

От переводчика lehsi:

Перевод не профессиональный делал для себя. Буду рад если полученный текст принесёт пользу в изучения темы, номера страниц в меню кликабельны.



Red Hat Enterprise Linux 8.0 RH134

Red Hat System Administration II

Edition 1 20190531

Publication date 20190531

Authors: Fiona Allen, Adrian Andrade, Herve Quatremain, Victor Costea,
Snehangshu Karmakar, Marc Kesler, Saumik Paul

Editor: Philip Sweany, Ralph Rodriguez, David Sacco, Seth Kenlon, Heather Charles

Copyright © 2019 Red Hat, Inc.

Содержание этого курса, всех его модулей и сопутствующих материалов, включая раздаточные материалы для слушателей, защищено авторским правом © Red Hat, Inc., 2019.

Эта учебная программа, включая весь предоставленный здесь материал, предоставляется без каких-либо гарантий со стороны Red Hat, Inc. Red Hat, Inc. не несет ответственности за ущерб или судебные иски, возникшие в результате использования или неправильного использования содержимого или деталей, содержащихся в данном документе.

Linux® является зарегистрированным товарным знаком Линуса Торвальдса в США и других странах.

Java® является зарегистрированным товарным знаком Oracle и / или ее дочерних компаний.

XFS® является зарегистрированным товарным знаком Silicon Graphics International Corp. или ее дочерних компаний в США и / или других странах.

Знак OpenStack® Word и логотип OpenStack являются зарегистрированными товарными знаками/знаками обслуживания или товарными знаками/знаками обслуживания OpenStack Foundation в США и других странах и используются с разрешения OpenStack Foundation. Мы не связаны, не поддерживаются и не спонсируются OpenStack Foundation или сообществом OpenStack.

Все остальные товарные знаки являются собственностью соответствующих владельцев.

Авторы:

Ачют Мадхусудан, Роб Локк, Рудольф Кацл, Прашант Растиги, Хайдер Соуза, Майкл Филлипс, Даллас Спон

Условные обозначения в документе

Введение	8
Системное администрирование Red Hat II	8
Ориентация на работу в классе	9
Управление вашими системами	10
Интернационализация	12
1. Повышение производительности командной строки	19
Написание простых сценариев Bash	20
Упражнения с пошаговыми инструкциями: Написание простых сценариев Bash	24
Более эффективное выполнение команд с помощью циклов	28
Упражнения с пошаговыми инструкциями: Более эффективное выполнение команд с помощью циклов	35
Сопоставление текста в выводе команды с регулярными выражениями	38
Упражнения с пошаговыми инструкциями: Сопоставление текста в выводе команды с регулярными выражениями	47
Лабораторная работа: Повышение производительности командной строки	51
Резюме	57
2. Планирование будущих задач	58
Планирование отложенного пользовательского задания	59
Упражнения с пошаговыми инструкциями: Планирование отложенного пользовательского задания	61
Планирование повторяющихся пользовательских заданий	65
Упражнения с пошаговыми инструкциями: Планирование повторяющихся пользовательских заданий	69
Планирование повторяющихся системных заданий	72
Упражнения с пошаговыми инструкциями: Планирование повторяющихся системных заданий	76
Управление временными файлами	80
Упражнения с пошаговыми инструкциями: Управление временными файлами	84
Контрольный опрос: Планирование будущих задач	88
Резюме	92

3. Настройка производительности системы	93
Регулировка профилей настройки	94
Упражнения с пошаговыми инструкциями: Регулировка профилей настройки	100
Влияние на процесс планирования.	103
Упражнения с пошаговыми инструкциями: Влияние на процесс планирования.	107
Лабораторная работа: Настройка производительности системы.	111
Резюме	117
4. Управление доступом к файлам с помощью списков ACL	118
Интерпретация файловых ACL	119
Контрольный опрос: Интерпретация файловых ACL	128
Защита файлов с помощью списков контроля доступа	130
Упражнения с пошаговыми инструкциями: Защита файлов с помощью списков контроля доступа	136
Лабораторная работа: Управление доступом к файлам с помощью списков контроля доступа	142
Резюме	152
5. Управление безопасностью SELinux	153
Изменение режима принудительного применения SELinux	154
Упражнения с пошаговыми инструкциями: Изменение режима принудительного применения SELinux	159
Управление контекстами файлов SELinux	162
Упражнения с пошаговыми инструкциями: Управление контекстами файлов SELinux	166
Настройка политики SELinux с помощью логических значений	169
Упражнения с пошаговыми инструкциями: Настройка политики SELinux с помощью логических значений	171
Исследование и решение проблем SELinux	174
Упражнения с пошаговыми инструкциями: Исследование и решение проблем SELinux	179
Лабораторная работа: Управление безопасностью SELinux	183
Резюме	190
6. Управление основным хранилищем	191
Добавление разделов, файловых систем и постоянных подключений	192

Упражнения с пошаговыми инструкциями: Добавление разделов, файловых систем и постоянных подключений	204
Управление пространством подкачки (SWOP)	209
Упражнения с пошаговыми инструкциями: Управление пространством подкачки	214
Лабораторная работа: Лабораторная работа: Управление основным хранилищем	219
Резюме	229
7. Управление логическими томами	230
Создание логических томов	231
Упражнения с пошаговыми инструкциями: Создание логических томов	240
Расширение логических объемов	246
Упражнения с пошаговыми инструкциями: Расширение логических объемов	253
Лабораторная работа: Управление логическими томами	257
Резюме	263
8. Реализация расширенных функций хранения	264
Управление многоуровневым хранилищем с помощью Stratis	265
Упражнения с пошаговыми инструкциями: Управление многоуровневым хранилищем с помощью Stratis	272
Сжатие и дедупликация хранилища с помощью VDO	278
Упражнения с пошаговыми инструкциями: Сжатие и дедупликация хранилища с помощью VDO	281
Лабораторная работа: Реализация расширенных функций хранения	286
Резюме	296
9. Доступ к сетевому хранилищу	297
Монтирование сетевого хранилища с помощью NFS	298
Упражнение под руководством: Монтирование сетевого хранилища с помощью NFS	303
Автоматическое монтирование сетевых хранилищ	308
Упражнения с пошаговыми инструкциями: Автоматическое монтирование сетевых хранилищ	312
Лабораторная работа: Доступ к сетевому хранилищу	319
Резюме	327
10. Управление процессом загрузки	328
Выбор цели загрузки	329

Пошаговое упражнение: Выбор цели загрузки	334
Сброс пароля root	338
Упражнения с пошаговыми инструкциями: Сброс пароля root	342
Устранение проблем с файловой системой при загрузке	344
Упражнения с пошаговыми инструкциями: Устранение проблем с файловой системой при загрузке	346
Лабораторная работа: Управление процессом загрузки	349
Резюме	355
11. Управление сетевой безопасностью	356
Управление межсетевыми экранами сервера	357
Упражнения с пошаговыми инструкциями: Управление межсетевыми экранами сервера	367
Управление маркировкой портов SELinux	372
Упражнения с пошаговыми инструкциями: Управление маркировкой портов SELinux	376
Лабораторная работа: Управление сетевой безопасностью	381
Резюме	390
12. Установка Red Hat Enterprise Linux	391
Установка Red Hat Enterprise Linux	392
Упражнения с пошаговыми инструкциями: Установка Red Hat Enterprise Linux	397
Автоматизация установки с помощью Kickstart	400
Упражнения с пошаговыми инструкциями: Автоматизация установки с помощью Kickstart	411
Установка и настройка виртуальных машин	415
Контрольный опрос: Установка и настройка виртуальных машин	420
Лабораторная работа: Установка Red Hat Enterprise Linux	422
Резюме	430
13. Всесторонний обзор	431
Всесторонний обзор	432
Лабораторная работа: Устранение проблем с загрузкой и обслуживание серверов	436
Лабораторная работа: Настройка и управление файловыми системами и хранилищами	444
Лабораторная работа: Настройка и управление безопасностью сервера	452

СОГЛАШЕНИЯ В ДОКУМЕНТЕ



РЕКОМЕНДАЦИИ

«Рекомендации» описывают, где найти внешнюю документацию по теме.



ПРИМЕЧАНИЕ

«Примечания» - это подсказки, ярлыки или альтернативные подходы к решению поставленной задачи. Игнорирование заметки не должно иметь негативных последствий, но вы можете упустить уловку, которая облегчит вам жизнь.



ВАЖНО

В блоках «Важно» подробно описаны вещи, которые легко упустить: изменения конфигурации, которые применяются только к текущему сеансу, или службы, которые необходимо перезапустить перед применением обновления. Игнорирование поля с надписью: «Важно» не приведет к потере данных, но может вызвать раздражение и разочарование.



ПРЕДУПРЕЖДЕНИЕ

«Предупреждения» нельзя игнорировать. Игнорирование предупреждений, скорее всего, приведет к потере данных.

ВВЕДЕНИЕ

АДМИНИСТРАЦИЯ СИСТЕМЫ RED HAT II

Этот курс специально разработан для студентов, окончивших Red Hat System Administration I (RH124). Red Hat System Administration II (RH134) фокусируется на ключевых задачах, необходимых для того, чтобы стать полноценным администратором Linux и подтвердить эти навыки с помощью экзамена Red Hat Certified System Administrator. Этот курс углубляется в администрирование Enterprise Linux, включая файловые системы и разделы, логические тома, SELinux, межсетевой экран и устранение неполадок.

ЦЕЛИ КУРСА

- Расширяйте и расширяйте навыки, полученные во время курса Red Hat System Administration I (RH124).
- Развитие навыков, необходимых системному администратору Red Hat Enterprise Linux, имеющему сертификат RHCSA.

АУДИТОРИЯ

- Этот курс специально разработан для студентов, окончивших Red Hat System Administration I (RH124). Организован таким образом, что учащимся нецелесообразно использовать RH134 в качестве отправной точки учебной программы. Студентам, которые не проходили предыдущий курс Red Hat, рекомендуется пройти либо Системное администрирование I (RH124), если они плохо знакомы с Linux, либо курс RHCSA Fast Track (RH200), если они имеют опыт администрирования Enterprise Linux..

НЕОБХОДИМЫЕ УСЛОВИЯ

- После прохождения курса Red Hat System Administration I (RH124) или аналогичных знаний.

ОРИЕНТАЦИЯ НА УЧЕБНУЮ СРЕДУ ДЛЯ ВЫПОЛНЕНИЯ ЗАДАНИЙ

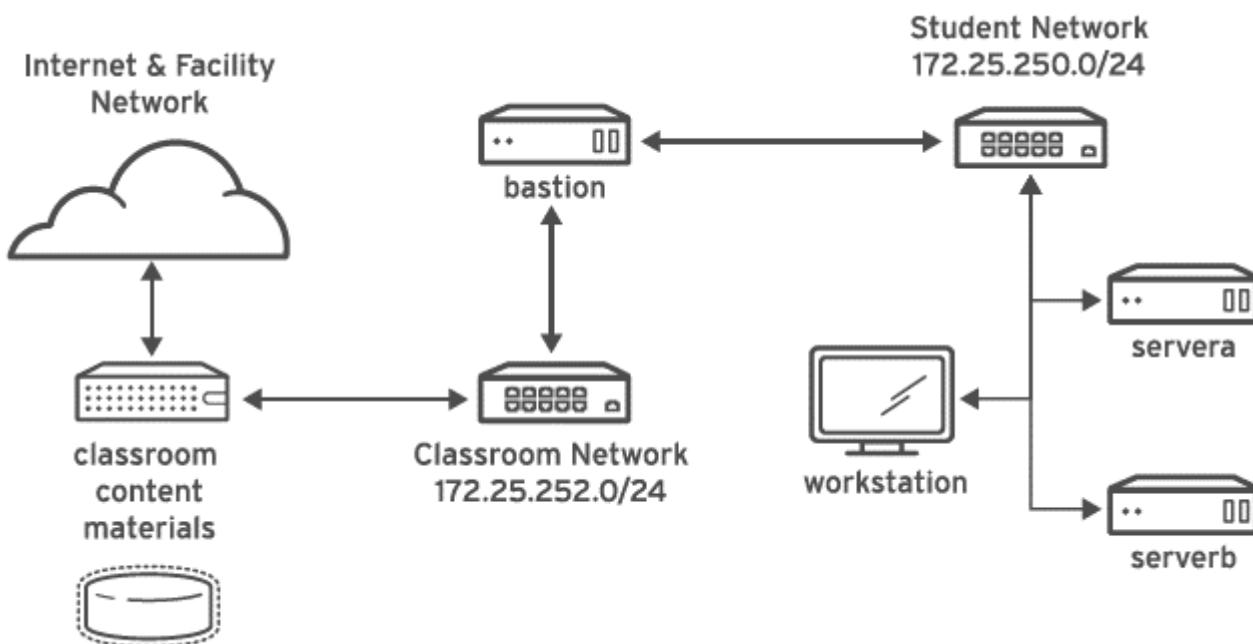


Рисунок 0.1: Среда для задач в классе

В этом курсе основной компьютерной системой, используемой для практического обучения, является рабочая станция (**workstation**). Студенты также используют две другие машины для этой деятельности: **servera**, и **serverb**. Все три из этих систем находятся в домене DNS **lab.example.com**.

Все компьютерные системы для учащихся имеют стандартную учетную запись пользователя **student** с паролем **student**. Пароль **root** во всех студенческих системах - **redhat**.

Назначение компьютеров в классной комнате

НАЗВАНИЕ МАШИНЫ	IP-АДРЕСА	РОЛЬ
bastion.lab.example.com	172.25.250.254	Система шлюза для подключения частной сети студента к классному серверу (всегда должна быть запущена)
workstation.lab.example.com	172.25.250.9	Графическая рабочая станция, используемая для системного администрирования
servera.lab.example.com	172.25.250.10	Первый сервер
serverb.lab.example.com	172.25.250.11	Второй сервер

Основная функция компьютера с именем **bastion** заключается в том, что он действует как маршрутизатор между сетью, соединяющей компьютеры учеников и сетью классной комнаты.

Если **bastion** не работает, другие компьютеры учеников смогут получить доступ к системам только в индивидуальной сети ученика.

Несколько систем в классе предоставляют вспомогательные услуги. Два сервера, **content.example.com** и **materials.example.com**, являются источниками программного обеспечения и лабораторных материалов, используемых в практических занятиях. Информация о том, как использовать эти серверы, содержится в инструкциях по этим действиям. Они предоставляются виртуальной машиной **classroom.example.com**. И **classroom**, и **bastion** должны всегда работать для правильного использования лабораторной среды.



ПРИМЕЧАНИЕ

При входе на **servera** или **serverb** вы можете увидеть сообщение об активации **cockpit**. Сообщение можно проигнорировать.

```
[student@workstation ~]$ ssh student@serverb
Warning: Permanently added 'serverb,172.25.250.11' (ECDSA) to the list
of known hosts.
Activate the web console with: systemctl enable --now cockpit.socket
[student@serverb ~]$
```

УПРАВЛЕНИЕ ВАШИМИ СИСТЕМАМИ

Студентам назначают удаленные компьютеры в классе онлайн-обучения Red Hat. Доступ к ним осуществляется через веб-приложение, размещенное на сайте rol.redhat.com [<http://rol.redhat.com>]. Студенты должны войти на этот сайт, используя свои учетные данные пользователя Red Hat Customer Portal.

Управление виртуальными машинами

Виртуальные машины в вашем классе управляются через веб-страницу. Состояние каждой виртуальной машины в классе отображается на странице под вкладкой **Online Lab**.

Состояния машины

СОСТОЯНИЕ ВИРТУАЛЬНОЙ МАШИНЫ	ОПИСАНИЕ
STARTING	Виртуальная машина загружается.
STARTED	Виртуальная машина запущена и доступна (или, при загрузке, скоро будет).

STOPPING	Виртуальная машина завершает работу.
STOPPED	Виртуальная машина полностью выключена. При запуске виртуальная машина загружается в том же состоянии, что и при выключении (диск будет сохранен).
PUBLISHING	Выполняется первоначальное создание виртуальной машины.
WAITING_TO_START	Виртуальная машина ожидает запуска других виртуальных машин.

В зависимости от состояния машины доступны следующие действия.

Classroom/Действия машины

КНОПКА ИЛИ ДЕЙСТВИЕ	ОПИСАНИЕ
ОБЕСПЕЧЕНИЕ ЛАБОРАТОРИИ (PROVISION LAB)	Создайте класс ROL. Создает все виртуальные машины, необходимые для работы в классе, и запускает их. Это может занять несколько минут.
УДАЛИТЬ ЛАБОРАТОРИЮ (DELETE LAB)	Удалите класс ROL. Уничтожает все виртуальные машины в классе. Внимание! Любая работа, выполненная на дисках, теряется.
НАЧАТЬ ЛАБОРАТОРИЮ (START LAB)	Запустите все виртуальные машины в классе.
ЗАКРЫТИЕ ЛАБОРАТОРИИ (SHUTDOWN LAB)	Остановите все виртуальные машины в классе.
ОТКРЫТАЯ КОНСОЛЬ (OPEN CONSOLE)	Откройте новую вкладку в браузере и подключитесь к консоли виртуальной машины. Студенты могут входить прямо в виртуальную машину и запускать команды. В большинстве случаев студенты должны войти в виртуальную машину рабочей станции и использовать <code>ssh</code> для подключения к другим виртуальным машинам.
ДЕЙСТВИЕ → Старт (ACTION → Start)	Запустите (включите) виртуальную машину.
ДЕЙСТВИЕ → Завершение работы (ACTION → Shutdown)	Изящно выключите виртуальную машину, сохранив содержимое ее диска.

ДЕЙСТВИЕ → Выключение (ACTION → Power Off)	Принудительно выключите виртуальную машину, сохранив содержимое ее диска. Это эквивалентно отключению питания от физического компьютера.
ДЕЙСТВИЕ → Сброс (ACTION → Reset)	Принудительно выключите виртуальную машину и верните диск в исходное состояние. Внимание: любая работа, созданная на диске, теряется.

В начале упражнения, если есть указание на сброс одного узла виртуальной машины, щелкните **ACTION → Reset** только для конкретной виртуальной машины.

В начале упражнения, если есть указание на сброс всех виртуальных машин, нажмите **ACTION → Reset**.

Если вы хотите вернуть среду класса в исходное состояние в начале курса, вы можете нажать **DELETE LAB**, чтобы удалить всю среду класса. После удаления лаборатории вы можете щелкнуть **PROVISION LAB**, чтобы подготовить новый набор классных систем.



ПРЕДУПРЕЖДЕНИЕ

Операцию **DELETE LAB** нельзя отменить. Любая работа, которую вы к этому моменту выполнили в классе, будет потеряна.

Таймер автостопа

Регистрация в Red Hat Online Learning дает студентам определенное количество компьютерного времени. Чтобы помочь сэкономить выделенное компьютерное время, в классе ROL есть связанный таймер обратного отсчета, который отключает среду класса, когда таймер истекает.

Чтобы настроить таймер, нажмите **MODIFY**, чтобы отобразить диалоговое окно «Новое время автостопа (*New Autostop Time*)». Установите количество часов, по истечении которых класс должен автоматически остановиться. Нажмите **ADJUST TIME**, чтобы применить это изменение к настройкам таймера.

ИНТЕРНАЦИОНАЛИЗАЦИЯ

ВЫБОР ЯЗЫКА ДЛЯ ПОЛЬЗОВАТЕЛЯ

Ваши пользователи могут предпочесть использовать для среды рабочего стола язык, отличный от общесистемного по умолчанию. Они также могут захотеть использовать другую раскладку клавиатуры или другой метод ввода для своей учетной записи.

Языковые настройки

В среде рабочего стола **GNOME** пользователю может быть предложено установить предпочтительный язык и метод ввода при первом входе в систему. Если нет, то самый простой способ для отдельного пользователя настроить предпочтительный язык и настройки метода ввода - использовать приложение «Регион и язык (**Region & Language**)».

Вы можете запустить это приложение двумя способами. Вы можете запустить команду **gnome-control-center region** из окна терминала или на верхней панели из системного меню в правом углу, выбрать кнопку настроек (на которой есть крестовая отвертка и гаечный ключ для значка) снизу слева от меню.

В открывшемся окне выберите Регион и язык (**Region & Language**). Щелкните поле «Язык (**Language**)» и выберите предпочтительный язык из появившегося списка. Это также обновит настройку форматов до значения по умолчанию для этого языка. При следующем входе в систему эти изменения вступят в силу в полной мере.

Эти настройки влияют на среду рабочего стола **GNOME** и любые приложения, такие как **gnometerinal**, которые запускаются внутри нее. Однако по умолчанию они не применяются к учетной записи, если доступ осуществляется через **ssh**-вход из удаленной системы или текстовый вход в виртуальную консоль (например, **tty5**).



ПРИМЕЧАНИЕ

Вы можете сделать так, чтобы в среде оболочки использовались те же настройки **LANG**, что и в графической среде, даже если вы входите в систему через текстовую виртуальную консоль или через **ssh**. Один из способов сделать это - разместить код, подобный приведенному ниже, в вашем файле **~/.bashrc**. В этом примере код языка, используемый при текстовом входе в систему, будет соответствовать языку, установленному в настоящее время для среды рабочего стола **GNOME** пользователя:

```
i=$(grep 'Language=' /var/lib/AccountsService/users/${USER} \
| sed 's/Language=//')
if [ "$i" != "" ]; then
export LANG=$i
fi
```

Японский, корейский, китайский и другие языки с нелатинским набором символов могут некорректно отображаться на текстовых виртуальных консолях

Отдельные команды можно заставить использовать другой язык, установив переменную **LANG** в командной строке:

```
[user@host ~]$ LANG=fr_FR.utf8 date
jeu. avril 25 17:55:01 CET 2019
```

Последующие команды вернутся к использованию языка системы по умолчанию для вывода. Команду **locale** можно использовать для определения текущего значения **LANG** и других связанных переменных среды.

Настройки метода ввода

GNOME 3 в Red Hat Enterprise Linux 7 или более поздней версии автоматически использует систему выбора метода ввода **IBus**, что упрощает быстрое изменение раскладки клавиатуры и методов ввода.

Приложение «Регион и язык (**Region & Language**)» также можно использовать для включения альтернативных методов ввода. В окне приложения «Регион и язык (**Region & Language**)» поле «Источники ввода (**Input Sources**)» показывает, какие методы ввода доступны в настоящее время. По умолчанию английский (США) (**English (US)**) может быть единственным доступным способом. Выделите Английский (США) (**English (US)**) и щелкните значок клавиатуры, чтобы увидеть текущую раскладку клавиатуры.

Чтобы добавить другой метод ввода, нажмите кнопку «+» в нижнем левом углу окна «Источники ввода(**Input Sources**)». Откроется окно «Добавить источник входного сигнала (**Add an Input Source**)». Выберите свой язык, а затем предпочтительный метод ввода или раскладку клавиатуры.

Если настроено более одного метода ввода, пользователь может быстро переключаться между ними, набирая **Super + Space** (иногда называемый **Windows + Space**). Индикатор состояния также появится на верхней панели GNOME, у которой есть две функции: он указывает, какой метод ввода активен, и действует как меню, которое можно использовать для переключения между методами ввода или выбора дополнительных функций более сложных методов ввода.

Некоторые методы отмечены шестерenkами, что указывает на то, что эти методы имеют расширенные параметры конфигурации и возможности. Например, метод ввода японский японский (Кана Кандзи (**Japanese (Kana Kanji)**)) позволяет пользователю предварительно редактировать текст на латинице и использовать клавиши со стрелками вниз и вверх для выбора правильных символов для использования.

Для носителей английского языка в США это также может оказаться полезным. Например, под **English (United States)** находится раскладка клавиатуры **English (international AltGr dead keys)**, при которой **AltGr** (или правый Alt) на 104/105-клавишной клавиатуре ПК рассматривается как клавиша-модификатор «вторичного сдвига» и неработающая клавиша. Ключ активации для набора дополнительных символов. Также доступны Дворжак (Dvorak) и другие альтернативные раскладки.



ПРИМЕЧАНИЕ

Любой символ **Unicode** можно ввести в среду рабочего стола **GNOME**, если вы знаете кодовый указатель **Unicode** символа. Введите **Ctrl+Shift+U**, а затем кодовый указатель. После нажатия **Ctrl+Shift+U** будет отображаться подчеркнутая буква **u**, указывающая на то, что система ожидает ввода кода указателя **Unicode**.

Например, строчная греческая буква лямбда имеет кодовый указатель **U+03BB**, и ее можно ввести, набрав **Ctrl+Shift+U**, затем **03BB**, и нажать клавишу **Enter**.

ОБЩИЕ СИСТЕМНЫЕ НАСТРОЙКИ ЯЗЫКА ПО УМОЛЧАНИЮ

В качестве языка системы по умолчанию используется английский (США) с использованием кодировки **Unicode UTF-8** в качестве набора символов (**en_US.utf8**), но это можно изменить во время или после установки.

Из командной строки пользователь **root** может изменить общесистемные настройки локали с помощью команды **localectl**. Если **localectl** запущен без аргументов, он отображает текущие общесистемные настройки локали.

Чтобы установить общесистемный язык по умолчанию, запустите команду **localectl set-locale LANG=locale**, где **locale** - это соответствующее значение переменной среды **LANG** из таблицы «Справочник кодов языков» в этой главе. Изменение вступит в силу для пользователей при их следующем входе в систему и хранится в **/etc/locale.conf**.

```
[root@host ~]# localectl set-locale LANG=fr_FR.utf8
```

В GNOME пользователь с правами администратора может изменить этот параметр в разделе «Регион и язык (**Region & Language**)», нажав кнопку «Экран входа в систему (**Login Screen**)» в правом верхнем углу окна. Изменение языка графического экрана входа также приведет к корректировке общесистемных языковых настроек по умолчанию, хранящихся в файле конфигурации **/etc/locale.conf**.



ВАЖНО

Текстовые виртуальные консоли, такие как **tty4**, более ограничены в шрифтах, которые они могут отображать, чем терминалы в виртуальной консоли, работающей в графической среде, или псевдотерминалы для сеансов **ssh**. Например, японские, корейские и китайские символы могут не отображаться должным образом на текстовой виртуальной консоли. По этой причине вам следует подумать об использовании английского или другого языка с набором латинских символов для общесистемного значения по умолчанию.

Точно так же виртуальные консоли на основе текста более ограничены в методах ввода, которые они поддерживают, это управляет отдельно от графической среды рабочего стола. Доступные глобальные параметры ввода можно настроить с помощью **localectl**, как для текстовых виртуальных консолей, так и для графической среды. См. Справочные страницы **localectl (1)** и **vconsole.conf (5)** для получения дополнительной информации.

ЯЗЫКОВЫЕ ПАКЕТЫ

Специальные пакеты RPM, называемые **langpacks**, устанавливают языковые пакеты, которые добавляют поддержку определенных языков. Пакеты **langpacks** используют зависимости для автоматической установки дополнительных пакетов RPM, содержащих локализации, словари и переводы для других пакетов программного обеспечения в вашей системе.

