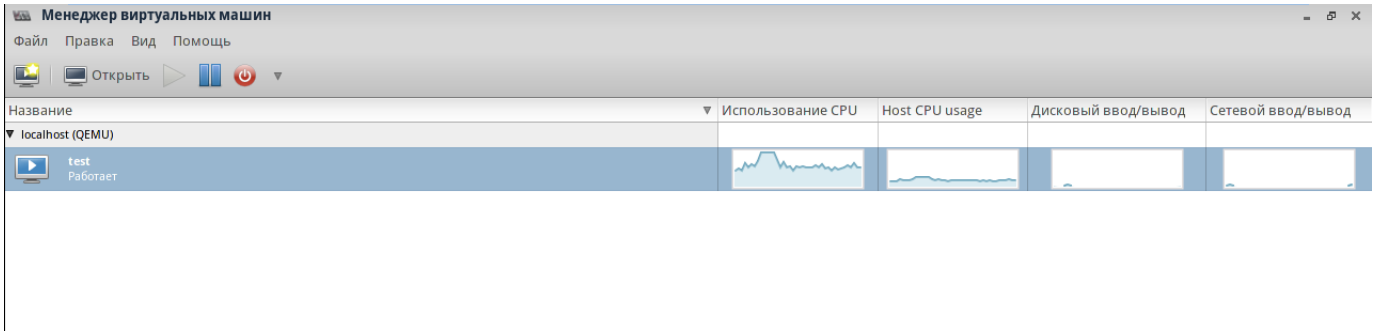

Рисунок 21 — Просмотр статистики

4.9 Миграция ВМ

В Менеджере виртуальных машин можно запустить процесс миграции любой виртуальной машины. Например, следующим образом:

1) Добавьте в Менеджере виртуальных машин подключение к удаленному хосту (рис. 21)

Сделать это можно, нажав на главной странице **Файл —> Добавить соединение**, введя параметры подключения, а также пароль пользователя удаленного хоста (рис. 22)

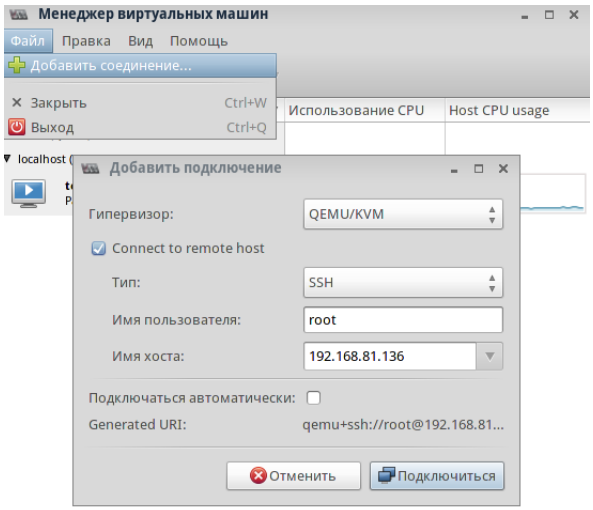

Рисунок 22 — Добавить соединение

2) Убедитесь, что подключение к удаленном хосту успешно добавлено (рис. 23)

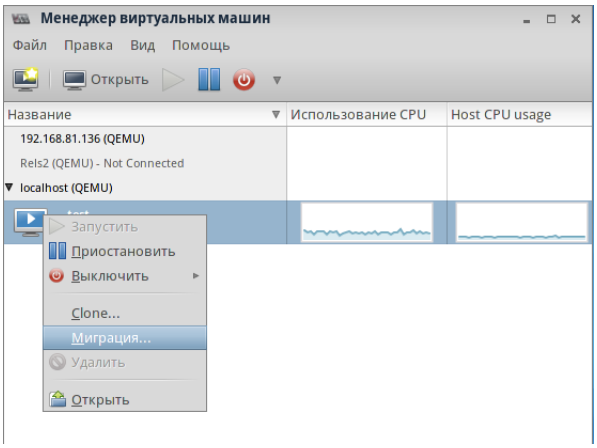

Рисунок 23 — Добавленное подключение

3) Выберите ВМ, которую нужно перенести, нажмите на нее правой кнопкой мыши и выберите пункт «Миграция» (рис. 24)

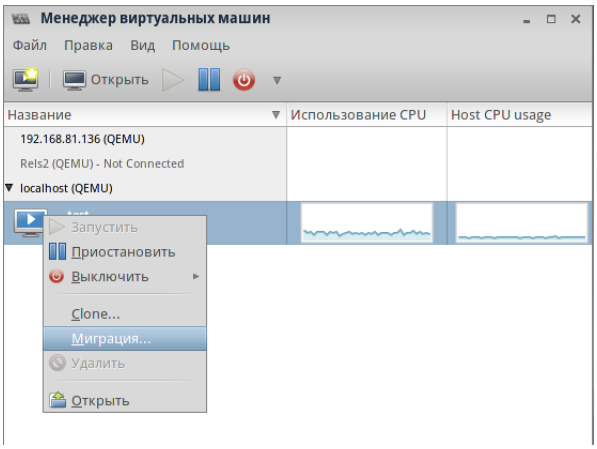

Рисунок 24 — Запуск миграции

4) Выберите хост, на который нужно выполнить миграцию (рис. 25) и нажмите кнопку *Миграция*. Прим. Перенести можно ВМ, которая выполняется.

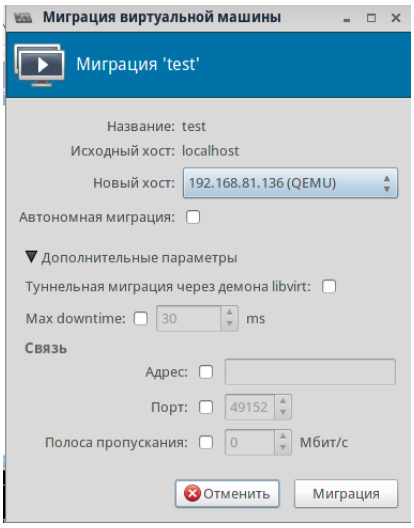

Рисунок 25 — Параметры миграции

Далее начинается процесс живой миграции (рис. 26). Обратите внимание, что в процессе миграции ВМ продолжает выполняться, т.е. все ИТ-сервисы, запущенные в ВМ, сохраняют полную работоспособность.

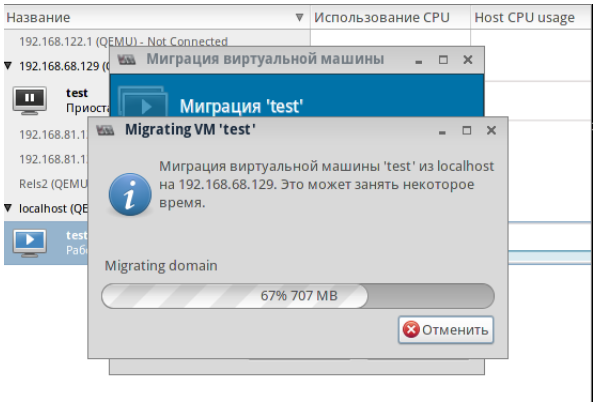

Рисунок 26 — Процесс миграции

По завершении процесса миграции можно будет увидеть ВМ, продолжающую выполнение на новом хосте (рис. 26).

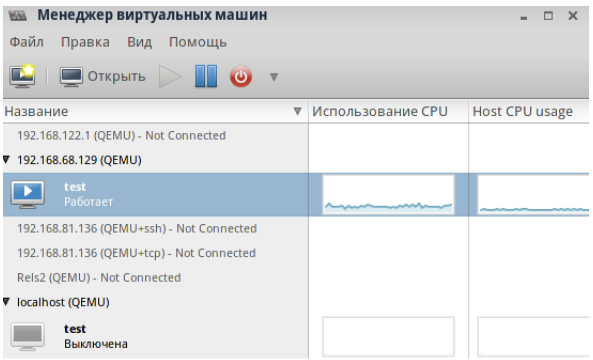

Рисунок 27 — ВМ на новом хосте

Можете подключиться к ВМ на новом хосте можно, нажав *Открыть*. При этом потребуется подтвердить пароль и авторизоваться в консоли удаленного хоста (рис. 28).

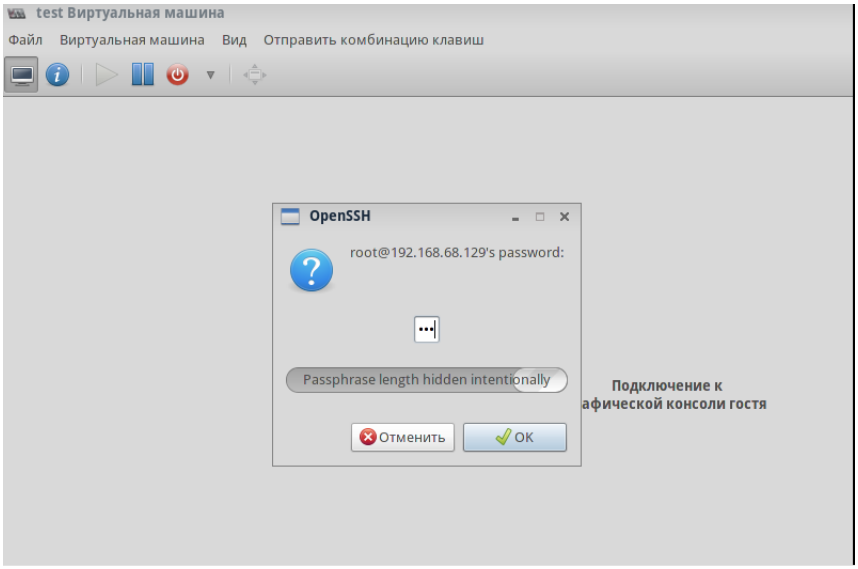

Рисунок 28 — Авторизация

После успешного прохождения авторизации можно будет увидеть ВМ в графической консоли (рис. 29).

