Глава 2

Виртуальные машины и Автоматизированные Установки

ЦЕЛИ СЕРТИФИКАЦИИ

2.01 Настройка KVM для Red Hat
2.02 Настройка виртуальной машины в KVM
2.03 Опции автоматической установки
2.04 Администрирование с помощью Secure Shell и Secure Copy
2.05. Рассмотрите возможность добавления Инструментами командной строки
✓ Двухминутная тренировка
Q & A Самопроверка

Для управления виртуальными машинами (**VM**) и установками **Kickstart** требуются навыки RHCSA. Другими словами, вы должны быть готовы установить RHEL 7 на виртуальную машину по сети, вручную и с помощью Kickstart.

Глава 1 охватила основы процесса установки. Предполагалось, что вы также можете настроить виртуализацию во время процесса установки. Но возможно, что вам нужно будет установить и настроить KVM после завершения установки.

Kickstart - это система Red Hat **для автоматизированных установок**. Он работает из текстового файла, который предоставляет ответы на программу установки RHEL 7. С этими ответами программа установки RHEL 7 может работать автоматически, без дальнейшего вмешательства.

После завершения установки в системах, используемых для тестирования, обучения и обслуживания, вы сможете управлять ими удаленно. Требование RHCSA - это не только понимание SSH-соединений, но и превосходный навык в реальном мире. Ссылки на пункты меню в этой книге основаны на среде **рабочего стола GNOME**. Если вы используете другую среду рабочего стола, такую как KDE, шаги несколько иные.

ЦЕЛЬ СЕРТИФИКАЦИИ 2.01

Настройте KVM для Red Hat

В главе 1 вы настроили физическую 64-битную систему RHEL 7 с пакетами, необходимыми для настройки виртуальных машин. Если ничего не помогает, эта конфигурация может помочь вам настроить несколько установок RHEL 7. Но если вы столкнулись с установкой RHEL без необходимых пакетов, что вы будете делать?

С правильными пакетами вы можете **настроить модули KVM**, получить доступ к командам конфигурации виртуальной машины и настроить подробную конфигурацию для группы виртуальных машин. Некоторые из команд, описанных в этом разделе, являются, в некотором смысле, предварительным просмотром будущих глав. Например, инструменты, связанные с обновлениями, описаны в **главе 7**. Но сначала важно обсудить, почему кто-то захочет использовать виртуальную машину, когда физическое оборудование гораздо более ощутимо.

ВНУТРИ ЭКЗАМЕНА

Управление виртуальными машинами
Цели RHCSA предполагают, что вам нужно знать, как сделать

- Доступ к консоли виртуальной машины

- Запуск и остановка виртуальных машин
- Настройка систем для запуска виртуальных машин при загрузке
- Установите системы Red Hat Enterprise Linux в качестве виртуальных гостей

Разумно предположить, что рассматриваемые виртуальные машины основаны на стандартном решении **виртуальной машины Red Hat KVM**. В **главе 1** вы настроили это решение в процессе установки в 64-битной системе; тем не менее, вам также может понадобиться установить соответствующие пакеты в действующей системе во время экзамена. Кроме того, есть графическая консоль **Virtual Machine Manager**, используемая Red Hat для управления такими виртуальными машинами. Конечно, этот менеджер виртуальных машин является **интерфейсом API управления**, предоставляемым **библиотекой libvirt**. Его также можно использовать для установки и настройки системы для автоматического запуска во время процесса загрузки.

Хотя блог об экзаменах Red Hat, упомянутый в главе 1, предполагает, что вы будете сдавать экзамен по «предустановленной» системе, это не исключает возможности установки на виртуальные машины. Поэтому в этой главе вы узнаете, как настроить установку RHEL 7 в **KVM**.

Установка при помощи Kickstart файла

Цели RHCSA утверждают, что вам нужно знать, как

- Установите **Red Hat Enterprise Linux** автоматически с помощью **Kickstart**

С этой целью каждая установка RHEL включает в себя **образец файла Kickstart**, основанный на данной установке. В этой главе вы узнаете, как использовать этот файл для автоматизации процесса установки. Это немного сложнее, чем кажется, потому что образец файла **Kickstart** должен быть изменен в первую очередь за пределами уникальных настроек для разных систем. Но как только он будет настроен, вы сможете настроить столько установок RHEL, сколько вам нужно, **используя базовый Kickstart файл**.

Доступ к удаленным системам и безопасная передача файлов

Цели RHCSA утверждают, что вам нужно знать, как

- Доступ к удаленным системам **с использованием SSH**
- Безопасно передавать файлы между системами

Если бы системным администраторам приходилось поддерживать физический контакт с каждой системой, половина их жизни была бы потрачена на пути от системы к системе. С помощью таких инструментов, как **Secure Shell (SSH),** администраторы могут удаленно выполнять свою работу и безопасно передавать файлы. Хотя **SSH** автоматически устанавливается в стандартной конфигурации в RHEL 7, пользовательские параметры конфигурации, такие как аутентификация на основе ключей, будут рассмотрены позже в книге.

Почему виртуальные машины

Кажется, что все хотят войти в игру VM. И они должны. Предприятия когда-то выделяли разные физические системы для каждой услуги. На самом деле, для обеспечения надежности, они могут выделять две или более систем для каждой из этих услуг. Конечно, можно настроить несколько служб в одной системе. На самом деле, вы можете сделать это на экзаменах Red Hat. Но на предприятиях, которые заботятся о безопасности, системы часто предназначены для отдельных сервисов, чтобы снизить риск, если одна система или сервис скомпрометированы.

При правильно настроенных системах каждый сервис может быть сконфигурирован на отдельной выделенной виртуальной машине. Вы можете найти 10 виртуальных машин, установленных на одной физической системе хоста. Поскольку разные службы обычно используют циклы ОЗУ и ЦП в разное время, часто разумно «перегружать» ОЗУ и ЦП в локальной физической системе. Например, в системе с 256 ГБ ОЗУ часто разумно выделить 16 ГБ каждая для 20 виртуальных машин, настроенных в этой системе.

На практике администратор может заменить 20 физических машин в более старой сети двумя физическими системами. Каждая из 20 виртуальных машин будет установлена на томе общего хранилища, отформатирована в кластерной файловой системе, такой как GFS2, и смонтирована в каждой физической системе. Конечно, эти две физические системы требуют мощного оборудования. Но в остальном экономия огромна не только в общих затратах на оборудование, но и в оборудовании, энергопотреблении и многом другом.

Если вам нужно установить KVM

Если вам нужно установить какое-либо программное обеспечение на RHEL 7, инструмент «Программное обеспечение GNOME» может быть очень полезным. Войдите в GUI как обычный пользователь. Чтобы открыть его из графического интерфейса, нажмите Приложения | Системные инструменты | Программное обеспечение (**Applications | System Tools | Software**.). Пока существует подходящее соединение с репозиториями, такими как RHN или связанные с третьими сторонами, поиск займет несколько минут. На левой панели щелкните стрелку рядом с Виртуализация. Должны появиться четыре группы пакетов виртуализации. Щелкните группу пакетов Virtualization Hypervisor, а затем первый пакет в этой группе, чтобы увидеть экран, подобный показанному на **рисунке 2-1**.

РИСУНОК 2-1 Добавить/Удалить программное средство

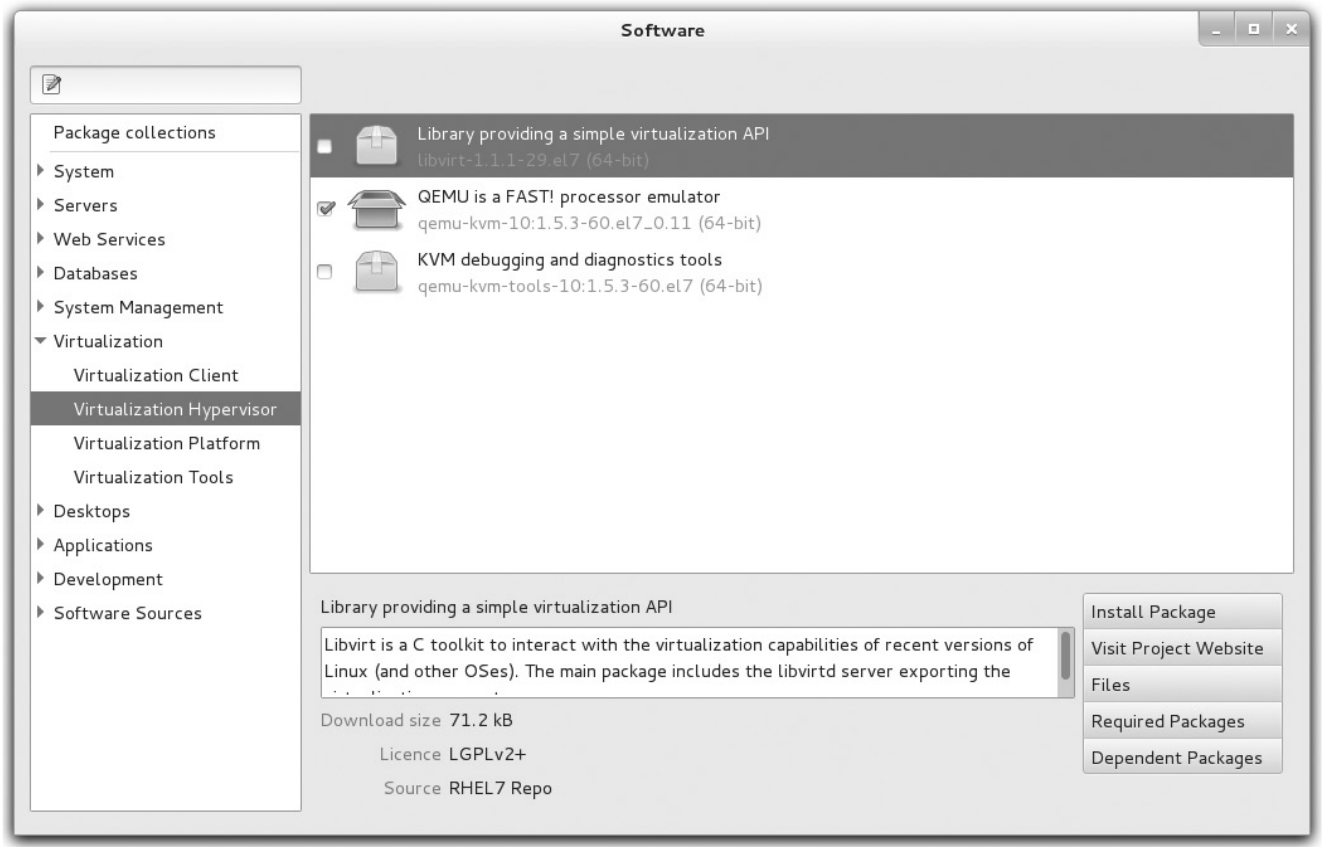


ТАБЛИЦА 2-1 Пакеты, связанные с виртуализацией

Пакет	Описание
qemu-kvm	Основной пакет **KVM**
libvirt	Служба **libvirtd** для управления гипервизорами
libvirt-client	Команда **virsh** и клиентский API для управления виртуальными машинами
virt-install	Инструменты командной строки для создания виртуальных машин
virt-manager	**GUI VM** инструмент администрирования
virt-top	Команда для отображения статистики виртуализации
virt-top	Графическая консоль для подключения к виртуальным машинам

Все, что вам нужно сделать для установки KVM, это выбрать соответствующие пакеты из групп пакетов **Virtualization Hypervisor**, **Virtualization Client** и **Virtualization Platform**. Если вы не помните список, показанный в **Таблице 2-1**, просто установите последнюю версию всех пакетов виртуализации.

Это всего семь пакетов! Конечно, в большинстве конфигураций они используют другие пакеты как зависимости. Но это все, что вам действительно нужно для настройки виртуальных машин в физической системе RHEL 7. Хотя группа «Инструменты виртуализации» не имеет обязательных пакетов, она включает в себя программное обеспечение, которое может оказаться полезным в реальной жизни, например, инструменты, которые могут помочь в чтении образов дисков виртуальных машин и управлении ими. Если вы хотите отобразить содержимое диска виртуальной машины или управлять разделами и файловыми системами виртуальной машины с хоста гипервизора, вам нужен **пакет libgtestsfs-tools**.

Установка с помощью инструмента **GNOME Software** довольно проста. Просто выберите (или отмените выбор) нужные пакеты и нажмите Применить. Если существуют зависимые пакеты, которые также требуют установки, вам будет предложено указать полный список этих пакетов. Конечно, из интерфейса командной строки вы можете устанавливать эти пакеты по одному с помощью команды **yum install *packagename***. В качестве альтернативы установите группы **Virtualization Host** и **Virtualization Client**, как показано здесь:

```
# yum group install "Virtualization Host" "Virtualization Client"
```

Вы узнаете больше о группах yum и package в главе 7.

Правильные модули KVM

В большинстве случаев установка нужных пакетов достаточно хороша. Соответствующие модули ядра должны быть загружены автоматически. Прежде чем KVM сможет работать, соответствующие модули ядра должны быть загружены. Запустите следующую команду:

```
# lsmod | grep kvm
```

Если модули KVM загружены правильно, вы увидите один из следующих двух наборов модулей:

```
kvm_intel 138567 0
kvm 441119 1 kvm_intel
```

или

```
kvm_amd 59887 0
kvm 261575 1 kvm_amd
```

Как следует из названий модулей, вывод зависит от производителя процессора. Если вы не получаете этот вывод, сначала убедитесь, что оборудование подходит. И, как предлагается в **Главе 1, убедитесь, что флаг svm или vmx указан в содержимом файла /proc/cpuinfo**. В противном случае может потребоваться дополнительная настройка в системном **BIOS** или меню **UEFI**. Некоторые меню включают определенные опции для аппаратной виртуализации, которые должны быть включены.

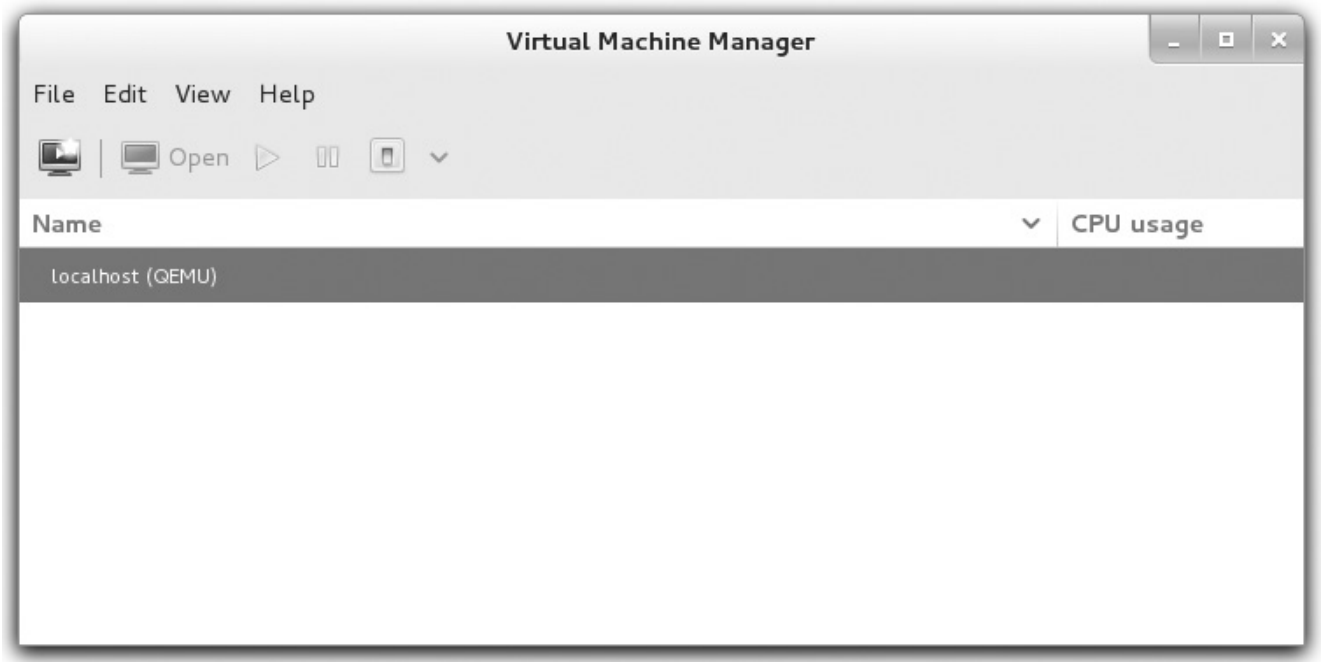
Если в файле **/proc/cpuinfo** есть один из отмеченных флагов, следующим шагом является попытка загрузки соответствующих модулей. Самый простой способ - с помощью команды **modprobe**. Следующая команда также должна **загрузить зависимый модуль KVM**. Если в системе установлен процессор AMD, замените **kvm_intel** на **kvm_amd**:

 # modprobe kvm_intel

Настройте диспетчер виртуальных машин

Диспетчер виртуальных машин является частью пакета **virt-manager**. И вы можете запустить его в графическом интерфейсе с помощью команды с тем же именем. Либо на рабочем столе GNOME щелкните «Приложения | Системные инструменты | Диспетчер виртуальных машин» (**Applications | System Tools | Virtual Machine Manager**). Откроется окно диспетчера виртуальных машин, показанное на **рисунке 2-2.**

РИСУНОК 2-2 Диспетчер виртуальных машин



При желании виртуальные машины на основе **KVM** можно настроить и администрировать удаленно. Все, что вам нужно сделать, это подключиться к удаленному гипервизору. Для этого **нажмите «Файл | Добавить соединение» (File | Add Connection)**. Откроется окно «**Добавить соединение**» **(Add Connection)**, в котором можно выбрать следующее:

- Контейнер Linux или гипервизор, обычно **KVM** или **Xen**. (Xen был гипервизором по умолчанию на RHEL 5, но не поддерживается с RHEL 6.)
- Соединение, которое может быть локальным или удаленным, с использованием метода соединения, **такого как SSH**.

Удаленные соединения могут быть заданы с помощью **имени хоста** или **IP-адреса** удаленной системы.

Конфигурация гипервизором

Каждый гипервизор может быть настроен в некоторых деталях. Щелкните правой кнопкой мыши гипервизор **localhost (QEMU)** и выберите **Details** во всплывающем меню или **«Панель меню | Редактировать | Детали соединения»**, которое появляется. Откроется окно подробностей, названное в честь хоста локальной системы, как показано на **рисунке 2-3**.

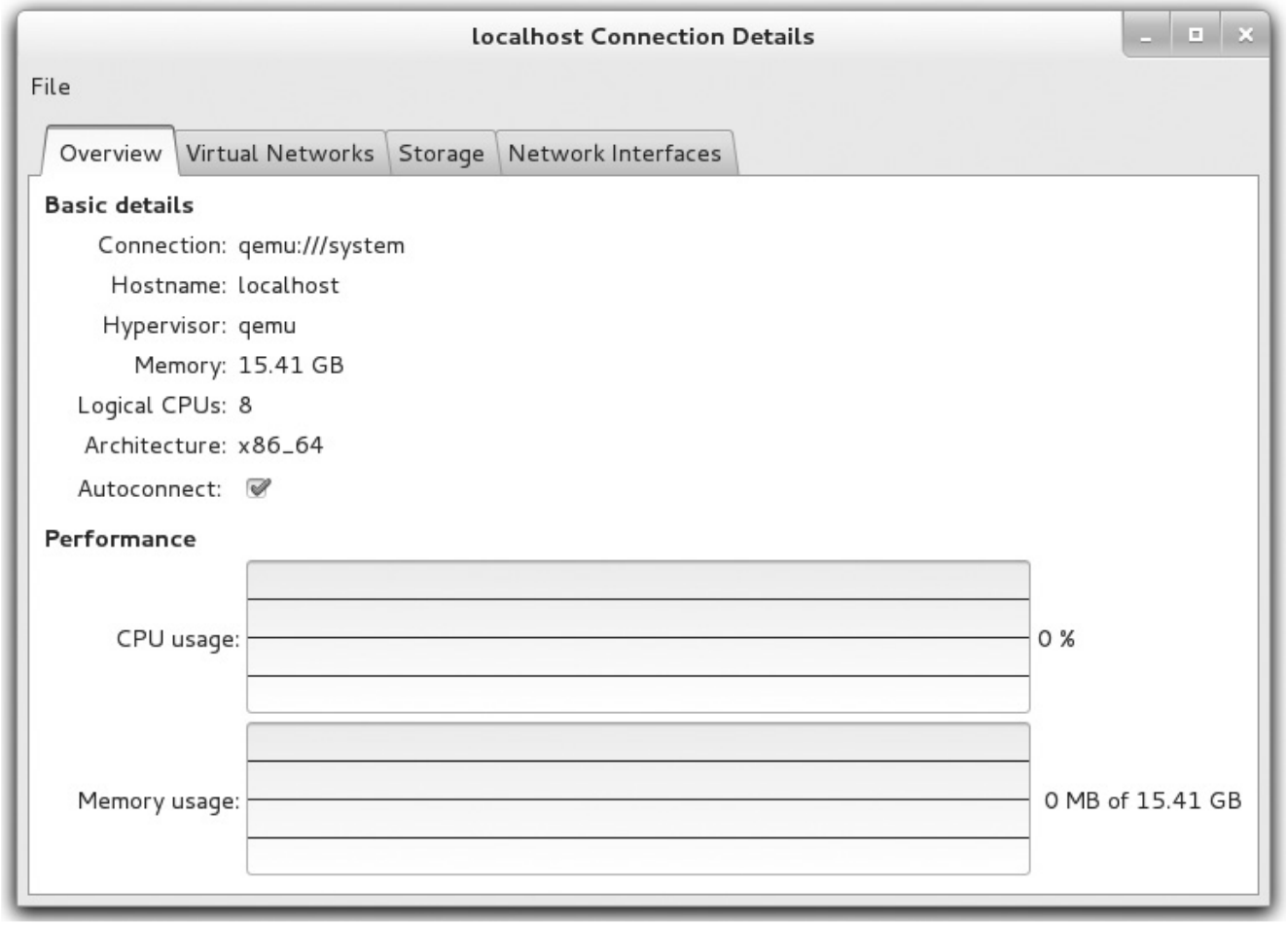


ТАБЛИЦА 2-2 Информация о хосте VM

Установка	Описание
Connection	Универсальный идентификатор ресурса (URI) для гипервизора.
Hostname	Имя хоста для хоста VM.
Hypervisor	**QEMU** используется **KVM**.
Memory	Доступная оперативная память от физической системы для виртуальных машин.
Logical CPUs	Доступные логические процессоры; это четырехъядерная система с гиперпоточность включена, что дает восемь логических процессоров.
Architecture	Архитектура процессора.
Autoconnect	Указывает, следует ли автоматически подключаться к гипервизору во время процесса загрузки

.