Чтобы вывести список установленных и которые могут быть установлены пакеты **langpack**, используйте **yum list langpacks- ***:

```
[root@host ~]# yum list langpacks-*  
Updating Subscription Management repositories.  
Updating Subscription Management repositories.  
Installed Packages  
langpacks-en.noarch      1.0-12.el8      @AppStream  
Available Packages  
langpacks-af.noarch       1.0-12.el8      rhel-8-for-x86_64-appstream-rpms  
langpacks-am.noarch       1.0-12.el8      rhel-8-for-x86_64-appstream-rpms  
langpacks-ar.noarch       1.0-12.el8      rhel-8-for-x86_64-appstream-rpms  
langpacks-as.noarch       1.0-12.el8      rhel-8-for-x86_64-appstream-rpms  
langpacks-ast.noarch      1.0-12.el8      rhel-8-for-x86_64-appstream-rpms  
...output omitted...
```

Чтобы добавить языковую поддержку, установите соответствующий пакет **langpacks**. Например, следующая команда добавляет поддержку французского языка:

```
[root@host ~]# yum install langpacks-fr
```

Используйте **yumrepoquery --whatsonplements**, чтобы определить, какие пакеты RPM могут быть установлены с помощью **langpack**:

```
[root@host ~]# yum repoquery --whatsonplements langpacks-fr  
Updating Subscription Management repositories.  
Updating Subscription Management repositories.  
Last metadata expiration check: 0:01:33 ago on Wed 06 Feb 2019 10:47:24 AM CST.  
glibc-langpack-fr-0:2.28-18.el8.x86_64  
gnome-getting-started-docs-fr-0:3.28.2-1.el8.noarch  
hunspell-fr-0:6.2-1.el8.noarch  
hyphen-fr-0:3.0-1.el8.noarch  
libreoffice-langpack-fr-1:6.0.6.1-9.el8.x86_64  
man-pages-fr-0:3.70-16.el8.noarch  
mythes-fr-0:2.3-10.el8.noarch
```



ВАЖНО

Пакеты **Langpacks** используют RPM *weak dependencies*, чтобы устанавливать дополнительные пакеты только тогда, когда также установлен основной пакет, который в этом нуждается.

Например, при установке **langpacks-fr**, как показано в предыдущих примерах, пакет **mythes-fr** будет установлен только в том случае, если тезаурус **mythes** также установлен в системе.

Если впоследствии в этой системе будет установлен **mythes**, пакет **mythes-fr** также будет установлен автоматически из-за слабой зависимости от уже установленного пакета **langpacks-fr**.



РЕКОМЕНДАЦИИ

Страницы руководства [page: locale \(7\), localectl \(1\), locale.conf \(5\), vconsole.conf \(5\), unicode \(7\) и utf-8 \(7\)](#)

Преобразования между именами макетов **X11** графической среды рабочего стола и их именами в **localectl** можно найти в файле **/usr/share/X11/xkb/rules/base.lst**

СПРАВОЧНИК ЯЗЫКОВЫЕ КОДЫ



ПРИМЕЧАНИЕ

Эта таблица может не отражать все языковые пакеты, доступные в вашей системе. Используйте **yum info langpacks-SUFFIX**, чтобы получить дополнительную информацию о любом конкретном пакете **langpacks**.

Коды языков

ЯЗЫК	СУФФИКС ЯЗЫКОВЫХ ПАКЕТОВ	\$ LANG VALUE
English (US)	en	en_US.utf8
Assamese	as	as_IN.utf8
Bengali	bn	bn_IN.utf8
Chinese (Simplified)	zh_CN	zh_CN.utf8
Chinese (Traditional)	zh_TW	zh_TW.utf8
French	fr	fr_FR.utf8
German	de	de_DE.utf8
Gujarati	gu	gu_IN.utf8

Hindi	hi	hi_IN.utf8
Italian	it	it_IT.utf8
Japanese	ja	ja_JP.utf8
Kannada	kn	kn_IN.utf8
Korean	ko	ko_KR.utf8
Malayalam	ml	ml_IN.utf8
Marathi	mr	mr_IN.utf8
Odia	or	or_IN.utf8
Portuguese (Brazilian)	pt_BR	pt_BR.utf8
Punjabi	pa	pa_IN.utf8
Russian	ru	ru_RU.utf8
Spanish	es	es_ES.utf8
Tamil	ta	ta_IN.utf8
Telugu	te	te_IN.utf8

ГЛАВА 1

ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ КОМАНДНОЙ СТРОКИ

ЦЕЛЬ

Выполняйте команды более эффективно с помощью расширенных функций оболочки **Bash**, сценариев оболочки и различных утилит, предоставляемых Red Hat Enterprise Linux.

ЗАДАЧИ

- Автоматизируйте последовательность команд, написав простой сценарий оболочки.
- Эффективное выполнение команд над списками элементов в сценарии или из командной строки, используя цикл **for** и условные выражения.
- Найдите текст, соответствующий шаблону, в файлах журнала и выводе команд с помощью команды **grep** и регулярных выражений.

РАЗДЕЛЫ

- Написание простых сценариев **Bash** (и упражнения с пошаговыми инструкциями)
- Более эффективное выполнение команд с помощью циклов (и упражнения с пошаговыми инструкциями)
- Сопоставление текста в выводе команды с регулярными выражениями (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Повышение производительности командной строки.

НАПИСАНИЕ ПРОСТЫХ СКРИПТОВ BASH

ЗАДАЧИ

После завершения этого раздела вы сможете автоматизировать последовательность команд, написав простой сценарий оболочки.

СОЗДАНИЕ И ВЫПОЛНЕНИЕ СКРИПТОВ ОБОЛОЧКИ BASH

Многие простые общие задачи системного администрирования выполняются с помощью инструментов командной строки.

Задачи с большей сложностью часто требуют объединения нескольких команд, которые передают результаты между ними. Используя среду оболочки **Bash** и функции сценариев, команды Linux объединяются в сценарии оболочки, чтобы легко решать повторяющиеся и сложные проблемы реального мира.

В своей простейшей форме сценарий оболочки **Bash** представляет собой исполняемый файл, содержащий список команд и, возможно, с программной логикой для управления принятием решений в общей задаче. Хорошо написанный сценарий оболочки сам по себе является мощным инструментом командной строки и может использоваться другими сценариями.

Умение писать сценарии оболочки необходимо для успешного системного администрирования в любой операционной среде. Практическое знание сценариев оболочки имеет решающее значение в корпоративных средах, где использование сценариев может повысить эффективность и точность выполнения рутинных задач.

Вы можете создать сценарий оболочки **Bash**, открыв новый пустой файл в текстовом редакторе. Вы можете использовать любой текстовый редактор, расширенные редакторы, такие как **vim** или **emacs**, понимают синтаксис оболочки **Bash** и могут обеспечивать цветовую подсветку. Это выделение помогает выявить распространенные ошибки, такие как неправильный синтаксис, непарные кавычки, незакрытые круглые скобки, фигурные и квадратные скобки и многое другое.

Указание интерпретатора команд

Первая строка сценария начинается с обозначения «#!», Обычно называемого **sh-bang** или **shebang**, от названий этих двух символов, «диэз (**sharp**)» и «бэнг». Это конкретное двухбайтовое обозначение магического числа указывает на интерпретируемый сценарий; синтаксис, следующий за обозначением, - это полное имя файла для правильного интерпретатора команд, необходимого для обработки строк этого сценария. Чтобы понять, как магические числа обозначают типы файлов в Linux, см. Справочные страницы **file(1)** и **magic(5)**. Для файлов сценариев, использующих синтаксис сценариев **Bash**, первая строка сценария оболочки начинается следующим образом:

```
#!/bin/bash
```

Выполнение сценария оболочки Bash

Завершенный сценарий оболочки должен быть исполняемым, чтобы запускаться как обычная команда. Используйте команду **chmod**, чтобы добавить разрешение на выполнение, возможно, вместе с командой **chown**, чтобы изменить владельца файла сценария. Предоставьте разрешение на выполнение только предполагаемым пользователям сценария.

Если вы поместите сценарий в один из каталогов, перечисленных в переменные среды оболочки **PATH**, то вы можете вызвать сценарий оболочки shell, используя только имя файла, как и любую другую команду. Оболочка shell использует первую найденную команду с этим именем файла; **избегайте использования существующих имен команд для имени файла сценария оболочки**. Кроме того, вы можете вызвать сценарий оболочки, указав путь к сценарию в командной строке. Команда **which**, за которой следует имя файла исполняемого сценария, отображает путь к команде, которая будет выполнена.

```
[user@host ~]$ which hello  
~/bin/hello  
[user@host ~]$ echo $PATH  
/home/user/.local/bin:/home/user/bin:/usr/share/Modules/bin:/usr/local/bin:/usr/  
bin:/usr/local/sbin:/usr/sbin
```

Экранирование специальных символов

Ряд символов и слов имеют особое значение для оболочки Bash. Однако иногда вы захотите использовать эти символы для их буквальных значений, а не для их особого назначения. Для этого воспользуйтесь одним из трех инструментов для удаления (или выхода) специального значения: **обратной косой чертой (\)**, **одинарных кавычек ("")** или **двойных кавычек ("")**.

Экранирующий символ обратной косой черты удаляет особое значение одного следующего за ним символа. Например, чтобы отобразить буквальную строку с хэштегом **#**, а не комментарий с командой **echo**, знак **#** не должен интерпретироваться Bash, как имеющий особое значение. Поместите обратную косую черту перед знаком **#**.

```
[user@host ~]$ echo # not a comment  
[user@host ~]$ echo \# not a comment  
# not a comment
```

Если вам нужно экранировать более одного символа в текстовой строке, используйте экранирующие-символы несколько раз или используйте одинарные кавычки (**"**). Одиночные кавычки сохраняют буквальное значение всех символов, которые они заключают. Обратите внимание на экранирующий-символ и одинарные кавычки в действии:

```
[user@host ~]$ echo # not a comment #  
  
[user@host ~]$ echo \# not a comment #  
# not a comment  
[user@host ~]$ echo \# not a comment \#
```

```
# not a comment #
[user@host ~]$ echo '# not a comment #'
# not a comment #
```

Используйте двойные кавычки, чтобы отключение файловых шаблонов и подстановок командного интерпретатора (оболочки), за исключением подстановок команд и переменных. Подстановка переменных концептуально идентична подстановке команд, но может использовать необязательный синтаксис фигурных скобок. Обратите внимание на приведенные ниже примеры использования кавычек в различных формах.

Используйте одинарные кавычки для буквального толкования всего текста. Помимо преобразования глобализации и расширения оболочки, кавычки предписывают оболочке дополнительно подавлять подстановку команд и переменных. Вопросительный знак (?) - это метасимвол, который также нуждается в защите от расширения.

```
[user@host ~]$ var=$(hostname -s); echo $var
host
[user@host ~]$ echo "***** hostname is ${var} *****"
***** hostname is host *****
[user@host ~]$ echo Your username variable is \$USER.
Your username variable is $USER.
[user@host ~]$ echo "Will variable $var evaluate to $(hostname -s)?"
Will variable host evaluate to host?
[user@host ~]$ echo 'Will variable $var evaluate to $(hostname -s)?'
Will variable $var evaluate to $(hostname -s)?
[user@host ~]$ echo '\"Hello, world\"'
"Hello, world"
[user@host ~]$ echo '\"Hello, world\"'
"Hello, world"
```

Обеспечение вывода из сценария shell

Команда **echo** отображает произвольный текст, передавая текст в качестве аргумента команды. По умолчанию текст отображается в стандартном выводе **standard output (STDOUT)**, но его также можно направить в стандартную ошибку **standard error (STDERR)** с помощью перенаправления вывода. В следующем простом сценарии **Bash** команда **echo** отображает сообщение «Hello, world» для **STDOUT**.

```
[user@host ~]$ cat ~/bin/hello
#!/bin/bash

echo "Hello, world"

[user@host ~]$ hello
Hello, world
```



ПРИМЕЧАНИЕ

Этот пользователь может просто запустить скрипт **hello** в командной строке, потому что каталог **~/bin** (**/home/user/bin**) находится в переменной **PATH** пользователя, а сценарий **hello** в нем является исполняемым. Оболочка shell автоматически находит там сценарий, пока нет другого исполняемого файла с именем **hello** ни в одном из каталогов, перечисленных перед **/home/user/bin** в переменной **PATH**.

Команда **echo** широко используется в сценариях оболочки для отображения информационных сообщений или сообщений об ошибках. Эти сообщения могут быть полезным индикатором хода выполнения сценария и могут быть направлены либо на стандартный вывод (**standard output**), стандартную ошибку (**standard error**), либо в файл журнала для архивирования. При отображении сообщений об ошибках рекомендуется направлять их в **STDERR**, чтобы было легче отличить сообщения об ошибках от обычных сообщений о состоянии.

```
[user@host ~]$ cat ~/bin/hello
#!/bin/bash

echo "Hello, world"
echo "ERROR: Houston, we have a problem." >&2

[user@host ~]$ hello 2> hello.log
Hello, world
[user@host ~]$ cat hello.log
ERROR: Houston, we have a problem.
```

Команда **echo** также может быть очень полезной при отладке проблемного сценария оболочки. Добавление операторов **echo** к той части скрипта, которая ведет себя не так, как ожидалось, может помочь прояснить выполняемые команды, а также значения вызываемых переменных.



РЕКОМЕНДАЦИИ

Справочные страницы **man bash(1)**, **magic(5)**, **echo(1)**, и **echo(1p)**.

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

Написание простых скриптов BASH

В этом упражнении вы напишете простой сценарий **Bash**, содержащий последовательность команд, и запустите его из командной строки.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Написать и выполнить простой сценарий **Bash**.
- Перенаправить вывод простого сценария **Bash** в файл.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните команду **lab console-write start**. Команда запускает сценарий, который определяет, доступна ли машина **servera** в сети. Сценарий предупредит вас, если он недоступен. При необходимости он также устанавливает расширенный пакет **vim**.

```
[student@workstation ~]$ lab console-write start
```

1. С хоста **workstation** откройте сеанс **SSH** на сервер **servera**, как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Создайте и выполните простой сценарий **Bash**.

2.1. Используйте текстовый редактор **vim**, чтобы создать новый текстовый файл в вашем домашнем каталоге, назовите его **firstscript.sh**

```
[student@servera ~]$ vim firstscript.sh
```

2.2. Вставьте следующий текст и сохраните файл. Обратите внимание, что количество знаков решетки (#) произвольно.

```
#!/bin/bash  
echo "This is my first bash script" > ~/output.txt
```

```
echo "" >> ~/output.txt
echo "#####" >> ~/output.txt
```

2.3. Используйте команду **sh** для выполнения сценария.

```
[student@servera ~]$ sh firstscript.sh
```

2.4. Просмотрите выходной файл, созданный сценарием.

```
[student@servera ~]$ cat output.txt
This is my first bash script
#####
```

3. Добавьте дополнительные команды в сценарий **firstscript.sh**, выполните его и просмотрите вывод.

3.1. Используйте текстовый редактор **vim** для редактирования **firstscript.sh**

```
[student@servera ~]$ vim firstscript.sh
```

3.2. Добавьте следующие строки полужирным шрифтом в файл **firstscript.sh**.

```
#!/bin/bash
#
echo "This is my first bash script" > ~/output.txt
echo "" >> ~/output.txt
echo "#####" >> ~/output.txt
echo "LIST BLOCK DEVICES" >> ~/output.txt
echo "" >> ~/output.txt
lsblk >> ~/output.txt
echo "" >> ~/output.txt
echo "#####" >> ~/output.txt
echo "FILESYSTEM FREE SPACE STATUS" >> ~/output.txt
echo "" >> ~/output.txt
df -h >> ~/output.txt
echo "#####" >> ~/output.txt
```

3.3. Сделайте файл **firstscript.sh** исполняемым с помощью команды **chmod**.

```
[student@servera ~]$ chmod a+x firstscript.sh
```

3.4. Выполните сценарий **firstscript.sh**.

```
[student@servera ~]$ ./firstscript.sh
```

3.5. Просмотрите выходной файл, созданный сценарием.

```
[student@servera ~]$ cat output.txt
This is my first bash script
#####
LIST BLOCK DEVICES

NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0     11:0    1 1024M  0 rom
vda     252:0   0   10G  0 disk
└─vda1  252:1   0   10G  0 part /
vdb     252:16  0    5G  0 disk

#####
FILESYSTEM FREE SPACE STATUS

Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        892M    0  892M  0% /dev
tmpfs          915M    0  915M  0% /dev/shm
tmpfs          915M   17M  899M  2% /run
tmpfs          915M    0  915M  0% /sys/fs/cgroup
/dev/vda1       10G  1.5G  8.6G 15% /
tmpfs         183M    0  183M  0% /run/user/1000
#####
```

4. Удалите файлы упражнений и выйдите из сервера.

4.1. Удалите файл сценария **firstscript.sh** и выведите файл **output.txt**.

```
[student@servera ~]$ rm firstscript.sh output.txt
```

4.2. Выйти из сервера **servera**.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation** выполните скрипт **lab console-write finish**, чтобы завершить данное упражнение.

```
[student@workstation ~]$ lab console-write finish
```

На этом пошаговое упражнение завершено.

БОЛЕЕ ЭФФЕКТИВНОЕ ИСПОЛЬЗОВАНИЕ ЦИКЛОВ ДЛЯ РАБОТЫ КОМАНД

ЗАДАЧИ

После завершения изучения данного раздела вы сможете:

- Перебирать списки с помощью циклов **for**.
- Оценивайте коды выхода из команд и сценариев.
- Выполняйте тесты с помощью операторов.
- Создавайте условные структуры с помощью операторов **if**.

Использование циклов для повторения команд

Системные администраторы часто сталкиваются с повторяющимися задачами в своей повседневной деятельности. Повторяющиеся задачи могут принимать форму многократного выполнения действия над целью, например, проверки процесса каждую минуту в течение 10 минут на предмет его завершения. Повторение задачи также может принимать форму однократного выполнения действия для нескольких целей, например, резервного копирования каждой базы данных в системе. Цикл **for** является одной из множества конструкций цикла оболочки, предлагаемых **Bash**, и может использоваться для повторных задач.

Обработка элементов из командной строки

Конструкция цикла **for** в **Bash** использует следующий синтаксис.

```
for VARIABLE in LIST; do  
COMMAND VARIABLE  
Done
```

Цикл обрабатывает строки, представленные в *LIST*, по порядку, одну за другой, и завершает работу после обработки последней строки в списке. Каждая строка в списке временно сохраняется как значение *VARIABLE*, в то время как цикл **for** выполняет блок команд, содержащихся в его конструкции. Имя переменной произвольно. Обычно на значение переменной ссылаются команды в командном блоке.

Список строк, предоставляемых циклу **for**, может быть предоставлен несколькими способами. Это может быть список строк, введенных пользователем напрямую, или созданный из различных типов расширения оболочки, таких как переменная, скобки, расширение имени файла или подстановка команд. Некоторые примеры, демонстрирующие различные способы предоставления строк, за которыми следуют циклы **for**.

```
[user@host ~]$ for HOST in host1 host2 host3; do echo $HOST; done  
host1  
host2
```

```
host3
[user@host ~]$ for HOST in host{1,2,3}; do echo $HOST; done
host1
host2
host3
[user@host ~]$ for HOST in host{1..3}; do echo $HOST; done
host1
host2
host3
[user@host ~]$ for FILE in file*; do ls $FILE; done
filea
fileb
filec
[user@host ~]$ for FILE in file{a..c}; do ls $FILE; done
filea
fileb
filec
[user@host ~]$ for PACKAGE in $(rpm -qa | grep kernel); \
do echo "$PACKAGE was installed on \
$(date -d @$($rpm -q --qf "%{INSTALLTIME}\n" $PACKAGE))"; done
abrt-addon-kerneloops-2.1.11-12.el7.x86_64 was installed on Tue Apr 22 00:09:07
EDT 2014
kernel-3.10.0-121.el7.x86_64 was installed on Thu Apr 10 15:27:52 EDT 2014
kernel-tools-3.10.0-121.el7.x86_64 was installed on Thu Apr 10 15:28:01 EDT 2014
kernel-tools-libs-3.10.0-121.el7.x86_64 was installed on Thu Apr 10 15:26:22 EDT
2014
[user@host ~]$ for EVEN in $(seq 2 2 10); do echo "$EVEN"; done
2
4
6
8
10
```

ИСПОЛЬЗОВАНИЕ КОДОВ ВЫХОДА В СКРИПТЕ

После того, как сценарий обработал все свое содержимое, он переходит к процессу, который его вызвал. Однако могут быть случаи, когда желательно выйти из сценария до его завершения, например, когда возникает ошибка. Это можно сделать с помощью команды **exit** в сценарии. Когда сценарий встречает команду **exit**, он немедленно завершает работу и не обрабатывает оставшуюся часть сценария.

Команда **exit** может быть выполнена с необязательным целочисленным аргументом от **0** до **255**, который представляет код выхода. Код выхода - это код, который возвращается после завершения процесса. Значение кода выхода **0** означает отсутствие ошибки. Все остальные ненулевые значения указывают на код выхода из ошибки. Вы можете использовать разные ненулевые значения, чтобы различать разные типы обнаруженных ошибок. Этот код выхода передается обратно родительскому процессу, который сохраняет его в переменной **?**, доступ к которой можно получить с помощью **\$?** как показано в следующих примерах.

```
[user@host bin]$ cat hello
#!/bin/bash
echo "Hello, world"
exit 0
```

```
[user@host bin]$ ./hello
Hello, world
```

```
[user@host bin]$ echo $?
0
```

Если команда **exit** вызывается без аргумента, то сценарий завершается и передает статус выхода последней выполненной команды родительскому процессу.

ТЕСТИРОВАНИЕ ВХОДНЫХ ДАННЫХ СКРИПТА

Чтобы гарантировать, что сценарии не будут легко нарушены неожиданными условиями, рекомендуется не делать предположений относительно вводимых данных, таких как аргументы командной строки, ввод данных пользователем, подстановки команд, расширения переменных и расширения имен файлов. Проверка целостности может быть выполнена с помощью команды **Bash test**.

Как и все команды, тестовая команда по завершении выдает код выхода, который сохраняется как значение **\$?**. Чтобы увидеть результат выполнения команды **test**, отобразите значение **\$?** сразу после выполнения команды. Опять же, значение статуса выхода 0 указывает, что проверка прошла успешно, а ненулевые значения указывают, что тест завершился с ошибками.

Тесты проводятся с использованием различных операторов. Операторы могут использоваться, чтобы определить, больше ли число, больше или равно, меньше, меньше или равно, или равно другому числу. Их можно использовать для проверки того, совпадает ли строка текста с другой строкой текста или нет. Операторы также могут использоваться для оценки того, имеет ли переменная значение или нет.



ПРИМЕЧАНИЕ

В сценариях оболочки используются многие типы операторов в дополнение к операторам сравнения, описанным здесь. На странице руководства для команды **test(1)** перечислены важные операторы условных выражений с описаниями. Справочная страница **bash(1)** также объясняет использование и оценку операторов, но ее очень трудно читать новичкам. Студентам рекомендуется развивать свои навыки в написании сценариев оболочки с помощью книг и курсов, посвященных программированию оболочки **shell**.

В следующих примерах демонстрируется использование команды **test** с использованием числовых операторов сравнения **Bash**.

```
[user@host ~]$ test 1 -gt 0 ; echo $?
0
[user@host ~]$ test 0 -gt 1 ; echo $?
1
```

Тесты могут выполняться с использованием синтаксиса команды тестирования **Bash**, [**<TESTEXPRESSION>**]. Они также могут быть выполнены с использованием нового расширенного синтаксиса тестовых команд **Bash**, [[**<TESTEXPRESSION>**]], который доступен с версии **Bash 2.02** и предоставляет такие функции, как сопоставление глобальных шаблонов и сопоставление шаблонов регулярных выражений.

Следующие примеры демонстрируют использование синтаксиса тестовой команды Bash и числовых операторов сравнения Bash.

```
[user@host ~]$ [ 1 -eq 1 ]; echo $?
0
[user@host ~]$ [ 1 -ne 1 ]; echo $?
1
[user@host ~]$ [ 8 -gt 2 ]; echo $?
0
[user@host ~]$ [ 2 -ge 2 ]; echo $?
0
[user@host ~]$ [ 2 -lt 2 ]; echo $?
1
[user@host ~]$ [ 1 -lt 2 ]; echo $?
0
```

Следующие примеры демонстрируют использование операторов сравнения строк **Bash**.

```
[user@host ~]$ [ abc = abc ]; echo $?
0
[user@host ~]$ [ abc == def ]; echo $?
1
[user@host ~]$ [ abc != def ]; echo $?
```

Следующие примеры демонстрируют использование строковых унарных операторов Bash.

```
[user@host ~]$ STRING=''; [ -z "$STRING" ]; echo $?
0
[user@host ~]$ STRING='abc'; [ -n "$STRING" ]; echo $?
0
```



ПРИМЕЧАНИЕ

Пробелы внутри тестовых скобок являются обязательными, поскольку они разделяют слова и элементы в тестовом выражении. Процедура синтаксического анализа команд оболочки делит все командные строки на слова и операторы, распознавая пробелы и другие метасимволы, используя встроенные правила синтаксического анализа. Полное описание этой продвинутой концепции см. На странице руководства **getopt(3)**. Символ левой квадратной скобки ([) сам по себе является встроенным псевдонимом для тестовой команды. Слова оболочки, будь то команды, подкоманды, параметры, аргументы или другие элементы-токены, всегда разделяются пробелами.

УСЛОВНЫЕ КОНСТРУКЦИИ

Простые сценарии оболочки представляют собой набор команд, которые выполняются от начала до конца. Условные структуры позволяют пользователям включать принятие решений в сценарии оболочки, так что определенные части сценария выполняются только при соблюдении определенных условий.

Использование конструкция if/then

Самой простой из условных структур в Bash является конструкция **if/then**, имеющая следующий синтаксис.

```
if <CONDITION>; then
<STATEMENT>
...
<STATEMENT>
fi
```

С помощью этой конструкции, если заданное условие выполняется, выполняется одно или несколько действий. Если данное условие не выполняется, никаких действий не предпринимается. Показанные ранее числовые, строковые и файловые тесты часто используются для проверки условий в операторах **if/then**. Оператор **fi** в конце закрывает конструкцию **if/then**. В следующем разделе кода демонстрируется использование конструкции **if/then** для запуска службы **psacct**, если она не активна.

```
[user@host ~]$ systemctl is-active psacct > /dev/null 2>&1
[user@host ~]$ if [ $? -ne 0 ]; then
> sudo systemctl start psacct
> fi
```

Использование конструкции if/then/else

Конструкция **if/then** может быть дополнительно расширена, чтобы можно было выполнять различные наборы действий в зависимости от того, выполняется ли условие. Это достигается с помощью конструкции **if/then/else**.

```
if <CONDITION>; then
<STATEMENT>
...
<STATEMENT>
else
<STATEMENT>
...
<STATEMENT>
fi
```

В следующем разделе кода демонстрируется использование оператора **if/then/else** для запуска службы **psacct**, если она неактивна, и для ее остановки, если она активна.

```
[user@host ~]$ systemctl is-active psacct > /dev/null 2>&1
[user@host ~]$ if [ $? -ne 0 ]; then
> sudo systemctl start psacct
> else
> sudo systemctl stop psacct
> fi
```

Использование конструкции **if/then/elif/then/else**

Наконец, конструкция **if/then/else** может быть дополнительно расширена для проверки более чем одного условия, выполняя другой набор действий при выполнении условия. Конструкция для этого показана в следующем примере:

```
if <CONDITION>; then
<STATEMENT>
...
<STATEMENT>
elif <CONDITION>; then
<STATEMENT>
...
<STATEMENT>
else
<STATEMENT>
...
<STATEMENT>
fi
```

В этой условной структуре Bash проверяет условия в представленном порядке. Когда он находит условие, которое является истинным, Bash выполняет действия, связанные с условием, а затем пропускает оставшуюся часть условной структуры. Если ни одно из условий не выполняется, Bash выполняет действия, перечисленные в условие **else**.