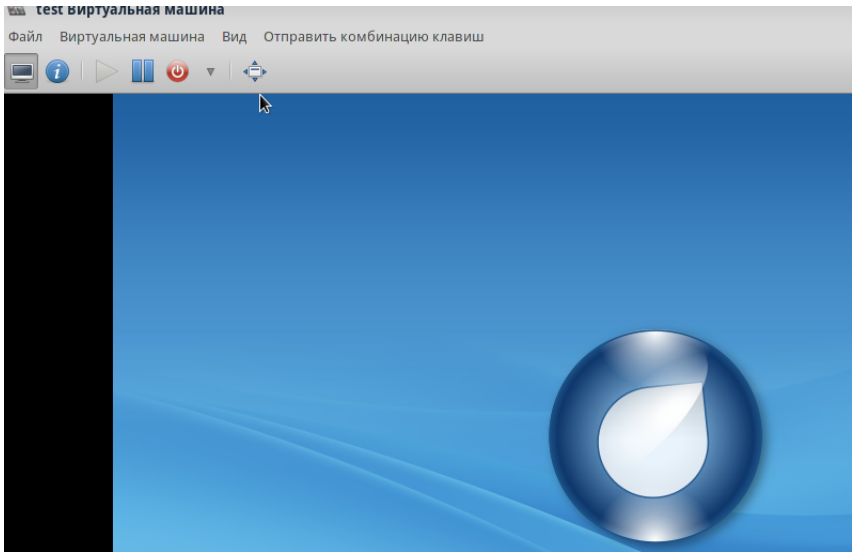

Рисунок 29 — ВМ доступ со старого хоста

Может быть проще, однако, открыть Менеджер виртуальных машин на новом хосте и в нем открыть мигрировавшую ВМ, это показано на рис. 30.

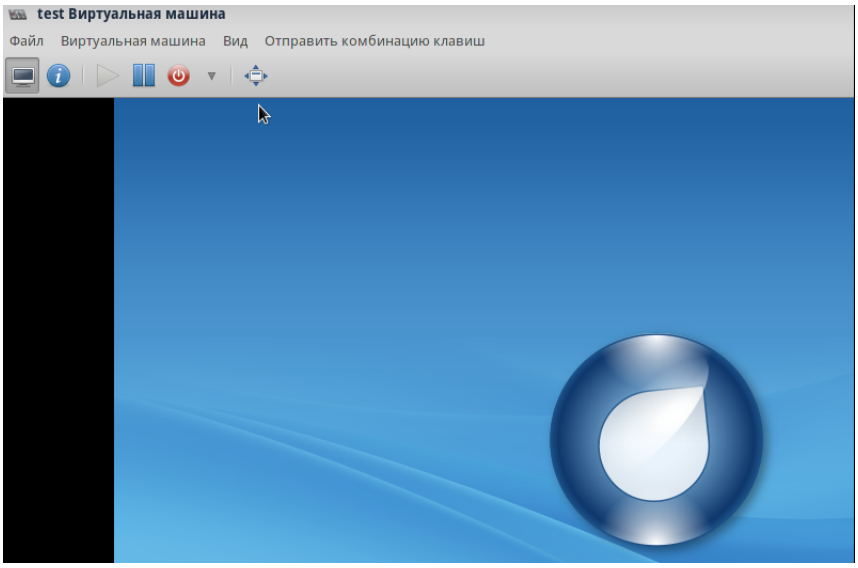

Рисунок 30 — Мигрировавшая ВМ

Отдельно остановимся на типовых недостатках конфигурирования хостов и сетей, приводящих к ошибках в процессе миграции.

4.9.1. **Отсутствие общего дискового пространства**

Помните, что хотя при миграции ВМ происходит только перенос процессоров и оперативной памяти, миграция невозможна без доступного обоим хостам хранилища. В случае, если оно не доступно, при запуске процесса миграции будет получена ошибка: «не удалось открыть файл <путь к файлу с образом: нет такого файла или каталога> ».

Как создать общую папку?

Прежде всего установите службы nfs на обоих хостах (это можно сделать с помощью ROSA Directory Server), запустите службу nfs на обоих хостах: `/sbin/service nfs start`

На nfs-сервере:

1) Создайте каталог /exports/images:

```
mkdir /exports/images
```

2) 2. В файле /etc/exports пропишите: `/var/lib/libvirt/images 192.168.68.129(rw,async, no_root_squash)`

– `/var/lib/libvirt/images` – путь к каталогу с образами ВМ;

– `192.168.68.129` — адрес nfs-клиента (второго хоста в процессе миграции), может быть указан как собственно ip-адреса, так и подсеть: `192.168.68.0/24` .

Примечание: на nfs-клиенте просмотреть расшаренные папки можно командой

```
showmount -e 192.168.68.128
```

где 192.168.68.128 — адрес nfs-сервера.

3) Перезапустите сервис nfs

`/sbin/service nfs start`

На nfs-клиенте:

1) В файле `/etc/fstab` пропишите:

```
192.168.68.128:/exports/images /var/lib/libvirt/images nfs auto 0 0
```

Где 192.168.68.128 — адрес nfs-сервера.

2) Перезапустите сервис nfs:

`/sbin/service nfs restart`

3) В консоли смонтируйте общий каталог: `mount 192.168.68.128:/var/lib/libvirt/i`
`/var/lib/libvirt/images`

Где /var/lib/libvirt/images — каталог с образами ВМ

Будьте внимательны! В случае возникновения ошибок не забудьте посмотреть логи на nfs-сервере в файле `/var/log/messages`.

При ошибке монтирования в логах могут присутствовать сообщения, подобные нижеследующим:

NOV 18 10:07:23 server1 rpc.mountd: refused mount request from 192.168.68.129 for /home (/): no export entry

4.9.2. Не разрешен сетевой адрес

Ошибка такого типа (точнее ‘‘unable to migrate guest unable to resolve address“) возникает, когда в сети неправильно происходит разрешение DNS. Настройте DNS в подсети, в которой находятся хосты или пропишите в каждом файле `/etc/hosts` прямое указание на другой хост, например, в файле хоста 192.168.68.128 пропишите `192.168.68.129 rosa.int`, где `rosa.int` – FQDN хоста, на который происходит миграция.

4.10 Просмотр хранилищ хоста

Для просмотра хранилищ, доступных на хосте, нажмите в главном окне менеджера виртуальных машин **Правка —> Свойства подключения** (рис. 31).

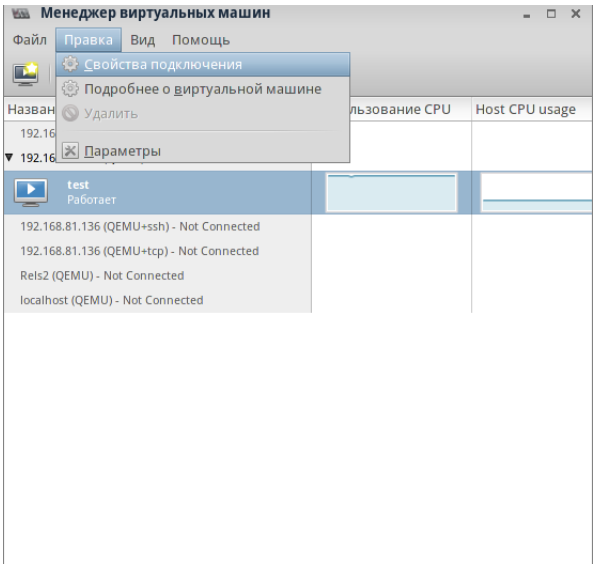

Рисунок 31 — Открыть свойства подключения

Откроется форма **Свойства подключения** с обзором основных деталей (информации) о хосте (рис. 32).