Как показано в **Таблице 2-2**, на вкладке Overview перечислены основные сведения о конфигурации виртуальной машины. Для следующего раздела оставайтесь в окне сведений о хосте для текущего гипервизора.

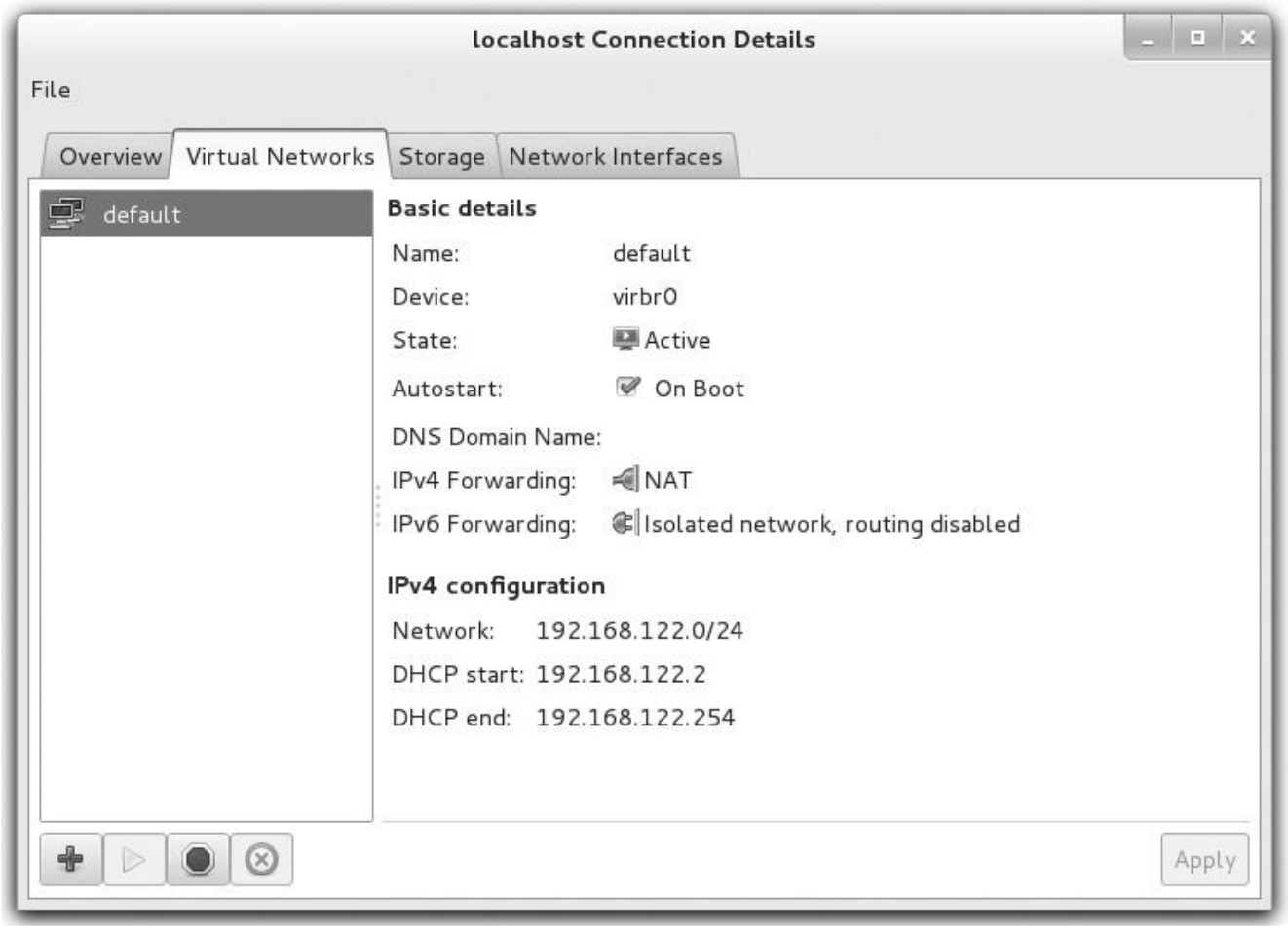
Виртуальные сети на гипервизоре

Теперь вы изучите сети, настроенные для виртуальных машин, в диспетчере виртуальных машин. В окне сведений о хосте текущего гипервизора перейдите **на вкладку Виртуальные сети**. Виртуальная сеть по умолчанию, показанная на **рисунке 2-4**, иллюстрирует стандартную сеть для виртуальных машин, созданную с помощью этого гипервизора.

Вы заметите, что данная сеть настроена на автоматический запуск при загрузке виртуальной машины. Поэтому, если на виртуальной машине настроен соответствующий виртуальный сетевой адаптер, а также команда клиента, связанная с протоколом динамической конфигурации хоста (DHCP), ему автоматически присваивается IP-адрес из указанного диапазона. Как отмечено на рисунке, назначенные адреса настроены для преобразования с использованием трансляции сетевых адресов (**NAT**), когда трафик передается на физический сетевой адаптер.

С помощью кнопок в левой нижней части экрана вы можете добавить новую виртуальную сеть, запустить и остановить существующую виртуальную сеть и удалить эту сеть. В **упражнении 2-1** вы создадите вторую виртуальную сеть.

РИСУНОК 2-4 Детали сети VM



УПРАЖНЕНИЕ 2-1

Создайте вторую виртуальную сеть

В этом упражнении вы создадите вторую виртуальную сеть на стандартном гипервизоре **KVM** в **GUI Virtual Machine Manager**. Для этого упражнения требуется система RHEL 7, настроенная как узел виртуализации, как обсуждалось ранее в этой главе.

1. Если у вас нет открытого окна сведений, щелкните правой кнопкой мыши стандартный гипервизор localhost (QEMU). В появившемся всплывающем меню выберите «Подробнее».
2. В открывшемся окне «Сведения о хосте» с именем локальной системы выберите вкладку «Виртуальные сети».
3. Нажмите знак «плюс» в левом нижнем углу вкладки «Виртуальные сети», чтобы открыть мастер создания новой виртуальной сети.
4. Прочтите инструкции, которым вы будете следовать в следующих шагах. Нажмите «Вперед», чтобы продолжить.
5. Назначьте имя для новой виртуальной сети. Для целей этой книги введите имя **outsider**. Нажмите «Вперед», чтобы продолжить.
6. Если он еще не введен, введите сетевой адрес **192.168.100.0/24** в текстовом поле **Сеть**. Система автоматически рассчитывает вероятные записи для другой сетевой информации, как показано на рисунке.

!!!! On the Job !!!!
Старайтесь избегать конфликтов IP-адресов с существующим оборудованием в локальной сети, например с маршрутизаторами и точками беспроводного доступа. Например, если кабельный модем использует IP-адрес 192.168.100.1 на своем интерфейсе, отмеченная сеть 192.168.100.0/24 на гипервизоре сделает этот кабельный модем недоступным с хоста Linux. Если у вас есть такое оборудование, измените сетевой адрес, показанный на рисунке.
!!!!!

7. Теперь вы можете выбрать диапазон IP-адресов в настроенной сети, который может быть назначен DHCP-клиенту. В соответствии с **таблицей 1-2 главы 1** вы настраиваете статический IP-адрес для системы **outsider1.example.org** в этой сети. Пока указанный IP-адрес 192.168.100.100 находится за пределами диапазона назначаемых DHCP IP-адресов, никаких изменений не требуется. Внесите необходимые изменения и нажмите «Вперед», чтобы продолжить.
8. При желании вы можете определить диапазон адресов IPv6. IPv6 является частью целей RHCE и будет рассмотрен в главе 12. Нажмите «**Вперед**», чтобы продолжить.
9. Теперь вам понадобится система, которая пересылает сетевые пакеты в физическую сеть, хотя бы потому, что именно так системы в этой сети взаимодействуют с системами в разных виртуальных сетях, возможно, на разных виртуальных хостах. Назначением может быть любое физическое устройство **в режиме NAT**, чтобы скрыть эти системы от удаленных хостов. Если вы не хотите ограничивать маршрутизацию от виртуальных машин до конкретной физической сетевой карты, по умолчанию в Пересылке в физическую сеть должны работать. Варианты описаны далее в этой главе, в обсуждении вкладки «**Сетевые интерфейсы**». Сделайте соответствующий выбор и нажмите «**Вперед**», чтобы продолжить.
10. Просмотрите сводку того, что было настроено. Если вы удовлетворены, нажмите «**Готово**». Внешняя сеть теперь будет доступна для использования новыми системами виртуальных машин и сетевыми картами.

Виртуальное хранилище на гипервизоре

Теперь вы изучите виртуальное хранилище, настроенное для виртуальных машин, в диспетчере виртуальных машин. В окне сведений о хосте для текущего гипервизора перейдите **на вкладку Хранилище** (**Storage**). Каталог файловой системы по умолчанию, показанный на **рисунке 2-5**, настраивает каталог **/var/lib/libvirt/images** для виртуальных образов. Такие образы представляют собой огромные файлы зарезервированного пространства, используемые в качестве жестких дисков для виртуальных машин.

Эти огромные файлы могут легко сокрушить многие системы. Один из способов получить контроль над такими файлами - выделить раздел или логический том этому каталогу **/var/lib/libvirt/images.**

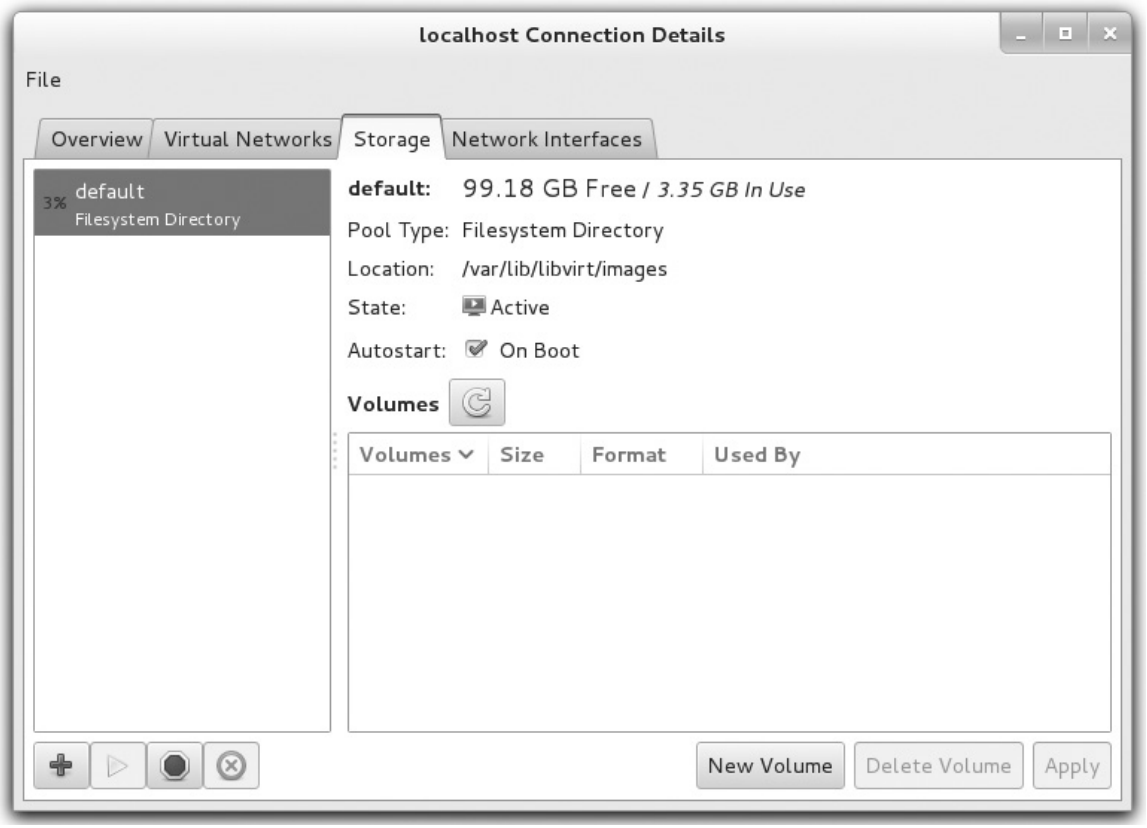
Поскольку вы, возможно, уже выделили наибольшее количество свободного места для раздела для вашего каталога **/home**, вы можете создать выделенное хранилище в этой области. Например, пользователь «**michael**» может иметь каталог **/home/michael/KVM**, в котором содержатся **его файлы VM**, используемые для виртуальных жестких дисков.

Следующие команды создадут соответствующий каталог как **обычный пользователь**, войдут в систему с правами пользователя **root**, установят соответствующие **контексты SELinux**, удалят каталог **/var/lib/libvirt/images** и заново создадут этот каталог как ссылку на соответствующий каталог пользователя:

```
$ mkdir /home/michael/KVM
$ su - root
# semanage fcontext -a -t virt_image_t '/home/michael/KVM(/.*)?'
# restorecon /home/michael/KVM
```

 # rmdir /var/lib/libvirt/images
 # ln -s /home/michael/KVM /var/lib/libvirt/images

РИСУНОК 2-5. Детали место хранения VM



Одним из преимуществ этой настройки является то, что она сохраняет настройки **SELinux** по умолчанию для каталога **/lib/libvirt/images**, как определено в файле **file_contexts** в каталоге **/etc/selinux/target/contexts/files**. Другими словами, эта конфигурация переживает перезгрузку **SELinux**, как объяснено в **Главе 4**.

Сетевые интерфейсы на гипервизоре

Теперь вы изучите сетевые интерфейсы, настроенные для виртуальных машин, в диспетчере виртуальных машин. В окне сведений о хосте для текущего гипервизора перейдите на вкладку **Сетевые интерфейсы**. Устройство сетевого интерфейса, показанное на **рисунке 2-6**, определяет только интерфейс обратной связи. Вы можете увидеть другие интерфейсы, такие как **адаптер Ethernet**, если он установлен в вашей системе.

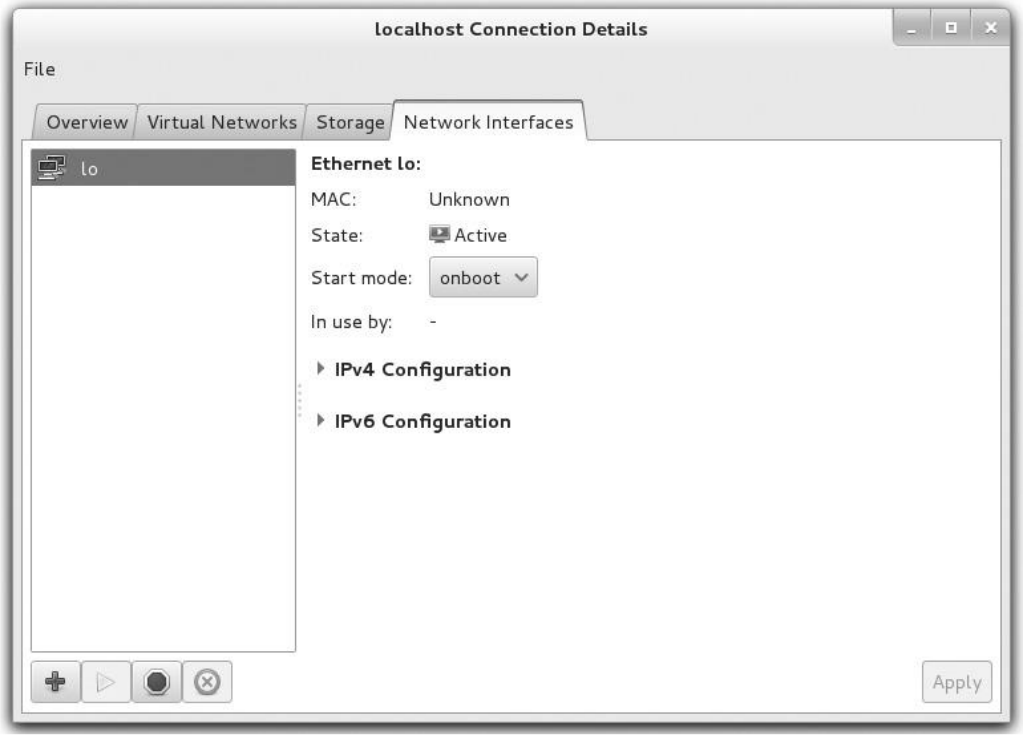
Если локальная система подключается через сетевую карту Ethernet или беспроводной адаптер, конфигурации по умолчанию должно быть достаточно. Правильно настроенная виртуальная машина должна иметь доступ к внешние сети с учетом параметров **конфигурации брандмауэра и пересылки IP**, описанных в главе 1. В RHEL 7 каждая виртуальная сеть **связана с виртуальным коммутатором, таким как virbr0**. Виртуальные коммутаторы по умолчанию работают в режиме **NAT**, когда трафик передается за пределы физического хоста.

Таким же образом, как на вкладках «**Виртуальная сеть**» и «**Хранилище**», вы можете настроить другой сетевой интерфейс, щелкнув знак «**плюс**» в левом нижнем углу вкладки «**Сетевые интерфейсы**». Это открывает окно **Configure Network Interfaces**, которое может помочь вам настроить один из четырех различных типов сетевых интерфейсов:

- **Bridge** Мосты физического и виртуального интерфейса

- **Bond** Объединяет два или более интерфейсов в одном логическом интерфейсе для резервирования.
- **Ethernet** Настраивает интерфейс
- **VLAN** Конфигурирует интерфейс с тегами IEEE 802.1Q VLAN

РИСУНОК 2-6. Сетевые карты VM



ЦЕЛЬ СЕРТИФИКАЦИИ 2.02

Настройте виртуальную машину на KVM

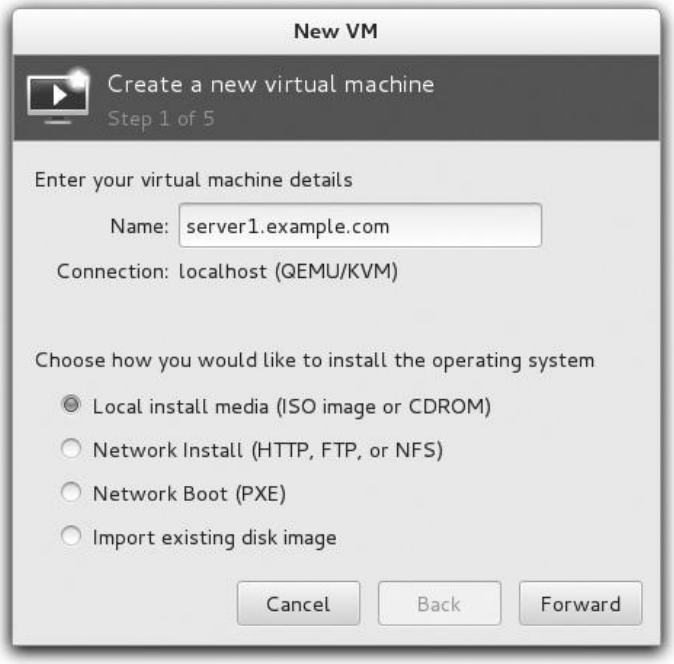
Процесс настройки **виртуальной машины в KVM** прост, особенно в диспетчере виртуальных машин. По сути, все, что вам нужно сделать, - это щелкнуть правой кнопкой мыши гипервизор QEMU, нажать «**Создать**» и следовать отображаемым подсказкам. Тем не менее, поскольку важно понимать процесс подробно, вы будете читать о нем шаг за шагом. Новые виртуальные машины могут быть настроены не только из графического интерфейса, но и из интерфейса командной строки. Как обычно для служб **Linux**, полученная конфигурация виртуальной машины сохраняется в виде текстового файла.