Следующий раздел кода демонстрирует использование оператора **if/then/elif/then/else** для запуска клиента **mysql**, если активна служба **mariadb**, запуска клиента **psql**, если активна служба **postgresql**, или запуска клиента **sqlite3**, если службы **mariadb** и **postgresql** не активны.

```
[user@host ~]$ systemctl is-active mariadb > /dev/null 2>&1
MARIADB_ACTIVE=$?
[user@host ~]$ sudo systemctl is-active postgresql > /dev/null 2>&1
POSTGRESQL_ACTIVE=$?
[user@host ~]$ if [ "$MARIADB_ACTIVE" -eq 0 ]; then
> mysql
> elif [ "$POSTGRESQL_ACTIVE" -eq 0 ]; then
> psql
> else
> sqlite3
> fi
```



РЕКОМЕНДАЦИИ

Справочные страницы **man bash(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

БОЛЕЕ ЭФФЕКТИВНОЕ ИСПОЛЬЗОВАНИЕ ЦИКЛОВ ДЛЯ ВЫПОЛНЕНИЯ КОМАНД

В этом упражнении вы будете использовать циклы для эффективной печати имени хоста с нескольких серверов.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность создать цикл `for` для перебора списка элементов из командной строки и в сценарии оболочки.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab console-commands start**. Команда запускает сценарий, который определяет, доступны ли в сети хосты **servera** и **serverb**. Сценарий предупредит вас, если они недоступны.

```
[student@workstation ~]$ lab console-commands start
```

1. Используйте команды **ssh** и **hostname** для вывода имени хоста **servera** и **serverb** на стандартный вывод.

```
[student@workstation ~]$ ssh student@servera hostname  
servera.lab.example.com  
[student@workstation ~]$ ssh student@serverb hostname  
serverb.lab.example.com
```

2. Создайте цикл **for** для более эффективного выполнения той же задачи.

```
[student@workstation ~]$ for HOST in servera serverb  
do  
ssh student@${HOST} hostname  
done  
servera.lab.example.com  
serverb.lab.example.com
```

3. Создайте сценарий оболочки для выполнения того же цикла **for**.

- 3.1. Создайте каталог **/home/student/bin**, в котором будет находиться сценарий оболочки.

```
[student@workstation ~]$ mkdir ~/bin
```

3.2. Убедитесь, что вновь созданный каталог находится в переменной окружения **PATH**.

```
[student@workstation ~]$ echo $PATH  
/home/student/.local/bin:/home/student/bin:/usr/local/bin:/usr/bin:/  
usr/local/sbin:/usr/sbin
```

3.3. Создайте сценарий оболочки в **/home/student/bin/printhostname.sh** для выполнения цикла **for**. Используйте команду **cat**, чтобы проверить содержимое **printhostname.sh**.

```
[student@workstation ~]$ vim ~/bin/printhostname.sh  
[student@workstation ~]$ cat ~/bin/printhostname.sh
```

```
#!/bin/bash  
#Execute for loop to print server hostname.  
for HOST in servera serverb  
do  
ssh student@$HOST hostname  
done  
exit 0
```

3.4. Убедитесь, что вновь созданный сценарий является исполняемым.

```
[student@workstation ~]$ chmod +x ~/bin/printhostname.sh
```

3.5. Запустите сценарий из домашнего каталога.

```
[student@workstation ~]$ printhostname.sh  
servera.lab.example.com  
serverb.lab.example.com
```

3.6. Убедитесь, что код выхода вашего скрипта равен 0.

```
[student@workstation ~]$ echo $?  
0
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab console-commands finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab console-commands finish
```

На этом пошаговое упражнение завершено.

СООТВЕТСТВИЕ ТЕКСТАМ В ВЫВОДЕ КОМАНДЫ С ОБЫЧНЫМИ ВЫРАЖЕНИЯМИ

ЗАДАЧИ

По завершении этого раздела учащиеся должны уметь:

- Создавайте регулярные выражения, соответствующие желаемым данным.
- Применяйте регулярные выражения к текстовым файлам с помощью команды **grep**.
- Поиск файлов и данных из команд, переправленных конвейером на вход команды **grep**.

НАПИСАНИЕ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ

Регулярные выражения предоставляют механизм сопоставления с образцом, который упрощает поиск определенного содержимого. Команды **vim**, **grep** и **less** могут использовать регулярные выражения. Все языки программирования, такие как **Perl**, **Python** и **C**, могут использовать регулярные выражения при использовании критериев сопоставления с образцом.

Регулярные выражения - это отдельный язык, а это значит, что у них есть собственный синтаксис и правила. В этом разделе рассматривается синтаксис, используемый при создании регулярных выражений, а также показаны некоторые примеры регулярных выражений.

Описание простого регулярного выражения

Самое простое регулярное выражение - точное совпадение. Точное совпадение - это когда символы в регулярном выражении соответствуют типу и порядку в данных, в которых выполняется поиск.

Предположим, пользователь просматривает следующий файл в поисках всех вхождений в соответствие с шаблоном **cat**:

```
cat
dog
concatenate
dogma
category
educated
boondoggle
vindication
chilidog
```

cat - это точное совпадение **c**, за которым следует **a**, за которым следует **t** без каких-либо других символов между ними. Использование **cat** в качестве регулярного выражения для поиска в предыдущем файле возвращает следующие совпадения:

```
cat
```

concatenate
category
educated
vindication

Соответствие начала и конца строки

В предыдущем разделе для файла использовалось регулярное выражение с точным соответствием. Обратите внимание, что регулярное выражение будет соответствовать строке поиска независимо от того, где оно встречается: в начале, конце или середине слова или строки. Используйте привязку к строке, чтобы указать, где регулярное выражение ищет совпадение.

Для поиска в начале строки используйте символ вставки (^). Для поиска в конце строки используйте знак доллара (\$).

Используя тот же файл, что и выше, регулярное выражение ^**cat** будет соответствовать двум словам. Регулярное выражение \$**cat** не найдет подходящих слов.

cat
dog
concatenate
dogma
category
educated
boondoggle
vindication
chilidog

Чтобы найти строки в файле, заканчивающиеся на **dog**, используйте это точное выражение и привязку конца строки для создания регулярного выражения **dog\$**. Применение выражение **dog\$** к файлу обнаружит два совпадения:

dog
chilidog

Чтобы найти единственное слово в строке, используйте привязки как в начале, так и в конце строки. Например, чтобы найти слово «**cat**», когда это единственное слово в строке, используйте выражение ^**cat\$**.

cat dog rabbit
cat
horse cat cow
cat pig

Добавление подстановочных знаков и множителей к регулярным выражениям

В регулярных выражениях точка (.) используются для соответствия любому одиночному символу, за исключением символа новой строки. Регулярное выражение **c.t** ищет строку, содержащую **c**, за которым следует любой одиночный символ, за которым следует **t**. Примеры совпадений включают **cat**, **concatenate**, **vindication**, **c5t** и **c\$t**.

Используя неограниченный подстановочный знак, вы не можете предсказать символ, который будет соответствовать подстановочному знаку. Чтобы соответствовать определенным символам, замените неограниченный подстановочный знак допустимыми символами. Изменение регулярного выражения на **c[aou]t** соответствует шаблонам, которые начинаются с **c**, за которым следует **a**, **o** или **u**, за которым следует **t**.

Другой тип множителя будет указывать количество предыдущих символов, желаемых в шаблоне. Примером использования явного множителя может быть '**c.\{2\}t**'. Это регулярное выражение будет соответствовать любому слову, начинающемуся с буквы **c**, за которым следуют ровно два символа и заканчиваются буквой **t**. '**c.\{2\}t**' соответствует двум словам в примере ниже:

```
cat
coat convert
cart covert
cypher
```



ПРИМЕЧАНИЕ

Рекомендуется использовать одинарные кавычки для инкапсуляции регулярного выражения, поскольку они часто содержат метасимволы оболочки (такие как \$, * и {}). Это гарантирует, что символы интерпретируются командой, а не оболочкой.



ПРИМЕЧАНИЕ

В этом курсе представлены две различные системы синтаксического анализа текста метасимволов: сопоставление с шаблоном (*pattern*) оболочки (shell) (также известное как подстановка файлов или расширение имени файла) и регулярные выражения (*regular expressions*). Поскольку обе системы используют одинаковые метасимволы, такие как звездочка (*), но имеют различия в интерпретации метасимволов и правилах, эти две системы могут сбивать с толку, пока каждая из них не будет освоена в достаточной степени.

Сопоставление с образцом (pattern) - это метод синтаксического анализа командной строки, предназначенный для простого указания многих имен файлов, и в основном он поддерживается только для представления образцов имен файлов в командной строке. **Регулярные выражения** предназначены для представления любой формы или шаблона в текстовых строках, независимо от их сложности. Регулярные выражения внутренне поддерживаются многочисленными командами обработки текста, такими как **grep**, **sed**, **awk**, **python**, **perl** и многими приложениями, с некоторыми минимальными вариациями в правилах интерпретации, зависящими от команд.

Регулярные выражения

ОПЦИЯ	ОПИСАНИЕ
.	Точка (.) Соответствует любому одиночному символу.
?	Предыдущий элемент является необязательным и будет сопоставлен не более одного раза.
*	Предыдущий элемент будет найден ноль или более раз.
+	Предыдущий элемент будет сопоставлен один или несколько раз.
{n}	Предыдущий элемент соответствует ровно n раз.
{n,}	Предыдущий элемент встречается n или более раз.
{, m}	Предыдущий элемент соответствует не более m раз.
{n, m}	Предыдущий элемент встречается не менее n раз, но не более m раз.
[:alnum:]	Буквенно-цифровые символы: '[:alpha:]' и '[:digit:]'; в языковом стандарте «C» и кодировке символов ASCII это то же самое, что и '[0-9A-Za-z]'.
[:alpha:]	Буквенные символы: '[:lower:]' и '[:upper:]'; в языковом стандарте «C» и кодировке символов ASCII это то же самое, что и '[A-Za-z]'.
[:blank:]	Пустые символы: пробел и табуляция.
[:cntrl:]	Управляющие персонажи. В ASCII эти символы имеют восьмеричные коды от 000 до 037 и 177 (DEL). В других наборах символов это эквивалентные символы, если таковые имеются.
[:digit:]	Цифры: 0 1 2 3 4 5 6 7 8 9.
[:graph:]	Графические символы: '[:alnum:]' и '[:punct:]'.
[:lower:]	Строчные буквы; в языковом стандарте С и кодировке символов ASCII это a b c d e f g h i j k l m n o p q r s t u v w x y z.
[:print:]	Печатные символы: '[:alnum:]', '[:punct:]' и пробел.
[:punct:]	Знаки препинания; в языковом стандарте 'C' и кодировке символов ASCII это так! "# \$% & '() * +, -. /; <=>? @ [\\] ^ _ { } ~. В других наборах символов это эквивалентные символы, если таковые имеются.
[:space:]	Пробелы: в языковом стандарте 'C' это табуляция, новая строка, вертикальная табуляция, подача формы, возврат каретки и пробел.
[:upper:]	Заглавные буквы: в языковом стандарте 'C' и кодировке символов ASCII это A B C D E F G H I J K L M N O P Q R S T U V W X Y Z.
[:xdigit:]	Шестнадцатеричные цифры: 0 1 2 3 4 5 6 7 8 9 A B C D E F a b c d e f.
b	Сопоставьте пустую строку на краю слова.
B	Сопоставьте пустую строку, если она не находится на краю слова.

<	Соответствует пустой строке в начале слова.
>	Сопоставьте пустую строку в конце слова.
w	Составное слово соотвествия. Синоним ' [:alnum:] '.
W	Соответствие несловесной составляющей. Синоним для ' ^_ [:alnum:] '.
s	Соответствие пустому пространству. Синоним для ' [[:space:]] '.
S	Соответствует не пробелам. Синоним для ' ^ [[:space:]] '.

ИСПОЛЬЗОВАНИЕ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ С КОМАНДОЙ GREP

Команда **grep**, предоставляемая как часть дистрибутива, использует регулярные выражения для нахождения совпадающих данных.

Нахождение данных с помощью команды grep

Команда **grep** указывает регулярное выражение и файл, в котором это регулярное выражение должно быть сопоставлено.

```
[user@host ~]$ grep '^computer' /usr/share/dict/words
computer
computerese
computerise
computerite
computerizable
computerization
computerize
computerized
computerizes
computerizing
computerlike
computernik
computers
```



ПРИМЕЧАНИЕ

Рекомендуется использовать одинарные кавычки для инкапсуляции регулярного выражения, поскольку они часто содержат метасимволы оболочки (такие как \$, * и {}). Это гарантирует, что символы интерпретируются **grep**, а не оболочкой shell.

Команду **grep** можно использовать вместе с другими командами с помощью оператора вертикальной черты ()).

Например:

```
[root@host ~]# ps aux | grep chrony
chrony      662  0.0  0.1  29440  2468 ?          S     10:56   0:00 /usr/sbin/chronyd
```

Параметры grep

У команды **grep** есть много полезных опций для настройки того, как она использует предоставленное регулярное выражение с данными.

Таблица общих параметров grep

ВАРИАНТ	НАЗНАЧЕНИЕ
-i	Используйте предоставленное регулярное выражение, но не применяйте чувствительность к регистру (без учета регистра).
-v	Отображать только строки, не содержащие совпадений с регулярным выражением.
-r	Рекурсивно применить поиск данных, соответствующих регулярному выражению, к группе файлов или каталогов.
-A NUMBER	Отобразить ЧИСЛО строк после совпадения регулярного выражения.
-B NUMBER	Отобразить ЧИСЛО строк до совпадения регулярного выражения.
-e	При использовании нескольких опций -e можно указать несколько регулярных выражений, которые будут использоваться с логическим ИЛИ (OR).

Есть много других вариантов использования **grep**. Используйте справочную страницу **man**, для их изучения.

Примеры использования команды grep

В следующих примерах используются различные файлы конфигурации и файлы журналов. По умолчанию в регулярных выражениях учитывается регистр. Используйте параметр **-i** с командой **grep**, чтобы выполнить поиск без учета регистра. В следующем примере выполняется поиск по шаблону **serverroot**.

```
[user@host ~]$ cat /etc/httpd/conf/httpd.conf
...output omitted...
ServerRoot "/etc/httpd"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
```

```
#  
# Change this to Listen on specific IP addresses as shown below to  
# prevent Apache from glomming onto all bound IP addresses.  
#  
#Listen 12.34.56.78:80  
Listen 80
```

```
[user@host ~]$ grep -i serverroot /etc/httpd/conf/httpd.conf  
# with "/", the value of ServerRoot is prepended -- so 'log/access_log'  
# with ServerRoot set to '/www' will be interpreted by the  
# ServerRoot: The top of the directory tree under which the server's  
# ServerRoot at a non-local disk, be sure to specify a local disk on the  
# same ServerRoot for multiple httpd daemons, you will need to change at  
ServerRoot "/etc/httpd"
```

В случаях, когда вы знаете, что не ищете, очень полезна опция **-v**. Параметр **-v** отображает только строки, не соответствующие регулярному выражению. В следующем примере возвращаются все строки, независимо от регистра, которые не содержат регулярного выражения **server**.

```
[user@host ~]$ cat /etc/hosts  
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  
  
172.25.254.254 classroom.example.com classroom  
172.25.254.254 content.example.com content  
172.25.254.254 materials.example.com materials  
172.25.250.254 workstation.lab.example.com workstation  
### rht-vm-hosts file listing the entries to be appended to /etc/hosts  
  
172.25.250.10 servera.lab.example.com servera  
172.25.250.11 serverb.lab.example.com serverb  
172.25.250.254 workstation.lab.example.com workstation
```

```
[user@host ~]$ grep -v -i server /etc/hosts  
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4  
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6  
  
172.25.254.254 classroom.example.com classroom  
172.25.254.254 content.example.com content  
172.25.254.254 materials.example.com materials  
172.25.250.254 workstation.lab.example.com workstation  
### rht-vm-hosts file listing the entries to be appended to /etc/hosts  
  
172.25.250.254 workstation.lab.example.com workstation
```

Чтобы просмотреть файл, не отвлекаясь на строки комментариев, используйте параметр **-v**. В следующем примере регулярное выражение соответствует всем строкам, начинающимся с символа # или ; (типичные символы, обозначающие строку, которая интерпретироваться как комментарий). Эти строки будут исключены из вывода текста.

```
[user@host ~]$ cat /etc/ethertypes
#
# Ethernet frame types
#      This file describes some of the various Ethernet
#      protocol types that are used on Ethernet networks.
#
# This list could be found on:
#          http://www.iana.org/assignments/ethernet-numbers
#          http://www.iana.org/assignments/ieee-802-numbers
#
# <name>    <hexnumber> <alias1>...<alias35> #Comment
#
IPv4        0800     ip ip4      # Internet IP (IPv4)
X25        0805
ARP         0806     ether-arp   #
FR_ARP      0808           # Frame Relay ARP          [RFC1701]
```

```
[user@host ~]$ grep -v '^#[;]' /etc/ethertypes
IPv4        0800     ip ip4      # Internet IP (IPv4)
X25        0805
ARP         0806     ether-arp   #
FR_ARP      0808           # Frame Relay ARP          [RFC1701]
```

Команда **grep** с параметром **-e** позволяет вам искать более одного регулярного выражения за раз. В следующем примере с использованием комбинации **less** и **grep** обнаруживаются все вхождения **pam_unix**, **user root** и **Accepted publickey** в файле журнала **/var/log/secure**.

```
[root@host ~]# cat /var/log/secure | grep -e 'pam_unix' \
-e 'user root' -e 'Accepted publickey' | less
Mar 19 08:04:46 jegui sshd[6141]: pam_unix(sshd:session): session opened for user
root by (uid=0)
Mar 19 08:04:50 jegui sshd[6144]: Disconnected from user root 172.25.250.254 port
41170
Mar 19 08:04:50 jegui sshd[6141]: pam_unix(sshd:session): session closed for user
root
Mar 19 08:04:53 jegui sshd[6168]: Accepted publickey for student from
172.25.250.254 port 41172 ssh2: RSA SHA256:M8ikhcEDm2tQ95Z0o7ZvufqEixCFCT
+wowZLNzNlBT0
```

Для поиска текста в файле, открытом с использованием **vim** или **less**, используйте символ косой черты (/) после введите шаблон для поиска. Нажмите **Enter**, чтобы начать поиск. После нахождения первого соответствия, нажмите **N**, чтобы найти следующее совпадение.

```
[root@host ~]# vim /var/log/boot.log
...output omitted...
[^[[[0;32m OK ^[[0m] Reached target Initrd Default Target.^M
Starting dracut pre-pivot and cleanup hook...^M
[^[[[0;32m OK ^[[0m] Started dracut pre-pivot and cleanup hook.^M
Starting Cleaning Up and Shutting Down Daemons...^M
Starting Plymouth switch root service...^M
Starting Setup Virtual Console...^M
[^[[[0;32m OK ^[[0m] Stopped target Timers.^M
[^[[[0;32m OK ^[[0m] Stopped dracut pre-pivot and cleanup hook.^M
[^[[[0;32m OK ^[[0m] Stopped target Initrd Default Target.^M
```

```
[root@host ~]# less /var/log/messages
...output omitted...
Feb 26 15:51:07 jegui NetworkManager[689]: <info> [1551214267.8584] Loaded device
plugin: NMTeamFactory (/usr/lib64/NetworkManager/1.14.0-14.el8/libnm-device-
plugin-team.so)
Feb 26 15:51:07 jegui NetworkManager[689]: <info> [1551214267.8599] device (lo):
carrier: link connected
Feb 26 15:51:07 jegui NetworkManager[689]: <info> [1551214267.8600] manager:
(lo): new Generic device (/org/freedesktop/NetworkManager/Devices/1)
Feb 26 15:51:07 jegui NetworkManager[689]: <info> [1551214267.8623] manager:
(ens3): new Ethernet device (/org/freedesktop/NetworkManager/Devices/2)
Feb 26 15:51:07 jegui NetworkManager[689]: <info> [1551214267.8653] device
(ens3): state change: unmanaged -> unavailable (reason 'managed', sys-iface-
state: 'external')
/device
```



РЕКОМЕНДАЦИИ

Справочные страницы **man regex(7)** и **grep(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ИСПОЛЬЗОВАНИЯ РЕГУЛЯРНОГО ВЫРАЖЕНИЯ ДЛЯ НАХОЖДЕНИЯ СООТВЕТСТВИЯ ТЕКСТАМ В ВЫВОДЕ КОМАНДЫ.

В этой лабораторной работе вы будете искать текст в системных журналах и выводе команд, используя более эффективные способы.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность эффективно искать текст в файлах журнала и файлах конфигурации.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab console-regex**. Эта команда запускает сценарий, который определяет, доступна ли машина **servera** в сети. Он также устанавливает пакет **postfix**.

```
[student@workstation ~]$ lab console-regex start
```

1. Используйте команду **ssh** для входа на сервер **servera**, как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Пароль для пользователя **student** слово **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

3. Пакет **postfix** был установлен стартовым скриптом. Используйте команду **grep**, чтобы найти **GID** и **UID** для групп и пользователей **postfix** и **postdrop**. Чтобы уменьшить вывод команды **grep**, отобразите все журналы с определенным временем начала.

- 3.1. Используйте команду **date**, чтобы определить текущее время.

```
[root@servera ~]# date  
Fri Mar 22 08:23:56 CET 2019
```

- 3.2. Используйте команду **grep** с параметрами даты, времени начала и **GID**, чтобы найти **GID** и **UID** пользователя **postfix** и **postdrop**. Сценарий настройки лаборатории был запущен на несколько минут раньше текущего времени. Учтите это при поиске в файле журнала **/var/log/secure**.

```
[root@servera ~]# grep '^Mar 22 08:2.*GID' /var/log/secure  
Mar 22 08:20:04 servera groupadd[2514]: group added to /etc/  
group: name=postdrop, GID=90  
Mar 22 08:20:04 servera groupadd[2514]: new group: name=postdrop,  
GID=90  
Mar 22 08:20:04 servera groupadd[2520]: group added to /etc/  
group: name=postfix, GID=89  
Mar 22 08:20:04 servera groupadd[2520]: new group: name=postfix,  
GID=89  
Mar 22 08:20:04 servera useradd[2527]: new user: name=postfix, UID=89,  
GID=89, home=/var/spool/postfix, shell=/sbin/nologin
```

4. Измените регулярное выражение, чтобы найти первые два сообщения в файле **/var/log/maillog**. Обратите внимание, что в этом поиске вы не используете символ вставки (^), потому что вы не ищете первый символ в строке.

```
[root@servera ~]# grep 'postfix' /var/log/maillog | head -n 2  
Mar 22 08:21:02 servera postfix/postfix-script[3879]: starting the Postfix  
mail system  
Mar 22 08:21:02 servera postfix/master[3881]: daemon started -- version  
3.3.1, configuration /etc/postfix
```

5. Вам необходимо найти имя каталога **queue** для сервера **Postfix**. Найдите в файле конфигурации **/etc/postfix/main.cf** всю информацию об очередях (**queue**). Используйте параметр **-i**, чтобы игнорировать различия в регистре.

```
[root@servera ~]# grep -i 'queue' /etc/postfix/main.cf  
# testing. When soft_bounce is enabled, mail will remain queued that  
# The queue_directory specifies the location of the Postfix queue.  
queue_directory = /var/spool/postfix  
# QUEUE AND PROCESS OWNERSHIP  
# The mail_owner parameter specifies the owner of the Postfix queue  
# is the Sendmail-compatible mail queue listing command.  
# setgid_group: The group for mail submission and queue management
```

6. Убедитесь, что **postfix** записывает сообщения в **/var/log/messages**. Используйте команду **less**, а затем косую черту (/) для поиска в файле. Нажмите **n**, чтобы перейти к следующей записи, соответствующей поиску. Используйте клавишу **q**, чтобы выйти из команды **less**.

```
[root@servera ~]# less /var/log/messages
...output omitted...
Mar 22 07:58:04 servera systemd[1]: Started Postfix Mail Transport Agent.
...output omitted...
Mar 22 08:12:26 servera systemd[1]: Stopping Postfix Mail Transport Agent...
Mar 22 08:12:26 servera systemd[1]: Stopped Postfix Mail Transport Agent.
...output omitted...
/Postfix
```

7. Используйте команду **ps aux**, чтобы убедиться, что сервер **postfix** в настоящее время запущен. Уменьшите вывод **ps aux**, объединив его с командой **grep**.

```
[root@servera ~]# ps aux | grep postfix
root      3881  0.0  0.2 121664  5364 ?        Ss   08:21   0:00 /usr/
libexec/postfix/master -w
postfix    3882  0.0  0.4 147284  9088 ?        S    08:21   0:00 pickup -l -
t unix -u
postfix    3883  0.0  0.4 147336  9124 ?        S    08:21   0:00 qmgr -l -t
unix -u
```

8. Убедитесь, что очереди **qmgr**, **cleanup** и **pickup** правильно настроены. Используйте команду **grep** с параметром **-e**, чтобы сопоставить несколько записей в одном файле. Файл конфигурации **- /etc/postfix/master.cf**

```
[root@servera ~]# grep -e qmgr -e pickup -e cleanup /etc/postfix/master.cf
pickup    unix n      -      n      60      1      pickup
cleanup   unix n      -      n      -      0      cleanup
qmgr      unix n      -      n      300     1      qmgr
#qmgr    unix n      -      n      300     1      oqmgr
```

9. Выйдите из сервера **servera**.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Окончание

На рабочей станции **workstation** запустите скрипт **lab console-regex finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab console-regex finish
```

На этом пошаговое упражнение завершено.

ЛАБОРАТОРНАЯ РАБОТА

ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ КОМАНДНОЙ СТРОКИ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы создадите сценарий **Bash**, который может фильтровать и получать соответствующую информацию с разных хостов.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создать сценарий **Bash** и перенаправьте его вывод в файл.
- Используйте циклы для упрощения кода.
- Отфильтруйте соответствующий контент с помощью команды **grep** и регулярных выражений.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab console-review start**. Эта команда запускает сценарий, который определяет, доступны ли в сети рабочая станция **workstation**, серверы **servera** и **serverb**. Сценарий предупредит вас, если они недоступны. Он также при необходимости устанавливает пакеты **vim-Enhanced** и **util-linux**, настраивает **sudo** и подготавливает содержимое **/var/log/secure** на **servera** и **serverb**.

```
[student@workstation ~]$ lab console-review start
```

1. Создайте файл сценария **/home/student/bin/bash-lab** на рабочей станции **workstation**.
2. Отредактируйте только что созданный файл сценария, чтобы он соответствовал следующей запрошенной информации от серверов **servera** и **serverb**. Системы настроены на использование ключей **SSH** для аутентификации; пароль не требуется.

КОМАНДА ИЛИ ФАЙЛ	ЗАПРАШИВАЕМОЕ СОДЕРЖИМОЕ
hostname -f	Получить все выходные данные.
echo "#####"	Получить все выходные данные.
lscpu	Получить только те строки, которые начинаются со строки CPU.
echo "#####"	Получить все выходные данные.
/etc/selinux/config	Игнорируйте пустые строки. Игнорируйте строки, начинающиеся с #.
echo "#####"	Получить все выходные данные.

/var/log/secure	Получить все записи «Неудачный пароль (Failed password)».
echo "#####"	Получить все выходные данные.



ПРИМЕЧАНИЕ

Вы можете использовать **sudo**, не требуя пароля на хостах servera и serverb. Не забудьте использовать цикл, чтобы упростить сценарий. Вы также можете использовать несколько команд **grep**, объединенных с помощью вертикальной черты (|).

3. Выполните сценарий **/home/student/bin/bash-lab** и просмотрите выходной контент на рабочей станции.

Оценка

На рабочей станции **workstation** запустите команду **lab console-review grade**, чтобы подтвердить успех выполнения данного упражнения.