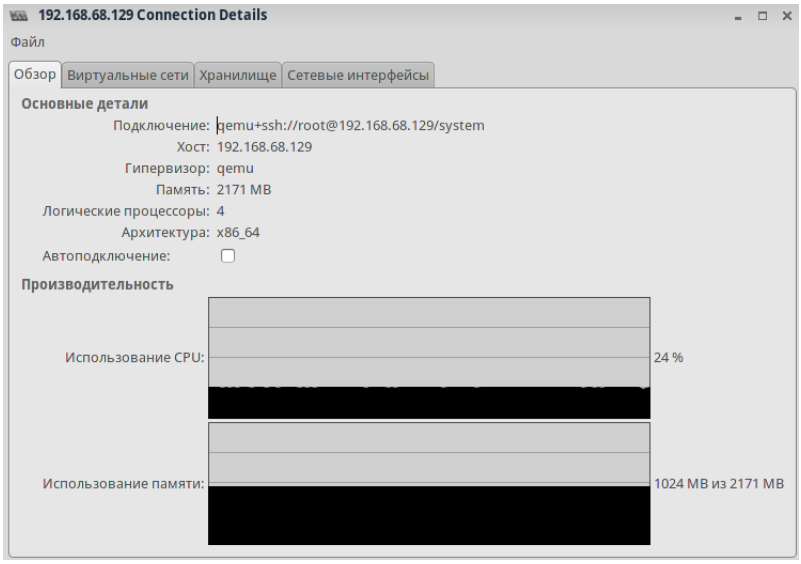

Рисунок 32 — Свойства подключения

Для просмотра информации о хранилищах выберите вкладку **Хранилище**. Откроется форма с перечнем хранилищ, доступных на хосте и перечнем томов в каждом хранилище (рис. 33).

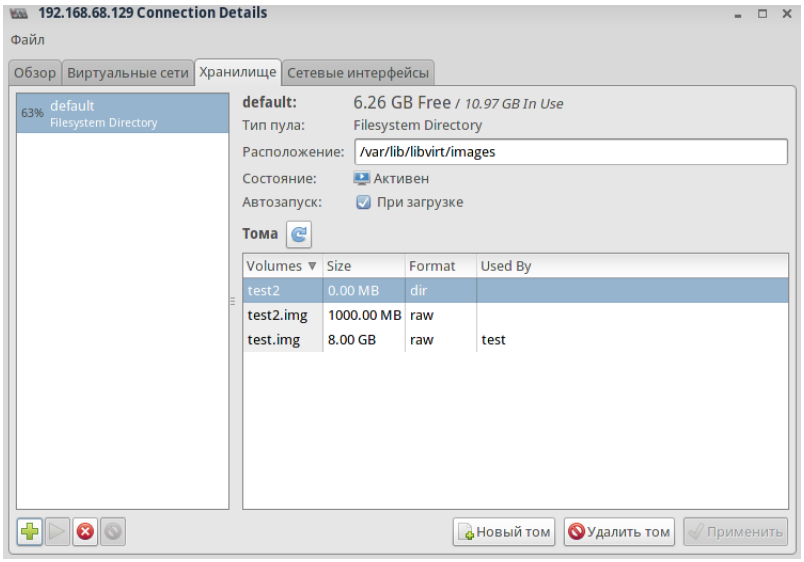

Рисунок 33 — Хранилище

4.11 Создание пула хранилища на дисковом разделе

Возможно создать хранилище, используя менеджер виртуальных машин или интерфейс командной строки (CLI). Здесь мы приводим описание как создать пул хранения с использованием графического интерфейса менеджера виртуальных машин.

1) Откройте хранилища хоста (см. Просмотр хранилищ хоста)
2) Создайте новый пул хранения

Для этого необходимо в окне **Хранилища** (рис. 34) нажать на кнопку  .

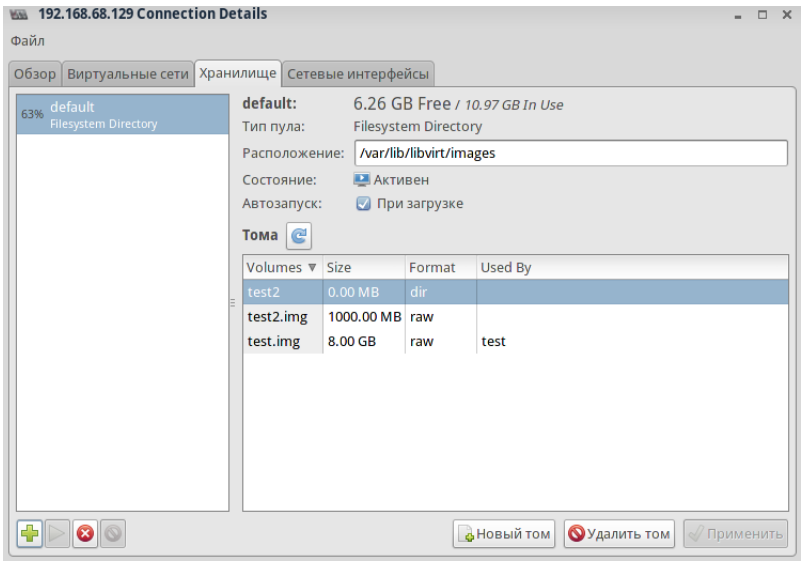

Рисунок 34 — Окно хранилища

Откроется окно **Добавить пул хранения шаг 1** (рис. 35), в котором необходимо указать:

– **Название** — название пула хранения;

– **Тип** — тип пула хранения.

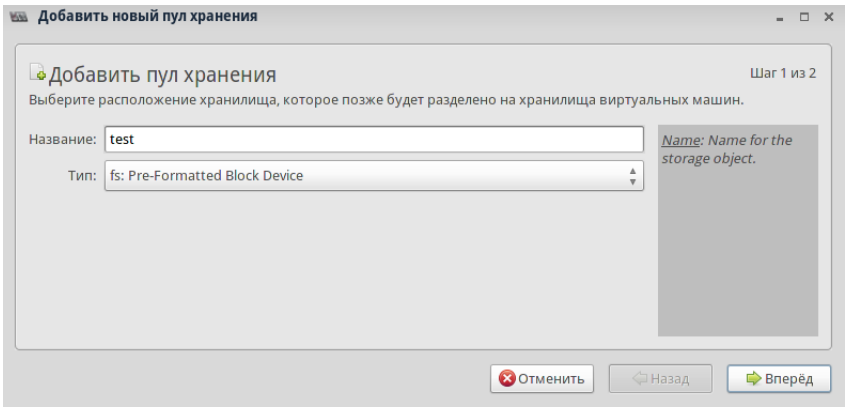

Рисунок 35 — Добавить пул хранения шаг 1

На шаге 2 добавления пула хранения (рис. 36) необходимо указать:

– **Путь к приемнику** — путь, по которому будет примонтирован создаваемый пул хранения;

– **Формат** — здесь обычно указывается тип файловой системы;

– **Путь к источнику** — путь к устройству, которое будет добавлено в пул хранения;

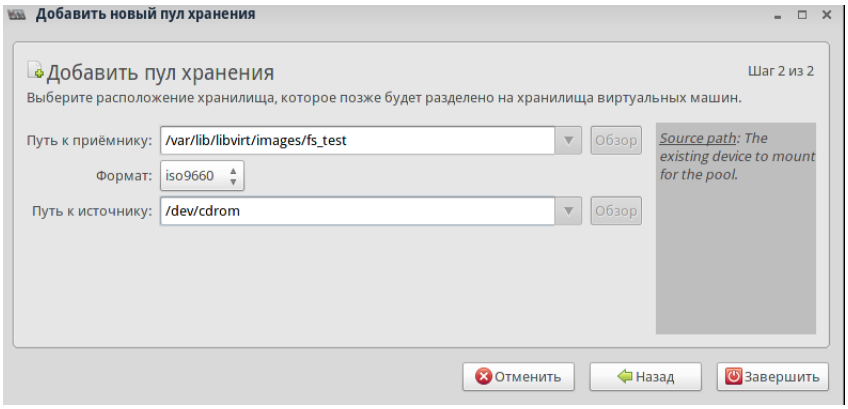

Рисунок 36 — Добавить пул хранения шаг 2

Убедитесь, что новый пул хранения создан в окне **Хранилище**, а его статус активен (рис. 37). Также можно установить/снять галочку в поле **Автозапуск**. Установленная галочка означает, что пул будет автоматически запускаться при запуске демона **libvirt**.

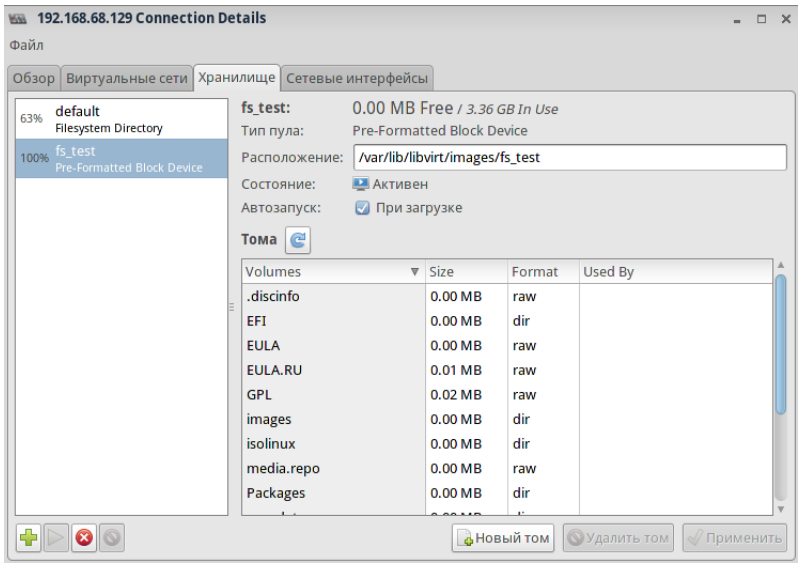

Рисунок 37 — Новый пул хранения

4.12 Создание пула хранения на базе директории

Для создании такого пула хранения в Менеджере виртуальных машин прежде всего необходимо произвести комплекс предварительных действий, таких, как создание директории, установка прав доступа к директории.