Настройте виртуальную машину на KVM

Чтобы следовать этому разделу, откройте диспетчер виртуальных машин в графическом интерфейсе. Еще один способ сделать это из командной строки на основе графического интерфейса (с помощью команды **virt-manager**). При появлении запроса **введите пароль администратора root**. Если гипервизор **localhost (QEMU)** отображается как не подключенный, щелкните его правой кнопкой мыши и выберите «**Подключиться**» во всплывающем меню. Следующими шагами вы настроите виртуальную машину, связанную с системой **server1.example.com**, описанной в **главе 1**. Теперь, чтобы настроить новую виртуальную машину, выполните следующие действия:

1. Щелкните правой кнопкой мыши гипервизор **localhost (QEMU).** Во всплывающем меню, которое появляется, нажмите **New**, чтобы открыть окно **New VM**, показанное на **рисунке 2-7.**

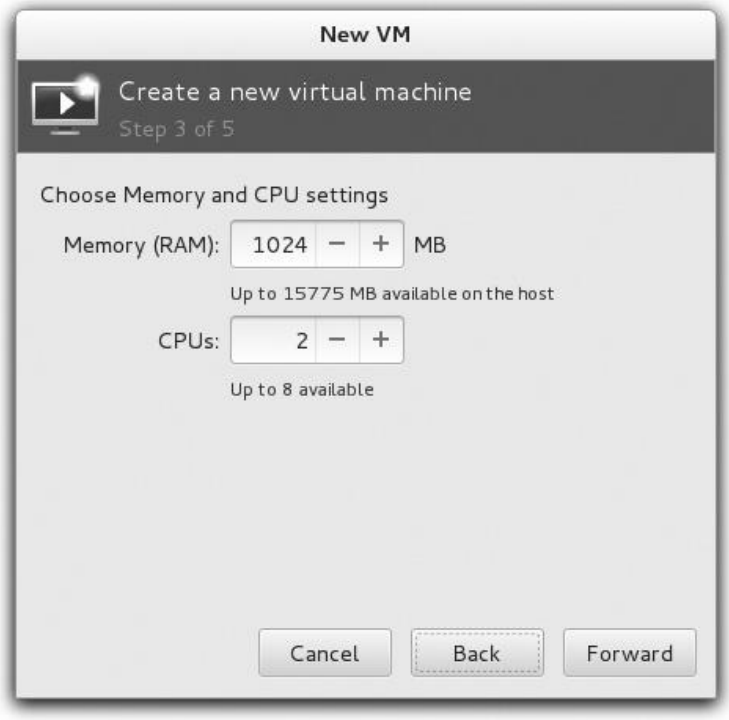
РИСУНОК 2-7 Создать новую ВМ



2. Введите имя для новой виртуальной машины; чтобы соответствовать обсуждению в оставшейся части этой книги, вы должны назвать эту виртуальную машину **server1.example.com**.
3. Теперь выберите, доступен ли установочный носитель на локальном установочном носителе (**образ ISO или компакт-диск**) или на сервере сетевой установки. Этот сервер может быть связан с **протоколом HTTP, NFS** или **FTP**. Выберите опцию **Local Install Media** и нажмите «**Вперёд**»**Forward**, чтобы продолжить. (В **лабораторной работе 1** вы перезапустите этот процесс с помощью опции «**Сетевая установка**».)
4. Если носитель доступен в локальном приводе **CD/DVD**, будет доступна опция для этого, как показано на **рисунке 2-8**. Но для выполнения этих шагов выберите «**Использовать образ ISO**» и нажмите «**Обзор**», чтобы перейти к расположению **образа ISO RHEL 7 DVD или сетевой загрузки**. Кроме того, вам нужно использовать раскрывающиеся текстовые поля «**Тип и версия ОС**», чтобы выбрать тип и дистрибутив операционной системы, как показано на рисунке.
5. Выберите объем оперативной памяти и количество процессоров, выделяемых виртуальной машине. Помните о минимумах, описанных ранее в этой главе и **главе 1** для RHEL 7. Как показано на **рисунке 2-9**, мелким шрифтом вы увидите информацию о доступной оперативной памяти и процессорах. Сделайте соответствующий выбор и нажмите «**Вперед**», чтобы продолжить.
6. Теперь вы настроите жесткий диск для виртуальной машины на экране, показанном на **рисунке 2-10**. Хотя его можно настроить на выделенных физических томах, стандартным является установка больших файлов в качестве виртуальных жестких дисков. В то время как местоположение по умолчанию для таких файлов каталог - **/var/lib/libvirt/images**, его можно изменить, как обсуждалось ранее в этой главе. На экзамене, вероятно, у вас будет более чем достаточно места в каталоге **/var/lib/libvirt /images**. Параметр «**Выбрать управляемое или другое существующее хранилище**» (**Select Managed or Other Existing Storage**) поддерживает создание виртуального жесткого диска в другом предварительно настроенном пуле хранения.
7. Убедитесь, что размер виртуального **диска составляет 16 ГБ**, и выбран параметр «**Выделить весь диск сейчас**» и нажмите «**Вперед**», чтобы продолжить.

РИСУНОК 2-8 Варианты установки носителя виртуальной машины

РИСУНОК 2-9 Выбор ОЗУ и ЦП виртуальной машины



8. В следующем окне подтвердите выбранные параметры. Нажмите **Advanced Options**, чтобы открыть варианты, показанные на **рисунке 2-11**.
   У вас могут быть варианты выбора одной из доступных виртуальных сетей. если вы выполнил **упражнение 2-1**, виртуальную сеть «**outsider**», связанную с IP-подсеть 192.168.100.0/24, также должна быть доступна.
9. Системе может потребоваться немного времени для создания виртуальной машины, включая большой файл, который будет служить виртуальным жестким диском. По завершении диспетчер виртуальных машин должен автоматически запустить систему с установочного DVD RHEL 7 в окне консоли.

РИСУНОК 2-10 Создать виртуальный жесткий диск

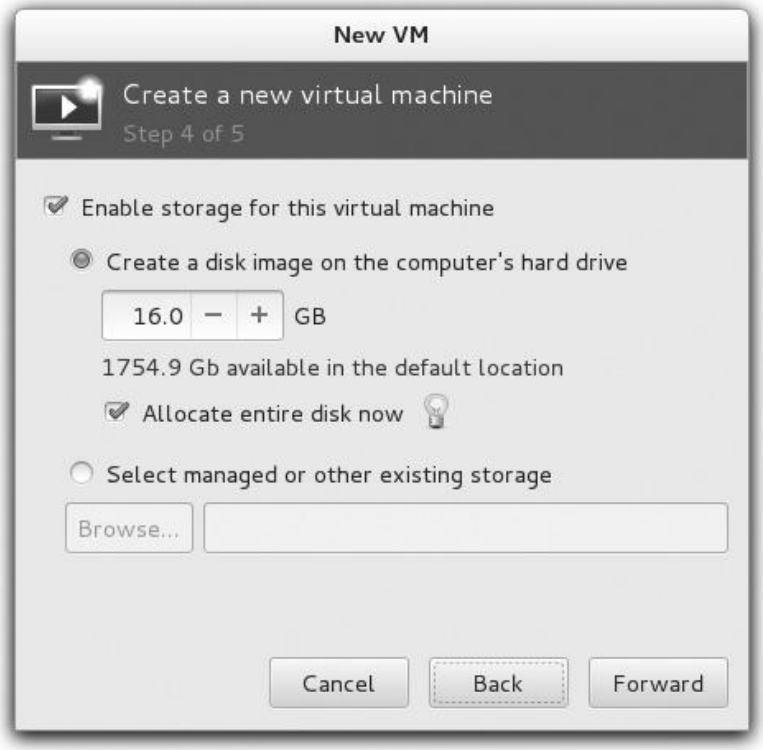
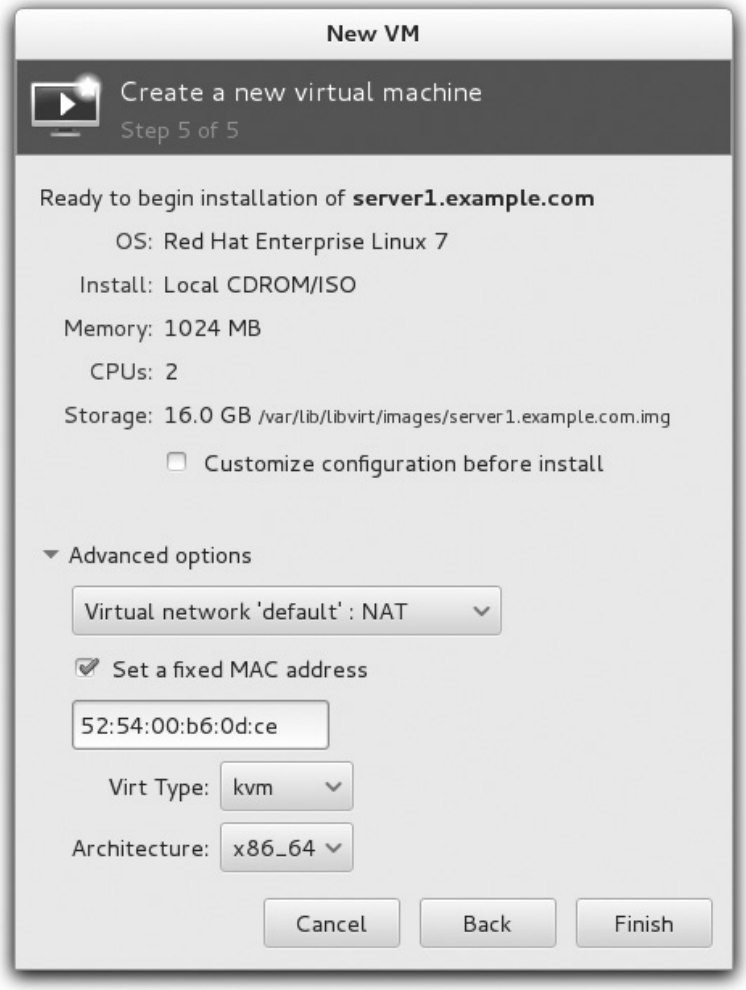
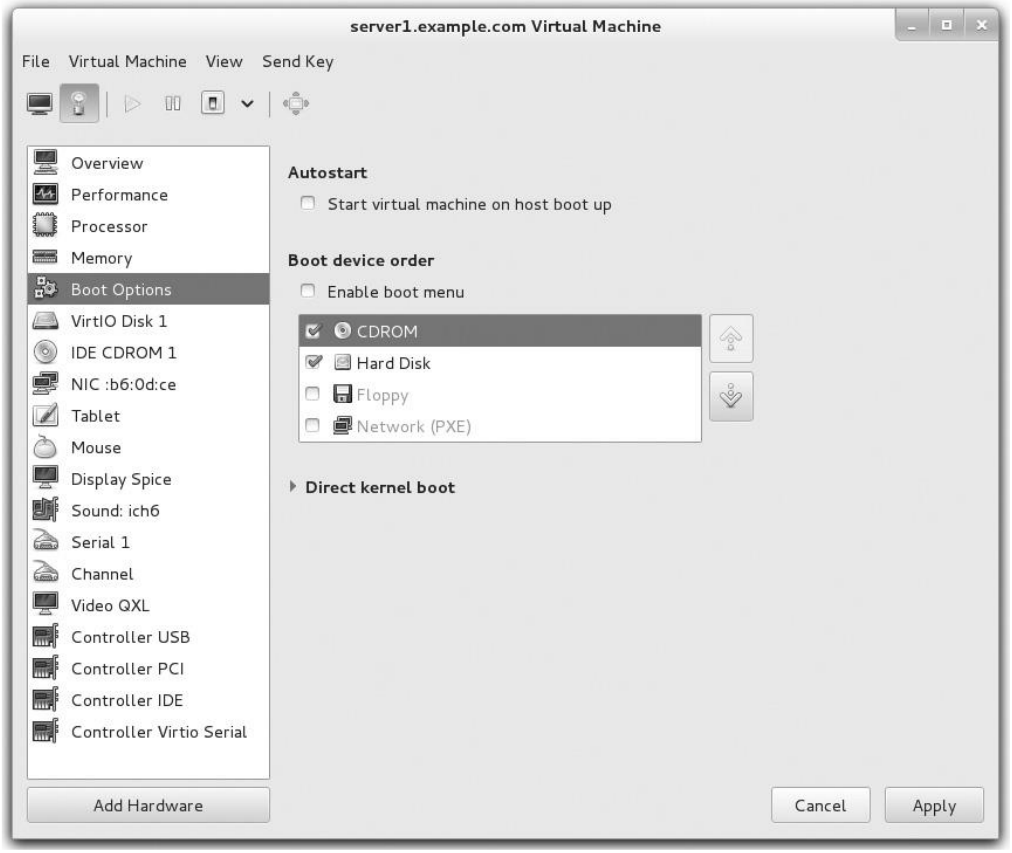


РИСУНОК 2-11 Просмотрите параметры конфигурации



10. Если новая система не запускается автоматически, эта ВМ должна быть указана в Диспетчере виртуальных машин, **показанном на рисунке 2-2**. После этого вы сможете выделить новую виртуальную машину (в данном случае с именем **server1.example.org**) и нажать «**Открыть**».
11. Теперь вы сможете приступить к установке RHEL 7 в **VM**, как **описано в главе 1**.
12. Если вы перезагрузите виртуальную машину, программа установки «**вытолкнет**» **DVD**. Если вы хотите повторно подключить DVD позже, вы должны нажать **«Вид | Детали,** подробно» **«View | Details»**, выберите IDE CDROM 1выберите «**Отключить**» (**Disconnect**), а затем нажмите «**Подключить**». В открывшемся окне «**Выбрать носитель**» выберите соответствующий файл с ISO-образом DVD или компакт-диском для физического носителя.
13. Помните, что когда вы выбираете программное обеспечение для установки, эта система является виртуальным гостем, а не виртуальным хостом, настроенным в главе 1. Нет необходимости добавлять какие-либо пакеты виртуализации в установку. Выберите **Сервер с графическим интерфейсом Server with GUI** без указания каких-либо дополнительных надстроек и нажмите «**Готово**».
14. После завершения установки нажмите «**Перезагрузить**» (**Reboot**). Если система снова попытается загрузиться с DVD-привода, вам необходимо изменить порядок загрузки между DVD-диском и жестким диском. Если система загружается прямо с жесткого диска, все готово!
15. Если система пытается загрузиться с DVD, необходимо выключить систему. Для этого нажмите **(Виртуальная машина | Выключить | Неисправность) (Virtual Machine | Shut Down | Shut Down)**.
16. Если вы запускаете эту последовательность команд впервые, диспетчер виртуальных машин запрашивает подтверждение. Нажмите **Да**.
17. Теперь нажмите «**Вид | Детали» (View | Details)**.
18. На левой панели выберите **Boot Options**, как показано на **рисунке 2-12**.

РИСУНОК 2-12 Параметры загрузки в ВМ



19. Один из способов изменить порядок загрузки - выделить CDROM и нажать кнопку со стрелкой вниз. Нажмите **Применить** (**Apply**); в противном случае изменения не будут записаны.
20. Теперь нажмите **Вид** | **Консоль** (**View | Console**), а затем **Виртуальная машина | Запустить (Virtual Machine | Run)**. Теперь система должна нормально загрузиться на экране начальной настройки, **описанном в главе 1**.

Еще одной причиной использования виртуальных машин является простота добавления дополнительных виртуальных жестких дисков. Процесс зависит от решения виртуальной машины. Для менеджера виртуальных машин RHEL 7 по умолчанию с решением KVM вы можете сделать это из окна машины, нажав «**Вид | Детали, Подробно**» **View | Details**. На этом экране вы увидите опцию «**Добавить оборудование**» (**Add Hardware**).

!!!! Exam watch !!!
Шаги, обсуждаемые в этом разделе, описывают, как выполнить задачу RHCSA для «доступа к консоли виртуальной машины». Также предлагается один метод, который можно использовать для «запуска и остановки виртуальных машин».
!!!!!

УПРАЖНЕНИЕ 2-2 Добавить виртуальные жесткие диски

В этом упражнении вы создадите дополнительный виртуальный жесткий диск на виртуальной машине на основе KVM. Мы предполагаем, что для этой цели существует виртуальная машина KVM, а также диспетчер виртуальных машин с графическим интерфейсом.

1. Откройте диспетчер виртуальных машин. Из командной строки в графическом интерфейсе запустите

virt-manager command

2. При появлении запроса введите пароль администратора и нажмите **Аутентификация**.
3. Выделите гипервизор **localhost (QEMU)**. Если он еще не подключен, щелкните его правой кнопкой мыши и выберите «**Подключиться**»(**Connect**) во всплывающем меню. Этот шаг может произойти автоматически.
4. Щелкните правой кнопкой мыши существующую виртуальную машину и выберите «**Открыть**» (**Open**) во всплывающем меню.
5. Нажмите «**Просмотр | Подробности» «View | Details»**. В левом нижнем углу открывшегося окна нажмите «**Добавить оборудование**» **«Add Hardware»**.
6. В появившемся окне «**Добавить новое виртуальное оборудование**» **«Add New Virtual Hardware»** выберите «**Хранилище**» **«Storage»** в меню слева.
7. В окне «**Хранилище**» **«Storage»** (показано далее) настройте диск объемом 1,0 ГБ, выберите «**Выделить весь диск**» (**Allocate Entire Disk Now**) и выберите тип устройства **Virtio Disk** в режиме кэширования по умолчанию. (Вы также можете выбрать диск SATA или IDE.) Выберите нужные параметры и нажмите «**Вперед**» (**Forward**), чтобы продолжить.
8. Вы можете увидеть подтверждение выбранных настроек. Если все в порядке, нажмите «**Готово**» (**Finish**), чтобы создать новый виртуальный жесткий диск.
9. Повторите предыдущие шаги, чтобы создать второй жесткий диск объемом 1 ГБ.
10. При следующей загрузке этой системы запустите команду **fdisk -l** от имени учетной записи **root**. Следует подтвердить соответствующую информацию о настроенных устройствах жесткого диска.