```
[student@workstation ~]$ lab console-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab console-review finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab console-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ КОМАНДНОЙ СТРОКИ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы создадите сценарий **Bash**, который может фильтровать и получать соответствующую информацию с разных хостов.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создать сценарий **Bash** и перенаправьте его вывод в файл.
- Используйте циклы для упрощения кода.
- Отфильтруйте соответствующий контент с помощью команды **grep** и регулярных выражений.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab console-review start**. Эта команда запускает сценарий, который определяет, доступны ли в сети рабочая станция **workstation**, серверы **servera** и **serverb**. Сценарий предупредит вас, если они недоступны. Он также при необходимости устанавливает пакеты **vim-Enhanced** и **util-linux**, настраивает **sudo** и подготавливает содержимое **/var/log/secure** на **servera** и **serverb**.

```
[student@workstation ~]$ lab console-review start
```

1. Создайте файл сценария **/home/student/bin/bash-lab** на рабочей станции **workstation**.

1.1. На рабочей станции **workstation** при необходимости создайте папку **/home/student/bin/**.

```
[student@workstation ~]$ mkdir -p /home/student/bin
```

1.2. Используйте **vim** для создания и редактирования файла сценария **/home/student/bin/bash-lab**.

```
[student@workstation ~]$ vim ~/bin/bash-lab
```

1.3. Вставьте следующий текст и сохраните файл.

```
#!/bin/bash
```

1.4. Сделайте свой файл сценария исполняемым.

```
[student@workstation ~]$ chmod a+x ~/bin/bash-lab
```

2. Отредактируйте только что созданный файл сценария, чтобы он соответствовал следующей запрошенной информации от серверов **servera** и **serverb**. Системы настроены на использование ключей **SSH** для аутентификации; пароль не требуется.

КОМАНДА ИЛИ ФАЙЛ	ЗАПРАШИВАЕМОЕ СОДЕРЖИМОЕ
hostname -f	Получить все выходные данные.
echo "#####"	Получить все выходные данные.
lscpu	Получить только те строки, которые начинаются со строки CPU .
echo "#####"	Получить все выходные данные.
/etc/selinux/config	Игнорируйте пустые строки. Игнорируйте строки, начинающиеся с #.
echo "#####"	Получить все выходные данные.
/var/log/secure	Получить все записи «Неудачный пароль (Failed password)».
echo "#####"	Получить все выходные данные.



ПРИМЕЧАНИЕ

Вы можете использовать **sudo**, не требуя пароля на хостах **servera** и **serverb**. Не забудьте использовать цикл, чтобы упростить сценарий. Вы также можете использовать несколько команд **grep**, объединенных с помощью вертикальной черты ()).

2.1. Используйте **vim**, чтобы открыть и отредактировать файл сценария **/home/student/bin/bash-lab**.

```
[student@workstation ~]$ vim ~/bin/bash-lab
```

2.2. Добавьте следующие строки полужирным шрифтом в файл **/home/student/bin/bash-lab**.



ПРИМЕЧАНИЕ

Ниже приведен пример того, как можно выполнить запрошенный сценарий. В сценариях Bash вы можете использовать разные подходы и получить тот же результат.

```

#!/bin/bash
#
USR='student'
OUT='/home/student/output'
#
for SRV in servera serverb
do
ssh ${USR}@${SRV} "hostname -f" > ${OUT}-${SRV}
echo ##### >> ${OUT}-${SRV}
ssh ${USR}@${SRV} "lscpu | grep '^CPU'" >> ${OUT}-${SRV}
echo ##### >> ${OUT}-${SRV}
ssh ${USR}@${SRV} "grep -v '^$' /etc/selinux/config|grep -v '^#' >>
${OUT}-${SRV}"
echo ##### >> ${OUT}-${SRV}
ssh ${USR}@${SRV} "sudo grep 'Failed password' /var/log/secure" >>
${OUT}-${SRV}
echo ##### >> ${OUT}-${SRV}
done

```

3. Выполните сценарий **/home/student/bin/bash-lab** и просмотрите выходной контент на рабочей станции **workstation**.

3.1. На рабочей станции **workstation** выполните сценарий **/home/student/bin/bash-lab**.

```
[student@workstation ~]$ bash-lab
```

3.2. Просмотрите содержимое **/home/student/output-servera** и **/home/student/output-serverb**.

```
[student@workstation ~]$ cat /home/student/output-servera
servera.lab.example.com
#####
CPU op-mode(s):      32-bit, 64-bit
CPU(s):              2
CPU family:          21
CPU MHz:             2294.670
#####
SELINUX=enforcing
SELINUXTYPE=targeted
#####
Mar 21 22:30:28 servera sshd[3939]: Failed password for invalid user
operator1 from 172.25.250.9 port 58382 ssh2
Mar 21 22:30:31 servera sshd[3951]: Failed password for invalid user
sysadmin1 from 172.25.250.9 port 58384 ssh2
Mar 21 22:30:34 servera sshd[3953]: Failed password for invalid user
manager1 from 172.25.250.9 port 58386 ssh2
#####

```

```
[student@workstation ~]$ cat /home/student/output-serverb
serverb.lab.example.com
#####
CPU op-mode(s):      32-bit, 64-bit
CPU(s):              2
CPU family:          6
CPU MHz:             2294.664
#####
SELINUX=enforcing
SELINUXTYPE=targeted
#####
Mar 21 22:30:37 serverb sshd[3883]: Failed password for invalid user
operator1 from 172.25.250.9 port 39008 ssh2
Mar 21 22:30:39 serverb sshd[3891]: Failed password for invalid user
sysadmin1 from 172.25.250.9 port 39010 ssh2
Mar 21 22:30:43 serverb sshd[3893]: Failed password for invalid user
manager1 from 172.25.250.9 port 39012 ssh2
#####
```

Оценка

На рабочей станции **workstation** запустите команду **lab console-review grade**, чтобы подтвердить успех выполнения данного упражнения.

```
[student@workstation ~]$ lab console-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab console-review finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab console-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- Как создавать и выполнять простые сценарии **Bash**.
- Как использовать циклы для перебора списка элементов из командной строки и в сценарии оболочки.
- Как искать текст в файлах журналов и конфигурационных файлах с помощью регулярных выражений и **grep**.

ГЛАВА 2

ПЛАНИРОВАНИЕ БУДУЩИХ ЗАДАЧ

ЦЕЛЬ

Планируйте задачи для автоматического выполнения в будущем.

ЗАДАЧИ

- Настройте команду, которая запускается один раз в будущем.
- Запланировать выполнение команд по повторяющемуся расписанию с использованием пользовательского файла **crontab**.
- Запланировать выполнение команд по повторяющемуся расписанию с использованием системного файла **crontab** и каталогов.
- Включение и отключение таймеров **systemd** и настройка таймера для управления временными файлами.

РАЗДЕЛЫ

- Планирование отложенного пользовательского задания (и упражнения с пошаговыми инструкциями)
- Планирование повторяющихся пользовательских заданий (и упражнения с пошаговыми инструкциями)
- Планирование повторяющихся системных заданий (и упражнения с пошаговыми инструкциями)
- Управление временными файлами (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Планирование будущих задач.

ПЛАНИРОВАНИЕ ОТЛОЖЕННОГО ЗАДАНИЯ ПОЛЬЗОВАТЕЛЯ

ЗАДАЧИ

После завершения этого раздела вы сможете настроить команду, которая будет выполняться один раз в какой-то момент в будущем.

ОПИСАНИЕ ОТЛОЖЕННЫХ ЗАДАЧ ПОЛЬЗОВАТЕЛЯ

Иногда вам может потребоваться запустить команду или набор команд в заданной точке в будущем. Примеры включают людей, которые хотят запланировать электронное письмо своему боссу, или системного администратора, работающего над конфигурацией брандмауэра, который выполняет «безопасную» работу по сбросу настроек брандмауэра через десять минут, если они не деактивируют это задание заранее.

Эти запланированные команды часто называют задачами или заданиями, а термин отложенный указывает, что эти задачи или задания будут выполняться в будущем.

Одно из решений, доступных пользователям Red Hat Enterprise Linux для планирования отложенных задач, это **at**. Пакет **at** предоставляет системный демон (**atd**) вместе с набором инструментов командной строки для взаимодействия с демоном (**at**, **atq** и т.д.). В стандартной установке Red Hat Enterprise Linux демон **atd** устанавливается и включается автоматически.

Пользователи (включая root) могут ставить в очередь задания для демона **atd** с помощью команды **at**. Демон **atd** предоставляет 26 очередей, от **a** до **z**, причем задания в более поздних по алфавиту очередях получают более низкий системный приоритет (более высокие значения **nice**, обсуждаемые в следующей главе).

Планирование отложенных пользовательских задач

Используйте команду **at TIMESPEC**, чтобы запланировать новое задание. Затем команда **at** считывает команды для выполнения из канала стандартного ввода. При ручном вводе команды вы можете завершить ввод, нажав **Ctrl+D**. Для более сложных команд, подверженных типографским ошибкам, часто проще использовать перенаправление ввода из файла сценария, например, **at now +5min <myscript**, чем вводить все команды вручную в окне терминала.

Аргумент **TIMESPEC** с командой **at** принимает множество мощных комбинаций, позволяя пользователям точно описывать, когда должно выполняться задание. Как правило, они начинаются с времени, например **02:00pm**, **15:59** или ещё **teatime**, за которым следует необязательная дата или количество дней в будущем. Ниже перечислены некоторые примеры комбинаций, которые можно использовать.

- **now +5min**
- **teatime tomorrow (teatime is 16:00)**
- **noon +4 days**
- **5pm august 3 2021**

Указание интерпретатора команд

Для получения полного списка допустимых спецификаций времени обратитесь к **timespec**, как указано в справочных материалах.

ИНСПЕКТИРОВАНИЕ И УПРАВЛЕНИЕ ОТЛОЖЕННЫМ ПОЛЬЗОВАТЕЛЕМ ЗАДАНИЙ

Для обзора отложенных заданий текущего пользователя, используйте команду **atq** или **at -l**.

```
[user@host ~]$ atq
❶ 28 ❷ Mon Feb  2 05:13:00 2015 ❸ a ❹ user
29 Mon Feb  3 16:00:00 2014 h user
27 Tue Feb  4 12:00:00 2014 a user
```

В представленных выше выходных данных каждая строка представляет отдельное задание, которое планируется запустить в будущем.

- ❶ Уникальный номер задания для этого задания.
- ❷ Дата и время выполнения запланированного задания.
- ❸ Указывает, что задание запланировано с очередью по умолчанию **a**. Различные задания могут быть запланированы с разными очередями.
- ❹ Владелец задания (и пользователь, от имени которого будет запущено задание).



ВАЖНО

Непrivилегированные пользователи могут видеть и контролировать только свои рабочие задачи. Пользователь **root** может видеть все задания и управлять ими.

Чтобы проверить фактические команды, которые будут выполняться при выполнении задания, используйте команду **at -c *JOBNUMBER***. Команда показывает среду для настраиваемого задания, отражающую среду пользователя, создавшего задание во время его создания, с последующими фактическими командами, которые нужно запустить.

Удаление заданий

Команда **atrm *JOBNUMBER*** удаляет запланированное задание. Удалите запланированное задание, когда оно больше не нужно, например, когда удаленная конфигурация брандмауэра прошла успешно и не требует сброса.



РЕКОМЕНДАЦИИ

Справочные страницы **man at(1)** и **atd(8)**
/usr/share/doc/at/timespec

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ПЛАНИРОВАНИЕ ОТЛОЖЕННОГО ЗАДАНИЯ ПОЛЬЗОВАТЕЛЯ

В этом упражнении вы будете использовать команду at, чтобы запланировать выполнение нескольких команд в указанное время в будущем.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Запланируйте запуск задания в указанное время в будущем.
- Проверьте команды, запускаемые запланированным заданием.
- Удалить запланированные задания.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию как студент, используя в качестве пароля студент.

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab scheduling-at start**, чтобы начать упражнение. Данный сценарий гарантирует, что среда чистая и правильно настроена.

```
[student@workstation ~]$ lab scheduling-at start
```

1. С рабочей станции **workstation** откройте сеанс SSH на сервер **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Запланируйте запуск задания через три минуты с помощью команды **at**. Задание должно сохранить вывод команды **date** в **/home/student/myjob.txt**.

- 2.1. Используйте команду **echo**, чтобы передать строку **date >> /home/student/myjob.txt** в качестве входных данных для команды **at**, чтобы задание запускалось через три минуты.

```
[student@servera ~]$ echo "date >> /home/student/myjob.txt" | at now  
+3min  
warning: commands will be executed using /bin/sh  
job 1 at Thu Mar 21 12:30:00 2019
```

- 2.2. Используйте команду **atq** для вывода списка запланированных заданий.

```
[student@servera ~]$ atq  
1 Thu Mar 21 12:30:00 2019 a student
```

- 2.3. Используйте команду **watch atq**, чтобы контролировать очередь отложенных заданий в реальном времени. После выполнения задание удаляется из очереди.

```
[student@servera ~]$ watch atq  
Every 2.0s: atq      servera.lab.example.com: Thu Mar 21 12:30:00  
2019  
  
1 Thu Mar 21 12:30:00 2019 a student
```

Предыдущая команда **watch** по умолчанию обновляет вывод **atq** каждые две секунды. После удаления отложенного задания из очереди нажмите **Ctrl+c**, чтобы выйти из **watch** и вернуться в командную строку оболочки.

- 2.4. Используйте команду **cat**, чтобы убедиться, что содержимое **/home/student/myjob.txt** соответствует выходным данным команды **date**.

```
[student@servera ~]$ cat myjob.txt  
Thu Mar 21 12:30:00 IST 2019
```

Предыдущий вывод совпадает с выводом команды **date**, подтверждая, что запланированное задание выполнено успешно.

3. Используйте команду **at** для интерактивного планирования задания с очередью **g, teatime** (16:00). Задание должно выполнить команду, которая выводит сообщение «**It's teatime**» в **/home/student/tea.txt**. Новые сообщения должны быть добавлены в файл **/home/student/tea.txt**.

```
[student@servera ~]$ at -q g teatime  
warning: commands will be executed using /bin/sh  
at> echo "It's teatime" >> /home/student/tea.txt  
at> Ctrl+d  
job 2 at Thu Mar 21 16:00:00 2019
```

4. Используйте команду **at** для интерактивного планирования другого задания с очередью **b**, которая запускается в **16:05**. Задание должно выполнить команду, которая печатает сообщение **The cookies are good** в **/home/student/cookies.txt**. Новые сообщения должны быть добавлены в файл **/home/student/cookies.txt**.

```
[student@servera ~]$ at -q b 16:05  
warning: commands will be executed using /bin/sh
```

```
at> echo "The cookies are good" >> /home/student/cookies.txt
at> Ctrl+d
job 3 at Thu Mar 21 16:05:00 2019
```

5. Проверьте команды в ожидающих заданиях.

5.1. Используйте команду **atq** для просмотра номеров ожидающих заданий.

```
[student@servera ~]$ atq
2 Thu Mar 21 16:00:00 2019 g student
3 Thu Mar 21 16:05:00 2019 b student
```

Обратите внимание на номера заданий в предыдущих выходных данных. Эти номера вакансий могут отличаться в зависимости от вашей системы.

5.2. Используйте команду **at**, чтобы просмотреть команды в незавершенном задании номер **2**.

```
[student@servera ~]$ at -c 2
...output omitted...
echo "It's teatime" >> /home/student/tea.txt
marcinDELIMITER28d54caa
```

Обратите внимание, что предыдущее запланированное задание выполняет команду **echo**, которая добавляет сообщение «**It's teatime**» в **/home/student/tea.txt**.

5.3. Используйте команду **at**, чтобы просмотреть команды в незавершенном задании номер **3**.

```
[student@servera ~]$ at -c 3
...output omitted...
echo "The cookies are good" >> /home/student/cookies.txt
marcinDELIMITER1d2b47e9
```

Обратите внимание, что предыдущее запланированное задание выполняет команду **echo**, которая добавляет сообщение **The cookies are good** в **/home/student/cookies.txt**.

6. Используйте команду **atq**, чтобы просмотреть номер задания, **teatime** (16:00), и удалите его с помощью команды **atrm**.

```
[student@servera ~]$ atq
2 Thu Mar 21 16:00:00 2019 g student
3 Thu Mar 21 16:05:00 2019 b student
[student@servera ~]$ atrm 2
```

7. Убедитесь, что задание, запланированное как **teatime** в (16:00), больше не существует.

7.1. Используйте команду **atq**, чтобы просмотреть список ожидающих заданий и убедиться, что задание, запланированное для запуска как **teatime** (16:00), больше не существует.

```
[student@servera ~]$ atq  
3 Thu Mar 21 16:05:00 2019 b student
```

7.2. Выйти из сервера **servera**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation** выполните скрипт **lab scheduling-at finish**, чтобы завершить упражнение. Сценарий удаляет файлы, созданные в ходе упражнения, и обеспечивает чистоту среды.

```
[student@workstation ~]$ lab scheduling-at finish
```

На этом пошаговое упражнение завершено.

ПЛАНИРОВАНИЕ ПОВТОРЯЮЩИХСЯ РАБОТ ПОЛЬЗОВАТЕЛЯ

ЗАДАЧИ

После завершения этого раздела вы сможете запланировать выполнение команд по повторяющемуся расписанию, используя файл пользователя **crontab**.

ОПИСАНИЕ ПОВТОРНЫХ ЗАДАНИЙ ПОЛЬЗОВАТЕЛЯ

Задания, запланированные к повторному запуску, называются повторяющимися заданиями. Системы Red Hat Enterprise Linux поставляются с демоном **crond**, предоставляемым пакетом **cronie**, который по умолчанию включен и запускается специально для повторяющихся заданий. Демон **crond** считывает несколько файлов конфигурации: по одному для каждого пользователя (редактируется с помощью команды **crontab**) и набор общесистемных файлов. Эти файлы конфигурации предоставляют пользователям и администраторам точный контроль над тем, когда должны выполняться их повторяющиеся задания.

Если запланированная команда производит какой-либо вывод или ошибку, которая не перенаправляется, демон **crond** пытается отправить этот вывод или ошибку по электронной почте пользователю, которому принадлежит это задание (если оно не отменено), используя почтовый сервер, настроенный в системе. В зависимости от среды может потребоваться дополнительная настройка. Вывод или ошибка запланированной команды могут быть перенаправлены в разные файлы.

ПЛАНИРОВАНИЕ ПОВТОРЯЮЩИХСЯ РАБОТ ПОЛЬЗОВАТЕЛЯ

Обычные пользователи могут использовать команду **crontab** для управления своими заданиями. Эту команду можно вызвать четырьмя разными способами:

Примеры crontab

КОМАНДА	НАЗНАЧЕНИЕ
crontab -l	Список заданий для текущего пользователя.
crontab -r	Удалить все задания для текущего пользователя.
crontab -e	Редактировать задания для текущего пользователя.
crontab <i>filename</i>	Удалите все задания и замените заданиями, прочитанными из файла. Если файл не указан, используется стандартный ввод.



ПРИМЕЧАНИЕ

Суперпользователь может использовать параметр **-u** с командой **crontab** для управления заданиями другого пользователя. Не следует использовать команду **crontab** для управления системными заданиями; вместо этого используйте методы, описанные в следующем разделе.

ОПИСАНИЕ ФОРМАТА ЗАДАНИЯ ПОЛЬЗОВАТЕЛЯ

Команда **crontab -e** по умолчанию вызывает **Vim**, если для переменной среды EDITOR не установлено иное значение. Введите по одному заданию в строку. Другие допустимые записи включают: пустые строки, обычно для удобства чтения; комментарии, обозначенные строками, начинающимися со знака числа (#); а также переменные среды, использующие формат **NAME=value**, который влияет на все строки ниже строки, в которой они объявлены. Общие настройки переменных включают переменную **SHELL**, которая объявляет, какую оболочку использовать для интерпретации оставшихся строк файла **crontab**; и переменная **MAILTO**, которая определяет, кто должен получать любой вывод по электронной почте.



ВАЖНО

Для отправки электронной почты может потребоваться дополнительная настройка локального почтового сервера или ретрансляции **SMTP** в системе.

Поля в файле **crontab** отображаются в следующем порядке:

- Minutes (Минуты)
- Hours (Часы)
- Day of month (День месяца)
- Month (Месяц)
- Day of week (День недели)
- Command (Команда)



ВАЖНО

Если оба поля «День месяца» и «День недели» отличны от *, команда выполняется, когда любое из этих двух полей удовлетворяется. Например, чтобы запускать команду **15 числа каждого месяца и каждую пятницу в 12:15**, используйте следующий формат задания:

```
15 12 15 * Fri command
```

Первые пять полей используют одни и те же правила синтаксиса:

- для «Не обращайте внимания»/всегда.
- Число для указания количества минут или часов, даты или дня недели. Для будних дней 0 означает воскресенье, 1 - понедельник, 2 - вторник и т. д. 7 также равно воскресенью.
- **x-y** для диапазона, от x до у включительно.
- **x,y** для списков. Списки также могут включать диапазоны, например 5,10–13,17 в столбце «Минуты», чтобы указать, что задание должно выполняться через 5, 10, 11, 12, 13 и 17 минут после часа.
- ***/x** для обозначения интервала x, например, ***/7** в столбце «Минуты» запускает задание каждые семь минут.

Кроме того, трехбуквенные сокращения на английском языке можно использовать как для месяцев, так и для будних дней, например, **Jan**, **Feb**, **b** и **Mon**, **Tue**.

Последнее поле содержит команду для выполнения с использованием оболочки по умолчанию. Переменная среды **SHELL** может использоваться для изменения оболочки для запланированной команды. Если команда содержит неэкранированный знак процента (%), то этот знак процента рассматривается как символ новой строки, и все, что находится после знака процента, передается команде на стандартном вводе (**stdin**).

Пример повторяющихся пользовательских заданий

В этом разделе описаны некоторые примеры повторяющихся заданий.

- Следующее задание выполняет команду **/usr/local/bin/yearly_backup** ровно в **9** часов утра 2 февраля каждого года.

```
0 9 2 2 * /usr/local/bin/yearly_backup
```

- Следующее задание отправляет электронное письмо со словом **Chime** владельцу этого задания каждые пять минут с 9:00 до 17:00 каждую пятницу июля.

```
*/5 9-16 * Jul 5 echo "Chime"
```

Предыдущий диапазон часов с **9-16** означает, что таймер задания начинается с девятого часа (09:00) и продолжается до конца шестнадцатого часа (16:59). Задание начинает выполняться в **09:00** с последним выполнением в **16:55**, потому что пять минут с **16:55 - 17:00**, что выходит за рамки указанного диапазона часов.

```
58 23 * * 1-5 /usr/local/bin/daily_report
```

- Следующее задание запускает команду **/usr/local/bin/daily_report** каждый будний день за две минуты до полуночи.

```
58 23 * * 1-5 /usr/local/bin/daily_report
```

- Следующее задание выполняет команду mutt для отправки почтового сообщения Check in получателю **boss@example.com** каждый рабочий день (с понедельника по пятницу) в 9:00.

```
0 9 * * 1-5 mutt -s "Checking in" boss@example.com % Hi there boss, just  
checking in.
```



РЕКОМЕНДАЦИИ

Справочные страницы **crond(8)**, **crontab(1)** и **crontab(5)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ПЛАНИРОВАНИЕ ПОВТОРЯЮЩИХСЯ РАБОТ ПОЛЬЗОВАТЕЛЯ

В этом упражнении вы запланируете выполнение команд по повторяющемуся расписанию от имени непrivилегированного пользователя, используя команду **crontab**.

В РЕЗУЛЬТАТЕ

Вы должны быть способен:

- Запланируйте повторяющиеся задания для запуска от имени непrivилегированного пользователя.
- Проверьте команды, запускаемые запланированным повторяющимся заданием.
- Удалить запланированные повторяющиеся задания.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab net-configure start**, чтобы начать упражнение. Этот сценарий гарантирует, что среда чистая и правильно настроена.

```
[student@workstation ~]$ lab scheduling-cron start
```

1. С рабочей станции **workstation** откройте сеанс **SSH** на сервер **servera**, как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Запланируйте повторяющуюся работу в качестве пользователя **student**, которая добавляет текущую дату и время в **/home/student/my_first_cron_job.txt** каждые две минуты с 8:00 до 21:00. Задание должно выполняться только с понедельника по пятницу, а не в субботу или воскресенье.



ВАЖНО

Если вы работаете в этой лабораторной работе вне дня и времени, указанных в предыдущей инструкции, вам следует соответствующим образом настроить системное время и/или дату, чтобы задание выполнялось во время работы.

- 2.1. Используйте команду **crontab -e**, чтобы открыть **crontab** с помощью текстового редактора по умолчанию.

```
[student@servera ~]$ crontab -e
```

- 2.2. Вставьте следующую строку.

```
*/2 08-20 * * Mon-Fri /usr/bin/date >> /home/student/my_first_cron_job.txt
```

- 2.3. В текстовом редакторе нажмите **Esc** и введите **:wq**, чтобы сохранить изменения и выйти из редактора. Когда редактор закроется, вы должны увидеть следующий результат:

```
...output omitted...
crontab: installing new crontab
[student@servera ~]$
```

Предыдущие выходные данные подтверждают, что задание было запланировано успешно.

3. Используйте команду **crontab -l**, чтобы вывести список запланированных повторяющихся заданий. Проверьте команду, которую вы запланировали запускать как повторяющееся задание на предыдущем шаге.

```
[student@servera ~]$ crontab -l
*/2 08-20 * * * /usr/bin/date >> /home/student/my_first_cron_job.txt
```

Обратите внимание, что предыдущее запланированное задание запускает команду **/usr/bin/date** и добавляет свои выходные данные в **/home/student/my_first_cron_job.txt**.

4. Используйте команду **while**, чтобы приглашение оболочки не переходило в спящий режим до тех пор, пока файл **/home/student/my_first_cron_job.txt** не будет создан в результате успешного выполнения запланированного вами повторяющегося задания. Подождите, пока не вернется приглашение оболочки.

```
[student@servera ~]$ while ! test -f my_first_cron_job.txt; do sleep 1s;
done
```

Предыдущая команда **while** использует **! test -f**, чтобы продолжить выполнение цикла команд **sleep 1s** до тех пор, пока файл **my_first_cron_job.txt** не будет создан в каталоге **/home/student**.

5. Используйте команду **cat**, чтобы убедиться, что содержимое **/home/student/my_first_cron_job.txt** соответствует выходным данным команды **date**.

```
[student@servera ~]$ cat my_first_cron_job.txt  
Fri Mar 22 13:56:01 IST 2019
```

Предыдущий вывод может отличаться в вашей системе.

6. Удалите все повторяющиеся задания, которые будут выполняться от имени студента.

6.1. Используйте команду **crontab -r**, чтобы удалить все запланированные повторяющиеся задания для учащегося.

```
[student@servera ~]$ crontab -r
```

6.2. Используйте команду **crontab -l**, чтобы убедиться, что для пользователя **student** не существует повторяющихся заданий.

```
[student@servera ~]$ crontab -l  
no crontab for student
```

6.3. Выйти из сервера **servera**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation** запустите **lab scheduling-cron finish**, чтобы завершить упражнение. Сценарий удаляет файлы, созданные в ходе упражнения, и обеспечивает чистоту среды.