1) Для создания директории воспользуйтесь командой:

```
makedir /guest_images
```

2) Установите владельца директории (это должен быть суперпользователь **root**):

```
chown root:root /guest_images
```

3) Установите права доступа к директории: `chmod 700 /guest_images`

Примечание: по умолчанию selinux в RELS выключен, чтобы проверить статус selinux, воспользуйтесь командой: `/usr/sbin/sestatus`

При отключённом selinux будет показано:

```
SELinux status: disabled
```

4) Если же selinux активен, то дополнительно для подготовки директории необходимо установить контекст selinux для директории:

```
semanage fcontext -a -t virt_image_t /guest_images
```

Теперь можно перейти непосредственно к действиям по созданию хранилища в Менеджере виртуальных машин.

1) Откройте хранилища хоста (см. Просмотр хранилищ хоста).

2) Создайте новый пул хранения. Для этого необходимо в окне **Хранилища** (рис. 38) нажать на кнопку

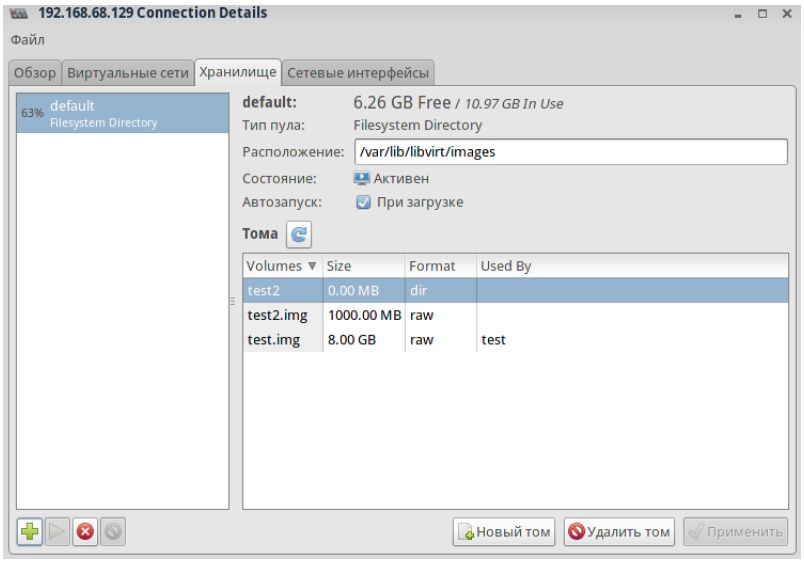

Рисунок 38 — Окно хранилища

Откроется окно **Добавить пул хранения шаг 1** (рис. 39), в котором необходимо указать:

1) **Название** — название пула хранения;
2) **Тип** — тип пула хранения (необходимо указать `dir: Filesystem Directory`).

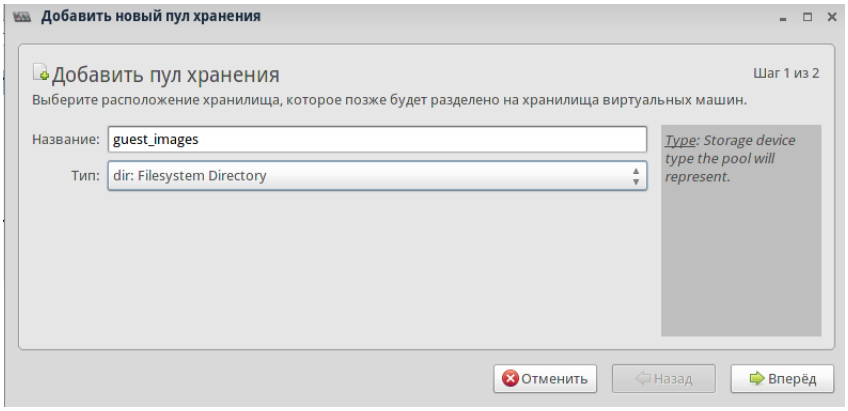

Рисунок 39 — Добавить пул хранения шаг 1

На шаге 2 добавления пула хранения (рис. 40) необходимо указать:

1) **Путь к приемнику** — путь к директории, в которой будет расположен пул хранения;

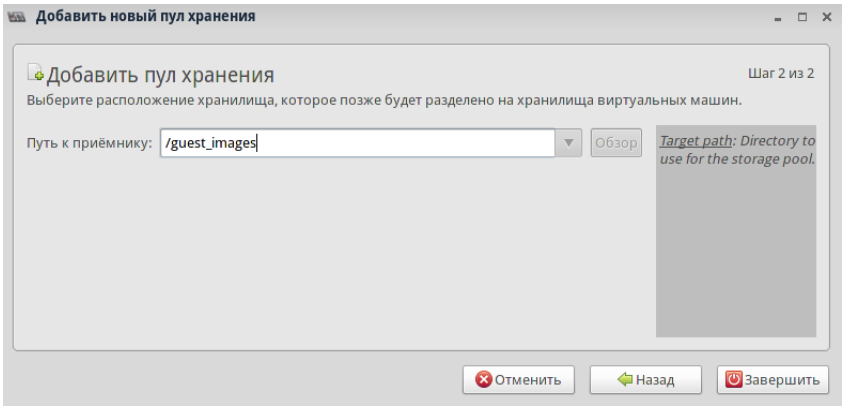

Рисунок 40 — Добавить пул хранения шаг 2

2) Убедитесь, что новый пул хранения создан в окне **Хранилище**, а его статус **Активен** (рис. 41). Также установить/снять галочку в поле **Автозапуск**. Установленная галочка означает, что пул будет автоматически запускаться при запуске демона `libvirt`.

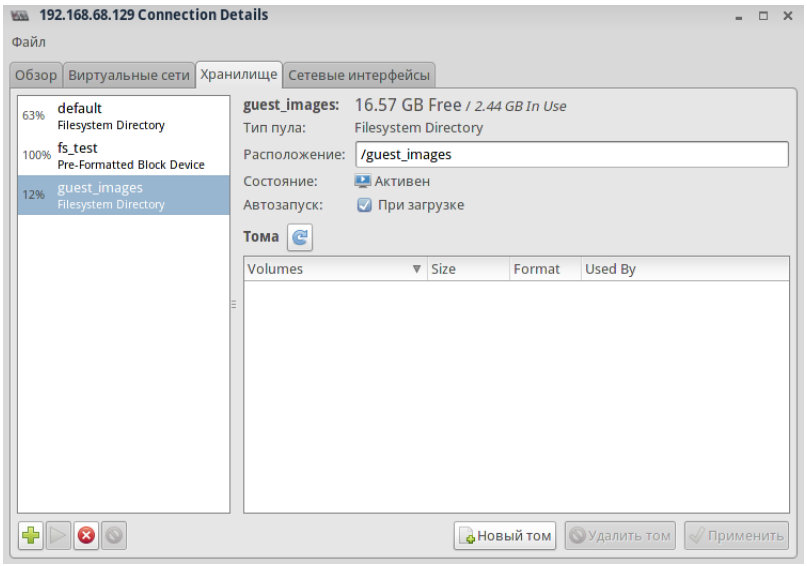

Рисунок 41 — Новый пул хранения

4.13 Создание LVM-пула хранения

Для создания пула хранения LVM выполните следующую последовательность действий:

1) Подготовьте раздел диска LVM (в простейшей конфигурации, LVM Group примонтирован к `/` и имеет тома `/lvm/root`, `/lvm/swap`).

2) Запустите процесс добавления пула хранения (в Менеджере виртуальных машин нужно нажать **Правка —> Свойства подключения**, выбрать вкладку **Хранилище**, нажать на кнопку  .

3) В открывшемся окне (рис. 42) ввести название хранилища и в типе выбрать `logical: LVM Volume Group`.

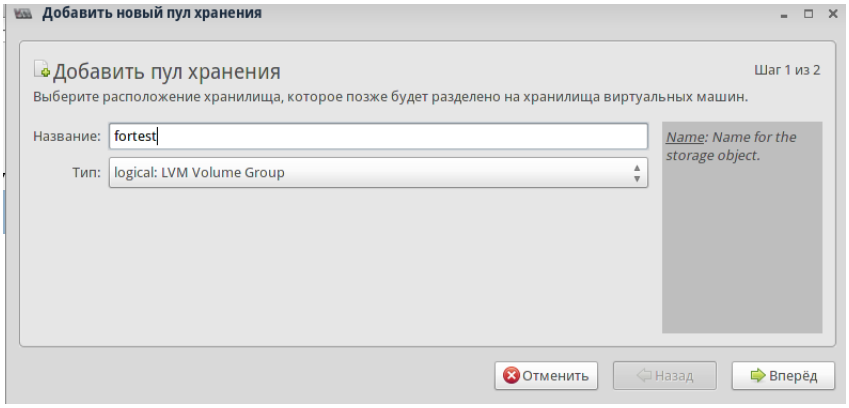

Рисунок 42 — Создание LVM-пула 1

4) В открывшемся окне выбрать пункт к приемнику, указав заранее созданную директорию и путь к источнику (в нашем случае /). Нажать на кнопку **Завершить** (рис. 43).