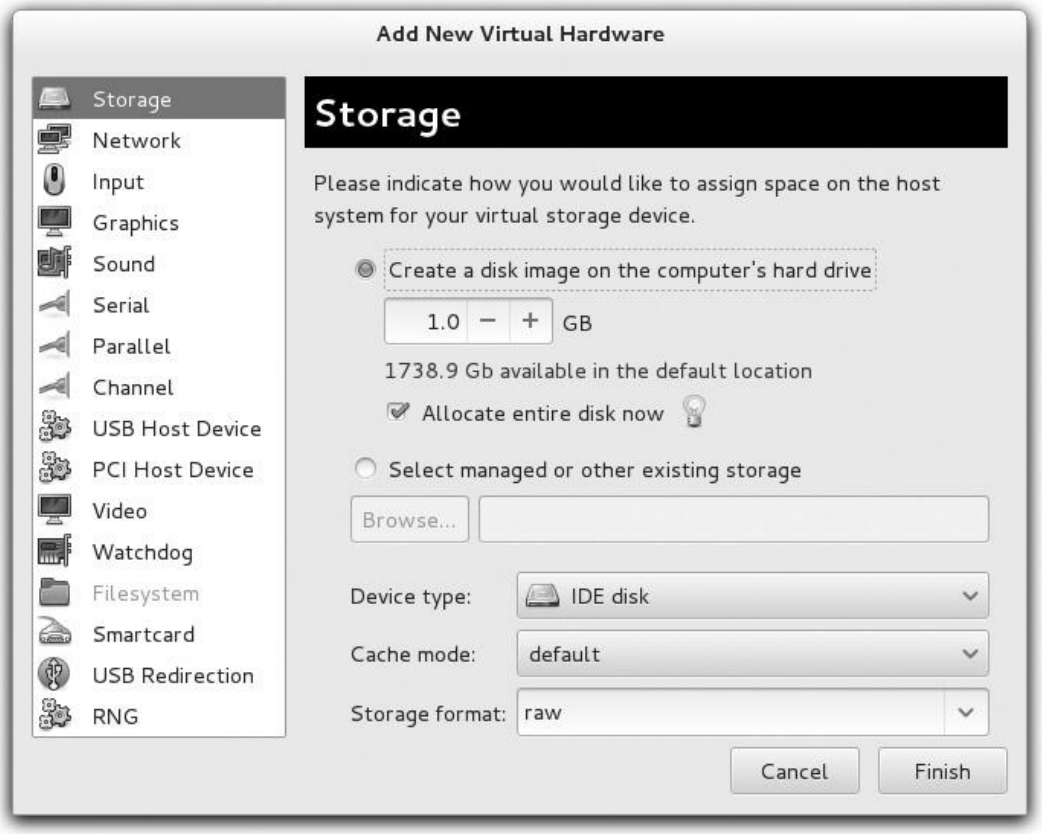


РИСУНОК 2-13 Файл конфигурации для виртуальной машины KVM

```
<domain type='kvm'>
  <name>server1.example.com</name>
  <uuid>7782a007-60eb-4292-b731-8b2b60594933</uuid>
  <memory unit='KiB'>1048576</memory>
  <currentMemory unit='KiB'>1048576</currentMemory>
  <vcpu placement='static'>2</vcpu>
  <os>
    <type arch='x86_64' machine='pc-i440fx-rhel7.0.0'>hvm</type>
    <boot dev='hd'/>
    <bootmenu enable='no'/>
  </os>
  <features>
    <acpi/>
    <apic/>
    <pae/>
  </features>
  <clock offset='utc'/>
  <on_poweroff>destroy</on_poweroff>
  <on_reboot>restart</on_reboot>
  <on_crash>restart</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/server1.example.com.img'/>
      <target dev='vda' bus='virtio'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x06' function='0x0'/>
    </disk>
    <disk type='file' device='disk'>
      <driver name='qemu' type='raw' cache='none'/>
      <source file='/var/lib/libvirt/images/server1.example.com-1.img'/>
      <target dev='hda' bus='ide'/>
      <address type='drive' controller='0' bus='0' target='0' unit='0'/>
    </disk>
```

Конфигурационные файлы KVM

Виртуальные машины на основе **KVM** обычно настраиваются в двух разных каталогах: **/etc/libvirt** и **/var/lib/libvirt**. Когда виртуальная машина **KVM** настроена, она устанавливается в файлах в формате XML в каталоге **/etc/libvirt/qemu**. Например, **на рисунке 2-13** показан соответствующий фрагмент файла конфигурации для основной виртуальной машины, которую мы использовали для подготовки этой книги (**server1.example.com.xml**).

Важные параметры для **VM** помечены. Например, объем памяти отображается в КиБ (1 КиБ=1024 байта), выделены два виртуальных ЦП, эмулятор KVM, диск находится в файле **server1.example.com.img** в Каталоге **/var/lib/libvirt/images** и т.д. Хотя вы можете редактировать этот файл конфигурации напрямую, изменения не будут реализованы до тех пор, пока служба **libvirtd** не будет перезапущена с помощью такой команды, как **systemctl restart libvirtd**.

Управление виртуальными машинами из командной строки

Конечно, инструменты командной строки можно использовать для создания, клонирования, преобразования и установки виртуальных машин на RHEL 7. Ключевыми командами для этого являются **virt-install, virsh и virt-clone**. Команда **virsh** - это особенно полезный способ решения двух разных задач RHCSA.

Команда virt-install

Вы можете выполнить те же шаги, что и ранее в этой главе, используя Диспетчер виртуальных машин. Все, что вам нужно, это команда **virt-install**. Команда с параметром **--help** показывает все параметры для требуемой информации, описанной ранее. Посмотрите на экран справки команды и сравните с примером, показанным на **рисунке 2-14**.

Для многих это проще, чем настройка **GUI Virtual Machine Manager**. Сообщение **Create Domain** в конце примера на **рисунка 2-14** запускает консольное окно с графическим представлением данной программы установки. Если вы получаете сообщение об ошибке «**Не удается открыть дисплей» (cannot open display)**, убедитесь, что вы открыли сеанс рабочего стола GNOME с правами root.

Если вы допустили ошибку с помощью команды **virt-install**, вы можете прервать процесс, **нажав ctrl-c**. Но имейте в виду, что вновь созданная виртуальная машина все еще работает. И теперь есть файл конфигурации и виртуальный диск для этой виртуальной машины. Если вы попытаетесь повторно запустить команду **virt-install** с тем же именем для виртуальной машины, появится сообщение об ошибке. Поэтому, если вы хотите использовать одно и то же имя для виртуальной машины, выполните следующие действия:

1. Остановите только что созданную виртуальную машину. Если это система **tester1.example.com**, показанная на **рисунке 2-14**, вы можете сделать это с помощью следующей команды:

   # virsh destroy tester1.example.com

2. Удалите связанный файл конфигурации XML в каталоге **/etc/libvirt/qemu** и файл виртуального диска, обычно создаваемый в каталоге **/var/lib/libvirt/images**. Однако в этом нет необходимости, если вы хотите повторно использовать файл.

   # virsh undefine tester1.example.com --remove-all-storage
3. Теперь вы сможете снова запустить команду **virt-install** с тем же именем для виртуальной машины.

РИСУНОК 2-14 Настройте виртуальную машину с помощью команды virt-install

```
[root@Maui ~]# virt-install --name=tester1.example.com \
> --ram=1024 --vcpus=2 \
> --disk path=/var/lib/libvirt/images/tester1.example.com.img,size=16 \
> --graphics=spice \
> --location=ftp://192.168.122.1/pub/inst \
> --os-type=linux \
> --os-variant=rhel7

Starting install...
Retrieving file .treeinfo...                          | 4.2 kB     00:00 !!!
Retrieving file vmlinuz...                            | 9.3 MB     00:00 !!!
Retrieving file initrd.img...                         |  68 MB     00:00 !!!
Allocating 'tester1.example.com.img'                  |  16 GB     00:00
Creating domain...                                    |    0 B     00:00
Domain installation still in progress. You can reconnect to
the console to complete the installation process.
```

Команда virt-install и Kickstart

Для установок **Kickstart**, описанных далее в этой главе, команда **virt-install** может использоваться для цитирования файла конфигурации **Kickstart**. Для этого вам необходимо понять некоторые из ключевых опций(ключей комманды), связанных с командой **virt-install**, как показано в **таблице 2-3**.

Например, следующая команда **virt-install** установит систему с именем **outsider1.example.org** автоматически использует файл **Kickstart** с именем **ks1.cfg с FTP-сервера** на указанном IP-адресе, 1 ГБ ОЗУ и виртуальный диск **outsider1.example.org.img**.

**# virt-install -n outsider1.example.org -r 1024 --disk **
**path=/var/lib/libvirt/images/outsider1.example.org.img,size=16 -l ftp://192.168.122.1/pub/inst -x **
ks=ftp://192.168.122.1/pub/ks1.cfg

Эта команда содержит **несколько переключателей (опций).** Большинство показанных переключателей описаны в примерах, перечисленных на странице руководства для команды **virt-install**. Вы можете отметить дополнительные параметры команды, которые полезны, но не обязательны для установки RHEL 7. Однако, они должны искать данный файл быстрой установки **Kickstart**.

ТАБЛИЦА 2-3 Командные переключатели для virt-install

Переключатель Switch	Описание
-n (--name)	Устанавливает имя для **ВМ**.
--vcpus	Настраивает количество виртуальных процессоров.
-r (--ram)	Настраивает объем оперативной памяти в **МБ**.
--disk	Определяет виртуальный диск; часто используется с **path=/var/lib/libvirt/images/virt.img,size=size_in_GB**.
-l (--location)	Указывает каталог или URL с установочными файлами (**эквивалентно --location**).
--graphics	Определяет настройки графического отображения гостя; допустимыми параметрами являются **vnc, spice и none**.
-x (--extra-args =)	Включает в себя дополнительные данные, такие как URL-адрес файла **Kickstart**.

Запомните формат дополнительных аргументов с кавычками, которые также могут быть выражены следующим образом:

--extra-args="ks=ftp://192.168.122.1/pub/ks1.cfg"

Команда virsh

Команда **virsh** запускает интерфейс для существующих виртуальных машин **KVM**. При запуске в одиночку он перемещается из обычной командной строки в следующую строку:

virsh #

Из этой подсказки запустите команду справки. Он включает доступ к ряду команд, некоторые из которых перечислены в **таблице 2-4**. Не все команды, показанные в выводе на экран справки, активны для KVM. Те команды **virsh**, которые можно использовать, также можно запускать непосредственно из командной строки **bash**; например, команда **virsh list --all** выводит список всех сконфигурированных виртуальных машин, независимо от того, работают они в данный момент или нет. В контексте **KVM** экземпляром операционной системы, работающей на ВМ, является домен. На доменные имена ссылаются разные команды **virsh**.

Посмотрите на вывод команды **virsh list --all** в нашей системе:

```
Id    Name                           State
----------------------------------------------------
-     server1.example.com            shut off
-     tester1.example.com            shut off
```

С помощью правильных команд **virsh** вы можете достичь двух целей RHCSA. Сначала следующая команда запускает указанную систему **server1.example.com**:

 # virsh start server1.example.com

Команда **virsh shutdown** корректно завершает работу операционной системы и выключает виртуальную машину:

 # virsh shutdown server1.example.com

Чтобы немедленно отключить виртуальную машину, вам нужно выполнить несколько более строгую команду:

 # virsh destroy server1.example.com

ТАБЛИЦА 2-4 Избранные команды в командной строке virsh

Команд virsh	Описание
autostart <domain>	Настраивает домен для запуска во время процесса загрузки хост-системы
capabilities	Перечисляет возможности местного гипервизора
edit <domain>	Редактирует файл конфигурации XML для домена
list --all	Перечисляет все домены
start <domain>	Загружает данный домен
shutdown <domain>	Коректно закрывает данный домен

Команда **virsh destroy** функционально эквивалентна отключению шнура питания в физической системе. Поскольку это может привести к различным проблемам, лучше всего остановить виртуальную машину, выполнив команду **poweroff** внутри виртуальной машины.

Даже в самых защищенных системах происходят сбои питания. Обновления ядра по-прежнему требуют перезагрузки системы. В этих случаях полезно автоматизировать запуск виртуальных машин на виртуальном хосте во время процесса загрузки.

Кроме того, команда **virsh** - это самый простой способ убедиться, что виртуальная машина запускается при следующей загрузке системы. Например, следующая команда настраивает указанную систему **tester1.example.com** для запуска во время процесса загрузки хост-системы:

 # virsh autostart tester1.example.com

Когда процесс загрузки завершится как для хоста, так и для виртуальной машины, вы сможете использовать такие команды, как **ssh**, для обычного подключения к этой системе виртуальной машины. Однако из графического интерфейса физического хоста вам все равно придется запустить диспетчер виртуальных машин и подключиться к связанному гипервизору, чтобы фактически получить доступ к виртуальной консоли для этой системы **tester1.example.com.**

Команда создает символически связанный файл в каталоге **/etc/libvirt/qemu/autostart**. Чтобы отменить процесс, либо выполните команду

 # virsh autostart --disable tester1.example.com

или удалите программно связанный файл с именем целевой виртуальной машины из этого каталога.

!!!! Exam watch !!!
Чтобы запустить и остановить виртуальную машину, вы можете запустить команды virsh start *vmname* и virsh destroy *vmname*, где *vmname* - это доменное имя виртуальной машины, как показано в выходных данных команды virsh list --all.
!!!!

!!!! Exam watch !!!
Чтобы настроить автоматический запуск виртуальной машины при загрузке системы, вы можете выполнить команду virsh autostart *vmname*, где *vmname* - это имя виртуальной машины, как показано в выводе команды virsh list --all.
!!!!!

Команда virt-clone

Команда **virt-clone** может использоваться для клонирования существующей виртуальной машины. Перед началом процесса убедитесь, что клонируемая система выключена. Это просто; один пример, **tester1.example.com** создана с **server1.example.com** показана на **рисунке 2-15.**

РИСУНОК 2-15 Клонирование виртуальной машины

```
[root@Maui ~]# virt-clone --original=server1.example.com \
> --name=tester1.example.com \
> --file=/var/lib/libvirt/images/tester1.example.com.img \
> --file=/var/lib/libvirt/images/tester1.example.com-1.img \
> --file=/var/lib/libvirt/images/tester1.example.com-2.img
Allocating 'tester1.example.com.img'                    |  16 GB     00:46
Allocating 'tester1.example.com-1.img'                  | 1.0 GB     00:00
Allocating 'tester1.example.com-2.img'                  | 1.0 GB     00:00

Clone 'tester1.example.com' created successfully.
[root@Maui ~]#
```

`Обратите внимание, что вы должны указать путь к виртуальному диску с помощью ключа **--file** для каждого диска исходной виртуальной машины, которую вы хотите клонировать. В этом случае **server1.example.com** имеет три виртуальных диска, потому что мы добавили два новых диска в **упражнении 2-2**.

После завершения процесса вы не только найдете отмеченные образы жесткого диска в указанных каталогах, но также найдете новый файл конфигурации XML для этой виртуальной машины в каталоге **/etc/libvirt/qemu**.

При первой загрузке клонированной машины лучше всего загрузить ее в **rescue target**. **Rescue target** не запускает большинство служб, включая сетевые (для получения дополнительной информации см. Главу 5). В этом случае вы сможете изменить любые сетевые параметры, такие как **имя хоста и IP-адрес**, перед запуском клонированного компьютера в производственной сети. Кроме того, вы должны убедиться, что аппаратный (**MAC**) адрес для соответствующей сетевой карты отличается от адреса исходной **VM**, чтобы избежать конфликтов с исходной сетевой картой.

Хотя этот процесс не может быть сложным для одной или двух виртуальных машин, представьте себе, что вы создадите несколько десятков виртуальных машин, каждая из которых впоследствии будет настроена для различных служб. Этой ситуации поможет большая автоматизация. Для этого Red Hat предоставляет систему, известную как **Kickstart**.

ЦЕЛЬ СЕРТИФИКАЦИИ 2.03

Опции автоматической установки

Kickstart - это решение Red Hat для автоматической установки RHEL. Думайте о каждом из шагов, выполняемых в процессе установки, как о вопросах. С помощью **Kickstart** на каждый из этих вопросов можно автоматически ответить одним текстовым файлом. С **Kickstart** вы можете очень быстро настроить одинаковые системы. Для этого файлы **Kickstart** полезны для быстрого развертывания и распространения систем Linux.

Кроме того, процесс установки дает возможность узнать больше о RHEL 7 - не только загрузочный носитель, но и разделы и логические тома, которые можно настроить после завершения установки. С появлением виртуальных машин нетрудно настроить автоматическую установку на новую виртуальную машину с помощью **Kickstart**.

Шаги, описанные в этом разделе, предполагают подключение к **FTP-серверу** с установочными файлами RHEL 7, созданными и настроенными в лабораторной **работе 2 главы 1**.

Kickstart Concepts

Одна из проблем, возникающих при установке на основе **Kickstart**, заключается в том, что в нее не входят пользовательские настройки, созданные после завершения базовой установки. Хотя эти параметры можно включить на основе сценариев после установки, это выходит за рамки экзамена RHCSA.

Существует два способа создания необходимого файла конфигурации **Kickstart**:

- Начните с файла **anaconda-ks.cfg** из домашнего каталога пользователя **root /root**.
- Используйте графический конфигуратор **Kickstart**, доступный через команду **system-config-kickstart.**

!!!! Exam watch !!!
Хотя рекомендуется отслеживать на https://bugzilla.redhat.com наличие ошибок, связанных с ключевыми компонентами, это может быть особенно важно в отношении Kickstart. Например, ошибка 1121008 позволяет предположить, что до Anaconda версии

19.31.83-1 установка Kickstart на основе NFS с настраиваемыми параметрами монтирования была проблематичной.
!!!!!

Первый вариант позволяет использовать файл шаблона **Kickstart**, созданный для локальной системы **Anaconda**: **anaconda-ks.cfg** в каталоге **/root**. Второй вариант, **Kickstart Configurator**, подробно обсуждается далее в этой главе.

Относительно легко настроить файл **anaconda-ks.cfg** для разных систем. Вскоре вы увидите, как настроить этот файл по мере необходимости для разных размеров жесткого диска, имен хостов, IP-адресов и многого другого.

Настройка локального доступа к файлу Kickstart

После настройки файла Kickstart его можно настроить на локальном носителе, таком как USB-ключ, компакт-диск, запасной раздел или даже дисковод гибких дисков. (Не смейтесь; многие системы виртуальных машин, включая KVM, упрощают использование виртуальных дисководов гибких дисков.) Для этого выполните следующие основные шаги:

1. Сконфигурируйте и отредактируйте файл **anaconda-ks.cfg** по желанию. Мы опишем этот процесс более подробно в ближайшее время.
2. Смонтируйте нужный локальный носитель. Вам может потребоваться выполнить команду, такую как **fdisk -l**, в качестве пользователя **root**, чтобы определить соответствующий файл устройства. Если диск не монтируется автоматически, вы можете подключить диск с помощью команды, например, **mount /dev/sdb1 /mnt**
3. Скопируйте файл **Kickstart** в **ks.cfg** на смонтированном локальном носителе. (С другими именами все в порядке; **ks.cfg** - это просто наиболее распространенное имя файла для этой цели в документации **Red Hat**.)
4. Убедитесь, что файл **ks.cfg** имеет как минимум разрешения на чтение для всех пользователей. Если **SELinux** активен в локальной системе, контексты обычно должны совпадать с контекстами других файлов в том же каталоге. Для получения дополнительной информации см. **Главу 4**.

   Помните, что файл конфигурации **Kickstart на FTP-сервере** может представлять угрозу безопасности. Это почти как ДНК системы. Если хакер-хакер завладеет этим файлом, он может использовать его для создания копии ваших систем и посмотреть, как взломать и скомпрометировать ваши данные. Поскольку этот файл обычно содержит пароль администратора, вы должны изменить пароль, как только система загрузится в первый раз.

!!!! On tht Job !!!
Будьте осторожны с файлом конфигурации Kickstart. Если прямой вход root не отключен, файл содержит пароль администратора root. Даже если этот пароль зашифрован, хакер с подходящими инструментами и копия этого файла конфигурации Kickstart могут выполнить атаку по словарю и расшифровать этот пароль, если он недостаточно безопасен.
!!!

Теперь вы должны быть готовы использовать носитель **Kickstart** в другой системе. Вскоре вы попробуете это снова в упражнении.

5. Теперь попробуйте получить доступ к файлу **Kickstart** на локальном носителе. Загрузите **установочный CD/DVD RHEL 7**. Когда появится первое меню, выделите «**Установить Red Hat Enterprise Linux 7.0**» (**Install Red Hat Enterprise Linux 7.0**)и нажмите «**Tab**». Должны появиться команды для **Anaconda**, аналогичные приведенным ниже, и курсор должен появиться в конце этой строки:

> vmlinuz initrd=initrd.img inst.stage2=hd:LABEL=RHEL-7.0 \x20Server.x86_64 quiet6.

6. Добавьте информацию о местоположении файла **Kickstart** в конец строки. Например, следующее дополнение находит этот файл в первом разделе второго жесткого диска, который может быть USB-накопителем:

ks=hd:sdb1:/ks.cfg

Или, если файл **Kickstart** находится на загрузочном компакт-диске, попробуйте добавить следующую команду:

ks=cdrom:/ks.cfg

В качестве альтернативы, если файл кикстарта находится на первом дисководе гибких дисков, введите следующее:

ks=hd:fd0:/ks.cfg

В этом методе могут быть проб и ошибок. Да, файлы устройств обычно назначаются в последовательности (**sda, sdb, sdc** и т. д.). Однако, если вы не загрузите Linux с заданным носителем, нет уверенности в том, какой файл устройства назначен конкретному диску.

Настройка доступа в сети для Kickstart

Процесс настройки файла **Kickstart** с локального носителя может занять много времени, особенно если вам нужно перейти из системы в систему для загрузки этого файла. Во многих случаях более эффективно настроить файл **Kickstart** на сетевом сервере. Одно логическое расположение - это тот же сетевой сервер, который используется для установочных файлов. Например, на основе **FTP-сервера**, созданного в главе 1, лабораторная работа 2, предположим, что в каталоге **/var/ftp/pub FTP-сервера** находится файл **ks.cfg**. Контексты **SELinux** должны соответствовать контексту этого каталога, что может быть подтверждено следующими командами:

 # ls -Zd /var/ftp/pub
 # ls -Z /var/ftp/pub

Как только соответствующий файл **ks.cfg** находится в каталоге **/var/ftp/pub**, вы можете получить к нему доступ, добавив следующую директиву в конец строки **vmlinuz**, описанной ранее **в шаге 5**:

ks=ftp://192.168.122.1/pub/ks.cfg

Аналогичные параметры возможны для файла кикстарта на NFS и HTTP-сервере, а именно:

ks=nfs:192.168.122.1:/ks.cfg
ks=http://192.168.122.1/ks.cfg

Если в локальной сети есть работающий **DNS-сервер**, вы можете заменить IP-адрес именем хоста или полным доменным именем целевого сервера.