```
[student@workstation ~]$ lab scheduling-cron finish
```

На этом пошаговое упражнение завершено.

ПЛАНИРОВАНИЕ ПОВТОРЯЮЩИХСЯ СИСТЕМНЫХ ЗАДАНИЙ

ЗАДАЧИ

После завершения этого раздела вы сможете запланировать выполнение команд по повторяющемуся расписанию, используя системный файл и каталоги **crontab**.

ОПИСАНИЕ ПОВТОРЯЮЩИХСЯ СИСТЕМНЫХ ЗАДАНИЙ

Системным администраторам часто приходится выполнять повторяющиеся задания. Лучше всего запускать эти задания из системных учетных записей, а не из учетных записей пользователей. То есть не планируйте запуск этих заданий с помощью команды **crontab**, а используйте общесистемные файлы **crontab**. Записи о заданиях в общесистемных файлах **crontab** аналогичны записям в пользовательских записях **crontab**, за исключением того, что общесистемные файлы **crontab** имеют дополнительное поле перед полем команды; пользователь, от имени которого должна выполняться команда.

Файл **/etc/crontab** содержит полезную синтаксическую диаграмму во включенных комментариях.

```
# For details see man 4 crontabs

# Example of job definition:
# ----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .---- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .--- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue ...
# | | | | |
# * * * * * user-name command to be executed
```

Повторяющиеся системные задания определены в двух местах: в файле **/etc/crontab** и в файлах в каталоге **/etc/cron.d/**. Вы всегда должны создавать свои собственные файлы **crontab** в каталоге **/etc/cron.d** для планирования повторяющихся системных заданий. Поместите настраиваемый файл **crontab** в **/etc/cron.d**, чтобы защитить его от перезаписи, если произойдет какое-либо обновление пакета поставщика **/etc/crontab**, которое может перезаписать существующее содержимое в **/etc/crontab**. Пакеты, требующие повторяющихся системных заданий, помещают свои файлы **crontab** в **/etc/cron.d/**, содержащие записи заданий. Администраторы также используют это расположение для группировки связанных заданий в один файл.

Система **crontab** также включает репозитории для скриптов, которые необходимо запускать каждый час, день, неделю и месяц. Эти репозитории представляют собой каталоги, называемые **/etc/cron.hourly/**, **/etc/cron.daily/**, **/etc/cron.weekly/** и **/etc/cron.monthly/**. Опять же, эти каталоги содержат исполняемые сценарии оболочки, а не файлы **crontab**.



ВАЖНО

Не забудьте сделать любой сценарий, который вы помещаете в эти каталоги, исполняемым. Если сценарий не является исполняемым, он не запустится. Чтобы сделать скрипт исполняемым, используйте команду `chmod +x script_name`.

Команда `run-parts`, вызываемая из файла `/etc/cron.d/0hourly`, запускает сценарии `/etc/cron.hourly/*`. Команда `run-parts` также запускает ежедневные, еженедельные и ежемесячные задания, но вызывается из другого файла конфигурации с именем `/etc/anacrontab`.



ПРИМЕЧАНИЕ

Раньше для обработки файла `/etc/anacrontab` использовалась отдельная служба под названием `anacron`, но в Red Hat Enterprise Linux 7 и более поздних версиях обычно этот файл анализирует служба `crond`.

Назначение `/etc/anacrontab` - убедиться, что важные задания всегда выполняются и не пропущены случайно из-за того, что система была выключена или находится в режиме гибернации, когда задание должно было быть выполнено. Например, если системное задание, которое запускается ежедневно, не было выполнено в последний раз из-за перезагрузки системы, задание будет выполнено, когда система будет готова. Однако при запуске задания может быть задержка в несколько минут в зависимости от значения параметра «**Delay in minutes** (Задержка в минутах)», указанного для задания в `/etc/anacrontab`.

В `/var/spool/anacron/` есть разные файлы для ежедневных, еженедельных и ежемесячных заданий, чтобы определить, было ли выполнено конкретное задание. Когда `crond` запускает задание из `/etc/anacrontab`, он обновляет отметки времени этих файлов. Эта же отметка времени используется для определения того, когда задание было выполнено в последний раз. Синтаксис `/etc/anacrontab` отличается от обычных файлов конфигурации `crontab`. Он содержит ровно четыре поля в строке, как показано ниже.

- **Period in days** (Период в днях)

Интервал в днях для задания, которое выполняется по повторяющемуся расписанию. Это поле принимает целое число или макрос в качестве значения. Например, макрос `@daily` эквивалентен целому числу **1**, что означает, что задание выполняется ежедневно. Точно так же макрос `@weekly` эквивалентен целому числу **7**, что означает, что задание выполняется еженедельно.

- **Delay in minutes** (Задержка в минутах)

Время, в течение которого демон `crond` должен ждать перед запуском этого задания.

- **Job identifier** (Идентификатор работы)

Уникальное имя задания идентифицируется как в сообщениях журнала.

- **Command** Команда

Команду, которую нужно выполнить.

Файл **/etc/anacrontab** также содержит объявления переменных среды с использованием синтаксиса **NAME=value**. Особый интерес представляет переменная **START_HOURS_RANGE**, которая определяет временной интервал для выполнения заданий. За пределами этого диапазона задания не запускаются. Если в определенный день задание не запускается в течение этого временного интервала, оно должно дождаться следующего дня для выполнения.

ВВЕДЕНИЕ ТАЙМЕРА SYSTEMD

С появлением **systemd** в Red Hat Enterprise Linux 7 теперь доступна новая функция планирования: **systemd timer units**. Блок таймера **systemd** активирует другой блок (**unit**) другого типа (например, службу (**service**)), имя которого совпадает с именем **timer units**. Блок таймера позволяет активировать другие блоки по таймеру. Для упрощения отладки **systemd** регистрирует события таймера в системных журналах (**system journals**).

Пример Timer Unit

Пакет **sysstat** предоставляет **timer unit systemd** под названием **sysstat-collect.timer** для сбора системной статистики каждые 10 минут. Следующий вывод показывает строки конфигурации **/usr/lib/systemd/system/sysstat-collect.timer**.

```
...output omitted...
[Unit]
Description=Run system activity accounting tool every 10 minutes

[Timer]
OnCalendar=*:00/10

[Install]
WantedBy=sysstat.service
```

Параметр **OnCalendar=*:00/10** означает, что этот **timer unit** активирует соответствующий блок (**sysstat-collect.service**) каждые 10 минут. Однако вы можете указать более сложные временные интервалы. Например, значение **2019-03-* 12:35,37,39:16** по отношению к параметру **OnCalendar** заставляет **timer unit** активировать соответствующий блок обслуживания в **12:35:16**, **12:37:16** и **12:39:16** каждый день в течение всего марта 2019 г. Вы также можете указать относительные таймеры с помощью таких параметров, как **OnUnitActiveSec**. Например, параметр **OnUnitActiveSec=15min** заставляет **timer unit** запускать соответствующий блок через 15 минут после того, как последний раз **timer unit** активировал соответствующий блок.



ВАЖНО

Не изменяйте файл конфигурации модуля в каталоге **/usr/lib/systemd/system**, потому что любое обновление пакета поставщика файла конфигурации может переопределить изменения конфигурации, внесенные вами в этот файл. Итак, сделайте копию файла конфигурации юнита, который вы собираетесь изменить, в каталоге **/etc/systemd/system**, а затем измените копию, чтобы изменения конфигурации, которые вы вносите в отношении юнита, не были отменены каким-либо обновлением поставщика программы. Если в каталогах **/usr/lib/systemd/system** и **/etc/systemd/system** существуют два файла с одинаковым именем, **systemd** анализирует файл в каталоге **/etc/systemd/system**.

После изменения файла конфигурации **timer unit** используйте команду **systemctl daemon-reload**, чтобы убедиться, что **systemd** знает об изменениях. Данная команда перезагружает конфигурацию менеджера **systemd**.

```
[root@host ~]# systemctl daemon-reload
```

После перезагрузки конфигурации менеджера **systemd** используйте следующую команду **systemctl**, чтобы активировать юнит таймера.

```
[root@host ~]# systemctl enable --now <unitname>.timer
```



РЕКОМЕНДАЦИИ

Справочные страницы **man crontab(5)**, **anacron(8)**, **anacrontab(5)**, **systemd.time(7)**, **systemd.timer(5)**, и **cron(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ПЛАНИРОВАНИЕ ПОВТОРЯЮЩИХСЯ СИСТЕМНЫХ ЗАДАНИЙ

В этом упражнении вы запланируете запуск команд по разным расписаниям, добавив файлы конфигурации в системные каталоги crontab.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Запланируйте повторяющееся системное задание для подсчета количества активных пользователей.
- Обновите таймер **systemd**, который собирает данные о деятельности системы.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab scheduling-system start**, для того чтобы начать упражнение. Этот сценарий гарантирует, что среда чистая и правильно настроена.

```
[student@workstation ~]$ lab scheduling-system start
```

1. С рабочей станции **workstation**, откройте сеанс **SSH** на сервере **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Используйте команду **sudo -i**, чтобы переключиться на учетную запись пользователя **root**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

3. Запланируйте повторяющееся системное задание, которое создает журнал сообщение с указанием количества активных пользователей в системе. Работа должна выполняться ежедневно. Вы можете использовать команды **w -h | wc -l** для получения количества активных пользователей в настоящее время в системе. Также используйте команду **logger** для создания сообщения журнала.

- 3.1. Создайте файл сценария с именем **/etc/cron.daily/usercount** со следующим содержимым. Вы можете использовать команду **vi /etc/cron.daily/usercount** для создания файла сценария.

```
#!/bin/bash
USERCOUNT=$(w -h | wc -l)
logger "There are currently ${USERCOUNT} active users"
```

- 3.2. Используйте команду **chmod**, чтобы включить разрешение на выполнение (**x**) для файла **/etc/cron.daily/usercount**.

```
[root@servera ~]# chmod +x /etc/cron.daily/usercount
```

4. Пакет **sysstat** предоставляет юниты **systemd**, называемые **sysstat-collect.timer** и **sysstat-collect.service**. Timer unit запускает служебный юнит каждые 10 минут для сбора данных о деятельности системы с помощью сценария оболочки, называемого **/usr/lib64/sa/sa1**. Убедитесь, что пакет **sysstat** установлен, и измените файл конфигурации юнит таймера, чтобы данные о деятельности системы собирались каждые две минуты.

- 4.1. Используйте команду **yum** для установки пакета **sysstat**.

```
[root@servera ~]# yum install sysstat
...output omitted...
Is this ok [y/N]: y
...output omitted...
Installed:
  sysstat-11.7.3-2.el8.x86_64          lm_sensors-libs-
  3.4.0-17.20180522git70f7e08.el8.x86_64

Complete!
```

- 4.2. Скопируйте **/usr/lib/systemd/system/sysstat-collect.timer** в **/etc/systemd/system/sysstat-collect.timer**.

```
[root@servera ~]# cp /usr/lib/systemd/system/sysstat-collect.timer \
/etc/systemd/system/sysstat-collect.timer
```



ВАЖНО

Вы не должны редактировать файлы в каталоге **/usr/lib/systemd**. Вы можете скопировать файл юнита в каталог **/etc/systemd/system** и отредактировать копию. Процесс **systemd** анализирует вашу настроенную копию вместо файла в каталоге **/usr/lib/systemd**.

4.3. Отредактируйте **/etc/systemd/system/sysstat-collect.timer**, чтобы таймер запускался каждые две минуты. Кроме того, замените любое вхождение строки **10 минут** на **2 минуты** во всем файле конфигурации юнита, включая строки в комментариях. Вы можете использовать команду **vi /etc/systemd/system/sysstatcollect.timer** для редактирования файла конфигурации.

```
...
#          Activates activity collector every 2 minutes

[Unit]
Description=Run system activity accounting tool every 2 minutes

[Timer]
OnCalendar=*:00/02

[Install]
WantedBy=sysstat.service
```

Предыдущие изменения приводят к тому, что модуль **sysstat-collect.timer** запускает юнит **sysstat-collect.service** каждые две минуты, который запускает **/usr/lib64/sa/sa1 1 1**. Выполнение файла **/usr/lib64/sa/sa1 1 1** собирает систему данные об активности в двоичном файле в каталоге **/var/log/sa**.

4.4. Используйте команду **systemctl daemon-reload**, чтобы убедиться, что **systemd** знает об изменениях.

```
[root@servera ~]# systemctl daemon-reload
```

4.5. Используйте команду **systemctl**, чтобы активировать юнит таймера **sysstat-collect.timer**.

```
[root@servera ~]# systemctl enable --now sysstat-collect.timer
```

4.6. Используйте команду **while**, чтобы дождаться создания двоичного файла в каталоге **/var/log/sa**. Подождите, пока не вернется приглашение оболочки.

```
[root@servera ~]# while [ $(ls /var/log/sa | wc -l) -eq 0 ]; do sleep 1s; done
```

В приведенной выше команде **while ls /var/log/sa | wc -l** возвращает **0**, если файл не существует, и **1**, если он существует. Команда **while** определяет, равно ли **0**, и если да, то входит в цикл, который приостанавливается на одну секунду. Когда файл существует, цикл **while** завершается.

4.7. Используйте команду **ls -l**, чтобы убедиться, что двоичный файл в каталоге **/var/log/sa** был изменен в течение последних двух минут.

```
[root@servera ~]# ls -l /var/log/sa
total 8
-rw-r--r--. 1 root root 5156 Mar 25 12:34 sa25
[root@servera ~]# date
Mon Mar 25 12:35:32 +07 2019
```

Вывод предыдущих команд может отличаться в вашей системе.

4.8. Выйдите из оболочки пользователя **root**.

```
[root@servera ~]# exit
logout
[student@servera ~]$
```

4.9. Выйти из сервера **servera**.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, выполните **lab scheduling-system finish**, чтобы завершить упражнение. Данный сценарий удаляет файлы, созданные в ходе упражнения, и обеспечивает чистоту среды.

```
[student@workstation ~]$ lab scheduling-system finish
```

На этом пошаговое упражнение завершено.

УПРАВЛЕНИЕ ВРЕМЕННЫМИ ФАЙЛАМИ

ЗАДАЧИ

После завершения этого раздела вы сможете включать и отключать таймеры `systemd` и настраивать таймер, который управляет временными файлами.

УПРАВЛЕНИЕ ВРЕМЕННЫМИ ФАЙЛАМИ

Современная система требует большого количества временных файлов и каталогов. Некоторые приложения (и пользователи) используют каталог `/tmp` для хранения временных данных, в то время как другие используют более специфичное для задачи расположение, такое как каталоги демонов и изменчивых (*volatile*) пользовательских файлов в `/run`. В этом контексте изменчивость означает, что файловая система, хранящая эти файлы, существует только в памяти. Когда система перезагружается или теряет питание, все содержимое энергозависимой памяти исчезнет.

Чтобы система работала чисто, необходимо, чтобы эти каталоги и файлы создавались, когда они не существуют, потому что демоны и скрипты могут полагаться на их наличие, а также для очистки старых файлов, чтобы они не заполняли дисковое пространство и не предоставляли неверную информацию.

Red Hat Enterprise Linux 7 и более поздние версии включают новый инструмент под названием `systemd-tmpfiles`, который предоставляет структурированный и настраиваемый метод управления временными каталогами и файлами.

Когда `systemd` запускает систему, одним из первых запускаемых сервисных модулей является `systemd-tmpfiles-setup`. Эта служба запускает команду `systemd-tmpfiles --create --remove`. Эта команда считывает файлы конфигурации из `/usr/lib/tmpfiles.d/*.conf`, `/run/tmpfiles.d/*.conf` и `/etc/tmpfiles.d/*.conf`. Все файлы и каталоги, отмеченные для удаления в этих файлах конфигурации, удаляются, а все файлы и каталоги, отмеченные для создания (или исправления разрешений), при необходимости будут созданы с правильными разрешениями.

Очистка временных файлов с помощью таймера Systemd

Чтобы гарантировать, что долго работающие системы не заполняют свои диски устаревшими данными, юнит таймера `systemd`, называемый `systemd-tmpfiles-clean.timer`, запускает `systemd-tmpfiles-clean.service` через регулярные интервалы времени, который выполняет команду `systemd-tmpfiles --clean`.

В файлах конфигурации юнита таймера `systemd` есть раздел `[Timer]`, который указывает, как часто следует запускать службу с таким же именем.

Используйте следующую команду `systemctl` для просмотра содержимого файла конфигурации юнита `systemd-tmpfilesclean.timer`.

```
[user@host ~]$ systemctl cat systemd-tmpfiles-clean.timer
# /usr/lib/systemd/system/systemd-tmpfiles-clean.timer
# SPDX-License-Identifier: LGPL-2.1+
#
# This file is part of systemd.
#
```

```
# systemd is free software; you can redistribute it and/or modify it
# under the terms of the GNU Lesser General Public License as published
# by
# the Free Software Foundation; either version 2.1 of the License, or
# (at your option) any later version.
```

[Unit]

```
Description=Daily Cleanup of Temporary Directories
Documentation=man:tmpfiles.d(5) man:system-tmpfiles(8)
```

[Timer]

```
OnBootSec=15min
OnUnitActiveSec=1d
```

В предыдущей конфигурации параметр **OnBootSec=15min** указывает, что служебный юнит с именем **systemd-tmpfiles-clean.service** запускается через 15 минут после загрузки системы. Параметр **OnUnitActiveSec=1d** указывает, что любое дальнейшее срабатывание юнита **systemd-tmpfiles-clean.service** происходит через 24 часа после последней активации данного юнита.

В зависимости от ваших требований вы можете изменить параметры в файле конфигурации таймера **systemd-tmpfilesclean.timer**. Например, значение **30min** для параметра **OnUnitActiveSec** запускает служебный юнит **systemd-tmpfiles-clean.service** через 30 минут после последней активации служебного юнита. В результате **systemd-tmpfiles-clean.service** запускается каждые 30 минут после вступления изменений в силу.

После изменения файла конфигурации юнита таймера используйте команду **systemctl daemon-reload**, чтобы убедиться, что **systemd** знает об изменении. Эта команда перезагружает конфигурацию менеджера **systemd**.

```
[root@host ~]# systemctl daemon-reload
```

После перезагрузки конфигурации менеджера **systemd** используйте следующую команду **systemctl**, чтобы активировать юнит **systemd-tmpfiles-clean.timer**.

```
[root@host ~]# systemctl enable --now systemd-tmpfiles-clean.timer
```

Очистка временных файлов вручную

Команда **systemd-tmpfiles --clean** анализирует те же файлы конфигурации, что и команда **systemd-tmpfiles --create**, но вместо создания файлов и каталогов она очищает все файлы, к которым не осуществлялся доступ, которые не были изменены или изменены позже, чем максимальный возраст, определенный в файле конфигурации.

Формат файлов конфигурации для **systemd-tmpfiles** подробно описан на странице руководства **tmpfiles.d(5)**. Базовый синтаксис состоит из семи столбцов: Тип, Путь, Режим, UID, GID, Возраст и Аргумент (**Type, Path, Mode, UID, GID, Age, и Argument**). Тип относится к действию, которое должна предпринять **systemd-tmpfiles**; например, **d** для создания каталога, если

он еще не существует, или **Z** для рекурсивного восстановления контекстов **SELinux** и прав доступа к файлам и владения ими.

Ниже приведены некоторые примеры с пояснениями.

```
d /run/systemd/seats 0755 root root -
```

При создании файлов и каталогов создайте каталог **/run/systemd/seats**, если он еще не существует, принадлежащий пользователю **root** и группе **root**, с разрешениями, установленными на **rwxr-xr-x**. Этот каталог не будет очищаться автоматически.

```
D /home/student 0700 student student 1d
```

Создайте каталог **/home/student**, если он еще не существует. Если он существует, очистите его от всего содержимого. При запуске **systemd-tmpfiles --clean** удаляются все файлы, к которым не осуществлялся доступ, не изменялись или не изменялись более одного дня.

```
L /run/fstablink - root root - /etc/fstab
```

Создайте символьическую ссылку **/run/fstablink**, указывающую на **/etc/fstab**. Никогда не очищайте эту линию автоматически.

Приоритет файла конфигурации

Файлы конфигурации могут находиться в трех местах:

- **/etc/tmpfiles.d/*.conf**
- **/run/tmpfiles.d/*.conf**
- **/usr/lib/tmpfiles.d/*.conf**

Файлы в **/usr/lib/tmpfiles.d/** предоставляются соответствующими пакетами **RPM**, и вам не следует редактировать эти файлы. Файлы в **/run/tmpfiles.d/** сами по себе являются изменчивыми, обычно используемыми демонами для управления своими собственными временными файлами во время выполнения. Файлы в **/etc/tmpfiles.d/** предназначены для администраторов, чтобы настроить пользовательские временные расположения и переопределить значения по умолчанию, предоставленные поставщиком.

Если файл в **/run/tmpfiles.d/** имеет то же имя, что и файл в **/usr/lib/tmpfiles.d/**, то используется файл в **/run/tmpfiles.d/**. Если файл в **/etc/tmpfiles.d/** имеет то же имя, что и файл в **/run/tmpfiles.d/** или **/usr/lib/tmpfiles.d/**, то используется файл в **/etc/tmpfiles.d/**.

Учитывая эти правила приоритета, вы можете легко переопределить настройки, предоставленные поставщиком, скопировав соответствующий файл в директорию **/etc/tmpfiles.d/**,

а затем отредактировав его. Работа таким образом гарантирует, что параметры, предоставленные администратором, могут легко управляться из центральной системы управления конфигурацией и не будут перезаписаны при обновление пакета программы.



ПРИМЕЧАНИЕ

При тестировании новых или измененных конфигураций может быть полезно применять команды только из одного файла конфигурации. Этого можно добиться, указав имя файла конфигурации в командной строке.



РЕКОМЕНДАЦИИ

Справочные страницы **man systemd-tmpfiles(8)**, **tmpfiles.d(5)**, **stat(1)**, **stat(2)**, и **systemd.timer(5)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ ВРЕМЕННЫМИ ФАЙЛАМИ

В этом упражнении вы настроите **systemd-tmpfiles**, чтобы изменить частоту удаления временных файлов из **/tmp**, а также периодичность удаление файлов из другого каталога.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Настроить **systemd-tmpfiles** для удаления неиспользуемых временных файлов из **/tmp**.
- Настроить **systemd-tmpfiles** для периодической очистки файлов из другого каталога.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab scheduling-tempfiles start**, чтобы начать упражнение. Данный сценарий создает необходимые файлы и обеспечивает правильную настройку среды.

```
[student@workstation ~]$ lab scheduling-tempfiles start
```

1. С рабочей станции **workstation**, откройте сеанс **SSH** на сервере **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Настройте **systemd-tmpfiles** для очистки каталога **/tmp**, чтобы он не содержал файлов, которые не использовались в течение последних пяти дней. Убедитесь, что конфигурация не будет перезаписана каким-либо обновлением пакета.

- 2.1. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

- 2.2. Скопируйте **/usr/lib/tmpfiles.d/tmp.conf** в **/etc/tmpfiles.d/tmp.conf**.

```
[root@servera ~]# cp /usr/lib/tmpfiles.d/tmp.conf /etc/tmpfiles.d/  
tmp.conf
```

- 2.3. Найдите строку конфигурации в **/etc/tmpfiles.d/tmp.conf**, которая относится к каталогу **/tmp**. Замените существующий возраст временных файлов в этой строке конфигурации новым сроком существования 5 дней. Удалите из файла все остальные строки, включая закомментированные. Вы можете использовать команду **vim /etc/tmpfiles.d/tmp.conf** для редактирования файла конфигурации. Файл **/etc/tmpfiles.d/tmp.conf** должен выглядеть следующим образом:

```
q /tmp 1777 root root 5d
```

Предыдущая конфигурация заставляет **systemd-tmpfiles** гарантировать, что каталог **/tmp** существует с восьмеричными разрешениями, установленными на **1777**. А также что, пользователем и владельцем группы **/tmp** является **root**. В каталоге **/tmp** не должно быть временных файлов, которые не использовались последние пять дней.

- 2.4. Используйте команду **systemd-tmpfiles --clean**, чтобы убедиться, что файл **/etc/tmpfiles.d/tmp.conf** содержит правильную конфигурацию.

```
[root@servera ~]# systemd-tmpfiles --clean /etc/tmpfiles.d/tmp.conf
```

Поскольку предыдущая команда не вернула никаких ошибок, она подтверждает правильность настроек конфигурации.

3. Добавьте новую конфигурацию, которая гарантирует, что каталог **/run/momentary** существует, а права собственности пользователя и группы - **root**. Восьмеричные разрешения для каталога должны быть **0700**. Конфигурация должна очищать любой файл в этом каталоге, который остается неиспользованным в течение последних 30 секунд.

- 3.1. Создайте файл **/etc/tmpfiles.d/momentary.conf** со следующим содержимым. Вы можете использовать команду **vim /etc/tmpfiles.d/momentary.conf** для создания файла конфигурации.

```
d /run/momentary 0700 root root 30s
```

Предыдущая конфигурация заставляет **systemd-tmpfiles** гарантировать, что каталог **/run/momentary** существует с восьмеричными разрешениями, установленными на **0700**. Пользователь и группа, владеющие **/run/momentary**, должны быть **root**. Любой файл в этом каталоге, который остается неиспользованным в течение последних 30 секунд, должен быть очищен.

- 3.2. Используйте команду **systemd-tmpfiles --create**, чтобы убедиться, что файл **/etc/tmpfiles.d/momentary.conf** содержит соответствующую конфигурацию. Команда создает каталог **/run/momentary**, если он не существует.

```
[root@servera ~]# systemd-tmpfiles --create /etc/tmpfiles.d/  
momentary.conf
```

Поскольку предыдущая команда не вернула никаких ошибок, она подтверждает правильность настроек конфигурации.

- 3.3. Используйте команду **ls**, чтобы убедиться, что каталог **/run/momentary** создан с соответствующими разрешениями, владельцем и владельцем группы.

```
[root@servera ~]# ls -ld /run/momentary  
drwx-----. 2 root root 40 Mar 21 16:39 /run/momentary
```

Обратите внимание, что восьмеричный набор разрешений **/run/momentary** равен **0700**, а права собственности пользователя и группы установлены на **root**.

4. Убедитесь, что все файлы в каталоге **/run/momentary**, которые не использовались в течение последних 30 секунд, были удалены в соответствии с конфигурацией **systemd-tmpfiles** для каталога.

- 4.1. Используйте команду **touch**, чтобы создать файл с именем **/run/momentary/testfile**.

```
[root@servera ~]# touch /run/momentary/testfile
```

- 4.2. Используйте команду **sleep**, чтобы настроить приглашение оболочки не возвращаться в течение 30 секунд.

```
[root@servera ~]# sleep 30
```

- 4.3. После возвращения приглашения оболочки используйте команду **systemd-tmpfiles --clean** для очистки устаревших файлов из **/run/momentary** в соответствии с правилом, упомянутым в **/etc/tmpfiles.d/momentary.conf**.

```
[root@servera ~]# systemd-tmpfiles --clean /etc/tmpfiles.d/  
momentary.conf
```

Предыдущая команда удаляет **/run/momentary/testfile**, потому что файл оставался неиспользованным в течение 30 секунд и должен был быть удален в соответствии с правилом, указанным в **/etc/tmpfiles.d/momentary.conf**.

- 4.4. Используйте команду **ls -l**, чтобы убедиться, что файл **/run/momentary/testfile** не существует.

```
[root@servera ~]# ls -l /run/momentary/testfile
```

```
ls: cannot access '/run/momentary/testfile': No such file or directory
```

4.5. Выйдите из оболочки пользователя **root**, чтобы вернуться к пользователю **student**.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$
```

4.6. Выйти из сервера **servera**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите скрипт **lab scheduling-tempfiles finish**, чтобы завершить это упражнение. Данный сценарий удаляет файлы, созданные в ходе упражнения, и обеспечивает чистоту среды.

```
[student@workstation ~]$ lab scheduling-tempfiles finish
```

На этом пошаговое упражнение завершено.