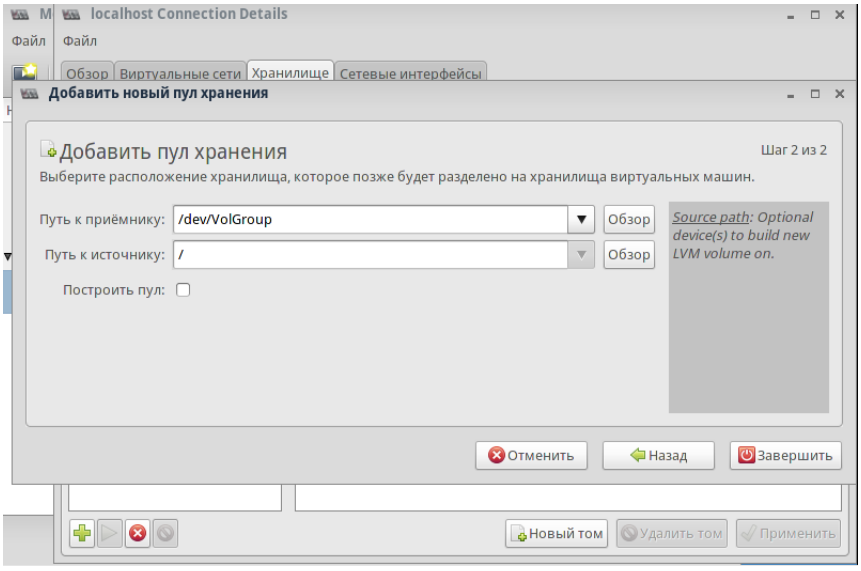

Рисунок 43 — Создание LVM-пула 2

5) Убедиться, что пул хранения создан (рис. 44).

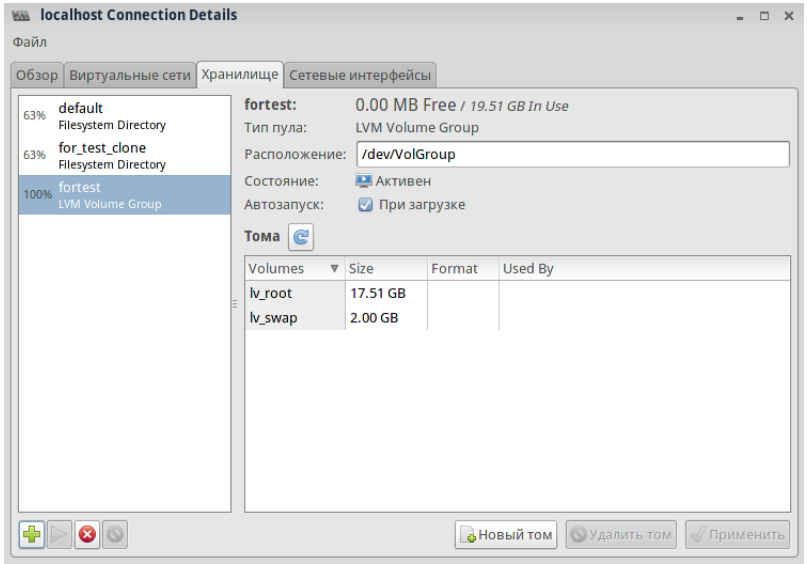

Рисунок 44 — Создание LVM-пула 3

Примечание: для удаления пула хранения необходимо выбрать нужный пул и нажать на кнопку  . Точно также можно создавать пулы хранения и другого типа, для этого на первом шаге создания пула необходимо выбрать нужный тип (см. например, рис. 42).

4.14 Клонирование ВМ

Чтобы выполнить клонирование виртуальной машины выполните в Менеджере виртуальных машин следующую последовательность действий:

Примечание: клонирование можно осуществить только той ВМ, выполнение которой приостановлено, или которая выключена. Для выполняющихся ВМ клонирование, в отличие от живой миграции, невозможно.

1) Выберите ВМ, которую необходимо клонировать, нажмите на нее правой кнопкой мыши и выберите **Clone** (рис. 45)

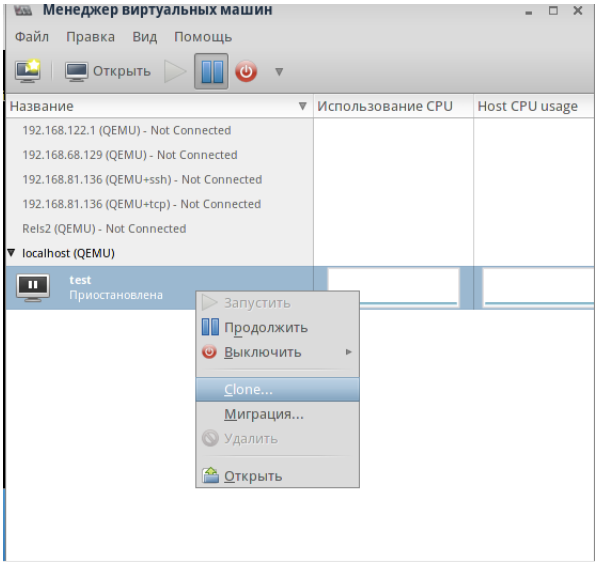

Рисунок 45 — Клонирование ВМ шаг 1

2) Введите название машины-клона, выберите диск, который будет клонирован (рис. 46)

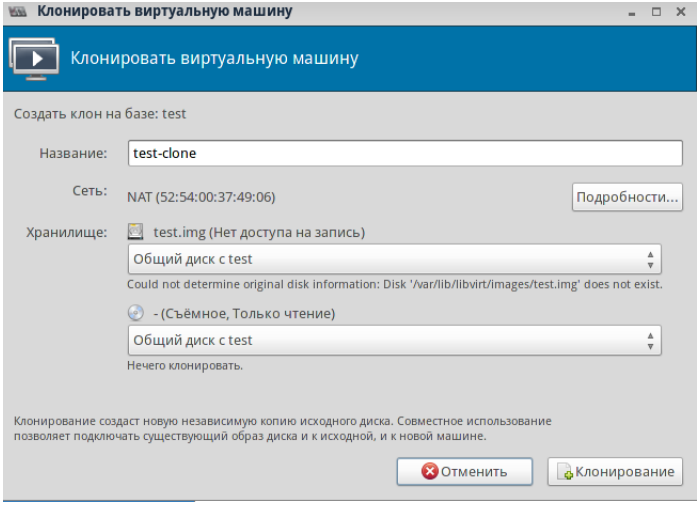

Рисунок 46 — Клонирование ВМ шаг 2

3) Убедитесь, что клонированная ВМ появилась в списке в Менеджере виртуальных машин (рис. 47).

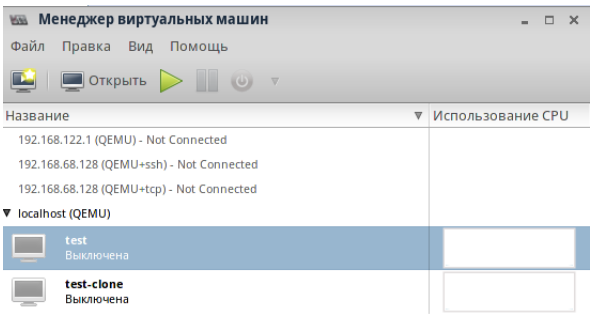

Рисунок 47 — Клонирование ВМ шаг 3

5 Управление виртуальными сетями

5.1 Введение в виртуальные сети

Важными понятиями для сетей являются IP-адрес и маска подсети.

IPv4-адрес состоит из 32 бит и является адресом сетевого интерфейса. В пакетах, передаваемых по сетям в заголовках пакета содержится IP-адрес источника и IP-адрес назначения. Пример IP-адреса: `192.168.68.134`.

Маска подсети, как и IP-адрес, состоит из 32 бит или 4-х октетов. В двоичном представлении маска подсети всегда записывается (в отличие от IP-адреса) таким образом, что в ее представлении сначала следует несколько единиц, а потом несколько нулей. Единицы и нули чередоваться не могут. Пример маски: `255.255.255.0`, что в двоичной форме эквивалентно `11111111.11111111.11111111.00000000`. Понятно, что маску можно определить просто количеством единиц в ее записи. Это позволяет записывать пару «IP-адрес/маска подсети» в сокращенном виде. Например, `192.168.68.134 255.255.255.0` — эквивалентно `192.168.68.134/24`. Такая сокращенная форма записи и используется в интерфейсе RELS при создании сетей для ВМ.

В каждой подсети есть два адреса, которые не назначаются сетевым интерфейсам, т.е. не могут быть присвоены ВМ в виртуальной сети. Это так называемый адрес подсети и так называемый «направленный бродкаст». Адрес подсети получается обнулением всех бит двоичной записи IP-адреса из подсети, стоящих на местах, соответствующих нулям в маске подсети. Например, для `192.168.68.134/24` IP-адресом подсети является `192.168.68.0`. Направленный бродкаст, наоборот, получается установкой единиц во всех битах IP-адреса, соответствующих нулевым битам маски. Для `192.168.68.134/24` — направленный бродкаст для `192.168.68.255`.