!!!!! On the Job !!!
Чтобы упростить процесс создания установочного сервера на основе Kickstart, см. Проект Cobbler по адресу http://cobbler.github.com. Cobbler использует профили и небольшие

блоки кода (так называемые «фрагменты») для динамического создания файлов Kickstart и автоматизации сетевых установок.
!!!!

Образец файла Kickstart

Мы основали этот раздел на файле **anaconda-ks.cfg**, созданном при установке RHEL 7 внутри виртуальной машины на основе KVM с несколькими добавленными комментариями. Несмотря на то, что вы можете использовать его в качестве образца файла, обязательно настройте его для своего оборудования и сети. В этом разделе рассказывается только о том, что вы можете сделать с помощью файла **Kickstart**; Ваша версия этого файла может отличаться.

!!!! Exam watch !!!
В отличие от того, что доступно для многих других пакетов Red Hat, документация по Kickstart, доступная в установленной системе RHEL 7, немного скудна. Другими словами, вы не можете полагаться на справочные страницы или файлы в каталоге /usr/share/doc для справки по кикстарту во время экзамена. Если вы не уверены в том, какие конкретные команды нужно включить в файл Kickstart, может помочь Конфигуратор Kickstart, описанный далее в этой главе.
!!!!

Хотя большинство параметров говорят сами за себя, мы включили наше объяснение каждой команды в файле. Этот файл иллюстрирует лишь небольшую часть доступных команд. Для получения дополнительной информации о каждой команде (и опциях) в этом файле ознакомьтесь с последним Руководством по установке RHEL 7, которое доступно в Интернете по адресу **https://access.redhat.com/documentation**.

При настройке файла **Kickstart** соблюдайте следующие основные правила и рекомендации:

- В общем, сохраняйте порядок директив. Однако возможны некоторые изменения в зависимости от того, идет ли установка с локального носителя или по сети.
- Вам не нужно использовать все варианты.
- Если вы пропустите обязательную опцию, пользователю будет предложено ответить.
- Не бойтесь вносить изменения; например, директивы, связанные с разделами, по умолчанию закомментированы.
- Перенос строки в файле допустим.

!!!! On the Job !!!
Если вы пропустите опцию, процесс установки остановится на этом этапе. Это простой способ проверить, правильно ли настроен файл кикстарта. Однако, поскольку некоторые параметры Kickstart изменяют разделы на жестком диске, даже тесты могут быть опасными. Поэтому лучше всего протестировать файл Kickstart на тестовой системе или, что еще лучше, на экспериментальной ВМ.
!!!!

Ниже приведен код из одного из наших **файлов anaconda-ks.cfg**. Первая строка говорит нам, что этот файл был создан для RHEL 7:

#version=RHEL7

Затем команда **auth** устанавливает **Shadow Password Suite (--enableshadow)** и 512-битный алгоритм шифрования SHA для шифрования пароля **(--passalgo = sha512).** Пароль, зашифрованный по алгоритму SHA512, начинается с **$6**:

authconfig --enableshadow --passalgo=sha512

команда проста; он запускает процесс установки с первого привода **DVD/CD** в системе:

cdrom

Следующим шагом является указание источника установочных файлов. Чтобы использовать **DVD RHEL 7**, оставьте существующую запись **cdrom**. Для установки с **сервера NFS** укажите URI следующим образом. Если есть надежный **DNS-сервер** для локальной сети, вы можете заменить имя хоста на IP-адрес.

nfs --server=192.168.122.1 --dir=/inst

Вы также можете настроить соединение с **сервером FTP или HTTP**, подставив одну из команд, показанных здесь. Указанные каталоги основаны на установочных серверах FTP и HTTP, созданных в главе 1:

url –url http://192.168.122.1/inst

или же

url –url ftp://192.168.122.1/pub/inst

Если **файл ISO**, представляющий DVD RHEL 7, существует в разделе локального жесткого диска, вы также можете указать это. Например, следующая директива указывает на компакт-диски или **DVD ISO в разделе /dev/sda10**:

harddrive --partition=/dev/sda10 --dir=/tmp/michael/

Firstboot --enable запускает агент установки во время первой установки. Если вы хотите избежать процесса **Firstboot**, вы также можете заменить эту строку директивой **firstboot --disabled**. Поскольку нет способа настроить файл **Kickstart** с ответами на запросы **Firstboot**, эта директива **--disabled** помогает автоматизировать процесс **Kickstart**.

firstboot --disabled

Следующая директива **ignoredisk** определяет тома только на указанном **диске vda**. Конечно, это работает, только если на целевой виртуальной машине есть указанный виртуальный диск. (На таких виртуальных машинах можно указывать **диски SAS или SCSI**, что может противоречить этим директивам.)

 # ignoredisk --only-use=vda

Команда **lang** устанавливает язык, который будет использоваться в процессе установки. Это имеет значение, если установка останавливается из-за отсутствия команды в этом файле. Команда клавиатуры не требует пояснений - она задает раскладку клавиатуры для настройки на этом компьютере.

keyboard --vckeymap=us --xlayouts='us'
lang en_US.UTF-8

Требуемая команда **network** является самой простой, если в локальной сети есть **DHCP-сервер network --device eth0 --bootproto dhcp**. Напротив, следующие две строки настраивают

информацию **статического IP-адреса**, с **указанными маской сети (--netmask),** адресом шлюза **(--gateway), DNS-сервером (--nameserver**) и именем компьютера **(--hostname).**

network --bootproto static --device=eth0 --gateway=192.168.122.1 □
--ip=192.168.122.150 --netmask=255.255.255.0 --noipv6 □
--nameserver==192.168.122.1 --activate
network --hostname tester1.example.com

Обратите внимание, что вся статическая информация о сети для команды **network** должна быть в одной строке. Перенос строки, если параметры превышают пространство в текстовом редакторе, является приемлемым. Если вы настраиваете этот файл для другой системы, **не забудьте** соответствующим образом **изменить IP-адрес и имя хоста**. Помните, что если вы не настроили сеть во время процесса установки, она не будет записана в файл **anaconda-ks.cfg**. Учитывая сложность директивы **network**, вы можете использовать **Kickstart Configurator**, чтобы помочь настроить эту директиву, или настроить сеть после завершения установки.

Поскольку пароль для пользователя root является частью процесса установки RHEL 7, файл конфигурации **Kickstart** может указывать этот пароль в зашифрованном формате. Хотя шифрование не требуется, оно может по крайней мере задержать хакера, который может взломать систему после завершения установки. Поскольку связанная криптографическая хеш-функция такая же, как и для файла **/etc/shadow**, вы можете скопировать нужный пароль из этого файла.

rootpw --iscrypted $6$5UrLfXTk$CsCW0nQytrUuvycuLT317/

Команда **timezone** связана с длинным списком часовых поясов. Они задокументированы в пакете **tzdata**. Для получения полного списка выполните команду **rpm -ql tzdata**. По умолчанию Red Hat устанавливает аппаратные часы на эквивалент среднего времени по Гринвичу с помощью ключа **--isUtc**. Этот параметр поддерживает автоматические изменения для перехода на летнее время. Следующий параметр можно найти в виде подкаталога и файла в каталоге **/usr/share/zoneinfo**:

timezone America/Los_Angeles --isUtc

Директива **user** может быть включена для создания пользователя во время процесса загрузки. Для этого требуется имя пользователя, зашифрованный пароль и, необязательно, список групп, к которым должен принадлежать пользователь, и информация GECOS для пользователя (обычно его полное имя). В следующем примере зашифрованный пароль опущен для краткости:

user --groups=wheel name=michael --password=... --iscrypted --gecos="MJ"

Что касается безопасности, необязательно можно добавить директиву **firewall**. В сочетании с **параметром --service = ssh** он указывает службы, которые разрешены через брандмауэр:

firewall --service=ssh

Директива **selinux** также является необязательной и может иметь значение **--enforcing,** **--permissive** или **--disabled**. По умолчанию используется **--enforcing**:

selinux --enforcing

Загрузчик по умолчанию - **GRUB 2**. Обычно его следует устанавливать в пространство между главной загрузочной записью (**MBR**) жесткого диска и первым разделом. Вы можете включить ключ **--boot-drive** для указания диска с загрузчиком и ключ **--append** для указания параметров для ядра:

bootloader --location=mbr --boot-drive=vda

Как следует из последующих комментариев, в первую очередь важно очистить некоторые существующие наборы разделов. Во-первых, **clearpart --all --initlabel --drives=vda** очищает все разделы виртуального жесткого диска **vda**. Если он не использовался ранее**, --initlabel** инициализирует этот диск:

clearpart --all --initlabel –-drives=vda

Изменения требуются в следующих директивах раздела (**part)**. Они должны указывать каталог, формат файловой системы **(--fstype**) и **--size в МБ**:

part /boot --fstype="xfs" --size=500
part swap --fstype="swap" --size=1000
part / --fstype="xfs" --size=10000
part /home --fstype="xfs" --size=1000

Помните, что ваша версия файла **anaconda-ks.cfg** может включать директиву **--onpart**, которая определяет файлы устройств раздела, такие как **/dev/vda1**. Это может привести к ошибке, если указанные разделы уже существуют. Поэтому, если вы видите какие-либо директивы **--onpart**, их проще всего удалить. В противном случае вам придется создать эти разделы до начала процесса установки, и это может быть сложно.

Хотя **для RAID-массивов и логических томов** могут использоваться другие параметры разделов, неявная задача экзаменов Red Hat - настроить такие тома после завершения установки. Если вы хотите опробовать другие параметры, такие как логические тома, создайте свой собственный файл **Kickstart**. Лучше всего, если вы настроите его с другой установки виртуальной машины. Просто имейте в виду, что файл **Kickstart** может настраивать физические тома (**PV**), группы томов (**VG**) и логические тома (**LV**) в том порядке (и порядок важен), как показано здесь:

part pv.01 --fstype="lvmpv" --ondisk=vda --size=11008
part /boot --fstype="xfs" --ondisk=vda --size=500
part swap --fstype="swap" --ondisk=vda --size=1000
volgroup rhel --pesize=4096 pv.01
logvol / --fstype="xfs" --size=10000 --name=root --vgname=rhel
logvol /home --fstype="xfs" --size=1000 -name=home --vgname=rhel

Для получения дополнительной информации о том, как настроены **LV, см. Главу 8**. Версия файла кикстарта по умолчанию может содержать директиву **repo**. Он будет указывать на источник сетевой установки **FTP из главы 1**, лабораторная работа 2 и должен быть удален или закомментирован из файла **Kickstart** следующим образом:

#repo --name="Red Hat Enterprise Linux" --baseurl=ftp://192.168.122.1/pub/inst --cost=100

Чтобы убедиться, что система фактически завершает процесс установки, это место для включения директивы, таких как **reboot, shutdown, halt,** или **poweroff**. Если вы повторно используете существующую виртуальную машину на основе **KVM**, может потребоваться отключить систему, чтобы сменить загрузочный носитель с **CD/DVD на жесткий диск**. Поэтому вы можете предпочесть использовать следующую директиву:

shutdown

Ниже приведен список групп пакетов, которые устанавливаются с помощью файла конфигурации **Kickstart**. Эти имена соответствуют именам, которые вы можете найти в файле ***-comps-Server.x86_64.xml file in the RHEL 7 DVD /repodata**, описанном в **главе 1**. Поскольку список длинный, приведенный ниже отрывок из групп пакетов (которые **начинаются с @)** и имен пакетов:

%packages
@base
@core
...
@print-client
@x11
%end

После установки групп пакетов вы можете указать команды после установки после следующей директивы. Например, вы можете настроить пользовательские файлы конфигурации. Однако директива **%post** и все, что следует, не требуется.

%post

Наконец, используйте утилиту **ksvalidator**, чтобы проверить синтаксис файла **Kickstart**. Пример показан здесь:

 # ksvalidator ks.cfg
The following problem occurred on line 32 of the kickstart file:

Unknown command: vogroup

УПРАЖНЕНИЕ 2-3

Создайте и используйте образец файла Kickstart

В этом упражнении вы будете использовать файл **anaconda-ks.cfg** для дублирования установки с одного компьютера на другой с одинаковым оборудованием. Это упражнение устанавливает все одинаковые пакеты с одинаковой конфигурацией разделов на втором компьютере. Кроме того, в этом упражнении даже настраивается **контекст SELinux** для файла **Kickstart**.

Поскольку цель состоит в том, чтобы установить те же пакеты, что и в текущей установке, не требуется вносить изменения в пакеты или группы пакетов из файла **anaconda-ks.cfg** по умолчанию в каталоге **/root**. Это предполагает доступ к источнику сетевой установки, например, созданному в **лабораторной работе 2 главы 1.**

Шаги в этом упражнении предполагают наличие достаточного пространства и ресурсов как минимум для двух разных виртуальных машин на основе KVM, как обсуждалось в главе 1:

1. Просмотрите файл **/root/anaconda-ks.cfg на сервере server1.example.com**. Скопируйте его **в ks.cfg**.
2. Если в файле есть директива **network**, измените ее, указав **IP-адрес 192.168.122.150** с именем **хоста tester1.example.com**. Если система с таким именем хоста и IP-адресом уже существует, используйте другое имя хоста и IP-адрес в той же сети. Это нормально, если такой директивы еще не существует; сеть может быть настроена после завершения установки, используя методы, описанные в главе 3.

3. Убедитесь, что директивы, связанные с дисками и разделами в **файле ks.cfg**, активны и не закомментированы. Обратите внимание на директиву **clearpart**; обычно он должен иметь значение **--all** для удаления всех разделов и **--initlabel** для инициализации вновь созданных дисков. Если к виртуальной машине подключено более одного жесткого диска, переключатель(ключ, опция) **--drives=vda** может сосредоточиться на первом виртуальном диске в виртуальной машине на основе **KVM**.
4. Удалите директиву **cdrom**, если она есть. Проверьте расположение сервера установки, связанного с директивой **url** или **nfs**. В этой лабораторной работе предполагается, что это **FTP-сервер**, доступный по **IP-адресу 192.168.122.1**, в подкаталоге **pub/inst/**. Если это другой **IP-адрес и каталог**, замените соответственно.

   url --url ftp://192.168.122.1/pub/inst

5. Убедитесь, что следующая директива включена непосредственно перед директивой **%packages** в конце файла:

   shutdown

6. Используйте утилиту **ksvalidator**, чтобы проверить синтаксис файла **Kickstart**. Если об ошибках не сообщается, перейдите к следующему шагу:

   ksvalidator ks.cfg

7. Скопируйте файл **ks.cfg** в базовый каталог сервера установки; если это **сервер vsFTP**, то это каталог **/var/ftp/pub.** Убедитесь, что файл доступен для чтения всем пользователям (по умолчанию он доступен только пользователю **root** с разрешениями 600). Например, вы можете использовать следующую команду:

   # chmod + r /var/ftp/pub/ks.cfg

8. Предполагая, что базовым каталогом является **/var/ftp/pub**, измените **контекст SELinux** этого файла с помощью следующей команды:

   # restorecon /var/ftp/pub/ks.cfg

9. Убедитесь, что все существующие брандмауэры не блокируют порт связи, связанный с сервером установки. Для получения подробной информации см. **Главу 4**. Самый простой способ сделать это - **открыть службу ftp** с помощью команды **firewall-cmd**:

   # firewall-cmd --permanent --add-service=ftp
   # firewall-cmd --reload

10. Создайте виртуальную машину на основе KVM на локальном хосте, чтобы на ней было достаточно места на жестком диске. Загрузите эту виртуальную машину, используя DVD RHEL 7.
11. В меню установки Red Hat выделите первую опцию и нажмите вкладку. Он отобразит директивы запуска в нижней части экрана. В конце этой строки добавьте следующую директиву:

    ks=ftp://192.168.122.1/pub/ks.cfg

Если файл **Kickstart** находится на другом сервере или на локальном носителе, замените его соответствующим образом.

Теперь вы должны увидеть, как при установке системы создаются те же базовые настройки, что и в первой системе. Если процесс установки останавливается до перезагрузки, то возникает проблема с файлом **Kickstart**, скорее всего, из-за недостатка информации.

Конфигуратор Kickstart

Даже пользователи, которые предпочитают работать в командной строке, могут учиться с помощью инструмента Red Hat GUI, известного, как **Kickstart Configurator**. Он включает большинство (но не все) основных параметров, связанных с настройкой файла конфигурации **Kickstart**. Вы можете установить его с помощью следующей команды:

 # yum install system-config-kickstart

Как инструмент с графическим интерфейсом, связанный с процессом установки, эта команда обычно включает ряд зависимостей.

!!!!! On the Job !!!
Те из вас, кто чувствителен к правильно написанному английскому языку, могут возразить против термина «Kickstart Configurator», но это имя, данное Red Hat упомянутому инструменту конфигурации GUI.
!!!!

Теперь, когда вы понимаете основы того, что входит в файл **Kickstart**, пришло время укрепить ваше понимание с помощью графического **Конфигуратора Kickstart**. Это может помочь вам узнать больше о том, как настроить файл Kickstart. После установки нужных пакетов его можно открыть из командной строки графического интерфейса с помощью команды **system-config-kickstart**. Чтобы запустить его с конфигурацией по умолчанию для локальной системы, приведите **файл anaconda-ks.cfg** следующим образом:

 # system-config-kickstart /root/anaconda-ks.cfg

Это должно открыть **Kickstart Configurator**, показанный на **рисунке 2-16**. (Конечно, сначала рекомендуется создать резервную копию файла **anaconda-ks.cfg**.)

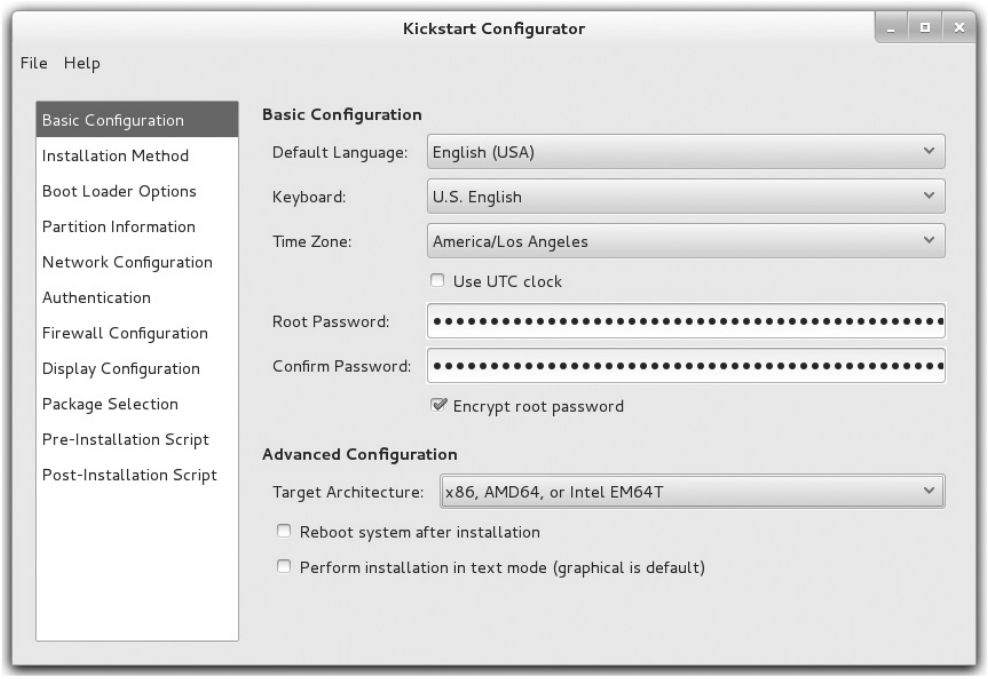
!!!! On the Job !!!!
Перед запуском Kickstart Configurator лучше убедиться, что имеется активное соединение с удаленным репозиторием RHEL 7 через RHN.
!!!!!

Экран, показанный на **рисунке 2-16**, иллюстрирует ряд основных этапов установки. Если вы уже установили RHEL, все эти шаги должны выглядеть знакомо.

На левой панели отображается ряд других параметров, каждый из которых связан с различными командами **Kickstart**. Чтобы узнать больше о **Kickstart**, поэкспериментируйте с некоторыми из этих настроек. Используйте Команду «Файл | Сохранить» (**File | Save**), чтобы сохранить эти настройки с именем файла по вашему выбору, который вы можете просмотреть в текстовом редакторе. Кроме того, вы можете выбрать «**Файл | Предварительный просмотр**» (**File | Preview**), чтобы увидеть влияние различных настроек на файл **Kickstart**.