КОНТРОЛЬНЫЙ ОПРОС

ПЛАНИРОВАНИЕ БУДУЩИХ ЗАДАЧ

Выберите правильные ответы на следующие вопросы

1. Какая команда отображает все пользовательские задания, которые в настоящее время запланированы для выполнения как отложенные?
 - a. **atq**
 - b. **atrm**
 - c. **at -c**
 - d. **at --display**
2. Какая команда удаляет отложенное пользовательское задание с номером 5?
 - a. **at -c 5**
 - b. **atrm 5**
 - c. **at 5**
 - d. **at --delete 5**
3. Какая команда отображает все повторяющиеся пользовательские задания, запланированные для текущего пользователя, вошедшего в систему?
 - a. **crontab -r**
 - b. **crontab -l**
 - c. **crontab -u**
 - d. **crontab -V**
4. В каком формате задание **/usr/local/bin/daily_backup hourly** выполняется ежечасно с 9 до 18 часов. во все дни с понедельника по пятницу?
 - a. **00 * * * Mon-Fri /usr/local/bin/daily_backup**
 - b. *** */9 * * Mon-Fri /usr/local/bin/daily_backup**
 - c. **00 */18 * * * /usr/local/bin/daily_backup**
 - d. **00 09-18 * * Mon-Fri /usr/local/bin/daily_backup**
5. В каком каталоге находятся сценарии оболочки, предназначенные для ежедневного запуска?
 - a. **/etc/cron.d**
 - b. **/etc/cron.hourly**
 - c. **/etc/cron.daily**
 - d. **/etc/cron.weekly**
6. Какой файл конфигурации определяет настройки для системных заданий, которые выполняются ежедневно, еженедельно и ежемесячно?
 - a. **/etc/crontab**
 - b. **/etc/anacrontab**

- c. /etc/inittab
- d. /etc/sysconfig/crond

7. Какой юнит **systemd** регулярно запускает очистку временных файлов?

- a. systemd-tmpfiles-clean.timer
- b. systemd-tmpfiles-clean.service
- c. dnf-makecache.timer
- d. unbound-anchor.timer

РЕШЕНИЕ

ПЛАНИРОВАНИЕ БУДУЩИХ ЗАДАЧ

Выберите правильные ответы на следующие вопросы

1. Какая команда отображает все пользовательские задания, которые в настоящее время запланированы для выполнения как отложенные?
 - a. **atq**
 - b. atrm
 - c. at -c
 - d. at --display

2. Какая команда удаляет отложенное пользовательское задание с номером 5?
 - a. at -c 5
 - b. atm 5**
 - c. at 5
 - d. at --delete 5

3. Какая команда отображает все повторяющиеся пользовательские задания, запланированные для текущего пользователя, вошедшего в систему?
 - a. crontab -r
 - b. crontab -l**
 - c. crontab -u
 - d. crontab -V

4. В каком формате задание **/usr/local/bin/daily_backup hourly** выполняется ежечасно с 9 до 18 часов. во все дни с понедельника по пятницу?
 - a. 00 * * * Mon-Fri /usr/local/bin/daily_backup
 - b. * */9 * * Mon-Fri /usr/local/bin/daily_backup
 - c. 00 */18 * * * /usr/local/bin/daily_backup
 - d. 00 09-18 * * Mon-Fri /usr/local/bin/daily_backup**

5. В каком каталоге находятся сценарии оболочки, предназначенные для ежедневного запуска?
 - a. /etc/cron.d
 - b. /etc/cron.hourly
 - c. /etc/cron.daily**
 - d. /etc/cron.weekly

6. Какой файл конфигурации определяет настройки для системных заданий, которые выполняются ежедневно, еженедельно и ежемесячно?
 - a. /etc/crontab
 - b. /etc/anacrontab**

- c. /etc/inittab
- d. /etc/sysconfig/crond

7. Какой юнит **systemd** регулярно запускает очистку временных файлов?

- a. **systemd-tmpfiles-clean.timer**
- b. systemd-tmpfiles-clean.service
- c. dnf-makecache.timer
- d. unbound-anchor.timer

РЕЗЮМЕ

В этой главе вы узнали:

- Задания, запуск которых запланирован один раз в будущем, называются отложенными заданиями или задачами.
- Повторяющиеся пользовательские задания выполняют задачи пользователя по повторяющемуся расписанию.
- Повторяющиеся системные задания выполняют административные задачи по повторяющемуся расписанию, что оказывает влияние на всю систему.
- Таймеры **systemd** могут выполнять как отложенные, так и повторяющиеся задания.

ГЛАВА 3

НАСТРОЙКИ ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМЫ

ЦЕЛЬ

Повысьте производительность системы, задав параметры настройки и изменив приоритет планирования процессов.

ЗАДАЧИ

- Оптимизируйте производительность системы, выбрав профиль настройки, управляемый демоном **tuned**.
- Установите приоритет или снимите приоритет с определенных процессов с помощью команд **nice** и **renice**.

РАЗДЕЛЫ

- Регулировка профилей настройки задания (и упражнения с пошаговыми инструкциями)
- Влияние на планирование процессов задания (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Настройка производительности системы.

РЕГУЛИРОВКА ПРОФИЛЕЙ НАСТРОЙКИ

ЗАДАЧИ

После завершения этого раздела вы сможете оптимизировать производительность системы, выбрав профиль настройки, управляемый настроенным демоном.

СИСТЕМЫ НАСТРОЙКИ

Системные администраторы могут оптимизировать производительность системы, регулируя различные параметры устройства в зависимости от вариантов рабочих нагрузок. Демон **tuned** применяет настройки как статически, так и динамически, используя профили настройки, которые отражают конкретные требования рабочей нагрузки.

Настройка статической настройки

Демон **tuned** применяет системные настройки при запуске службы или при выборе нового профиля настройки. Статическая настройка настраивает предопределенные параметры ядра в профилях, которые применяются во время выполнения. При статической настройке параметры ядра устанавливаются в соответствии с ожидаемой общей производительностью и не корректируются при изменении уровней активности.

Настройка динамической настройки

Благодаря динамической настройке демон **tuned** отслеживает активность системы и регулирует настройки в зависимости от изменений поведения во время выполнения. Динамическая настройка - это непрерывная настройка в соответствии с текущей рабочей нагрузкой, начиная с исходных настроек, заявленных в выбранном профиле настройки.

Например, устройства хранения часто используются во время запуска и входа в систему, но имеют минимальную активность, когда пользовательские рабочие нагрузки состоят из использования веб-браузеров и почтовых клиентов. Точно так же активность ЦП и сетевых устройств возрастает во время пиковой нагрузки в течение рабочего дня. Демон **tuned** отслеживает активность этих компонентов и регулирует настройки параметров, чтобы максимизировать производительность в периоды высокой активности и уменьшить настройки во время низкой активности. Демон **tuned** использует параметры производительности, указанные в предварительно определенных профилях настройки.

УСТАНОВКА И ВКЛЮЧЕНИЕ TUNED

Минимальная установка Red Hat Enterprise Linux 8 включает и включает настроенный пакет по умолчанию. Чтобы установить и включить пакет вручную:

```
[root@host ~]$ yum install tuned
[root@host ~]$ systemctl enable --now tuned
Created symlink /etc/systemd/system/multi-user.target.wants/tuned.service → /usr/
lib/systemd/system/tuned.service.
```

ВЫБОР ПРОФИЛЯ НАСТРОЙКИ

Приложение **Tuned** предоставляет профили, разделенные на следующие категории:

- Энергосберегающие профили
- Профили для повышения производительности

Профили повышения производительности включают профили, которые сосредоточены на следующих аспектах:

- Низкая задержка для хранилища и сети.
- Высокая пропускная способность для хранилища и сети.
- Производительность виртуальной машины
- Производительность хоста виртуализации

Профили настройки, распространяемые с Red Hat Enterprise Linux 8

НАСТРОЕННЫЙ ПРОФИЛЬ	ЦЕЛЬ
balanced	Идеально подходит для систем, требующих компромисса между энергосбережением и производительностью.
desktop	На основе профиля balanced . Обеспечивает более быстрый отклик интерактивных приложений.
throughput-performance	Настраивает систему на максимальную пропускную способность.
latency-performance	Идеально подходит для серверных систем, которым требуется низкая задержка за счет энергопотребления.
network-latency	Получено из профиля latency-performance . Это позволяет использовать дополнительные параметры настройки сети, чтобы обеспечить низкую задержку сети.
network-throughput	Получено из профиля throughput-performance . Дополнительные параметры настройки сети применяются для максимальной пропускной способности сети.
powersave	Настраивает систему для максимального энергосбережения.
oracle	Оптимизирован для загрузки базы данных Oracle на основе профиля throughput-performance .
virtual-guest	Настраивает систему на максимальную производительность, если она работает на виртуальной машине.
virtual-host	Настраивает систему на максимальную производительность, если она выступает в качестве хоста для виртуальных машин.

УПРАВЛЕНИЕ ПРОФИЛЯМИ ИЗ КОМАНДНОЙ СТРОКИ

Команда **tuned-adm** используется для изменения настроек демона **tuned**. Команда **tuned-adm** может запрашивать текущие настройки, перечислять доступные профили, рекомендовать профиль настройки для системы, напрямую изменять профили или отключать настройку.

Системный администратор идентифицирует активный в данный момент профиль настройки с помощью **tuned-adm active**.

```
[root@host ~]# tuned-adm active  
Current active profile: virtual-guest
```

Команда **tuned-adm list** перечисляет все доступные профили настройки, включая как встроенные, так и настраиваемые профили настройки, созданные системным администратором.

```
[root@host ~]# tuned-adm list  
Available profiles:  
- balanced  
- desktop  
- latency-performance  
- network-latency  
- network-throughput  
- powersave  
- sap  
- throughput-performance  
- virtual-guest  
- virtual-host  
Current active profile: virtual-guest
```

Используйте имя профиля, чтобы переключить активный профиль на другой, который лучше соответствует текущим требованиям настройки системы **tuned-adm profile *profilename***.

```
[root@host ~]$ tuned-adm profile throughput-performance  
[root@host ~]$ tuned-adm active  
Current active profile: throughput-performance
```

Команда **tuned-adm** может порекомендовать профиль настройки для системы. Этот механизм используется для определения профиля системы по умолчанию после установки.

```
[root@host ~]$ tuned-adm recommend  
virtual-guest
```



ПРИМЕЧАНИЕ

Вывод команды **tuned-adm recommend** основан на различных характеристиках системы, в том числе на том, является ли система виртуальной машиной и другими предопределенными категориями, выбранными во время установки системы.

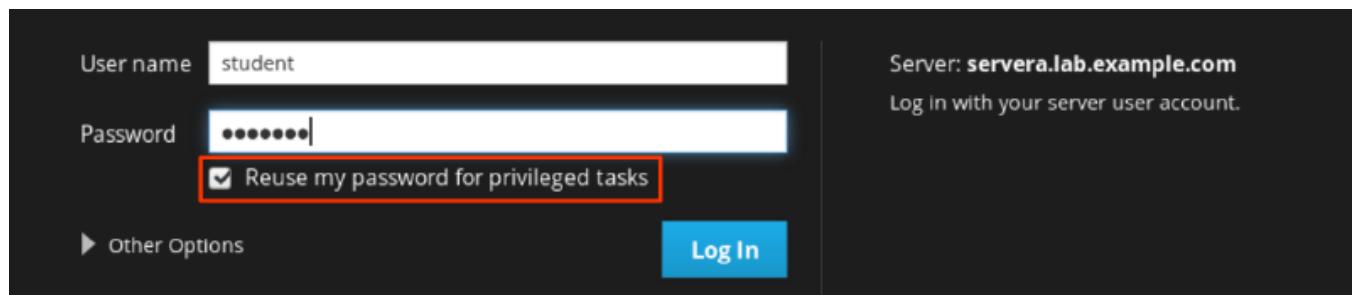
Чтобы отменить изменения настроек, сделанные текущим профилем, либо переключитесь на другой профиль, либо деактивируйте настроенный демон **tuned**. Отключите настройку **tuned** с помощью команды **tuned-adm off**.

```
[root@host ~]$ tuned-adm off  
[root@host ~]$ tuned-adm active  
No current active profile.
```

УПРАВЛЕНИЕ ПРОФИЛЯМИ С ВЕБ-КОНСОЛИ

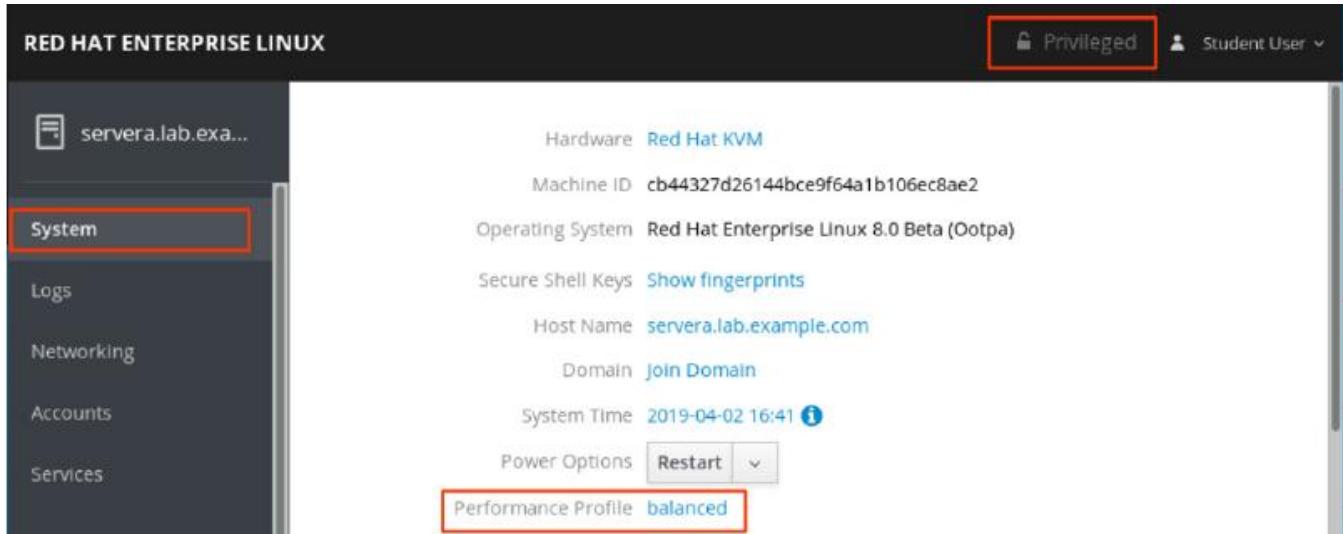
Чтобы управлять профилями производительности системы с помощью веб-консоли, войдите в систему с привилегированным доступом. Щелкните параметр «*Reuse my password for privileged tasks*» (Повторно использовать мой пароль) для привилегированных задач. Это позволяет пользователю выполнять команды с привилегиями **sudo**, которые изменяют профили производительности системы.

Рисунок 3.1: Привилегированный вход в веб-консоль



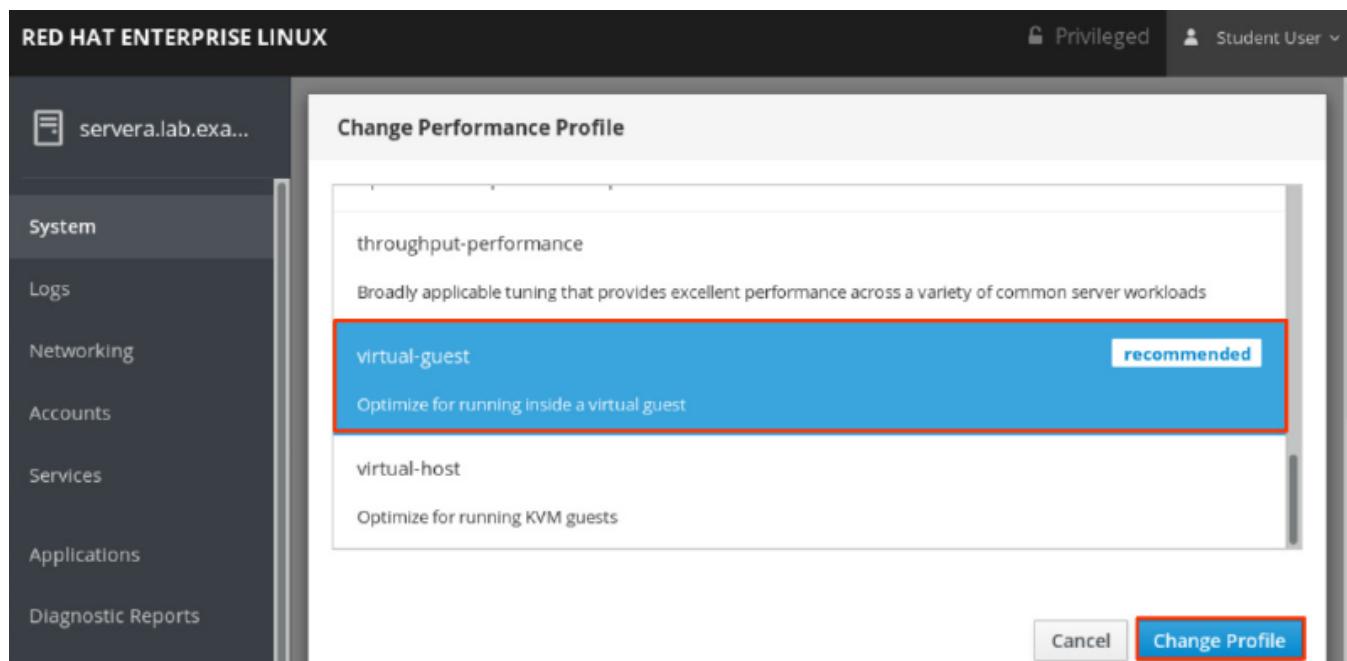
В качестве привилегированного пользователя щелкните пункт меню «*Systems*» на левой панели навигации. Текущий активный профиль отображается в поле «*Performance Profile*» (Профиль производительности). Чтобы выбрать другой профиль, щелкните ссылку активного профиля.

Рисунок 3.2: Активный профиль производительности



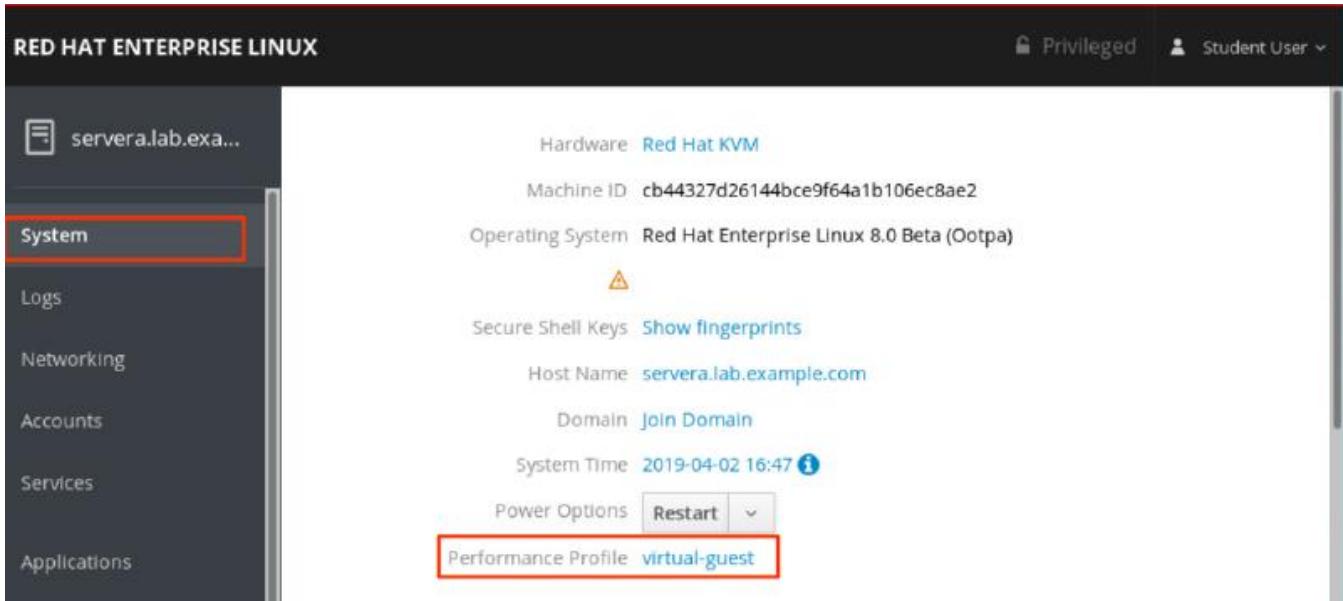
В пользовательском интерфейсе «*Change Performance Profile*» (изменения профиля производительности) прокрутите список профилей, чтобы выбрать тот, который лучше всего подходит для целей системы.

Рисунок 3.3: Выберите предпочтительный профиль производительности



Чтобы проверить изменения, вернитесь на главную страницу «*System*» (Система) и убедитесь, что в поле «*Performance Profile*» (Профиль производительности) отображается активный профиль.

Рисунок 3.4: Проверка активного профиля производительности



РЕКОМЕНДАЦИИ

Справочные страницы **man tuned(8)**, **tuned.conf(5)**, **tuned-main.conf(5)** и, **tuned-adm(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

РЕГУЛИРОВКА ПРОФИЛЕЙ НАСТРОЙКИ

В этом упражнении вы настроите производительность сервера, активировав службу tuned и применив профиль tuned.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность настроить систему для использования профиля настройки.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab tuning-profiles start**. Команда запускает сценарий, определяющий, доступен ли хост **servera** в сети.

```
[student@workstation ~]$ lab tuning-profiles start
```

1. С рабочей станции **workstation**, откройте сеанс **SSH** на сервере **servera** как пользователь **student**. Системы настроены на использование ключей SSH для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Убедитесь, что пакет **tuned** установлен, включен и запущен.

2.1. Используйте **yum**, чтобы подтвердить, что пакет **tuned** установлен.

```
[student@servera ~]$ yum list tuned  
...output omitted...  
Installed Packages  
tuned.noarch           2.10.0-15.el8  
@anaconda
```

2.2. Команда **systemctl is-enabled tuned; systemctl is-active tuned** отображает его включение и состояние выполнения.

```
[student@servera ~]$ systemctl is-enabled tuned; systemctl is-active
```

```
tuned  
enabled  
active
```

3. Перечислите доступные профили настройки и укажите активный профиль. Если **sudo** запрашивает пароль, введите "student" после приглашения.

```
[student@servera ~]$ sudo tuned-adm list  
[sudo] password for student: student  
Available profiles:  
- balanced - General non-specialized tuned profile  
- desktop - Optimize for the desktop use-case  
- latency-performance - Optimize for deterministic performance at the  
cost of increased power consumption  
- network-latency - Optimize for deterministic performance at the  
cost of increased power consumption, focused on low  
latency  
- network-throughput - Optimize for streaming network throughput,  
generally only necessary on older CPUs or 40G+ networks  
- powersave - Optimize for low power consumption  
- throughput-performance - Broadly applicable tuning that provides excellent  
performance across a variety of common server  
workloads  
- virtual-guest - Optimize for running inside a virtual guest  
- virtual-host - Optimize for running KVM guests  
Current active profile: virtual-guest
```

4. Измените текущий активный профиль настройки на **powersave**, затем подтвердите результаты. Если **sudo** запрашивает пароль, введите "student" после приглашения.

4.1. Измените текущий активный профиль настройки.

```
[student@servera ~]$ sudo tuned-adm profile powersave
```

4.2. Убедитесь, что **powersave** является активным профилем настройки.

```
[student@servera ~]$ sudo tuned-adm active  
Current active profile: powersave
```

5. Выход из сервера **servera**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab tuning-profiles finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab tuning-profiles finish
```

На этом пошаговое упражнение завершено.

ВЛИЯНИЕ НА ПЛАНИРОВАНИЕ ПРОЦЕССА

ЗАДАЧИ

После завершения этого раздела вы сможете расставлять приоритеты или отменять приоритеты определенных процессов с помощью команд **nice** и **renice**.

ПЛАНИРОВАНИЕ И МНОГОЗАДАЧНОСТЬ ПРОЦЕССОВ LINUX

Современные компьютерные системы варьируются от низкоуровневых систем с одним процессором, который может выполнять только одну инструкцию в любой момент времени, до высокопроизводительных суперкомпьютеров с сотнями процессоров каждый и десятками или даже сотнями процессорных ядер на каждом процессоре, что позволяет параллельное выполнение огромного количества инструкций. У всех этих систем по-прежнему есть одна общая черта: необходимость запускать больше потоков процессов, чем у них есть ЦП.

Linux и другие операционные системы запускают больше процессов, чем есть процессорных единиц, используя технику, называемую квантованием времени или многозадачностью. Планировщик процессов операционной системы быстро переключается между процессами на одном ядре, создавая впечатление, что одновременно выполняется несколько процессов.

ОТНОСИТЕЛЬНЫЕ ПРИОРИТЕТЫ

У разных процессов есть разные уровни важности. Планировщик процессов можно настроить для использования разных политик планирования для разных процессов. Политика планирования, используемая для большинства процессов, выполняемых в обычной системе, называется **SCHED_OTHER** (также называемая **SCHED_NORMAL**), но существуют другие политики для различных потребностей рабочей нагрузки.

Поскольку не все процессы одинаково важны, процессам, выполняющимся с политикой **SCHED_NORMAL**, может быть назначен относительный приоритет. Этот приоритет называется хорошей ценностью (*nice value*) процесса, которая организована в виде 40 различных уровней удобства для любого процесса.

Значения уровня **nice** находятся в диапазоне от -20 (высший приоритет) до 19 (низший приоритет). По умолчанию процессы наследуют свой **nice** уровень от своего родителя, который обычно **равен 0**. Более высокие уровни **nice** качества указывают на меньший приоритет (процесс легко отказывается от использования ЦП), в то время как более низкие уровни **nice** качества указывают на более высокий приоритет (процесс менее склонен давать вверх ЦП). Если нет конкуренции за ресурсы, например, когда активных процессов меньше, чем доступных ядер ЦП, даже процессы с высоким уровнем **nice** по-прежнему будут использовать все доступные ресурсы ЦП. Однако если процессоров, запрашивающих процессорное время, больше, чем доступных ядер, процессы с более высоким уровнем **nice** будут получать меньше процессорного времени, чем процессы с более низким уровнем **nice**.

УСТАНОВКА УРОВНЕЙ NICE И РАЗРЕШЕНИЙ

Поскольку установка низкого уровня **nice** для процесса, требовательного к процессору, может отрицательно повлиять на производительность других процессов, работающих в той же системе, только пользователь **root** может снизить уровень **nice** процесса.

Непrivилегированным пользователям разрешается только повышать хороший уровень в своих собственных процессах. Они не могут понизить уровень **nice** своих процессов или изменить хороший уровень процессов других пользователей.

ОТЧЕТНОСТЬ ОБ УРОВНЯХ

Несколько инструментов отображают уровни **nice** запущенных процессов. Инструменты управления процессами, такие как **top**, по умолчанию отображают уровень **nice**. Другие инструменты, такие, как команда **ps**, отображают уровни **nice** при использовании правильных параметров.

Отображение уровней nice с помощью Top

Используйте команду **top** для интерактивного просмотра и управления процессами. В конфигурации по умолчанию отображаются две интересные колонки с уровнями **nice** и приоритетами. В столбце **NI** отображается значение **nice** процесса, а в столбце **PR** отображается его запланированный приоритет. В верхнем интерфейсе уровень **nice** соответствует внутренней очереди системного приоритета, как показано на следующем рисунке. Например, **nice -20** соответствует 0 в столбце PR. Уровень **nice 19** соответствует приоритету 39 в столбце **PR**.

Рисунок 3.5: Уровни nice по данным программы top



Отображение уровней nice из командной строки

Команда **ps** отображает уровни **nice** процесса, но только путем включения правильных параметров форматирования.

Следующая команда **ps** перечисляет все процессы с их **PID**, именем процесса, уровнем **nice** и классом планирования, отсортированные в порядке убывания по уровню **nice**. Процессы, отображающие **TS** в столбце класса планирования **CLS**, выполняются в соответствии с политикой планирования **SCHED_NORMAL**. Процессы, отмеченные тире (-) в качестве своего уровня **nice**, выполняются в соответствии с другими политиками планирования и интерпретируются планировщиком как более высокий приоритет. Подробности дополнительных политик планирования выходят за рамки этого курса.