Как правило, IP-адрес маршрутизатора — это первый по порядку из возможных для

использования адресов в подсети. Например, для подсети `192.168.68.0/24` IP-адресом маршрутизатора будет `192.168.68.1`.

Важно помнить, что данные, передаваемые внутри подсети, передаются непосредственно между хостами и не проходят через шлюз (маршрутизатор).

5.2 Создание виртуальной сети

Для создания виртуальной сети выполните следующую последовательность действий:

1) В главном окне Менеджера виртуальных машин нажмите **Правка —> Свойства подключения**, перейдите во вкладку **Виртуальные сети** (рис. 48)

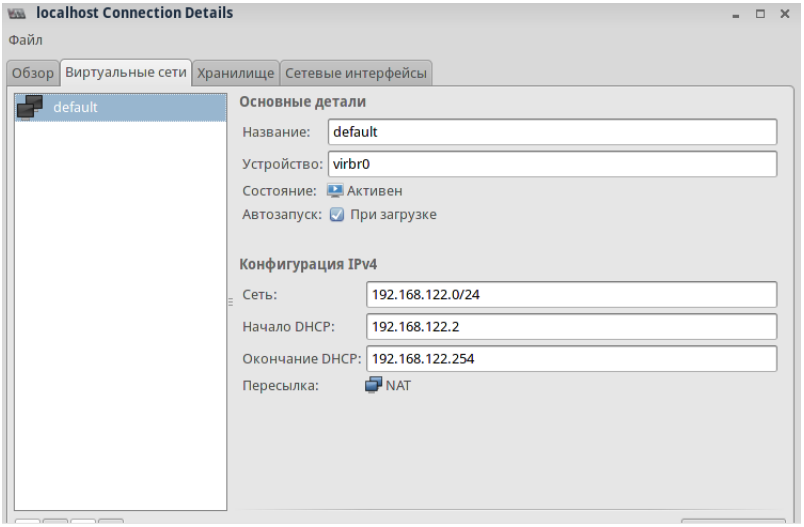

Рисунок 48 — Создание виртуальной сети шаг 1

2) Нажмите на кнопку  внизу формы, откроется мастер создания виртуальной сети (рис. 49), нажмите кнопку **Вперед**.

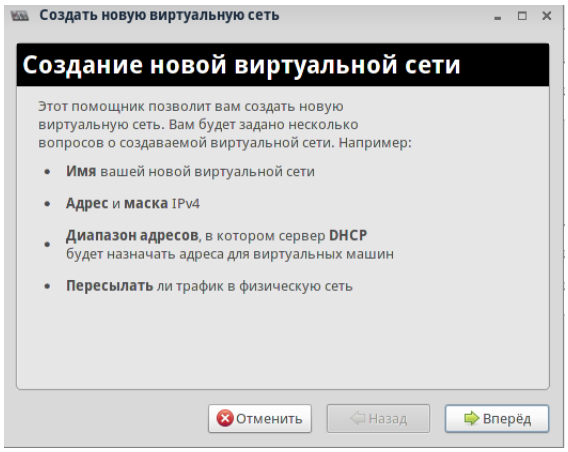

Рисунок 49 — Мастер создания виртуальной сети

3) Введите имя виртуальной сети (рис. 50).

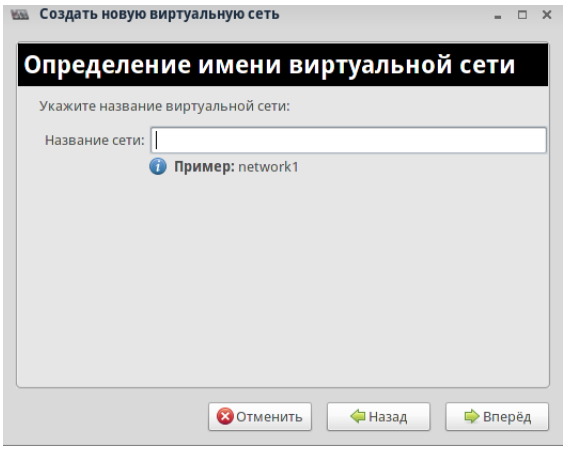

Рисунок 50 — Создание виртуальной сети шаг 2

4) Введите адрес сети (в данном случае это IPv4-адрес, который вводится в виде четырех октетов в десятичном виде, через слэш записывается маска подсети), см. рис. 51

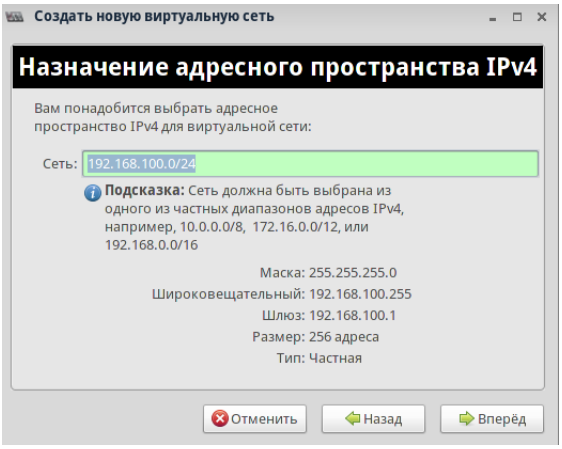

Рисунок 51 — Создание виртуальной сети шаг 3

Примечание: при необходимости можно ознакомиться с некоторой теорией об IP-адресах, масках подсетей и с другой полезной теоретической информацией в этой неплохой статье.

5) Введите начальный и конечный IP-адреса сети, укажите, необходимо ли включение DHCP (рис. 52)

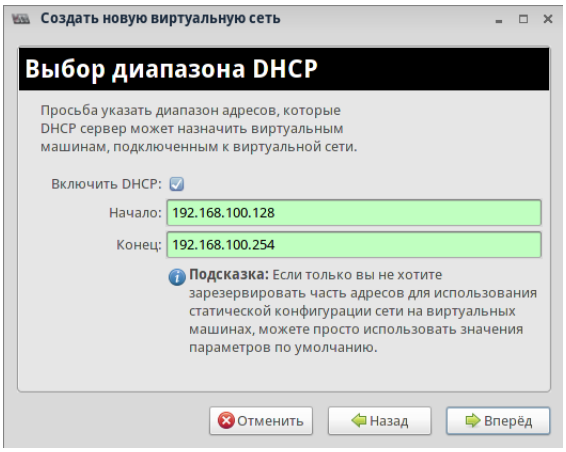

Рисунок 52 — Создание виртуальной сети шаг 4

6) Укажите режим доступа из создаваемой сети к физической сети, доступны режимы (рис. 53):

– Изолированная виртуальная сети (машины в этой сети не будут иметь доступ к интернету и внешним сетям)

– Пересылать на физическое сетевой устройство (у машин в этой сети будет доступ к внешним сетям)

– NAT — используется трансляция адресов (наиболее часто используемый подход)

– Маршрутиризированная (следует использовать при невозможности трансляции, например, в случае шифрования пакетов и, соответственно, невозможности прочитать заголовки пакетов или при отсутствии заголовков у пакетов)

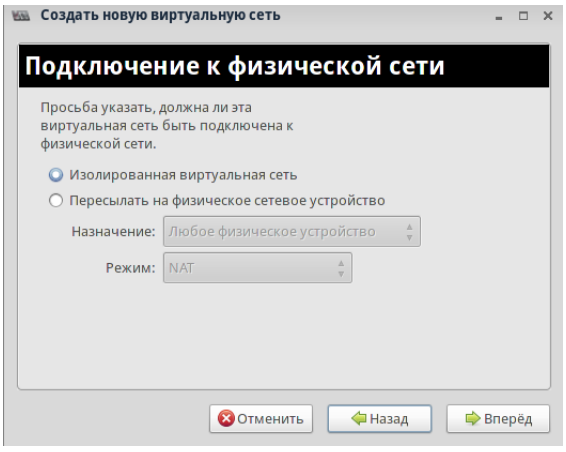

Рисунок 53 — Создание виртуальной сети шаг 5

Примечание: при выборе режима доступа **Пересылать на физическое сетевое устройство** необходимо будет выбрать физическое устройство из списка доступных на хосте интерфейсов.

7) Мастер создания виртуальных сетей выведет форму с указанием всех выбранных параметров создаваемой сети, если все указано верно, подтвердите создание сети (рис. 54)

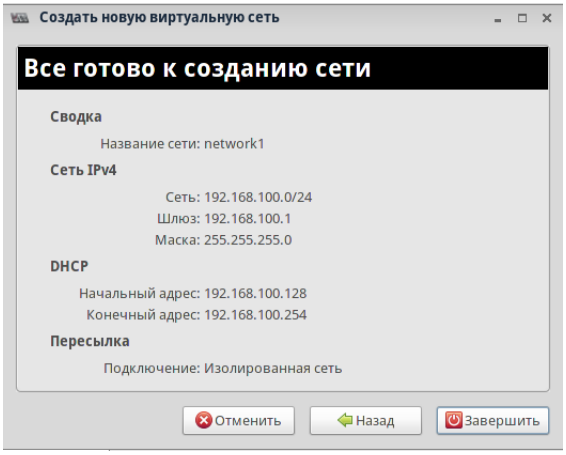

Рисунок 54 — Создание виртуальной сети шаг 6

8) Убедитесь, что созданная сеть появилась в списке сетей (**Свойства подключения**, вкладка **Виртуальные сети**), см. рис. 55.