В следующих разделах представлен краткий обзор каждой опции, показанной на левой панели. Детальное понимание **Kickstart Configurator** также может помочь вам понять процесс установки.

РИСУНОК 2-16 Конфигуратор Kickstart



Базовая конфигурация

На экране «Базовая конфигурация» вы можете назначить настройки для следующих компонентов:

- **Default Language** Задает язык по умолчанию для установки и операционной системы.
- **Keyboard** Устанавливает клавиатуру по умолчанию; обычно ассоциируется с языком.
- **Time Zone** Настройка местного часового пояса и указание, установлены ли аппаратные часы в формате UTC, что по сути совпадает со средним временем по Гринвичу.
- **Root Password** Определяет пароль для пользователя **root**; может быть зашифрован.
- **Target Architectu** Может помочь настроить файл кикстарта для разных систем.
- **Reboot System After Installation** Добавляет команду перезагрузки в конец файла **кикстарта**.
- **Perform System Installation in Text Mode**. Поддерживает автоматическую установку в текстовом режиме. После автоматизации режим установки не должен иметь значения.

Способ установки

Варианты метода установки просты. Вы **либо** устанавливаете Linux **впервые**, **либо обновляете** предыдущую установку. Способ установки и ваши записи основаны на расположении установочных файлов. Например, если вы выберете метод установки **NFS**, **Kickstart Configurator** запросит у вас имя или **IP-адрес сервера NFS и общий каталог с установочными файлами RHEL**.

Вы можете настроить файл **Kickstart** для установки RHEL с CD/DVD, с раздела локального жесткого диска или с одного из стандартных сетевых серверов: NFS, HTTP или FTP.

Параметры загрузчика

В следующем разделе перечислены параметры загрузчика. Загрузчик по умолчанию - **GRUB**, который поддерживает зашифрованные пароли для дополнительного уровня безопасности во время процесса загрузки.

Загрузчики Linux обычно устанавливаются на **MBR**. Если вы используете двойную загрузку **Linux и Microsoft Windows с GRUB**, вы можете настроить загрузчик **Windows** (или

альтернативный сторонний загрузчик), чтобы он указывал на **GRUB** в первом секторе раздела **Linux с каталогом /boot.**

Информация о разделах

Секция Информация о разделе определяет, как эта установка настраивает жесткие диски на соответствующих компьютерах. Хотя он поддерживает настройку стандартных разделов и разделов RAID, он еще не поддерживает настройку групп LVM. Параметр «Очистить основную загрузочную запись» позволяет стереть MBR со старого жесткого диска, на котором может быть проблема; он включает команду **zerombr** в файле **Kickstart**.

!!!! On the Job !!!!!
Не используйте опцию zerombr, если вы хотите сохранить в MBR альтернативный загрузчик, такой как Microsoft Windows Bootmgr.
!!!!

Вы можете удалить разделы в зависимости от того, были ли они созданы в файловой системе Linux. Если вы используете новый жесткий диск, важно также инициализировать метку диска. Нажмите кнопку **Добавить** (**Add**); это открывает диалоговое окно Параметры раздела.

Конфигурация сети

Раздел «**Конфигурация сети**» (**Network Configuration**)позволяет настроить IP-адресацию на сетевых картах на целевом компьютере. Вы можете настроить статическую IP-адресацию для конкретного компьютера или настроить использование DHCP-сервера. Просто нажмите **Добавить сетевое устройство** (**Network Configuration**) и откройте окно Информация о сетевом устройстве.

Аутентификация

В разделе «**Аутентификация**» (**Authentication**) вы можете настроить две формы безопасности для паролей пользователей: теневые пароли, которые шифруют пароли пользователей в файле **/etc/shadow**, и хэш шифрования для этих паролей. Если у вас установлен сканер отпечатков пальцев, вы можете установить соответствующий флажок, чтобы включить сканер отпечатков пальцев. Это позволяет выполнять двухфакторную аутентификацию, запрашивая у пользователей свои учетные данные при входе в систему и сканируя на сканере отпечатков пальцев.

Этот раздел также позволяет настроить информацию для аутентификации для различных протоколов:

- **NIS Network Information Service** используется для подключения к базе данных аутентификации при входе в сеть с компьютерами Unix и Linux.
- **LDAP** В этом контексте облегченный протокол доступа к каталогам - это служба каталогов, которая может использоваться в качестве альтернативной базы данных аутентификации при входе.
- **Kerberos 5** Система MIT для надежной криптографии используется для аутентификации пользователей в сети.
- **Hesiod Hesiod** - это сетевая база данных, которая может использоваться для хранения информации об учетной записи пользователя и пароле.
- **SMB Samba** подключается к сети в стиле **Microsoft Windows** для аутентификации при входе в систему.
- **Name Switch Cache**. Связан с **NIS** для поиска учетных записей и групп пользователей.

Конфигурация брандмауэра

Раздел «**Конфигурация брандмауэра**» позволяет вам настроить брандмауэр по умолчанию для данного компьютера. В большинстве систем вы хотите свести количество доверенных сервисов к минимуму. Однако в такой ситуации, как экзамены Red Hat, вас могут попросить настроить множество служб в одной системе, что потребует настройки множества доверенных служб на брандмауэре.

В этом разделе вы также можете настроить **основные параметры SELinux**. Варианты «Активный» и «Отключенный» (**Active and Disabled**) просты; опция **Warn** соответствует разрешающей реализации **SELinux**. Для получения дополнительной информации см. **Главу 4.**

Конфигурация дисплея

Раздел «**Конфигурация дисплея**» поддерживает установку базового графического интерфейса Linux. Фактическая установка зависит от пакетов и групп пакетов, выбранных в следующем разделе. Хотя существует много споров о превосходстве GUI по сравнению с текстовыми административными инструментами, текстовые инструменты более стабильны. По этой (и более) причине многие администраторы Linux даже не устанавливают графический интерфейс. Однако, если вы устанавливаете Linux на нескольких рабочих станциях, как это может быть сделано с рядом файлов **Kickstart**, вполне вероятно, что большинство пользователей не будут администраторами.

Кроме того, вы можете отключить или включить агент установки, также известный как процесс первой загрузки (Firstboot). Для полностью автоматической установки агент установки должен быть отключен.

Выбор пакета

В разделе «Выбор пакета» можно выбрать группы пакетов, которые устанавливаются с помощью этого файла **кикстарта**. Как отмечалось ранее, связанные экраны будут пустыми, если нет текущего соединения с удаленным репозиторием, таким как обновления из RHN. На момент написания вы столкнулись бы с той же проблемой, если бы использовали локальный источник установки. В этом случае вам нужно вручную отредактировать файл, созданный **Kickstart Configurator**, и добавить требуемый выбор пакета.

Сценарии установки

Вы можете добавить сценарии до и после установки в файл **Kickstart**. Сценарии после установки более распространены, и они могут помочь настроить другие части операционной системы Linux обычным способом. Например, если вы хотите установить каталог с информацией о вознаграждениях работникам, вы можете добавить сценарий после установки, который добавляет соответствующие команды **cp** для копирования файлов с сетевого сервера.

ЦЕЛЬ СЕРТИФИКАЦИИ 2.04

Администрирование с помощью Secure Shell и Secure Copy

Red Hat Enterprise Linux устанавливает пакеты **Secure Shell (SSH)** по умолчанию. Требование RHCSA относительно SSH простое; вам нужно знать, как использовать его для доступа к удаленным системам. Кроме того, вам также необходимо знать, как безопасно передавать файлы между системами. Поэтому в этом разделе вы узнаете, как использовать команды **ssh** и **scp** для доступа к удаленным системам и передачи файлов.

Как было предложено ранее, SSH уже установлен по умолчанию в стандартных установках RHEL 7. Хотя брандмауэры включены по умолчанию, стандартный брандмауэр RHEL 7 оставляет **TCP-порт 22** открытым **для доступа SSH**. Файлы связанные с конфигураций

SSH, хранятся в каталоге **/etc/ssh**. Конфигурация **сервера SSH** является частью требований RHCE. Связанные клиентские команды, такие как **ssh, scp и sftp**, рассматриваются в этом разделе.

Демон **Secure Shell** является безопасным, потому что он шифрует сообщения. Другими словами, пользователи, прослушивающие сеть, не могут прочитать данные, передаваемые между **SSH-клиентами и серверами**. И это важно в публичной сети, такой как Интернет. RHEL включает в себя **SSH версии 2**, которая поддерживает несколько методов обмена ключами и **несовместима со старой версией SSH 1**. Аутентификация на основе ключей для **SSH рассматривается в главе 4**. Если вы изучаете цели **RHCE по SSH**, прочитайте **главу 11.**

Настройте клиент SSH

Основной файл конфигурации **клиента SSH** - это **/etc/ssh/ssh_config**. Отдельные пользователи могут иметь собственные **конфигурации клиентов SSH в своих файлах ~/.ssh/config**. Четыре директивы включены по умолчанию. Во-первых, директива **Host *** применяет другие директивы ко всем соединениям:

Host *

Затем следует директива, поддерживающая аутентификацию с использованием интерфейса прикладного программирования **Generic Security Services (GSSAPI)** для **аутентификации клиент/сервер.** Это обеспечивает поддержку аутентификации **Kerberos**:

GSSAPIAuthentication yes

Следующая директива поддерживает удаленный доступ к приложениям с графическим интерфейсом. **X11** - это устаревшая ссылка на сервер **X Window System**, используемый в Linux.

ForwardX11Trusted yes

Следующие директивы позволяют клиенту устанавливать несколько переменных среды. Детали обычно тривиальны между двумя системами Red Hat Enterprise Linux.

SendEnv LANG LC_CTYPE LC_NUMERIC LC_TIME LC_COLLATE LC_MONETARY LC_MESSAGES
SendEnv LC_PAPER LC_NAME LC_ADDRESS LC_TELEPHONE LC_MEASUREMENT
SendEnv LC_IDENTIFICATION LC_ALL LC_LANGUAGE
SendEnv XMODIFIERS

Это создает основу для доступа к командной строке удаленных систем.

Доступ к командной строке

Этот раздел основан на стандартном доступе с помощью **команды ssh**. Для доступа к удаленной системе вам **потребуется имя пользователя и пароль** в этой удаленной системе. По умолчанию прямой **ssh-доступ к корневой учетной записи включен**. Например, следующая команда открывает оболочку с использованием этой учетной записи в указанной системе server1:

$ ssh root@server1.example.com

!!!! On the Job !!!!
Если при попытке доступа к удаленному хосту через SSH вы получаете сообщение об ошибке «Имя или служба неизвестна» (Name or service not known), это означает, что

система не может преобразовать имя хоста в IP-адрес. Мы настроим разрешение имен в главе 3. Тем временем, чтобы войти на server1.example.com через SSH, используйте его IP-адрес 192.168.122.50.
!!!!!

Следующая команда работает аналогичным образом:

$ ssh -l root server1.example.com

Без имени пользователя команда **ssh** предполагает, что вы входите удаленно, как имя пользователя в локальной системе. Например, если вы должны были запустить команду

$ ssh server1.example.com

от учетной записи пользователя michael команда ssh предполагает, что вы пытаетесь войти в систему server1.example.com как пользователь michael. При первом запуске команды между системами она выдает нечто похожее на следующее сообщение:

The authenticity of host 'server1.example.com (192.168.122.50)'
can't be established.
ECDSA key fingerprint is b6:80:5d:8c:1d:ab:18:ab:46:15:c5:c8:e3:ea:9f:1c.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'server1.example.com,192.168.122.50'
(ECDSA) to the list of known hosts.
michael@server1.example.com's password:

Подключившись через ssh, вы можете делать в удаленной системе все, что поддерживается вашими привилегиями пользователя на этом компьютере. Например, вы даже можете корректно завершить работу удаленной системы с помощью команды **poweroff**. После выполнения этой команды у вас обычно есть пара секунд для выхода из удаленной системы с помощью команды выхода.

Больше инструментов командной строки SSH

Если вы предпочитаете обращаться к удаленной системе с помощью FTP-подобного клиента, команда **sftp** для вас. Хотя **ключ -l** не имеет того же значения, что и команда **ssh**, он все же может использоваться для входа в учетную запись любого пользователя в удаленной системе. В то время как обычная **FTP-связь** происходит в виде открытого текста, связь с командой **sftp** может использоваться для передачи файлов в зашифрованном формате.

Кроме того, если вы просто хотите передать файлы по зашифрованному соединению, команда **scp** вам поможет. Например, мы создали несколько снимков экрана для этой книги на тестовых виртуальных машинах, настроенных в **главах 1 и 2**. Чтобы передать один из этих снимков экрана одной из наших систем, мы использовали команду, аналогичную следующей, которая скопировала Файл **F02-20.tif** из локального каталога в удаленную систему с указанным именем хоста **в /home/michael в каталог /RHbook/Chapter2**:

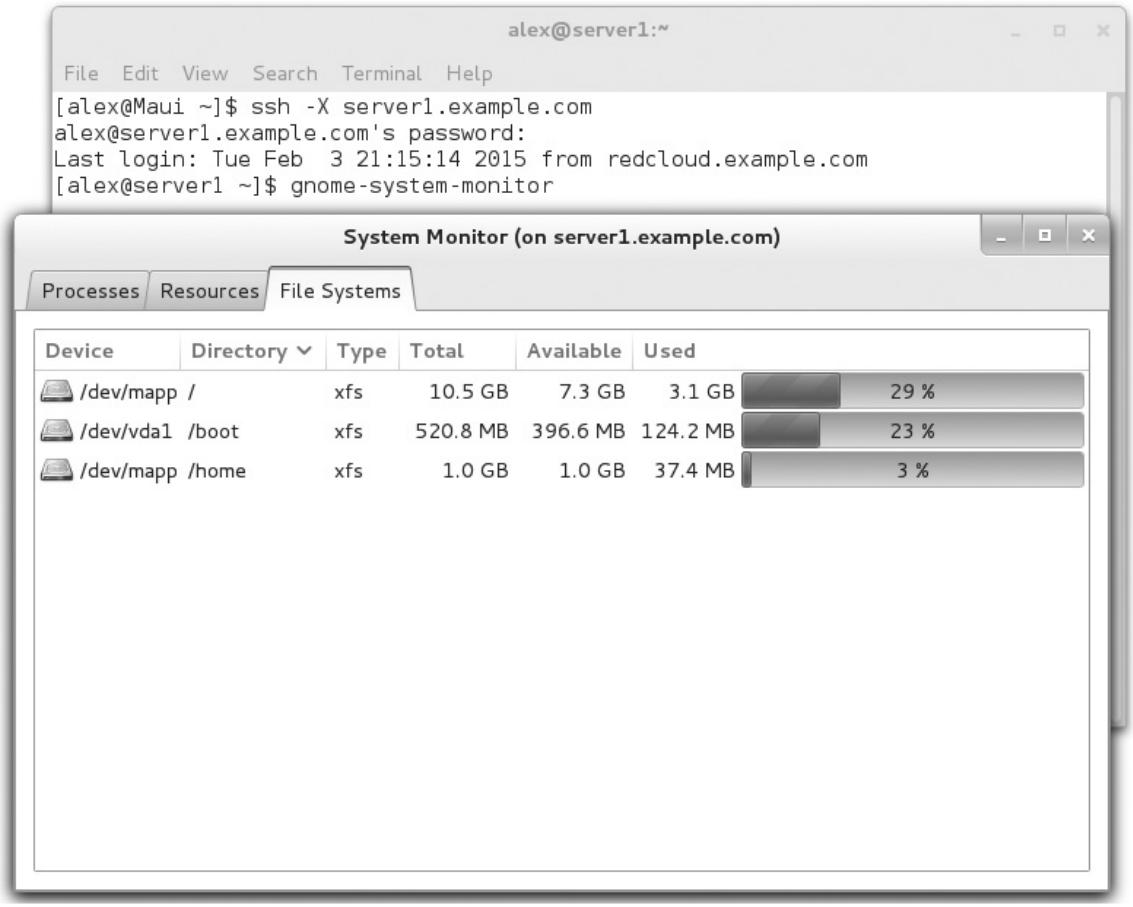
michael@server1:/home/michael/RHbook/Chapter2/

Если не настроена аутентификация на основе ключей (как описано в главе 4), команда запрашивает пароль пользователя **michael** в системе с именем **server1**. После подтверждения пароля команда **scp** передает файл **F02-20.tif** в зашифрованном формате в отмеченный каталог на удаленной системе с **именем server1**.

Графический безопасный доступ к оболочке

Команда **ssh** может использоваться для пересылки вывода приложения с графическим интерфейсом по сети. Как бы странно это ни звучало, это работает, если локальная система запускает **X-сервер**, когда вы вызываете удаленные клиентские приложения с графическим интерфейсом.

РИСУНОК 2-17 Удаленный доступ к GUI через SSH



По умолчанию файлы конфигурации как **сервера SSH**, так и клиента настроены для поддержки **связи X11 по сети**. Все, что вам нужно сделать, это подключиться к удаленной системе **с ключом -X** (или **-Y**, чтобы использовать доверенную переадресацию X11, которая обходит некоторые элементы управления расширения безопасности). Например, вы можете использовать последовательность команд, показанную на **рис. 2-17**, для мониторинга удаленной системы.

ЦЕЛЬ СЕРТИФИКАЦИИ 2.05

Рассмотрите возможность добавления этих инструментов командной строки. Возможно, вы захотите добавить несколько инструментов командной строки, чтобы помочь администрировать различные системы Linux. Эти инструменты будут использованы позже в этой книге, чтобы убедиться, что различные серверы действительно работают. Хотя лучше всего тестировать сервисы, такие, как **Postfix**, с реальными почтовыми клиентами (например, **Evolution** и **Thunderbird**), такие инструменты команд, как **telnet, nmap** и **mutt**, можно использовать для удаленной проверки этих сервисов из интерфейса командной строки. В целях экзамена вы можете использовать эти инструменты для тестирования, диагностики и решения системных проблем за время, необходимое для загрузки сложного инструмента, такого как

Evolution. Хотя команда **ssh** может помочь получить удаленный доступ к **инструментам GUI,** взаимодействие с такими инструментами может занять много времени.

Для административных целей интересующие инструменты включают следующее:

- Используйте **telnet** и **nmap** для проверки удаленного доступа к открытым портам.
- Используйте **Mutt** в качестве почтового клиента для проверки работоспособности почтового сервера.
- Используйте **elinks** в качестве веб-браузера, чтобы убедиться, что веб-сервисы доступны.
- Используйте **lftp** для доступа к FTP-серверам с завершением команды.

Проверка портов с помощью telnet

Команда **telnet** - удивительно мощный инструмент. Любой, кто знает о последствиях для безопасности клиентов с открытым текстом, может смущаться использовать **telnet**. Люди, которые используют **telnet** для входа на удаленные серверы, передают свои имена пользователей, пароли и другие команды в виде открытого текста. Любой, у кого есть анализатор протокола, такой как Wireshark, может легко прочитать эти данные.

Тем не менее, **telnet** может сделать больше. При локальном запуске он может проверить работу службы. Например, следующая команда проверяет работу **vsFTP** в локальной системе:

```
$ telnet localhost 21
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 (vsFTPd 3.0.2)
```

`«Escape-символ» - ctrl-] (одновременно нажимаются клавиша **ctrl и правая квадратная скобка**). Нажатие этой комбинации клавиш на указанном экране вызывает приглашение **telnet>**. Оттуда вы можете выйти с помощью команды **quit**.

```
^]
telnet> quit
```

В большинстве случаев вам даже не нужно выполнять символ **Escape** для выхода; просто введите команду **quit**.

Если **vsFTP** не запущен или настроен для связи через порт, отличный от 21, вы получите следующий ответ:

```
Trying 127.0.0.1...
telnet: connect to address 127.0.0.1: Connection refused
```

Если нет брандмауэра, вы получите тот же результат от удаленной системы. Однако, если брандмауэр блокирует связь **через порт 21**, вы можете получить сообщение, подобное следующему:

```
telnet: connect to address 192.168.122.50: No route to host
```

Некоторые службы, такие как почтовый сервер **Postfix**, по умолчанию настроены на прием соединений только из локальной системы. В этом случае с брандмауэром или без него вы получите сообщение «**Отказано в соединении**» (**connection refused**) при попытке подключения из удаленной системы.

Проверка портов с помощью nmap

Команда **nmap** - это мощный инструмент сканирования портов. Таким образом, веб-сайт разработчиков **nmap** гласит, что «при неправильном использовании **nmap** может (в редких случаях) получить иск, уволить, исключить, посадить в тюрьму или запрет вашим провайдером». Тем не менее, он включен в стандарт RHEL 7. хранилища. Таким образом, он поддерживается Red Hat для легального использования. Это быстрый способ получить представление об услугах, которые открыты локально и удаленно. Например, команда **nmap localhost**, показанная на **рис. 2-18**, обнаруживает и раскрывает те службы, которые работают в локальной системе.