```
[user@host ~]$ ps axo pid,comm,nice,cls --sort=-nice
 PID COMMAND      NI  CLS
 30 khugepaged    19  TS
 29 ksmd         5  TS
  1 systemd       0  TS
  2 kthreadd      0  TS
  9 ksoftirqd/0   0  TS
 10 rcu_sched     0  TS
 11 migration/0   - FF
 12 watchdog/0    - FF
...output omitted...
```

НАЧАЛО ПРОЦЕССОВ С РАЗЛИЧНЫХ УРОВНЕЙ NICE

Во время создания процесса процесс наследует уровень **nice** своего родителя. Когда процесс запускается из командной строки, он наследует свой **nice** уровень от процесса оболочки, в котором он был запущен. Обычно это приводит к запуску новых процессов с уровнем **nice** 0.

В следующем примере процесс запускается из оболочки и отображается значение **nice** процесса. Обратите внимание на использование параметра **PID** в **ps** для указания запрошенного вывода.

```
[user@host ~]$ sha1sum /dev/zero &
[1] 3480
[user@host ~]$ ps -o pid,comm,nice 3480
 PID COMMAND      NI
 3480 sha1sum     0
```

Команда **nice** может использоваться всеми пользователями для запуска команд с уровнем **nice** по умолчанию или выше. Без параметров команда **nice** запускает процесс со значением по умолчанию, равным **10**.

В следующем примере команда **sha1sum** запускается как фоновое задание с уровнем **nice** по умолчанию и отображает уровень **nice** процесса:

```
[user@host ~]$ nice sha1sum /dev/zero &
[1] 3517
[user@host ~]$ ps -o pid,comm,nice 3517
 PID COMMAND      NI
 3517 sha1sum     10
```

Используйте опцию **-n**, чтобы применить определенный пользователем уровень **nice** начальному процессу. По умолчанию к текущему уровню **nice** процесса добавляется 10. В

следующем примере команда запускается как фоновое задание с заданным пользователем значением **nice** и отображает уровень **nice** процесса:

```
[user@host ~]$ nice -n 15 sha1sum &  
[1] 3521  
[user@host ~]$ ps -o pid,comm,nice 3521  
 PID COMMAND      NI  
3521 sha1sum      15
```



ВАЖНО

Непrivилегированные пользователи могут только увеличить уровень **nice** с его текущего значения до максимума **19**. После увеличения непrivилегированные пользователи не могут уменьшить значение, чтобы вернуться к предыдущему уровню **nice**. Только пользователь **root** может снизить уровень **nice** с любого текущего уровня до минимального **-20**.

ИЗМЕНЕНИЕ УРОВНЯ NICE СУЩЕСТВУЮЩЕГО ПРОЦЕССА

Уровень **nice** существующего процесса можно изменить с помощью команды **renice**. В этом примере используется идентификатор **PID** из предыдущего примера для изменения с текущего приятного уровня **15** на желаемый уровень **nice 19**.

```
[user@host ~]$ renice -n 19 3521  
3521 (process ID) old priority 15, new priority 19
```

Команду **top** также можно использовать для изменения уровня **nice** процесса. В интерактивном интерфейсе **top** нажмите опцию **r**, чтобы получить доступ к команде **renice**, за которой следует **PID**, который нужно изменить, и новый уровень **nice**.



РЕКОМЕНДАЦИИ

Справочные страницы **man nice(1)**, **renice(1)**, **top(1)**, и **sched_setscheduler(2)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ВЛИЯНИЕ НА ПЛАНИРОВАНИЕ ПРОЦЕССОВ

В этом упражнении вы настроите приоритет планирования процессов с помощью команд **nice** и **renice** и понаблюдаете за тем, как это влияет на выполнение процесса.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность настраивать приоритеты планирования для процессов.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab tuning-procscheduling start**. Команда запускает сценарий, для определения, доступен ли хост **servera** в сети.

```
[student@workstation ~]$ lab tuning-procscheduling start
```

1. С хоста **workstation** откройте сеанс **SSH** на сервер **servera**, как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Определите количество ядер ЦП на сервере, а затем запустите два экземпляра команды **sha1sum /dev/zero &** для каждого ядра.

- 2.1. Используйте команду **grep** для анализа количества существующих виртуальных процессоров (ядер ЦП) из файла **/proc/cpuinfo**.

```
[student@servera ~]$ grep -c '^processor' /proc/cpuinfo  
2
```

- 2.2. Используйте команду цикла, чтобы запустить несколько экземпляров команды **sha1sum /dev/zero &**. Запустите два на виртуальный процессор, найденный на предыдущем шаге. В этом примере это четыре экземпляра. Значения **PID** в вашем выводе будут отличаться от примера.

```
[student@servera ~]$ for i in $(seq 1 4); do sha1sum /dev/zero & done  
[1] 2643  
[2] 2644  
[3] 2645  
[4] 2646
```

3. Убедитесь, что фоновые задания выполняются для каждого из процессов **sha1sum**.

```
[student@servera ~]$ jobs  
[1]  Running sha1sum /dev/zero &  
[2]  Running sha1sum /dev/zero &  
[3]- Running sha1sum /dev/zero &  
[4]+ Running sha1sum /dev/zero &
```

4. Используйте команды **ps** и **pgrep**, чтобы отобразить процент использования ЦП для каждого процесса **sha1sum**.

```
[student@servera ~]$ ps u $(pgrep sha1sum)  
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND  
student  2643 49.8  0.0 228360  1744 pts/0      R   11:15   6:09 sha1sum /  
dev/zero  
student  2644 49.8  0.0 228360  1780 pts/0      R   11:15   6:09 sha1sum /  
dev/zero  
student  2645 49.8  0.0 228360  1748 pts/0      R   11:15   6:09 sha1sum /  
dev/zero  
student  2646 49.8  0.0 228360  1780 pts/0      R   11:15   6:09 sha1sum /  
dev/zero
```

5. Завершите все процессы **sha1sum**, затем убедитесь, что нет запущенных заданий.

5.1. Используйте команду **pkill**, чтобы завершить все запущенные процессы с шаблоном имени **sha1sum**.

```
[student@servera ~]$ pkill sha1sum  
[2]  Terminated sha1sum /dev/zero  
[4]+ Terminated sha1sum /dev/zero  
[1]- Terminated sha1sum /dev/zero  
[3]+ Terminated sha1sum /dev/zero
```

5.2. Убедитесь, что нет запущенных заданий.

```
[student@servera ~]$ jobs  
[student@servera ~]$
```

6. Запустите несколько экземпляров **sha1sum /dev/zero &**, затем запустите еще один экземпляр **sha1sum /dev/zero &c** уровнем **nice 10**. Запустите по крайней мере столько экземпляров, сколько виртуальных процессоров в системе. В этом примере запускаются 3 обычных экземпляра плюс еще один с более высоким уровнем **nice**.

6.1. Используйте цикл, чтобы запустить три экземпляра **sha1sum /dev/zero &**.

```
[student@servera ~]$ for i in $(seq 1 3); do sha1sum /dev/zero & done  
[1] 1947  
[2] 1948  
[3] 1949
```

6.2. Используйте команду **nice**, чтобы запустить четвертый экземпляр с **10** уровнем **nice**.

```
[student@servera ~]$ nice -n 10 sha1sum /dev/zero &  
[4] 1953
```

7. Используйте команды **ps** и **pgrep**, чтобы отобразить **PID**, процент использования ЦП, значение **nice** и имя исполняемого файла для каждого процесса. Экземпляр со значением **nice**, равным **10**, должен отображать более низкий процент использования ЦП, чем другие экземпляры.

```
[student@servera ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum)  
PID %CPU NI COMMAND  
1947 66.0 0 sha1sum  
1948 65.7 0 sha1sum  
1949 66.1 0 sha1sum  
1953 6.7 10 sha1sum
```

8. Используйте команду **sudo renice**, чтобы понизить **nice** уровень процесса по сравнению с предыдущим шагом. Обратите внимание на значение **PID** из экземпляра процесса с уровнем **nice 10**. Используйте этот **PID** процесса, чтобы понизить его **nice** уровень до **5**.

```
[student@servera ~]$ sudo renice -n 5 1953  
[sudo] password for student:  
1953 (process ID) old priority 10, new priority 5
```

9. Повторите команды **ps** и **pgrep**, чтобы повторно отобразить процент процессора и **nice** уровень.

```
[student@servera ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum)
 PID %CPU  NI COMMAND
 1947 63.8   0 sha1sum
 1948 62.8   0 sha1sum
 1949 65.3   0 sha1sum
1953  9.1   5 sha1sum
```

10. Выход из сервера **servera**.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab tuning-procscheduling finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab tuning-procscheduling finish
```

На этом пошаговое упражнение завершено.

ЛАБОРАТОРНАЯ РАБОТА

НАСТРОЙКА ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМЫ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы примените определенный профиль настройки и отрегулируете приоритет планирования для существующего процесса с высокой загрузкой ЦП.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Активировать определенный профиль настройки для компьютерной системы.
- Настроить приоритет планирования ЦП для процесса.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab tuning-review start**. Команда запускает сценарий, который определяет, доступен ли хост **serverb** в сети.

```
[student@workstation ~]$ lab tuning-review start
```

1. Измените текущий профиль настройки для **serverb** на сбалансированный, общий неспециализированный настроенный профиль.
2. Два процесса на **serverb** потребляют высокий процент использования ЦП. Измените уровень **nice** каждого процесса на **10**, чтобы выделить больше процессорного времени для других процессов.

Оценка

На рабочей станции **workstation**, запустите команду **lab tuning-review grade**, чтобы подтвердить успех выполнения лабораторной работы.

```
[student@workstation ~]$ lab tuning-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab tuning-review finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab tuning-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

НАСТРОЙКА ПРОИЗВОДИТЕЛЬНОСТИ СИСТЕМЫ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы примените определенный профиль настройки и отрегулируете приоритет планирования для существующего процесса с высокой загрузкой ЦП.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Активировать определенный профиль настройки для компьютерной системы.
- Настроить приоритет планирования ЦП для процесса.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab tuning-review start**. Команда запускает сценарий, который определяет, доступен ли хост **serverb** в сети.

```
[student@workstation ~]$ lab tuning-review start
```

1. Измените текущий профиль настройки для **serverb** на **balanced**, общий неспециализированный профиль **tuned**.

- 1.1. С рабочей станции **workstation** откройте сеанс **SSH** на **serverb** как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 1.2. Используйте **yum**, чтобы подтвердить, что пакет **tuned** установлен.

```
[student@serverb ~]$ yum list tuned
...output omitted...
Installed Packages
tuned.noarch          2.10.0-15.el8
@anaconda
```

- 1.3.** Используйте команду **systemctl is-enabled tuned**, чтобы отобразить состояние службы **tuned**.

```
[student@serverb ~]$ systemctl is-enabled tuned  
enabled
```

- 1.4.** Перечислите все доступные профили **tuned** и их описания. Обратите внимание, что текущий активный профиль - **virtual-guest**.

```
[student@serverb ~]$ sudo tuned-adm list  
[sudo] password for student: student  
Available profiles:  
- balanced - General non-specialized tuned profile  
- desktop - Optimize for the desktop use-case  
- latency-performance - Optimize for deterministic performance at  
the cost of increased power consumption  
- network-latency - Optimize for deterministic performance at  
the cost of increased power consumption, focused on low  
latency  
- network-throughput - Optimize for streaming network throughput,  
generally network performance  
only necessary on older CPUs or 40G+  
networks  
- powersave - Optimize for low power consumption  
- throughput-performance - Broadly applicable tuning that provides  
excellent performance across a variety of common  
server workloads  
- virtual-guest - Optimize for running inside a virtual guest  
- virtual-host - Optimize for running KVM guests  
Current active profile: virtual-guest
```

- 1.5.** Измените текущий активный профиль **tuned** на профиль **balanced**.

```
[student@serverb ~]$ sudo tuned-adm profile balanced
```

- 1.6.** Выведите сводную информацию о текущем активном настроенном профиле. Используйте команду **tuned-adm profile_info**, чтобы убедиться, что активный профиль является **balanced**.

```
[student@serverb ~]$ sudo tuned-adm profile_info  
Profile name:  
balanced
```

Profile summary:
General non-specialized tuned profile
...output omitted...

2. Два процесса на **serverb** потребляют высокий процент использования ЦП. Измените уровень **nice** каждого процесса на **10**, чтобы выделить больше процессорного времени для других процессов.
- 2.1. Определите двух потребителей ЦП на **serverb**. Самые популярные потребители ЦП указываются в выводе команды последними. Значения процентного соотношения ЦП будут отличаться.

```
[student@serverb ~]$ ps aux --sort=pcpu
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME
COMMAND
...output omitted...
root      2983  100  0.0 228360  1744 ?          R<   21:08
0:23 md5sum /dev/zero
root      2967  101  0.0 228360  1732 ?          RN   21:08
0:23 sha1sum /dev/zero
[student@serverb ~]$
```

- 2.2. Определите текущий уровень **nice** для каждого из двух верхних потребителей ЦП.

```
[student@serverb ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum;pgrep
md5sum)
PID %CPU  NI COMMAND
2967 99.6   2 sha1sum
2983 99.7  -2 md5sum
```

- 2.3. Используйте команду **sudo renice -n 10 2967 2983**, чтобы настроить уровень **nice** для каждого процесса на **10**. Используйте значения **PID**, указанные в выходных данных предыдущей команды.

```
[student@serverb ~]$ sudo renice -n 10 2967 2983
[sudo] password for student: student
2967 (process ID) old priority 2,  new priority 10
2983 (process ID) old priority -2, new priority 10
```

- 2.4. Убедитесь, что текущий уровень **nice** для каждого процесса равен **10**.

```
[student@serverb ~]$ ps -o pid,pcpu,nice,comm $(pgrep sha1sum;pgrep
md5sum)
PID %CPU      NI      COMMAND
2967 99.6     10      sha1sum
```

2983 99.7

10

md5sum

2.5. Выйти с сервера **serverb**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

Оценка

На рабочей станции **workstation**, запустите команду **lab tuning-review grade**, чтобы подтвердить успех выполнения лабораторной работы.

```
[student@workstation ~]$ lab tuning-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab tuning-review finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab tuning-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- Служба **tuned** автоматически изменяет настройки устройства в соответствии с конкретными требованиями системы на основе предварительно определенного выбранного профиля **tuned**.
- Чтобы отменить все изменения, внесенные в настройки системы выбранным профилем, либо переключитесь на другой профиль, либо отключите службу **tuned**.
- Система назначает относительный приоритет процессу для определения доступа к ЦП. Этот приоритет называется **nice** процесса.
- Команда **nice** назначает приоритет процессу при его запуске. Команда **renice** изменяет приоритет запущенного процесса.

ГЛАВА 4

УПРАВЛЕНИЕ ДОСТУПОМ К ФАЙЛАМ С ПОМОЩЬЮ ACL

ЦЕЛЬ

Интерпретируйте и настройте списки управления доступом (**ACL**) к файлам для обработки ситуаций, требующих сложных прав доступа пользователей и групп.

ЗАДАЧИ

- Опишите варианты использования списков **ACL**, определите файлы, для которых установлены списки **ACL**, и интерпретируйте влияние этих списков **ACL**.
- Установка и удаление **ACL** для файлов и определение **ACL** по умолчанию, автоматически устанавливаемых каталогом для вновь создаваемых файлов.

РАЗДЕЛЫ

- Интерпретация файловых ACL (и упражнения с пошаговыми инструкциями)
- Защита файлов с помощью **ACL** (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Управление доступом к файлам с помощью ACL.

ИНТЕРПРЕТАЦИЯ ACL ФАЙЛОВ

ЗАДАЧИ

После заполнения этого раздела вы сможете:

Описать **ACL** и параметры монтирования файловой системы.

Просматривайте и интерпретируйте **ACL** с помощью **ls** и **getfacl**.

Описать маску **ACL** и приоритет разрешений **ACL**.

Определить, где Red Hat Enterprise Linux по умолчанию использует списки контроля доступа.

КОНЦЕПЦИИ СПИСКА КОНТРОЛЯ ДОСТУПА

Стандартные права доступа к файлам Linux удовлетворительны, когда файлы используются только одним владельцем и одной назначеннной группой людей. Однако в некоторых случаях требуется, чтобы доступ к файлам с разными наборами разрешений осуществлялся несколькими именованными пользователями и группами. Такую функцию предоставляют Списки Контроля Доступа (**ACL**).

С помощью списков ACL вы можете предоставлять разрешения некоторым пользователям и группам, определяемым по имени пользователя, имени группы, UID или GID, используя те же флаги разрешений, которые используются с обычными разрешениями на файлы: **чтение**, **запись** и **выполнить**. Эти дополнительные пользователи и группы, помимо владельца файла и принадлежности файла к группе, называются именованными пользователями (**named users**) и именованными группами (**named groups**) соответственно, потому что они указаны не в длинном списке, а в ACL.

Пользователи могут устанавливать ACL для файлов и каталогов, которыми они владеют. Привилегированные пользователи, которым назначена возможность CAP_FOWNER Linux, могут устанавливать ACL для любого файла или каталога. Новые файлы и подкаталоги автоматически наследуют настройки ACL от ACL родительского каталога по умолчанию, если они установлены. Подобно обычным правилам доступа к файлам, в иерархии родительского каталога необходимо установить, как минимум второе разрешение на поиск (**execute**), чтобы обеспечить доступ именованным пользователям и именованным группам.

Поддержка ACL файловыми системами

Файловые системы должны быть смонтированы с включенной поддержкой **ACL**. Файловые системы **XFS** имеют встроенную поддержку **ACL**. В других файловых системах, таких как **ext3** или **ext4**, созданных в Red Hat Enterprise Linux 8, опция **acl** включена по умолчанию, хотя в более ранних версиях вы должны подтвердить, что поддержка **ACL** включена. Чтобы включить поддержку **ACL** файловой системой, используйте параметр **ACL** с командой **mount** или в записи файловой системы в файле конфигурации **/etc/fstab**.

ПРОСМОТР И ИНТЕРПРЕТАЦИЯ РАЗРЕШЕНИЙ ACL

Команда **ls -l** выводит только минимальные сведения о настройке **ACL**:

```
[user@host content]$ ls -l reports.txt  
-rwxrwx---+ 1 user operators 130 Mar 19 23:56 reports.txt
```

Знак **плюс** (+) в конце 10-символьной строки разрешений указывает на то, что в этом файле существует расширенная структура с записями **ACL**.

user:

Показывает пользовательские настройки **ACL**, которые совпадают со стандартными пользовательскими настройками файлов; **rwx**.

group:

Показывает текущие настройки маски (*mask*) **ACL**, а не настройки владельца группы; **rw**.

other:

Показывает параметры **other ACL**, такие же, как и стандартные параметры других файлов; нет доступа.



ВАЖНО

Изменение разрешений группы для файла с помощью **ACL** с помощью **chmod** не изменяет разрешения владельца группы, но меняет маску **ACL**. Используйте **setfacl -m g :: perms file**, если намерение состоит в том, чтобы обновить права владельца группы файла.

Просмотр списков управления доступом к файлам

Чтобы отобразить настройки **ACL** для файла, используйте команду **getfacl file**:

```
[user@host content]$ getfacl reports.txt  
# file: reports.txt  
# owner: user  
# group: operators  
user::rwx  
user:consultant3:---  
user:1005:rwx      #effective:rwx-  
group::rwx        #effective:rwx-  
group:consultant1:r--  
group:2210:rwx    #effective:rwx-  
mask::rwx-  
other::---
```

Просмотрите каждый раздел предыдущего примера:

Прокомментированные записи:

```
# file: reports.txt  
# owner: user  
# group: operators
```

Первые три строки - это комментарии, в которых указывается имя файла, владелец (пользователь **user**) и владелец группы (операторы **operators**). Если есть какие-либо дополнительные флаги файла, такие как **setuid** или **setgid**, появится четвертая строка комментария, показывающая, какие флаги установлены.

Записи пользователей:

```
user::rwx  
user:consultant3:---  
user:1005:rwx      #effective:rw-
```

1. Разрешения владельца файла у **user rwx**.
2. Разрешения именованных пользователей. Одна запись для каждого указанного пользователя, связанного с этим файлом. **consultant3** не имеет разрешений.
3. Разрешения именованных пользователей. **UID 1005** имеет **rwx**, но маска ограничивает действующие разрешения только **rw**.

Строки относящиеся к group:

```
group::rwx      #effective:rw-  
group:consultant1:r--  
group:2210:rwx    #effective:rw-
```

1. Разрешения владельца группы. Операторы имеют **rwx**, но маска ограничивает действующие разрешения только **rw**.
2. Разрешения именованной группы. По одной записи для каждой именованной группы, связанной с этим файлом. группа **consultant1** имеет только права только на чтение **r**.
3. Разрешения именованной группы. **GID 2210** имеет права **rwx**, но маска ограничивает действующие разрешения только на чтение и запись **rw**.

Запись mask:

```
mask :: rw-
```

Параметры маски показывают максимально возможные разрешения для всех именованных пользователей, владельца группы и именованных групп. **UID 1005**, операторы (**operators**) и **GID 2210** не могут выполнить этот файл, даже если для каждой записи установлено разрешение на выполнение.

Строка other:

```
other :: ---
```

Прочие (**other**) или "мировые" разрешения. Все остальные **UID** и **GID** не имеют разрешений.

Просмотр списков ACL каталогов

Чтобы отобразить настройки **ACL** для каталога, используйте команду **getfacl directory**:

```
[user@host content]$ getfacl .
# file: .
# owner: user
# group: operators
# flags: -s-
user::rwx
user:consultant3:---
user:1005:rwx
group::rwx
group:consultant1:r-x
group:2210:rwx
mask::rwx
other::---
default:user::rwx
default:user:consultant3:---
default:group::rwx
default:group:consultant1:r-x
default:mask::rwx
default:other::---
```

Просмотрите каждый раздел предыдущего примера:

Начинается файл с записей комментариев:

```
# file: .
```

```
# owner: user
# group: operators
# flags: -s-
```

Первые три строки - это комментарии, которые определяют имя каталога, владельца (**user**) и владельца группы (**operators**). Если есть какие-либо дополнительные флаги каталога (**setuid**, **setgid**, **sticky**), то четвертая строка комментария показывает, какие флаги установлены; в этом случае **setgid**.

Стандартные записи ACL:

```
user::rwx
user:consultant3:---
user:1005:rwx
group::rwx
group:consultant1:r-x
group:2210:rwx
mask::rwx
other::---
```

Разрешения **ACL** для этого каталога такие же, как в примере файла, показанном ранее, но применяются к каталогу. Ключевым отличием является включение разрешения на выполнение для этих записей (при необходимости), для разрешения поиска в каталоге.

Записи пользователей по умолчанию:

```
default:user::rwx ①
default:user:consultant3:--- ②
```

1. Разрешения **ACL** владельца файла по умолчанию. Владелец файла получит **rwx**, **read/write** новые файлы и выполняет их в новых подкаталогах.
2. Разрешения **ACL** именованного пользователя по умолчанию. Одна запись для каждого указанного пользователя, который автоматически получит **ACL** по умолчанию, применяемый к новым файлам или подкаталогам. **consultant3** всегда по умолчанию не имеет разрешений.

Записи группы по умолчанию:

```
default:group::rwx ①
default:group:consultant1:r-x ②
```

1. Разрешения ACL владельца группы по умолчанию. Владелец группы файла получит права **rwx**, **read/write** новые файлы и выполняет их в новых подкаталогах.
2. Разрешения ACL именованной группы по умолчанию. Одна запись для каждой названной группы, которая автоматически получит **ACL** по умолчанию. **consultant1** получит **rx**, доступный только для чтения для новых файлов и выполнит его в новых подкаталогах.

Запись mask ACL по умолчанию:

```
default:mask::rwx
```

Параметры маски по умолчанию показывают начальные максимальные разрешения, возможные для всех новых файлов или каталогов, созданных с именованными пользовательскими **ACL**, **ACL** владельца группы или именованными групповыми **ACL**: чтение и запись для новых файлов и разрешение на выполнение в новых подкаталогах. Новые файлы никогда не получают разрешения на выполнение.

Строка other:

```
default:other::---
```

По умолчанию **other** или "мировые" разрешения. Все остальные **UID** и **GID** не имеют разрешений на новые файлы или новые подкаталоги.

Записи по умолчанию в предыдущем примере не включают именованного пользователя (**UID 1005**) и именованной группы (**GID 2210**); следовательно, они не будут автоматически добавлять начальные записи **ACL** в какие-либо новые файлы или новые подкаталоги. Это эффективно ограничивает их файлами и подкаталогами, для которых у них уже есть **ACL**, или если соответствующий владелец файла добавляет **ACL** позже, используя **setfacl**. Они по-прежнему могут создавать свои собственные файлы и подкаталоги.



ПРИМЕЧАНИЕ

Выходные данные команды **getfacl** можно использовать в качестве входных данных для **setfacl** для восстановления списков **ACL** или для копирования списков **ACL** из исходного файла или каталога и сохранения их в новый файл. Например, чтобы восстановить **ACL** из резервной копии, используйте **getfacl -R / dir1> file1**, чтобы сгенерировать рекурсивный выходной файл дампа **ACL** для каталога и его содержимого. Затем выходные данные можно использовать для восстановления исходных списков контроля доступа, передав сохраненные выходные данные в качестве входных данных для команды **setfacl**. Например, чтобы выполнить массовое обновление того же каталога по текущему пути, используйте следующую команду:

```
setfacl --set-file=file1
```

Маска ACL

Маска **ACL** определяет максимальные разрешения, которые вы можете предоставить именованным пользователям, владельцу группы и именованным группам. Он не ограничивает права владельца файла или других пользователей. Все файлы и каталоги, реализующие ACL, будут иметь маску ACL.

Маску можно просмотреть с помощью команды **getfacl** и явно установить с помощью команды **setfacl**. Она будет рассчитана и добавлена автоматически, если она не установлена явно, но она также может быть унаследована от настройки маски родительского каталога по умолчанию. По умолчанию маска пересчитывается всякий раз, когда любой из затронутых ACL добавляется, изменяется или удаляется.