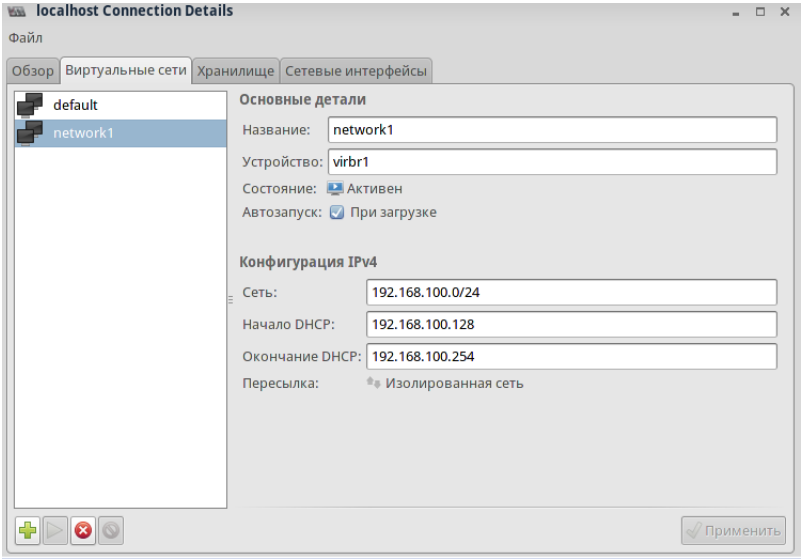

Рисунок 55 — Созданная виртуальная сеть

5.3 Подключение ВМ к сети

Примечание: подключение оборудование к ВМ, в т.ч. сетей, необходимо осуществлять при выключенной ВМ.

Чтобы подключить ВМ к виртуальной сети выполните следующие действия:

1) В главном окне Менеджера виртуальных машин выберите нужную ВМ, нажав на нее мышью, в верхнем меню выберите **Правка —> Подробнее о виртуальной машине**, перейдите к сведениям о ВМ, нажав на кнопку  , откроется окно со списком оборудования ВМ (рис. 56)

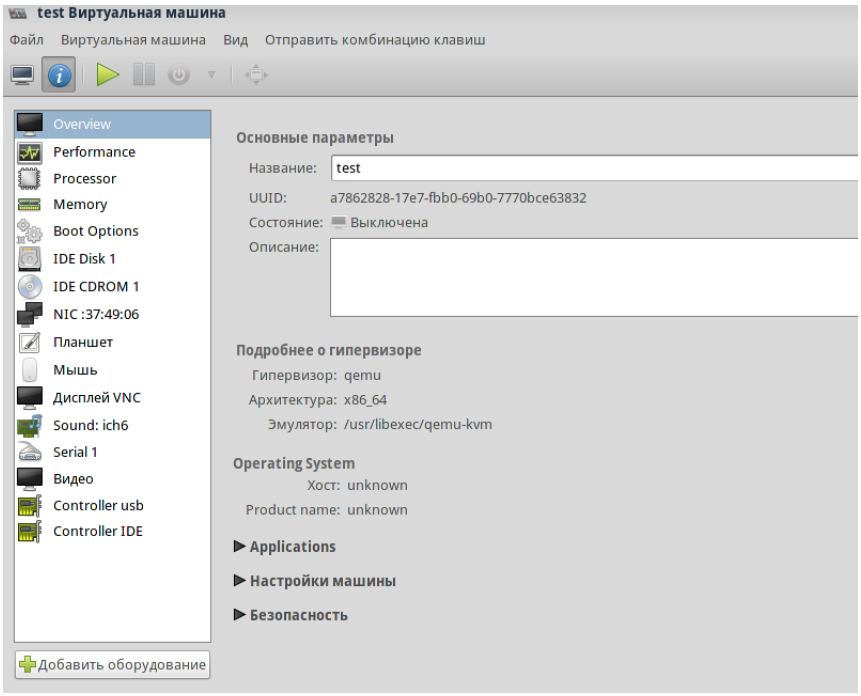

Рисунок 56 — Оборудование ВМ

2) Нажмите «Добавить оборудование», откроется форма добавления нового оборудования, в которой необходимо в левой части выбрать пункт «Network» (рис. 57)

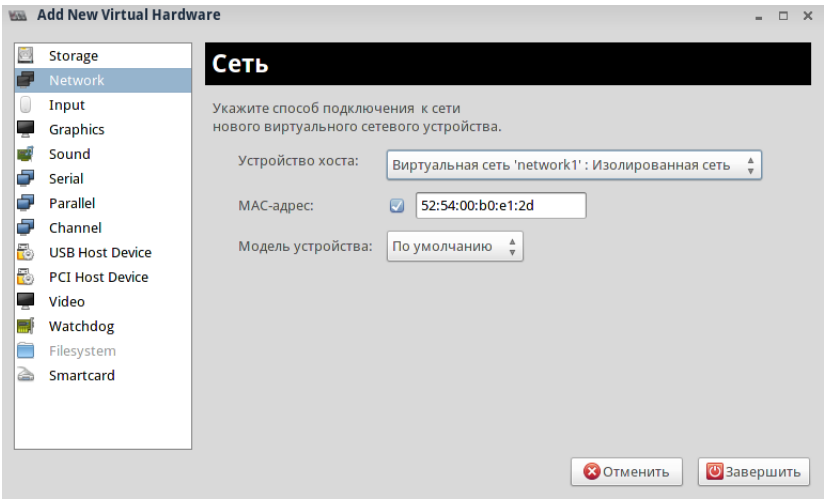

Рисунок 57 — Добавление оборудования

3) Необходимо выбрать сеть, к которой нужно подключить ВМ. Например, можно выбрать только что созданную в нашем примере сеть Network1 (см. Создание виртуальной сети). Также нужно указать MAC-адрес ВМ и подтвердить добавление сети, нажав на кнопку **Завершить**.

4) Убедитесь, что сеть подключена к ВМ (рис. 58)

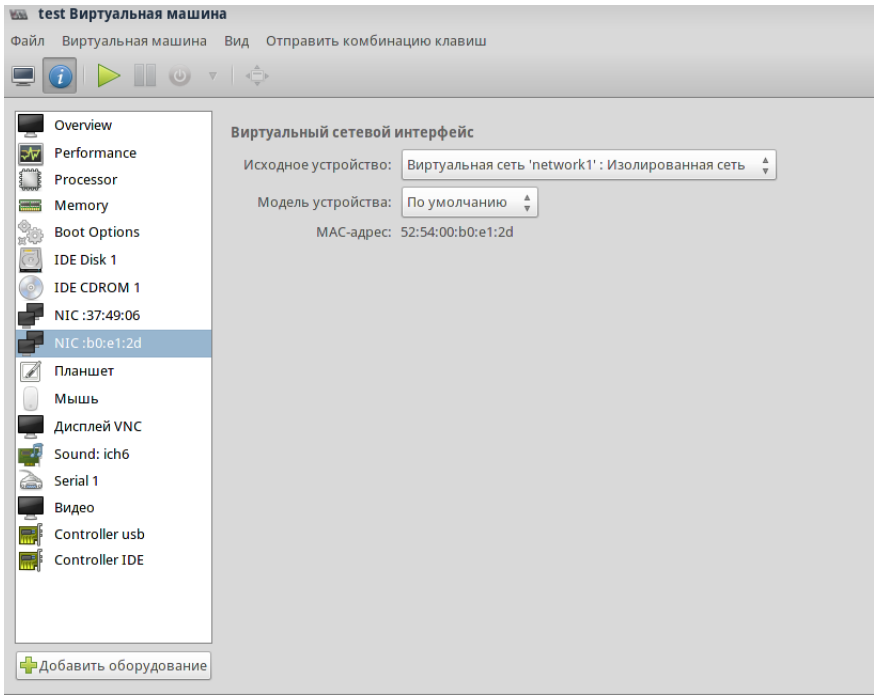

Рисунок 58 — Подключенная сеть

6 Управление виртуальными машинами из консоли

Гипервизорами и виртуальными машинами можно управлять из консоли, используя утилиту `virsh`. Утилита использует LibvirtAPI. Доступ непривилегированным пользователям предоставляется только в режиме чтения.

6.1 Консольные команды управления ВМ

В этом разделе мы приводим общий список команд, доступных с использованием Virsh, и подробно останавливаемся на использовании некоторых из них.