Но, напротив, когда сканер портов запускается из удаленной системы, похоже, что открыт только один порт. Это показывает влияние брандмауэра на сервере.

Starting Nmap 6.40 (http://nmap.org) at 2015-02-02 09:52 PST
Nmap scan report for server1.example.com (192.168.122.50)
Host is up (0.027s latency).
Not shown: 999 filtered ports
PORT STATE SERVICE
22/tcp open ssh

Настройте почтовый клиент

Процесс настройки почтового клиента с графическим интерфейсом должен быть простым для любого кандидата на сертификацию Red Hat. Однако, это не обязательно может быть верно для клиентов командной строки, и они полезны для тестирования функциональности стандартных служб сервера электронной почты, таких как **Postfix** и **Sendmail**. Например, если сервер настроен для почтового протокола **Post Office Protocol (POP)** e-mail - даже электронная почта, которая доставляется с использованием почти повсеместной версии 3 (POP3) - можно проверить с помощью следующей команды:

 # mutt -f pop://username@host

Поскольку почтовые клиенты с графическим интерфейсом должны быть тривиальными для читателей, оставшаяся часть этого раздела посвящена использованию почтовых клиентов из командной строки.

Почта с командной строки

Один из способов протестировать локальную почтовую систему - встроенная утилита **mail** командной строки. Он предоставляет простой текстовый интерфейс. Система хранит почту каждого пользователя в файлах каталога **/var/mail**, связанных с каждым именем пользователя. Пользователи, которые читают сообщения с помощью почтовой утилиты, также могут отвечать, пересылать или удалять связанные сообщения.

Вы, безусловно, можете использовать любые другие программы чтения почты, такие как **mutt** или менеджеры электронной почты, связанные с различными веб-браузерами с графическим интерфейсом, для тестирования вашей системы. Другие программы чтения почты хранят сообщения в разных каталогах. Считыватели почты, такие как **mutt** и m**ail**, можно использовать для отправки сообщений, если для локальной системы активен **сервер SMTP**.

Существует два основных способа использования почты. Сначала вы можете ввести тему, а затем текст сообщения. Когда закончите, **нажмите Ctrl-D**. Сообщение отправлено, а почтовая утилита возвращается в командную строку. Вот пример:

$ mail michael
Subject: Test Message
Text of the message

EOT
$

Кроме того, вы можете перенаправить файл в виде текста электронного письма другому пользователю. Например, следующая команда отправляет копию файла **/etc/hosts** пользователю **root** на сервере **server1** с темой сообщения «**hosts file**»:

$ mail -s 'hosts file' < /etc/hosts root@server1.example.com

Чтение почтовых сообщений

По умолчанию почтовая система не открывается для пользователя, если в соответствующем файле нет действительной электронной почты. Как только почтовая система открыта, пользователь увидит список новых и уже прочитанных сообщений. Если вы открыли почтовую систему для учетной записи, вы можете ввести номер сообщения и нажать клавишу ввода. Если вы нажмете ввод без аргумента, почтовая утилита предполагает, что вы хотите прочитать следующее непрочитанное сообщение. Чтобы удалить почтовое сообщение, **используйте команду d** после прочтения сообщения или используйте **d#** для удаления сообщения с **номером #**.

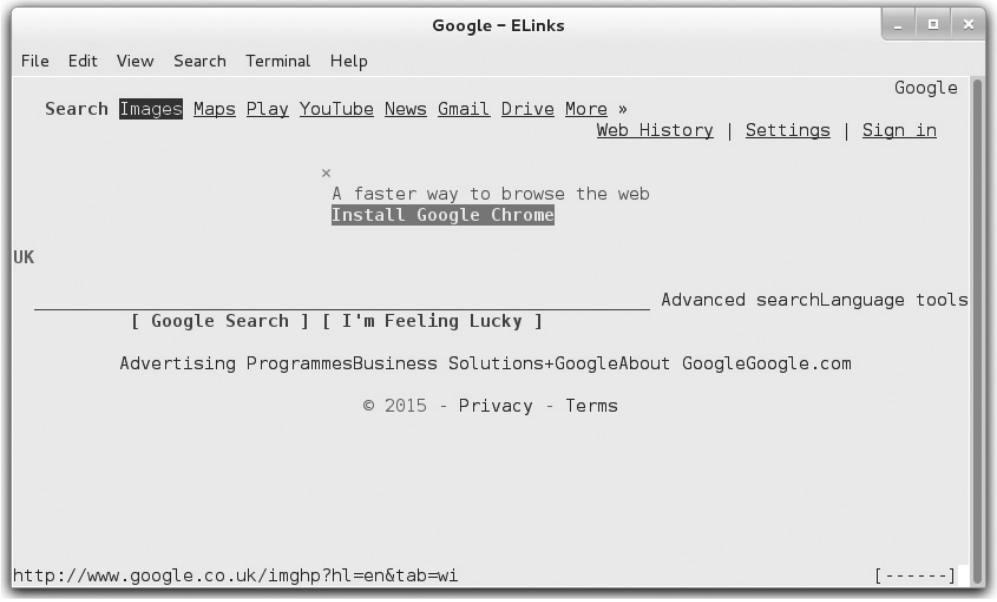
Кроме того, почтовые сообщения можно прочитать из указанного пользователем файла в локальном каталоге **/var/mail**. Файлы в этом каталоге названы по имени пользователя.

Использование текстовых и графических браузеров

Linux включает в себя множество графических браузеров. Доступ к обычным и защищенным веб-сайтам доступен через связанные с ними протоколы, протокол передачи гипертекста (**HTTP**) и его двоюродного брата, протокол передачи гипертекста, безопасный (**HTTPS**). Использование графических браузеров должно быть просто для любого серьезного пользователя Linux.

Возможно, вы не всегда имеете доступ к графическому интерфейсу, особенно при работе из удаленной системы. В любом случае текстовые браузеры работают быстрее. Стандартный текстовый браузер для **Red Hat - ELinks**. После установки пакета вы можете использовать его из командной строки, чтобы открыть веб-сайт по вашему выбору. Например, **рисунок 2-19** иллюстрирует результат команды **elinks http://www.google.com**.

РИСУНОК 2-19 Браузер ELinks



Чтобы выйти из **ELinks**, нажмите клавишу **esc** для доступа к строке меню, а затем нажмите **F | X** и примите приглашение выйти из браузера. В качестве альтернативы, **клавиша Q** может использоваться для быстрого выхода.

Если вы настраиваете веб-сервер, самый простой способ убедиться, что он работает, - это домашняя страница с простым текстом. HTML кодирование не требуется. Например, мы могли бы добавить следующий текст в home.html:

This is my home page

Затем мы можем запустить команду **elinks home.html**, чтобы просмотреть этот текст в браузере **ELinks**. Если вы настроили файловый **сервер Apache** в каталоге **/var/www/html/inst**, как описано в **главе 1**, вы также можете использовать **elinks** для просмотра файлов, скопированных на этот сервер, с помощью следующей команды:

$ elinks http://192.168.122.1/inst

Использование lftp для доступа к URL

Исходное клиентское программное обеспечение **FTP** представляло собой текстовое ориентированное клиентское приложение для командной строки с простым, но эффективным интерфейсом. Большинство веб-браузеров предлагают графический интерфейс и могут также использоваться в качестве **FTP-клиента**.

Любой **FTP-клиент** позволяет **просматривать дерево каталогов и файлов**. Использовать **ftp** в качестве клиента легко. Вы можете использовать команду **ftp** для подключения к серверу, такому как **ftp.redhat.com**, с помощью следующей команды:

 # ftp ftp.redhat.com

Однако этот клиент запрашивает имя пользователя и пароль. Вы можете ввести имя пользователя **anonymous** и свой адрес электронной почты в качестве пароля для доступа к **FTP-серверу Red Hat**. Но если вы случайно введете реальное имя пользователя и пароль, эти данные будут отправлены в виде открытого текста, доступного любому, кто случайно использует в сети нужные приложения сетевого анализатора. Как ни странно, клиент команды **ftp** не установлен на стандартных установках RHEL 7.

Это одна из причин, почему **lftp** лучше. Он автоматически пытается анонимный вход в систему, не запрашивая имя пользователя или пароль. Он также поддерживает завершение команд, что может особенно помочь вам получить доступ к файлам и каталогам с более длинными именами.

Конечно, существуют риски для большинства **FTP-клиентов**, поскольку они передают данные в виде открытого текста, но пока использование этой команды ограничено общедоступными серверами с анонимным доступом, риск минимален. В конце концов, если вы используете **lftp** для загрузки пакетов **Linux** с общедоступных серверов, это не значит, что вы подвергаете риску какую-либо личную информацию. Безусловно, существуют и другие риски безопасности для таких клиентов, но разработчики Red Hat постоянно работают над тем, чтобы поддерживать этого клиента в актуальном состоянии.

Если риски приемлемы, команда **lftp** может использоваться для входа на **FTP-сервер**, где разрешены имена пользователей и пароли. Пользователь Майкл может войти на такой сервер с помощью следующей команды:

$ lftp ftp.example.org -u michael

Клиент **lftp** может обрабатывать несколько различных команд, как показано на **рисунке 2-20**. Некоторые из этих команд **описаны в таблице 2-5**.

Почти все команды из приглашения **FTP** выполняются на удаленном хосте, подобно сеансу **telnet**. Из этой подсказки вы можете запускать обычные команды оболочки; просто запустите команду с восклицательным **знаком (!)**.

Это только часть команд, доступных через **lftp**. Если вы что-то не помните, команда **help *cmd*** выдает краткое описание указанной команды.

РИСУНОК 2-20 Команды в lftp

```
[root@Maui ~]# lftp ftp.redhat.com
lftp ftp.redhat.com:~> help
    !<shell-command>                    (commands)
    alias [<name> [<value>]]            attach [PID]
    bookmark [SUBCMD]                   cache [SUBCMD]
    cat [-b] <files>                    cd <rdir>
    chmod [OPTS] mode file...           close [-a]
    [re]cls [opts] [path/][pattern]     debug [<level>|off] [-o <file>]
    du [options] <dirs>                 exit [<code>|bg]
    get [OPTS] <rfile> [-o <lfile>]     glob [OPTS] <cmd> <args>
    help [<cmd>]                        history -w file|-r file|-c|-l [cnt]
    jobs [-v] [<job_no...>]             kill all|<job_no>
    lcd <ldir>                          lftp [OPTS] <site>
    ln [-s] <file1> <file2>             ls [<args>]
    mget [OPTS] <files>                 mirror [OPTS] [remote [local]]
    mkdir [-p] <dirs>                   module name [args]
    more <files>                        mput [OPTS] <files>
    mrm <files>                         mv <file1> <file2>
    [re]nlist [<args>]                  open [OPTS] <site>
    pget [OPTS] <rfile> [-o <lfile>]    put [OPTS] <lfile> [-o <rfile>]
    pwd [-p]                            queue [OPTS] [<cmd>]
    quote <cmd>                         repeat [OPTS] [delay] [command]
    rm [-r] [-f] <files>                rmdir [-f] <dirs>
    scache [<session_no>]               set [OPT] [<var> [<val>]]
    site <site-cmd>                     source <file>
    torrent [-0 <dir>] <file|URL>...    user <user|URL> [<pass>]
    wait [<jobno>]                      zcat <files>
    zmore <files>
lftp ftp.redhat.com:~>
```

ТАБЛИЦА 2-5 Стандартные команды клиента lftp

Команда	Описание
cd	Изменяет текущий рабочий каталог на удаленном хосте
ls	Перечисляет файлы на удаленном хосте
get	Получает один файл с удаленного хоста
mget	Извлекает много файлов с удаленного хоста с подстановочными знаками или полными именами файлов
put	Загружает один файл с вашего компьютера на удаленный хост
mput	Загружает группу файлов на удаленный хост
pwd	Перечисляет текущий рабочий каталог на удаленном хосте
quit	Завершает сеанс FTP
!ls	Перечисляет файлы на вашем главном компьютере в текущем каталоге
lcd	Изменяет локальный каталог хоста для загрузки / выгрузки
!pwd	Перечисляет текущий рабочий каталог на локальном хост-компьютере.

РЕЗЮМЕ СЕРТИФИКАЦИИ

Учитывая важность виртуализации в современной вычислительной среде, неудивительно, что Red Hat сделала KVM частью требований, связанных с RHCSA. Если предположить, действительное подключение к соответствующим репозиториям, установка пакетов, связанных с KVM, достаточно проста. Вам может понадобиться использовать команду, например **modprobe kvm**, чтобы убедиться, что модули загружены. Затем диспетчер виртуальных машин можно использовать для настройки виртуальных машин с помощью KVM в системе RHEL 7. Вы также можете использовать такие команды, как **virt-install, virt-clone** и **virsh** для установки, клонирования и управления этими виртуальными машинами.

Вы можете автоматизировать всю установку с помощью **Kickstart**. Каждая система RHEL имеет файл шаблона **кикстарта** в каталоге **/root**, который вы можете изменить и использовать для установки RHEL в других системах автоматически. Кроме того, вы можете использовать **GUI Kickstart Configurator,** что бы создать соответствующий файл **Kickstart**.

Со всеми этими системами удаленный доступ является обязательным. Команда **SSH** может помочь настроить удаленная зашифрованная связь между системами Linux. RHCSA требует, чтобы Вы знали, как использовать **клиент SSH** и безопасно передавать файлы между системами.

Команда ssh может использоваться для входа в удаленные системы; команда ssh -X может быть даже используется для доступа к удаленным приложениям с графическим интерфейсом. Команда **scp** может удаленно копировать файлы в зашифрованном соединение.

Когда вы просматриваете и устраняете неисправности сервисов RHEL, может быть полезно использовать некоторые инструменты командной строки. **Команда telnet** может подключиться к удаленному сервису на выбранных портах. **Команда nmap** может использоваться как сканер портов. Команда **mult** может проверить работоспособность почтового сервера. Команда **elinks** может используется в качестве браузера командной строки. Наконец, команда **lftp** является отличным **FTP-клиентом**, который поддерживает завершение команды.

ДВЕ МИНУТЫ ПОВТОРЕНИЯ

Настройте KVM для Red Hat

- Пакеты, необходимые для KVM, являются частью групп пакетов виртуализации.
- Виртуальные машины на основе KVM можно настроить с помощью диспетчера виртуальных машин (Virtual Machine Manager).
- Модули ядра, необходимые для KVM, включают **kvm** и **kvm_intel** или **kvm_amd**.

Настройте виртуальную машину на KVM

- Каталог по умолчанию для виртуальных машин на основе KVM - **/var/lib/libvirt/images.**.
- Файлы конфигурации виртуальной машины хранятся в различных подкаталогах **/etc/libvirt**.
- Консоли виртуальных машин доступны с помощью диспетчера виртуальных машин, который вы можете запустить в графическом интерфейсе с помощью команды **virt-manager**.
- Виртуальные машины можно устанавливать, клонировать и настраивать с помощью **virt-install**, **virt-clone** и команды **virsh**.
- Команда **virsh list --all** выводит список всех сконфигурированных виртуальных машин.
- Команда **virsh autostart *vmname*** настраивает домен виртуальной машины с именем ***vmname***
- запускаться автоматически при загрузке хост-системы.
- Команда **virsh start *vmname*** запускает процесс загрузки для домена VM по имени ***vmname***.
- Команда **virsh destroy *vmname*** фактически отключает питание домена виртуальной машины по имени ***vmname***.

Опции автоматической установки

- Установка системы быстрой установки **Kickstart** описана в текстовом файле **/root/anaconda-ks.cfg**.
- Файл **Kickstart** можно изменить напрямую или с помощью инструмента **Kickstart Configurator**.
- Файлы **Kickstart** можно вызывать с локальных медиа или сетевых серверов.

Администрирование с помощью Secure Shell и Secure Copy

- **SSH** установлен по умолчанию на RHEL 7. Он даже доступен через брандмауэры по умолчанию.
- **Команда ssh** может использоваться для безопасного доступа к удаленным системам. Это может даже включить доступ к удаленным утилитам GUI.
- Связанные команды включают **sftp** и **scp**.

Рассмотрите возможность добавления этих инструментов командной строки.

Администраторы иногда могут иметь только командную строку для проверки доступа к серверам.
Команды **telnet** и **nmap** можно использовать для проверки удаленного доступа к открытым портам.
Почтовый клиент mutt можно использовать для проверки работоспособности почтового сервера.
Веб-браузер консоли **elinks** может проверить работу веб-сервера.
Клиент lftp может использоваться для проверки доступа к **FTP-серверам** с преимуществом Завершение команды.

САМОПРОВЕРКА

Следующие вопросы помогут оценить ваше понимание материалов, представленных в этой главе.

Поскольку на экзаменах Red Hat не появляется вопросов с несколькими вариантами ответов, вопросы с несколькими вариантами ответов не появляются в этой книге. Эти вопросы исключительно проверяют ваше понимание главы. Получать результаты, а не запоминание пустяков, это то, что рассчитывает на экзамены Red Hat. Там может быть более одного ответа на многие из этих вопросов.

Настройте KVM для Red Hat

1. Назовите один модуль ядра, связанный с **KVM**.
   _____

2. Как называется инструмент, который может настраивать виртуальные машины на основе **KVM** в графическом интерфейсе?
   _____

Настройте виртуальную машину на KVM

3. Какая команда запускает диспетчер виртуальных машин в графическом интерфейсе?
   _____

4. В каком каталоге виртуальные диски по умолчанию хранятся в диспетчере виртуальных машин?

   _____

5. Какую команду можно использовать для создания новой виртуальной машины?

   _____

Опции автоматической установки

6. Какая команда запускает инструмент настройки **Kickstart** на основе графического интерфейса?

   _____

7. Как называется файл в каталоге **/root**, в котором указано, как был установлен **RHEL**?

   _____

8. Какая директива в файле конфигурации **Kickstart** связана с сетью?

   _____

9. Если установочный **FTP-сервер** находится по адресу **ftp://server1.example.com/pub/inst**, какая директива файл конфигурации **Kickstart** указывает на этот сервер?

_____

10. Какая директива в файле конфигурации **Kickstart** отключит систему после установки завершено?

    _____

Администрирование с помощью Secure Shell и Secure Copy

11. Какой командный ключ **ssh** обеспечивает доступ к удаленным **утилитам GUI**?

    _____

12. Какая команда инициирует безопасную копию файла **/etc/hosts** из системы **server1.example.com** в каталог **/tmp** на локальном хосте?

    _____

Рассмотрите возможность добавления этих инструментов командной строки.

13. Какую команду вы бы использовали, чтобы увидеть, работает ли сервер на порту 25 в системе с **IP-адрес 192.168.122.1**?

    _____

14. Какую команду можно использовать для проверки активных и доступных (от вашего клиента) сервисов на удаленном компьютере система с **IP-адресом 192.168.122.1?**

    _____

Лабораторная работа

Во время экзаменов Red Hat задания будут представлены в электронном виде. Таким образом, эта книга также представляет большинство лабораторий в электронном виде. Для получения дополнительной информации см. Раздел «Лабораторные вопросы» в конце главы 2. **Лабораторная работа 1 представлена в Учебном руководстве**, стр. 109.

Лабораторная работа 1

В этой лабораторной работе вы установите RHEL для создания базового сервера на виртуальной машине на основе KVM. Вам понадобится достаточно место для одного жесткого диска не менее 16 ГБ (с достаточным пространством для 11 ГБ данных плюс раздел подкачки, при условии, что для ВМ требуется не менее 512 МБ свободной оперативной памяти). Вам также понадобится место для дополнительных двух виртуальных жестких дисков по 1 ГБ каждый (всего 18 ГБ).

Шаги в этой лабораторной работе предполагают установку на виртуальной машине **на основе KVM**. Чтобы начать процесс, откройте графический интерфейс и запустите команду **virt-manager**. Если это не произойдет автоматически, щелкните правой кнопкой мыши на **localhost (QEMU)** и нажмите «**Подключиться**» (**Connect**) во всплывающем меню. Введите **root** административный пароль, если будет предложено сделать это. После подключения вы можете затем щелкнуть правой кнопкой мыши ту же опцию, а затем нажмите «**Новый**» (**New**). Это запустит мастер, который поможет настроить виртуальную машину.

Если вы настраиваете фактические виртуальные машины, которые будут использоваться в следующих главах, это будет Система **server1.example.com** обсуждается в главе 1.

В идеале на главном компьютере должно быть достаточно места как минимум для четырех разных виртуальных систем данного размера. Это включает три системы, указанные в главе 1, плюс одну запасную. Другими словами, логического тома или раздела с 75 ГБ свободного места было бы (едва) достаточно.