Таблица 1

Команда	Описание команды
`help`	Вывести справку по утилите
`list`	Просмотр списка доступных ВМ
`create`	Создать и запустить ВМ из конфигурационного файла
`start`	Запустить ВМ
`destroy`	Остановить ВМ
`reboot`	Перезагрузить ВМ
`restore`	Восстановить ВМ

Окончание таблицы 1

Команда	Описание команды
`resume`	Возобновить ВМ
`save`	Сохранить состояние ВМ в файл
`shutdown`	Выключить ВМ
`suspend`	Приостановить ВМ
`migrate`	Миграция ВМ
`dumpxml`	Вывести файл для заданной ВМ конфигурационный файл (в XML)
`define`	Задать для ВМ конфигурационный файл
`undefine`	Удалить файлы конфигурации ВМ и данные ВМ
`domid`	Просмотр идентификатора ВМ
`domuuid`	Просмотр UUID ВМ
`dominfo`	Просмотр сведений о ВМ
`domname`	Просмотр имени ВМ
`domstate`	Просмотр состояния ВМ
`setmem`	Изменить размер ОЗУ для ВМ
`setmaxmem`	Установить максимальный объем ОЗУ гипервизора
`setvcpus`	Установить число процессоров для ВМ
`vcpuinfo`	Просмотр информации о процессорах
`vcpupin`	Настройка процессоров
`domblkstat`	Просмотр блочных устройств ВМ
`domifstat`	Просмотр сетевых интерфейсов ВМ
`attach-device`	Подключить устройство к ВМ
`attach-disk`	Подключить новое дисковое устройство к ВМ
`attach-interface`	Подключить новый сетевой интерфейс к ВМ
`detach-device`	Отключить устройство от ВМ
`detach-disk`	Отключить дисковое устройство от ВМ
`detach-interface`	Отключить сетевой интерфейс от ВМ

6.2 Управление ВМ консольными командами

Подключение к хосту

```
# virsh connect адрес хоста
```

Определение идентификатора хоста

```
# virsh domid имя_домена или uuid_домена
```

Определение имени домена

```
# virsh domname идентификатор_домена или uuid_домена
```

Определение UUID ВМ

```
# virsh domuuid  идентификатор_домена или имя_домена
```

Пример:

```
# virsh domuuid 1
48f57e15-4ae8-703c-2e4b-51a9efbb9c04
```

Получение информации о хосте

```
# virsh nodeinfo
```

Пример:

```
Модель процессора: x86_64
CPU:                 4
Частота процессора: 2494 MHz
Сокеты:         2
Ядер на сокет: 2
Потоков на ядро: 1
Ячейки NUMA:   1
Объём памяти: 3140496 KiB
```

Получение информации о ВМ

```
# virsh dominfo  идентификатор_домена или имя_домена или
uuid_домена
```

Пример:

```
ID:             1
Имя:          tets
UUID:           48f57e15-4ae8-703c-2e4b-51a9efbb9c04
Тип ОС:    hvm
Статус:   работает
CPU:            2
Время CPU: 235,4s
Макс.память: 2258944 KiB
Занято памяти: 2258944 KiB
Persistent:     yes
Автозапуск: выкл.
```

```
Managed save:   no
Модель безопасности: none
```

Просмотр списка ВМ

```
# virsh list
```

Пример:

```
# virsh list --all
  ID    Имя                            Статус
----------------------------------------------------
  -     tets                           выключен
```

Столбец «Статус» может содержать следующие значения:

- **работает** — запущенные ВМ
- **приостановлен** — ВМ приостановлена
- **выключен** — выключенные ВМ

Получение информации о процессорах

```
# virsh vcpuinfo  идентификатор_домена или имя_домена или uuid_домена
```

Пример:

```
VCPU:           0
CPU:            3
Статус:   работает
Время CPU: 8,0s
Соответствие CPU: yyyy

VCPU:           1
CPU:            3
Статус:   работает
Время CPU: 8,0s
Соответствие CPU: yyyy
```

Управление виртуальными сетями

Для вывода списка доступных виртуальных сетей используйте команду:

```
# virsh net-list
```

Пример:

```
# virsh net-list
Имя                 Статус     Автозапуск      Persistent
----------------------------------------------------------
default           активен         yes                yes
```

Для просмотра информации по конкретной виртуальной сети используйте команду:

```
# virsh net-dumpxml имя_сети
```

Пример ответа:

```
<network connections='1'>
  <name>default</name>
  <uuid>ee05cecd-2177-4e1e-9982-69834d71d781</uuid>
  <forward mode='nat'/>
  <bridge name='virbr0' stp='on' delay='0' />
  <mac address='52:54:00:46:C9:B0'/>
  <ip address='192.168.122.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.122.2' end='192.168.122.254' />
    </dhcp>
  </ip>
</network>
```

Создание конфигурационного файла ВМ

```
# virsh dumpxml идентификатор ВМ > guest.xml
```

Где guest.xml — конфигурационный файл для ВМ. Если не указать перенаправление вывода, то ответ будет выведен в stdout.

Пример выводаvirsh dumpxml:

```
<domain type='qemu' id='1'>
  <name>tets</name>
  <uuid>48f57e15-4ae8-703c-2e4b-51a9efbb9c04</uuid>
  <memory unit='KiB'>2258944</memory>
  <currentMemory unit='KiB'>2258944</currentMemory>
  <vcpu placement='static'>2</vcpu>
  <os>
    <type arch='x86_64' machine='rhel6.5.0'>hvm</type>
    <boot dev='cdrom'/>
    <boot dev='hd'/>
  </os>
  <features>
    <acpi/>
```

```
    <apic/>
    <pae/>
  </features>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>destroy</on_reboot>
  <on_crash>destroy</on_crash>
  <devices>
    <emulator>/usr/libexec/qemu-kvm</emulator>
    <disk type='file' device='disk'>
      <driver name='qemu' type='raw' cache='none'/>
      <source file='/var/lib/libvirt/images/tets.img'/>
      <target dev='hda' bus='ide'/>
      <alias name='ide0-0-0'/>
      <address type='drive' controller='0' bus='0' target='0' unit
='0'/>
    </disk>
    <disk type='block' device='cdrom'>
      <driver name='qemu' type='raw'/>
      <source dev='/dev/sr0'/>
      <target dev='hdc' bus='ide'/>
      <readonly/>
      <alias name='ide0-1-0'/>
      <address type='drive' controller='0' bus='1' target='0' unit
='0'/>
    </disk>
    <controller type='usb' index='0'>
      <alias name='usb0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01'
function='0x2'/>
    </controller>
    <controller type='ide' index='0'>
      <alias name='ide0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x01'
function='0x1'/>
    </controller>
    <interface type='network'>
      <mac address='52:54:00:53:23:d1'/>
      <source network='default'/>
      <target dev='vnet0'/>
```

```
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03'
function='0x0'/>
    </interface>
    <serial type='pty'>
      <source path='/dev/pts/1'/>
      <target port='0'/>
      <alias name='serial0'/>
    </serial>
    <console type='pty' tty='/dev/pts/1'>
      <source path='/dev/pts/1'/>
      <target type='serial' port='0'/>
      <alias name='serial0'/>
    </console>
    <input type='mouse' bus='ps2'/>
    <graphics type='vnc' port='5900' autoport='yes' listen='127.0.0.1'>
      <listen type='address' address='127.0.0.1'/>
    </graphics>
    <sound model='ich6'>
      <alias name='sound0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x04'
function='0x0'/>
    </sound>
    <video>
      <model type='cirrus' vram='9216' heads='1'/>
      <alias name='video0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x02'
function='0x0'/>
    </video>
    <memballoon model='virtio'>
      <alias name='balloon0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x05'
function='0x0'/>
    </memballoon>
  </devices>
  <seclabel type='none'/>
</domain>
```

Создание и запуск ВМ

```
# virsh create конфигурационный файл.xml
```

Приостановка ВМ

```
# virsh suspend идентификатор ВМ
```

Параметры ВМ будут сохранены. Обратная команда: `resume`

Возобновление ВМ

```
# virsh resume  идентификатор ВМ
```

ВМ возобновляется мгновенно.

Сохранение ВМ

```
# virsh save  идентификатор ВМ  Имя_файла
```

Обратная команда: `restore`

Восстановление ВМ

```
# virsh restore имя_файла
```

Сохраненная машина будет восстановлена из файла и перезапущена. Это, в отличие от возобновления, занимает время.

Завершение работы ВМ

```
# virsh shutdown  идентификатор ВМ
```

Перезагрузка ВМ

```
# virsh reboot  идентификатор ВМ
```

Примечание: поведение ВМ при выключении и перезагрузке может быть настроено в конфигурационном файле ВМ.

Принудительная остановка ВМ

```
# virsh destroy  идентификатор ВМ
```

Мы не рекомендуем использовать эту команду без необходимости: работоспособность ВМ после принудительной остановки может быть нарушена.

Миграция ВМ

Для миграции ВМ на другой хост используется команда `migrate`. Можете использовать параметр `–live` для живой миграции.

Например:

```
# virsh migrate –live Имя_ВМ Адрес_удаленного_хоста
```

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

KVM (Kernel-based Virtual Machine): программное решение, обеспечивающее виртуализацию в среде Linux на платформе x86

MAC (Mandatory access control, мандатное управление доступом): разграничение доступа субъектов к объектам, основанное на назначении метки конфиденциальности для информации, содержащейся в объектах

RBAC (Role Based Access Control, управление доступом на основе ролей): политика избирательного контроля доступа, при которой доступа субъектов системы к объектам группируются с учетом специфики их применения, образуя роли

RELS: ROSA Enterprise Linux Server

Selinux (Security-Enhanced Linux, Linux с улучшенной безопасностью): реализация системы принудительного контроля доступа, реализован в ядре Linux

TCO (Total Cost of Ownership, совокупная стоимость владения): общая величина затрат на содержание ИТ-сервисов и сопутствующей инфраструктуры в период эксплуатации

ВМ: виртуальная машина

ОС: операционная система

ПО: программное обеспечение