Шаги, описанные в этой лабораторной работе, являются общими. К этому времени у вас должен быть опыт работы с установка RHEL 7. В любом случае, точные шаги зависят от типа установки и загрузочного носителя:

1. Начните с сетевого загрузочного компакт-диска RHEL 7 или установочного DVD.
2. На основе шагов, описанных в главе 1, запустите процесс установки для RHEL 7.
3. На экране «**Обзор установки**» (**Summary screen**) выберите «**Источник установки**» (**Installation Source**) и укажите Установочный сервер на **основе FTP**, созданный в главе 1. Если вы следовали инструкциям в этой главе, сервер будет на **ftp://192.168.122.1/pub/inst.**
4. На экране «**Обзор установки**» (**Summary screen**) нажмите «**Место установки**» (**Installation Destination**) и выберите «**Выборочная**» (**custom**) разделы.
5. Создайте первый раздел объемом около 500 МБ на диске, отформатированный в файловой **системе xfs**, и назначьте каталог **/boot**.
6. Создайте следующий раздел с 1 ГБ дискового пространства (или более, если пространство доступно), зарезервируйте место для раздела **swop (подкачки).**
7. Создайте третий раздел с объемом дискового пространства около 10 ГБ, отформатированный в файловой системе **xfs**, и назначьте его в корневой каталог верхнего уровня, (/).
8. Создайте другой раздел с объемом дискового пространства около 1 ГБ и назначьте его в каталог **/home**.
9. На экране **«Обзор установки» (Summary screen)** настройте локальную систему в сети, настроенной на KVM гипервизор. По умолчанию используется сеть 192.168.122.0/24; для системы **server1.example.com**, это будет **IP-адрес 192.168.122.50 и шлюз 92.168.122.1.** Настройте имя хоста **server1.example.com.**
10. На экране **«Обзор установки» (Summary screen)** нажмите **«Выбор программ» (Software Selection)**, а затем выберите **Сервер с GUI» (Server with GUI).** Установка пакетов виртуализации внутри виртуальной машины не требуется.
11. Продолжайте процесс установки, используя ваши лучшие решения.
12. Перезагрузитесь при появлении запроса и войдите в систему как пользователь **root**. Запустите команду **poweroff**, когда вы будете готовы закончить эту лабораторию.

Лабораторная работа 2

В этой лабораторной работе вы будете клонировать систему, созданную **в лабораторной работе 1**

1. Используйте методы, описанные в этой главе, чтобы клонировать эту систему. Процесс можно завершить либо в командной строке с помощью команды **virt-clone**, либо с помощью диспетчера виртуальных машин.

Кроме того, при перезагрузке системы вы хотите настроить эту систему как систему **clone1.example.org** по IP-адресу **192.168.100.50**. Вы можете заменить другой IP-адрес, если он находится в сети, отличной от системы **server1.example.com**.

Когда система клонируется, она переносит все из предыдущей системы. Таким образом, при первой загрузке клонированной системы лучше всего загрузить ее в режиме «**восстановления**» (**rescue target**), которая не запускает сеть. Для получения дополнительной информации о **targets systemd** и конфигурации сети см. **Главы 3 и 5**.

Лабораторная работа 3

Используйте команду **virt-install** для создания новой системы. Используйте модель, описанную в главе 1, для системы **tester1.example.com** с **IP-адресом 192.168.122.150**.

Лабораторная работа 4

В этой лабораторной работе вы измените файл **Kickstart**, созданный на виртуальной машине **server1.example.com** в **лабораторной работе 1**. Это файл **anaconda-ks.cfg** в домашнем каталоге пользователя **root**. Этот файл будет использоваться для автоматизации установки системы на виртуальной машине. Если вы еще не создали его, система будет с следующими значениями **tester1.example.com с IP-адресом 192.168.122.150**. Если вы настроили систему **server1.example.com** в другой сети, убедитесь, что система **tester1.example.com** находится в той же сети. Помните уроки главы при изменении этого файла. Вот несколько советов:

1. Измените директиву **network**, чтобы установить для новой системы соответствующий **IP-адрес и имя хоста**.
2. Убедитесь, что директивы раздела активны (без комментариев). Измените их по мере необходимости, чтобы убедиться, что размер указан.
3. Настройте систему на выключение (**shutdown**) после завершения установки.
4. Не бойтесь проб и ошибок. Если установка останавливается во время процесса, проверьте сообщения в разных виртуальных консолях (см. Пункт 11).
5. Скопируйте файл **Kickstart** в каталог установки, **созданный в главе 1**. Используйте методы, описанные в этой главе, чтобы убедиться, что файл **Kickstart** соответствует **контексту SELinux** других файлов в этом каталоге, и убедитесь, что он доступен для чтения всем пользователям.
6. Создайте новую виртуальную машину на основе KVM, используя методы, описанные в этой главе.
7. Загрузите систему с сетевого загрузочного компакт-диска RHEL 7.
8. На экране загрузки выделите опцию **Install Red Hat Enterprise Linux 7.0 и нажмите TAB**.
9. Добавьте директиву **ks** вместе **с URL-адресом источника сетевой установки**, созданного в **главе 1**.
10. Нажмите клавишу **ВВОД (ENTER)**. Теперь установка должна завершиться без вмешательства.
11. Если установка останавливается во время процесса, запишите, где она остановилась. Экраны, доступные во время процесса установки, могут помочь. Чтобы получить доступ к этим экранам, нажмите **CTRL-ALT-F3, CTRL-ALT-F4 и CTRL-ALT-F5**. Чтобы

вернуться к основному экрану установки, нажмите **ALT-F6 (ALT-F1, если вы используете текстовую программу установки**).

12. Сопоставьте точку остановки с соответствующей записью в файле конфигурации **Kickstart**.
13. **Измените** файл конфигурации **Kickstart** и перезапустите установку на основе исправленного файла **Kickstart**.

Лабораторная работа 5

В этой лабораторной работе вы измените файл **Kickstart**, созданный на виртуальной машине **server1.example.com** в **лабораторных работах 1** и **4**, используйте команду **virt-install**. Если вы его еще не создали, система будет использовать о**utsider1.example.org с IP-адресом 192.168.100.100**. (Если эта сеть IP-адресов используется другим компонентом, например кабельным модемом, допустим другой IP-адрес, например 192.168.101.100.)

Лабораторная работа 6

В этой лабораторной работе вы **протестируете клиент Secure Shell** на всех виртуальных машинах, которые вы создали. Надеюсь, вы запомните пароль администратора, настроенный в процессе установки.

1. Чтобы увидеть, как работает система, сначала выполните следующую команду в локальной системе (при желании замените 127.0.0.1 на localhost):

   # ssh localhost

2. Продолжите вход в систему на локальной системе. Выйдите из системы с помощью команды **exit**.
3. Просмотрите текущие известные хосты для системы с помощью команды **cat ~/.ssh/known_hosts**. Файл может показаться непонятным, но в конце файла вы увидите строку, такую как следующая, которая ссылается на **открытый ключ RSA** из системы **localhost**:

   localhost ecdsa-sha2-nistp256 AAAAE2VjZHNhL ...

4. Повторите процесс с удаленной системой, такой как **server1.example.com**, или эквивалентным IP-адресом, таким как 192.168.122.50. После возвращения в локальную систему, что вы видите добавленным в файл **known_hosts**? (Прямые соединения с системой **server1.example.com** могут не работать, пока вы не настроите файл **/etc/hosts**, как описано в главе 3.)

5. Для этого шага **требуется доступ к GUI** в локальной системе и приложениям GUI в удаленной системе. Если вы следовали **инструкциям**, изложенным в **главах 1 и 2**, у вас будет ряд систем, отвечающих этим требованиям. Имея это в виду, войдите в указанную удаленную систему с локального терминала на основе графического интерфейса пользователя с помощью следующей команды (при необходимости подставьте соответствующее имя хоста или IP-адрес):

   # ssh -X root@192.168.122.50

6. Теперь попробуйте открыть приложение с графическим интерфейсом в удаленной системе, возможно, даже **веб-браузер Firefox** (если он установлен) с помощью команды **firefox**. Чтобы подтвердить успех, посмотрите на строку заголовка появившегося окна.

Должно отображаться сообщение с указанием местоположения удаленного приложения, например:

Welcome to Firefox - Mozilla Firefox (on server1.example.com)

7. Выйдите из удаленного приложения, а затем выйдите из удаленной системы.

Лабораторная работа 7

В этой лабораторной работе вы будете выполнять три задачи, связанные с одной из виртуальных машин, созданных в предыдущих лабораторных работах:

- Запустите виртуальную машину из командной строки.
- Остановить виртуальную машину из командной строки.
- Настройте эту виртуальную машину из командной строки.

Вы можете распознать эти задачи по целям RHCSA. Процесс должен быть довольно простым. Сначала посмотрите настроенные в данный момент виртуальные машины с помощью следующей команды:

 # virsh list --all

Из виртуальных машин, отображаемых в выходных данных, выберите ту, которая в данный момент не работает. Если система **server1.example.com** не запущена, запустите ее. Убедитесь, что он запущен. Используйте команду **ssh** для доступа к удаленной консоли в этой системе виртуальных машин.

Теперь из командной строки физической системы хоста остановите виртуальную консоль. Какая команда на самом деле выполняет задачу? Подтвердите результат командой **virsh list --all.**

Если вы хотите настроить автоматический запуск системы **server1.example.com** во время процесса загрузки, выполните следующую команду:

 # virsh autostart server1.example.com

Чтобы подтвердить изменение, просмотрите содержимое каталога **/etc/libvirt/qemu/autostart**. Затем войдите в систему виртуальной машины и выполните команду **ip addr show**, чтобы подтвердить IP-адрес сетевой карты этой виртуальной машины. Если вы настроили эту конкретную систему **server1.example.com** в соответствии с инструкциями, описанными в **главе 1**, этот **IP-адрес должен быть 192.168.122.50**.

Теперь отключите все работающие в данный момент виртуальные машины и перезагрузите систему физического хоста. Когда система загрузится снова, войдите в локальную хост-систему. Не запускайте диспетчер виртуальных машин. **Запустите команды ssh**, описанные в **лабораторной работе 6**. Если это работает, тогда вы сможете подключиться к виртуальной машине, как если бы это была удаленная система.

Выйдите из удаленной системы и выполните команду **virsh list --all**. Вы должны увидеть вывод, похожий на следующий:

Id	**Name**	**State**
2	**server1.example.com**	**running**

Теперь выключите удаленную систему. Вы можете снова войти в удаленную систему и запустить команду **poweroff** прямо оттуда. Как отменить процесс, чтобы эта система не запускалась при следующей перезагрузке системы физического хоста?

Лабораторная работа 8

В этой лабораторной работе вы будете использовать команды, описанные в конце главы 2, для проверки соединений с доступными службами. Если вы создали серверы сетевой установки, описанные в главе 1, в этих системах будут активны **как минимум FTP и HTTP-серверы**. Порты по умолчанию для этих **служб - 21 и 80** соответственно. Попробуйте команду **telnet localhost 21** в локальной системе, где **активна служба vsFTP**. Посмотрите на следующий вывод:

Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 (vsFTPd 3.0.2)

Теперь выйдите из сеанса **telnet**. Подтвердите IP-адрес локальной системы с помощью команды **ip addr show dev virbr0**. Это должен быть адрес, такой как 192.168.122.1. Войдите на удаленную виртуальную машину, такую как **tester1.example.org**, с помощью команды, например **ssh root@192.168.122.150**. Теперь повторите ту же команду с этой удаленной системы; например, для системы **server1.example.com по IP-адресу 192.168.122.50** выполните следующую команду:

 # telnet 192.168.122.50 21

Вы получаете сообщение «**Отказано в соединении**» (**connection refused**) или «**нет маршрута к хосту**» (**no route to host**)? Что означает каждое из этих сообщений? Это приемлемо, если вы не уверены в том, как решить эту проблему сейчас, поскольку брандмауэры не рассматриваются до главы 4.

Теперь попробуйте **nmap** в локальной системе с помощью следующей команды:

 # nmap localhost

Из системы **tester1.example.com** просмотрите, что другие системы в локальной сети могут видеть из следующей команды. Обратите внимание на различия. Это даст вам подсказки о том, какие сервисы заблокированы брандмауэрами. Эти брандмауэры могут выходить за пределы того, что настроено с помощью команды **firewall-cmd, описанной в главе 4**.

 # nmap 192.168.122.50

ОТВЕТЫ НА САМОПРОВЕРКУ

Настройте KVM для Red Hat

1. Три модуля ядра связаны **с KVM: kvm, kvm_intel** и **kvm_amd**.
2. Инструментом, который может настраивать виртуальные машины на основе **KVM** в графическом интерфейсе, является **диспетчер виртуальных машин (Virtual Machine Manager**).

Настройте виртуальную машину на KVM

3. Команда, которая запускает диспетчер виртуальных машин в графическом интерфейсе, является **virt-manager**.
4. Каталог с виртуальными дисками по умолчанию для диспетчера виртуальных машин: **/var/lib/libvirt/images**.

5. Командой, которую можно использовать для создания новой виртуальной машины, является **virt-install**.

Опции автоматической установки

6. Командой, запускающей инструмент настройки **Kickstart** на основе графического интерфейса пользователя, является **system-config-kickstart**.
7. Имя файла **Kickstart** в каталоге **/root**, в котором указано, как был установлен RHEL это **anaconda-ks.cfg**.
8. Директива в файле конфигурации **Kickstart**, относящаяся к сети, является **network**.
9. Директива, которая указывает на данный **сервер установки FTP**: **url --url ftp://server1.example.com/pub/inst**.
10. Директива в файле конфигурации **Kickstart**, которая выключит систему после установки, **shutdown**.

Администрирование с помощью Secure Shell и Secure Copy

11. Командный **ключ ssh**, который разрешает доступ к утилитам удаленного графического интерфейса, **-X**. Переключатель **-Y** также приемлемый ответ.
12. Требуемая команда для безопасного копирования **/etc/hosts** с **server1** в **/tmp** на локальном хосте:

    scp server1.example.com:/etc/hosts /tmp/.

Рассмотрите возможность добавления этих инструментов командной строки

13. Команда, которую вы будете использовать, чтобы увидеть, работает ли сервер на порту 25 в системе с IP-адрес 192.168.122.1 является **telnet 192.168.122.1 25**.
14. Команда, которая может использоваться для проверки активных и доступных служб в удаленной системе с IP-адрес 192.168.122.1 - это **nmap 192.168.122.1**.

ОТВЕТЫ ЛАБОРОТОРНОЙ РАБОТЫ

Лабораторная работа 1

Хотя в этой лабораторной работе нет ничего действительно сложного, она должна повысить вашу уверенность в использовании виртуальных машин на основе KVM. По завершении вы сможете войти на виртуальную машину, как пользователь **root** и выполнить следующие проверки в системе:

1. Проверьте смонтированные файловые системы, а также доступное пространство. Следующие команды должны выполняться подтвердите те файловые системы, которые монтируются, а также свободное место, доступное на соответствующих томах:

   # mount
   # df -m

2. Предположим, у вас хорошее подключение к Интернету и подписка на Red Hat Portal, убедитесь, что система в актуальном состоянии. Если вы используете переделанный дистрибутив, доступ к их публичному доступу репозитории приемлемы. В любом случае, запустите следующую команду, чтобы убедиться, что локальная система соответствует последним обновлениям системы:

   # yum update

Эта лабораторная работа подтверждает вашу способность «устанавливать системы Red Hat Enterprise Linux в качестве виртуальных гостей» (**install Red Hat Enterprise Linux systems as virtual guests**).

Лабораторная работа 2

Помните, что эту и все будущие лабораторные работы в этой книге можно найти на DVD, который идет с этой книгой. Лабораторные работы со 2 по 8 можно найти в подкаталоге **Chapter2/** этого DVD.

Одна из проблем с клонированием системы заключается в том, как оно включает аппаратный MAC-адрес любой сетевой карточки. Такие конфликты могут привести к проблемам в сети. Так что не только вам придется сменить IP адрес, но вам также может потребоваться, чтобы уникальный виртуальный адрес был назначен для данного виртуального Сетевого адаптера. Из-за таких проблем KVM обычно устанавливает другой аппаратный MAC-адрес для клонированная система. Например, если исходная система имела сетевую **карту eth0** с одним аппаратным адресом, клонированная система будет иметь сетевую карту с другим аппаратным адресом.

Если это кажется слишком большой проблемой, не стесняйтесь удалять клонированную систему. В конце концов, нет ссылки клонирование ВМ в требованиях RHCSA. Однако может быть полезно иметь другую систему резервного копирования.

И это прекрасная возможность попрактиковаться в навыках, полученных в Лаборатории 4 с установками Kickstart.

Лабораторная работа 3

Цель этой лабораторной работы - показать вам метод настройки виртуальной машины на основе KVM через командную строку.

Если вы еще не настроили четыре разные виртуальные машины, предложенные в главе 1 (три виртуальные машины и резервная копия), сейчас есть отличная возможность сделать это. Один из способов выполнить работу - воспользоваться помощью команды **virt-install**. Укажите команде следующую информацию:

- Выделенная оперативная память (**--ram**) (Allocated RAM (--ram)) в мегабайтах, которая должна быть не менее 512.
- Путь к файлу виртуального диска (**--disk**), который может совпадать с виртуальным диском, созданным в Лабораторной работе 2 и ее размер в гигабайтах, если этот файл еще не существует.
- **URL (--location)** для сервера установки FTP, созданного в главе 1, лабораторной работе 2. В качестве альтернативы вы можете использовать сервер установки HTTP, также обсуждаемый в главе 1.
- Тип ОС (**--os-type=*linux***) и вариант (**--os-option=*rhel7***).

Теперь вы можете завершить эту установку в обычном режиме или запустить вариант этой установки в лабораторной среде 5.

Лабораторная работа 4

Если у вас нет опыта настройки **Kickstart**, могут потребоваться ряд проб и ошибок. Но лучше столкнуться с проблемами сейчас, а не во время экзамена Red Hat или на работе. Если вы можете настроить файл **Kickstart**, который можно использовать для установки системы без вмешательства, вы готовы решить эту проблему на экзамен RHCSA. во-первых

Одна общая проблема связана с только что созданными виртуальными дисками. Во-первых они должны быть инициализированы, именно с этой целью ключ **--initlabel** переключается на директиву **clearpart**.

Лабораторная работа 5

Если вы впервые запускаете установку **Kickstart**, лучше сделать это снова. Если вы попрактикуетесь сейчас, это означает, что вы сможете быстрее установить **Kickstart** во время экзамена. И это только начало. Представьте себе уверенность в том, что вашему боссу понадобится пара десятков виртуальных машин с одним и тем же программным обеспечением и объемами. Предполагая, что единственными различиями являются имя хоста и настройки сети, вы сможете выполнить эти задачи довольно быстро.

Если вы можете настроить установку **Kickstart** из командной строки с помощью команды **virt-install**, это гораздо проще сделать на удаленном виртуальном хосте. Вы сможете настроить новые системы с удаленного местоположения, таким образом увеличивая вашу ценность, как специалиста.

Если вы еще не настроили четыре виртуальные машины, предложенные в главе 1 (три в качестве тестовых систем, одна в качестве резервной), у вас есть возможность сделать это сейчас.

Чтобы использовать файл Kickstart с **virt-install**, вам необходимо использовать обычные командные ключи. Так как вам не разрешить принести эту книгу на экзамен, попробуйте выполнить эту лабораторную работу, не обращаясь к основной части этой главы. Вы сможете обратиться к странице справочника для команды **virt-install** для выяснения важных ключей. Обязательно поместите директиву **ks=** вместе с **URL-адресом файла Kickstart в кавычки**. Успех лабораторной работы - установка новой системы.

Лабораторная работа 6

Эта лабораторная работа предназначена для того, чтобы улучшить ваше понимание использования **команды ssh** в качестве клиента выполняемое шифрование должно быть прозрачным и не повлияет на команды, используемые **через SSH** подключение для администрирования удаленных систем.

Лабораторная работа 7

Эта лабораторная работа несколько критична в отношении нескольких различных целей RHCSA. Как только вы понимаете процесс, реальные задачи обманчиво просты. После завершения этой лаборатории вы должны быть уверены в ваших силах сделать следующее:

- Запуск и остановка виртуальных машин.
- Настройка систем для запуска виртуальных машин при загрузке.

Лабораторная работа также предлагает один метод для удаленного доступа к виртуальной машине.

Лабораторная работа 8

Эта лабораторная работа предназначена для повышения вашего знакомства с двумя важными инструментами для устранения неполадок в сети, **telnet** и **nmap**. Сетевые администраторы с некоторым опытом работы с Linux могут предпочесть другие инструменты. Если вы знаком с другими инструментами, такими как **nc**, отлично. Это результаты, которые имеют значение.