Приоритет разрешений ACL

При определении того, может ли процесс (запущенная программа) получить доступ к файлу, права доступа к файлу и списки управления доступом применяются следующим образом:

- Если процесс запущен от имени пользователя, владеющего файлом, то применяются права доступа пользователя **ACL** к файлу.
- Если процесс выполняется от имени пользователя, указанного в записи ACL именованного пользователя, то применяются разрешения **ACL** именованного пользователя (если это разрешено маской).
- Если процесс выполняется как группа, которая соответствует группе-владельцу файла, или как группа с явно названной записью группового ACL, тогда применяются соответствующие разрешения **ACL** (если это разрешено маской).
- В противном случае применяются другие разрешения **ACL** для файла.

ПРИМЕРЫ ИСПОЛЬЗОВАНИЯ ACL ОПЕРАЦИОННОЙ СИСТЕМОЙ

В Red Hat Enterprise Linux есть примеры, демонстрирующие типичное использование ACL для требований расширенных разрешений.

Списки контроля доступа к файлам журнала Systemd

systemd-journald использует записи **ACL**, чтобы разрешить доступ для чтения к файлу **/run/log/journal/cb44...8ae2/system.journal** группам **adm** и **wheel**. Этот **ACL** позволяет членам групп **adm** и **wheel** иметь доступ для чтения к журналам, управляемым **journalctl**, без необходимости предоставлять специальные разрешения для привилегированного содержимого внутри **/var/log/**, например, к таким как **messages**, **secure** или **audit**.

Из-за конфигурации **systemd-journald** родительская папка файла **system.journal** может измениться, но **systemd-journald** автоматически применяет **ACL** к новой папке и файлу.



ПРИМЕЧАНИЕ

Системные администраторы должны установить **ACL** для папки **/var/log/journal/**, когда **systemd-journald** настроен на использование постоянного хранилища.

```
[user@host ]$ getfacl /run/log/journal/cb44...8ae2/system.journal
getfacl: Removing leading '/' from absolute path names
# file: run/log/journal/cb44...8ae2/system.journal
# owner: root
# group: systemd-journal
user::rw-
group::r--
group:adm:r--
group:wheel:r--
mask::r--
other::---
```

ACL на устройствах, управляемых Systemd

systemd-udev использует набор правил **udev**, которые включают тег **uaccess** для некоторых устройств, таких как CD/DVD-плееры или записывающие устройства, USB-накопители, звуковые карты и многие другие. Упомянутые ранее правила **udev** устанавливают **ACL** на этих устройствах, чтобы позволить пользователям, вошедшим в систему с графическим пользовательским интерфейсом (например, **gdm**), иметь полный контроль над этими устройствами.

Списки **ACL** будут оставаться активными до тех пор, пока пользователь не выйдет из графического интерфейса. Следующий пользователь, который войдет в графический интерфейс, получит новый **ACL**-список для требуемых устройств.

В следующем примере вы можете видеть, что у пользователя есть запись **ACL** с разрешениями **rw**, примененными к устройству **/dev/sr0**, которое является приводом **CD/DVD**.

```
[user@host ]$ getfacl /dev/sr0
getfacl: Removing leading '/' from absolute path names
# file: dev/sr0
# owner: root
# group: cdrom
user::rwuser:
group:rwgroup::
rwmask::
rwother::---
```



РЕКОМЕНДАЦИИ

Справочные страницы **man acl(5)**, **getfacl(1)**, **journald.conf(5)**, **ls(1)**, **systemd-journald(8)** и **systemd-udevd(8)**

КОНТРОЛЬНЫЙ ОПРОС

ИНТЕРПРЕТАЦИЯ ACL-файлов

Сопоставьте следующие элементы с их аналогами в таблице.

default:m::rx /directory

default:user:mary:rx /directory

g::rw /directory

g::rw file

getfacl /directory

group:hug:rwx /directory

user::rx file

user:mary:rx file

ОПИСАНИЕ	РАБОТА ACL
Отобразить ACL в каталоге.	
Именованный пользователь с разрешениями на чтение и выполнение файла.	
Владелец файла с разрешениями на чтение и выполнение для файла.	
Разрешения на чтение и запись для каталога предоставлены владельцу группы каталогов.	
Права на чтение и запись для файла предоставлены владельцу файловой группы.	
Права на чтение, запись и выполнение для каталога предоставлены указанной группе.	
Права на чтение и выполнение установлены в качестве маски по умолчанию.	
Именованному пользователю предоставлено начальное разрешение на чтение для новых файлов, а также на чтение и выполнение для новых подкаталогов.	

РЕШЕНИЕ

ИНТЕРПРЕТАЦИЯ ACL-файлов

Сопоставьте следующие элементы с их аналогами в таблице.

ОПИСАНИЕ	РАБОТА ACL
Отобразить ACL в каталоге.	getfacl /directory
Именованный пользователь с разрешениями на чтение и выполнение файла.	user:mary:rx file
Владелец файла с разрешениями на чтение и выполнение для файла.	user::rx file
Разрешения на чтение и запись для каталога предоставлены владельцу группы каталогов.	g::rw /directory
Права на чтение и запись для файла предоставлены владельцу файловой группы.	g::rw file
Права на чтение, запись и выполнение для каталога предоставлены указанной группе.	group:hug:rwx /directory
Права на чтение и выполнение установлены в качестве маски по умолчанию.	default:m::rx /directory
Именованному пользователю предоставлено начальное разрешение на чтение для новых файлов, а также на чтение и выполнение для новых подкаталогов.	default:user:mary:rx /directory

ЗАЩИТА ФАЙЛОВ С ПОМОЩЬЮ ACL

ЗАДАЧИ

После завершения изучения раздела вы сможете:

- Измените права доступа к обычным файлам **ACL** с помощью **setfacl**.
- Контролировать права доступа к файлам **ACL** по умолчанию для новых файлов и каталогов.

ИЗМЕНЕНИЕ РАЗРЕШЕНИЙ ДЛЯ ФАЙЛА ACL

Используйте **setfacl** для добавления, изменения или удаления стандартных **ACL** для файлов и каталогов.

ACL используют обычное представление разрешений файловой системы: «**r**» - разрешение на чтение, «**w**» - разрешение на запись и «**x**» - разрешение на выполнение. Знак «**-**» (тире) означает, что соответствующее разрешение отсутствует. Когда (рекурсивно) устанавливаются списки управления доступом, верхний регистр «**X**» может использоваться для обозначения того, что разрешение на выполнение должно быть установлено только для каталогов, а не для обычных файлов, если файл уже не имеет соответствующего разрешения на выполнение. Это такое же поведение, что и команды **chmod**.

Добавление или изменение списков ACL

ACL могут быть установлены через командную строку с помощью опции **-m** или переданы через файл с помощью опции **-M** (используйте «**-**» (тире) вместо имени файла для **stdin**). Эти две опции являются опциями «изменения»; они добавляют новые записи **ACL** или заменяют определенные существующие записи **ACL** в файле или каталоге. Любые другие существующие записи **ACL** в файле или каталоге остаются нетронутыми.



ПРИМЕЧАНИЕ

Используйте параметры **--set** или **--set-file**, чтобы полностью заменить настройки **ACL** в файле.

При первом определении **ACL** для файла, если операция добавления не включает настройки для владельца файла, владельца группы или других разрешений, тогда они будут установлены на основе текущих стандартных разрешений для файла (они также известны как **базовый ACL**), записи и не могут быть удалены), а также будет вычислено и добавлено новое значение маски.

Чтобы добавить или изменить *пользовательский* или *именованный пользовательский* **ACL**:

```
[user@host ~]$ setfacl -m u:name:rX file
```

Если имя оставлено пустым, оно применяется к владельцу файла, в противном случае имя может быть **именем пользователя** или **значением UID**. В этом примере предоставленные разрешения будут доступны только для чтения, и, если они уже установлены, выполнить (если только файл не является каталогом, в этом случае каталог получит разрешение execute, чтобы разрешить поиск в каталоге).

Владелец файла **ACL** и стандартные разрешения владельца файла эквивалентны; следовательно, использование **chmod** для разрешений владельца файла эквивалентно использованию **setfacl** для разрешений владельца файла, **chmod** не влияет на права *именованных пользователей*.

Чтобы добавить или изменить **ACL** группы (*group*) или именованной группы (*named group*):

```
[user@host ~]$ setfacl -m g:name:rw file
```

Это происходит по той же схеме, что и при добавлении или изменении записи **ACL** пользователя. Если имя оставлено пустым, оно применяется к владельцу группы. В противном случае укажите имя группы или значение GID для именованной группы. В этом примере разрешения будут доступны для чтения и записи.

chmod не влияет на права группы для файлов с настройками **ACL**, но обновляет маску **ACL**.

Чтобы добавить или изменить права для прочих (*other*) **ACL**:

```
[user@host ~]$ setfacl -m o:::- file
```

Прочие (*other*) принимает только настройки разрешений. Типичные настройки разрешений для прочих (*other*): отсутствие разрешений вообще, указывается тире (-); и разрешения только на чтение устанавливаются как обычно с помощью **r**. Конечно, вы можете установить любые стандартные разрешения.

Разрешения **ACL** для прочих (*other*) и стандартные прочие (*other*) разрешения эквивалентны, поэтому использование команды **chmod** для разрешений *other* эквивалентно использованию **setfacl** для установления разрешений для *other*.

Вы **можете** добавить несколько записей одной командой; используйте список записей, разделенных запятыми:

```
[user@host ~]$ setfacl -m u::rwx,g:consultants:rX,o:::- file
```

В приведённом примере устанавливается право *владельца файла* на **чтение, запись и выполнение**, устанавливает для именованной группы **consultants** только **чтение** и условное выполнение и ограничивает всех прочих (*other*) пользователей разрешениями *no*. Владелец группы поддерживает существующие разрешения для файлов или ACL, а другие «именованные» записи остаются неизменными.

Использование **getfacl** в качестве ввода

Вы можете использовать вывод команды **getfacl** в качестве входных данных для команды **setfacl**:

```
[user@host ~]$ getfacl file-A | setfacl --set-file=- file-B
```

Параметр **--set-file** принимает ввод из файла или из стандартного ввода. Знак дефиса (-) указывает на использование стандартного ввода. В этом случае файл **-B** будет иметь те же настройки ACL, что и файл **-A**.

Установка явной маски ACL

Вы можете явно установить маску ACL для файла или каталога, чтобы ограничить максимальные эффективные разрешения для именованных пользователей, владельца группы и именованных групп. Это ограничивает любые существующие разрешения, превышающие маску, но не влияет на разрешения, которые являются менее разрешительными, чем маска.

```
[user@host ~]$ setfacl -m m:::r file
```

Команда добавляет значение маски, которое ограничивает всех *именованных пользователей, владельца группы* и все именованные группы разрешением только для чтения, независимо от их существующих настроек. Настройка маски не влияет на права *владельца файла* и права пользователей *other*.

Команда **getfacl** показывает эффективный комментарий рядом с записями, которые ограничены настройкой маски.



ВАЖНО

По умолчанию, каждый раз, когда одна из затронутых настроек ACL (именованные пользователи, владелец группы или именованные группы) изменяется или удаляется, маска ACL пересчитывается, потенциально сбрасывая предыдущую явную настройку маски.

Чтобы избежать пересчета маски, используйте параметр **-n** или включите параметр маски (**-m m :: perms**) с любой операцией **setfacl**, которая изменяет параметры **ACL**, связанные с маской.

Рекурсивные модификации ACL

При настройке **ACL** для каталога используйте параметр **-R** для рекурсивного применения **ACL**. Помните, что вы, вероятно, захотите использовать разрешение «**X**» (заглавная буква **X**) с рекурсией, чтобы файлы с набором разрешений на выполнение сохраняли настройку, а каталоги получали набор разрешений на выполнение, позволяющий выполнять поиск в каталогах. Считается хорошей практикой также использовать прописную букву «**X**» при не рекурсивной настройке списков управления доступом, поскольку это предотвращает случайное добавление администраторами разрешений на выполнение к обычному файлу.

```
[user@host ~]$ setfacl -R -m u:name:rX directory
```

Это добавляет имя пользователя (*name*) в каталог (*directory*) каталога и все выполняемые файлы, и подкаталоги, устанавливая разрешения только на чтение и условное выполнение.

Удаление ACL

Удаление определенных записей **ACL** выполняется в том же базовом формате, что и операция изменения, за исключением того, что "**:perms**" не указывается.

```
[user@host ~]$ setfacl -x u:name,g:name file
```

Предыдущая команда удаляет только названного пользователя и названную группу из **ACL** файла или каталога. Любые другие существующие записи **ACL** остаются активными.

Вы можете включить операции удаления (**-x**) и изменения (**-m**) в одну и ту же операцию **setfacl**.

Маска может быть удалена только в том случае, если не установлены другие **ACL** (за исключением базового (**base**) **ACL**, который нельзя удалить), поэтому ее необходимо удалить последней. У файла больше не будет **ACL**, и команда **ls -l** не будет отображать знак плюса (+) рядом со строкой разрешений. В качестве альтернативы, чтобы удалить все (*all*) записи **ACL** в файле или каталоге (включая **ACL** по умолчанию (**default**) для каталогов), используйте следующую команду:

```
[user@host ~]$ setfacl -b file
```

УПРАВЛЕНИЕ РАЗРЕШЕНИЯМИ ФАЙЛА ACL ПО УМОЛЧАНИЮ

Чтобы файлы и каталоги, созданные в каталоге, наследовали определенные ACL, используйте ACL по умолчанию (*default*) для каталога. Вы можете установить ACL по умолчанию и любые стандартные настройки ACL, включая маску по умолчанию.

Сам каталог по-прежнему требует стандартных ACL для управления доступом, потому что ACL по умолчанию не реализуют управление доступом для каталога; они предоставляют только поддержку наследования разрешений ACL. Например:

```
[user@host ~]$ setfacl -m d:u:name:rx directory
```

Команда в предыдущем примере добавляет именованного пользователя по умолчанию (**d:u:name**) с разрешением только для чтения и разрешением на выполнение в подкаталогах.

Команда **setfacl** для добавления **ACL** по умолчанию для каждого из типов **ACL** точно такая же, как и для стандартных **ACL**, но с префиксом **d:**. Или используйте опцию **-d** в командной строке.



ВАЖНО

При настройке **ACL** по умолчанию для каталога убедитесь, что пользователи смогут получить доступ к содержимому новых подкаталогов, созданных в нем, путем включения разрешения на выполнение в список управления доступом по умолчанию.

Пользователи не будут автоматически получать разрешения на выполнение для вновь создаваемых обычных файлов, потому что, в отличие от новых каталогов, маска **ACL** для нового обычного файла **rw-**.



ПРИМЕЧАНИЕ

Новые файлы и новые подкаталоги продолжают получать значения **UID** своего владельца и **GID** первичной группы от создавшего пользователя, за исключением случаев, когда установлен флаг **setgid** родительского каталога, и в этом случае **GID** первичной группы совпадает с **GID** родительского каталога.

Удаление записей **ACL** по умолчанию

Удаляются **ACL** по умолчанию так же, как вы удаляете стандартный **ACL**, с префиксом **d:** или используйте параметр **-d**.

```
[user@host ~]$ setfacl -x d:u:name directory
```

Команда в примере удаляет запись **ACL** по умолчанию, которая была добавлена в предыдущем примере.

Чтобы удалить все записи **ACL** по умолчанию в каталоге, используйте команду **setfacl -k directory**.



РЕКОМЕНДАЦИИ

Справочные страницы **man acl(5)**, **setfacl(1)**, и **getfacl(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ЗАЩИТА ФАЙЛОВ С ПОМОЩЬЮ ACL

В этом упражнении вы будете использовать записи **ACL**, чтобы предоставить доступ к каталогу для группы и запретить доступ для пользователя, установить **ACL** по умолчанию для каталога и подтвердить, что новые файлы, созданные в этом каталоге, наследуют **ACL** по умолчанию.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Используйте записи **ACL**, чтобы предоставить доступ группе и запретить доступ одному из ее членов.
- Убедиться, что существующие файлы и каталоги отражают новые разрешения **ACL**.
- Установить **ACL** по умолчанию для каталога и убедитесь, что новые файлы и каталоги наследуют его конфигурацию.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab acl-secure start**. Эта команда запускает сценарий, который определяет, доступна ли хост **servera** в сети. Скрипт также создает пользователей, группы, каталоги и файлы, использованные в этом упражнении.

```
[student@workstation ~]$ lab acl-secure start
```

Operators и **Consultants** являются членами компании по поддержке ИТ. Им нужно начать делиться информацией. **servera** содержит правильно настроенный общий каталог, расположенный в **/share/content**, в котором размещаются файлы.

В настоящее время только члены группы **operators** имеют доступ к этому каталогу, но и членам группы **consultants** необходим полный доступ к этому каталогу.

Пользователь **consultant1** является членом группы **consultants**, но неоднократно вызывал проблемы, поэтому у этого пользователя не должно быть доступа к каталогу.

Ваша задача - добавить соответствующие записи **ACL** в каталог и его содержимое, чтобы члены группы **consultants** имели полный доступ, но запретили пользователю **consultant1** в любом доступе. Убедитесь, что для будущих файлов и каталогов, хранящихся в **/share/content**, применяются соответствующие записи **ACL**.

Важная информация:

- Пользователи **sysadmin1** и **operator1** являются членами группы **operators**.
- Пользователи **consultant1** и **consultant2** являются членами группы **consultants**.

- Каталог **/share/content** содержит подкаталог с именем **server-info** и множество файлов для проверки **ACL**. Кроме того, каталог **/share/content** содержит исполняемый сценарий с именем **loadvg.sh**, который вы можете использовать для тестирования.
- Для пользователей **sysadmin1**, **operator1**, **consultant1** и **consultant2** установлены пароли слово **redhat**.
- Все изменения должны происходить в каталоге **/share/content** и его файлах; не изменяйте каталог **/share**.

1. Войдите на сервер **servera** и переключитесь на пользователя **root**.

1.1. Используйте команду **ssh** для входа на сервер **servera** в качестве **student**. Системы настроены на использование ключей **SSH**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

1.2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Пароль для пользователя **student** - **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

2. Добавьте именованный **ACL** в каталог **/share/content** и все его содержимое.

2.1. Используйте **setfacl** для рекурсивного обновления каталога **/share/content**, предоставляя группе **consultants** права на чтение, запись и условное выполнение.

```
[root@servera ~]# setfacl -Rm g:consultants:rwx /shares/content
```

Параметр **-R** означает рекурсивный, параметр **-m** означает изменение/добавление, **rwx** означает применение разрешений на чтение, запись и условное выполнение.

2.2. Используйте **setfacl** для рекурсивного обновления каталога **/share/content**, запрещая пользователю **consultant1** из группы **consultants** любой доступ.

```
[root@servera ~]# setfacl -Rm u:consultant1:- /shares/content
```

Параметр **-R** означает рекурсивный, параметр **-m** означает изменение/добавление, **-** означает запрет доступа.

3. Добавьте именованный **ACL** в качестве **ACL** по умолчанию для поддержки будущих добавлений файлов и каталогов.

3.1. Используйте **setfacl**, чтобы добавить правило доступа по умолчанию для группы **consultants**. Предоставьте разрешения на чтение, запись и выполнение для каталога **content**.

```
[root@servera ~]# setfacl -m d:g:consultants:rwx /shares/content
```

Параметры **-m** означают изменение/добавление, **d:g** означает группу по умолчанию, **rwx** означает применение разрешений на чтение/запись/выполнение (необходимо для правильного создания подкаталога и доступа)

3.2. Используйте **setfacl**, чтобы добавить правило доступа по умолчанию для пользователя **consultant1**. Запретив любой доступ к каталогу содержимого.

```
[root@servera ~]# setfacl -m d:u:consultant1:- /shares/content
```

Параметр **-m** означает изменение/добавление, **d:u** означает пользователя по умолчанию, **-** означает отсутствие разрешений.

4. Проверьте изменения **ACL**.

consultant2 должен иметь возможность читать любой файл и создавать новый каталог с новым файлом в нем.

consultant1 не должен иметь возможность читать, писать или выполнять какие-либо файлы; это включает невозможность перечислить содержимое каталога.

Используйте **su - user**, чтобы переключиться на ваших тестовых пользователей. Используйте **exit** или **Ctrl + D**, чтобы выйти из тестовой пользовательской оболочки.

```
[root@servera ~]# exit  
[student@servera ~]$ su - consultant2  
Password: redhat  
[consultant2@servera ~]$ cd /shares/content/
```

4.1. Используйте **cat**, чтобы проверить, может ли **consultant2** читать файл.

```
[consultant2@servera content]$ cat serverb-loadavg.txt  
#####  
serverb.lab.example.com  
#####  
Wed Mar 25 15:25:19 EDT 2019  
#####
```

```
ldavg 0.18, 0.06, 0.05
#####
#####
```

- 4.2. Используйте сценарий **loadavg.sh**, чтобы проверить, может ли **consultant2** выполнить файл.

```
[consultant2@servera content]$ ./loadavg.sh
ldavg 0.00, 0.00, 0.04
```

- 4.3. Создайте каталог с названием **reports**.

Используйте **echo** для создания файла с некоторым содержимым, назовите файл **test.txt** и поместите его в новый каталог.

Когда закончите, вернитесь к пользователю **student**.

```
[consultant2@servera content]$ mkdir reports
[consultant2@servera content]$ echo "TEST REPORT" > reports/test.txt
[consultant2@servera content]$ exit
logout
[student@servera ~]$
```

- 4.4. Авторизуйтесь как пользователь **consultant1**. Используйте команду **cd**, чтобы попытаться перейти в каталог как **consultant1**, а также попробуйте **ls** для вывода списка каталога. Обе команды должны завершиться ошибкой **Permission denied**.

Попробуйте одну или несколько команд, которые использовал **consultant2**, но как **consultant1**, чтобы дополнительно проверить отсутствие доступа. Используйте полный путь **/shares/content**, потому что вы не можете использовать **cd** для перехода в каталог.

Когда вы закончите тестирование **consultant1**, вернитесь к учётной записи **student**.

```
[student@servera ~]$ su - consultant1
Password: redhat
[consultant1@servera ~]$ cd /shares/content/
-bash: cd: /shares/content/: Permission denied
[consultant1@servera ~]$ ls /shares/content/
ls: cannot open directory '/shares/content/': Permission denied
[consultant1@servera ~]$ cat /shares/content/serverb-loadavg.txt
cat: /shares/content/serverb-loadavg.txt: Permission denied
[consultant1@servera ~]$ exit
Logout
```

- 4.5. Войдите в систему как пользователь **sysadmin1**. Используйте **getfacl**, чтобы просмотреть все записи **ACL** в **/shares/content** и записи **ACL** в **/share/content/reports**.

Когда вы закончите тестирование **consultant1**, вернитесь к учётной записи **student**.

```
[student@servera ~]$ su - sysadmin1
Password: redhat
[sysadmin1@servera ~]$ getfacl /shares/content
getfacl: Removing leading '/' from absolute path names
# file: shares/content/
# owner: root
# group: operators
# flags: -suser:::
rwx
user:consultant1:---
group::rwx
group:consultants:rwx
mask::rwx
other::---
default:user::rwx
default:user:consultant1:---
default:group::rwx
default:group:consultants:rwx
default:mask::rwx
default:other:---

[sysadmin1@servera ~]$ getfacl /shares/content/reports
getfacl: Removing leading '/' from absolute path names
# file: shares/content/reports
# owner: consultant2
# group: operators
# flags: -suser:::
rwx
user:consultant1:---
group::rwx
group:consultants:rwx
mask::rwx
other::---
default:user::rwx
default:user:consultant1:---
default:group::rwx
default:group:consultants:rwx
default:mask::rwx
default:other:---

[sysadmin1@servera ~]$ exit
logout
```

4.6. Выйти из сервера **servera**.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab acl-secure finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab acl-secure finish
```

На этом пошаговое упражнение завершено.

ЛАБОРАТОРНАЯ РАБОТА

УПРАВЛЕНИЕ ДОСТУПОМ К ФАЙЛАМ С ПОМОЩЬЮ ACL

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы создадите общий каталог для пользователей в двух группах, объединив разрешение **set-GID** и записи **ACL** по умолчанию, чтобы обеспечить правильные разрешения доступа.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

Настроить разрешение **set-GID** для папки, чтобы наследовать групповое владение файлами и папками внутри.

Настроить записи **ACL**, чтобы разрешить или запретить пользователям и группам доступ на **чтение/запись/выполнение** к файлам и каталогам.

Настроить **ACL** по умолчанию, чтобы автоматически получать правильные ACL и права доступа к новым файлам и каталогам.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab acl-review start**. Команда запускает сценарий, который определяет, доступен ли сервер **serverb** в сети. Он также создает пользователей, группы, каталоги и файлы, используемые в этом упражнении.

```
[student@workstation ~]$ lab acl-review start
```

Агентство по финансированию акций создает общий каталог для хранения файлов дел, для которых члены группы **managers** будут иметь разрешения на чтение и запись.

Соучредитель агентства **manager1** решил, что члены группы подрядчиков (**contractors**) также должны иметь возможность читать и писать в общий каталог. Однако **manager1** не доверяет пользователю **contractor3** (члену группы **contractors**), и поэтому **contractor3** должен иметь доступ к каталогу только для чтения.

manager1 создал пользователей и группы и начал процесс настройки общего каталога, копируя некоторые файлы шаблонов. Поскольку **manager1** был слишком занят, вам придется закончить работу.

Ваша задача - завершить настройку общего каталога. Каталог и все его содержимое должны принадлежать группе менеджеров (**managers**), а файлы обновляются для чтения и записи для владельца и группы (**managers**). У других пользователей не должно быть разрешений. Вам также

необходимо предоставить разрешения на чтение и запись для группы **contractors**, за исключением **contractor3**, который получает разрешения только на чтение. Убедитесь, что ваши настройки применимы к существующим и будущим файлам.

Важная информация:

- Общий каталог: **/shares/cases** на **serverB**.
- Пользователи **manager1** и **manager2** входят в группу **managers**.
- Пользователи «**contractor1**», «**contractor2**» и «**contractor3**» входят в группу **contractors**.
- В каталоге есть два файла: **shortlist.txt** и **backlog.txt**.
- Все пять паролей пользователей - **redhat**.
- Все изменения должны происходить в каталоге **/shares/cases** и его файлах; не изменяйте каталог **/share**.

1. Директория **cases** и его содержимое должны принадлежать группе **managers**. Новые файлы, добавленные в каталог **cases**, должны автоматически принадлежать группе **managers**. Владельцы пользователей и групп для существующих файлов должны иметь права на чтение и запись, а другие пользователи не должны иметь никаких разрешений вообще.



ПРИМЕЧАНИЕ

Подсказка: не используйте **setfacl**.

2. Добавьте записи **ACL** в каталог **cases** (и его содержимое), которые позволяют членам группы **contractors** иметь доступ на чтение/запись к файлам и выполнение разрешений в каталоге. Ограничите пользователя **contractor3** для чтения файлов и разрешения на выполнение в каталоге.
3. Добавьте записи **ACL**, которые гарантируют, что все новые файлы или каталоги в каталоге **cases** имеют правильные разрешения, применяемые для всех авторизованных пользователей и групп.
4. Убедитесь, что вы правильно внесли изменения в **ACL** и файловую систему.
Используйте команды **ls** и **getfacl**, чтобы просмотреть свои настройки в каталоге **/share/cases**. В качестве пользователя **student** используйте **su -user**, чтобы переключиться сначала на пользователя **manager1**, а затем на пользователя **contractor1**. Убедитесь, что вы можете записывать в файл, читать из файла, создавать каталог и записывать в файл в новом каталоге. Используйте команду **ls**, чтобы проверить новые права доступа к каталогу, и **getfacl**, чтобы просмотреть новый **ACL** каталога.
В качестве пользователя **student** используйте **su -contractor3** для переключения пользователя. Попробуйте записать в файл (он должен потерпеть неудачу) и попытаться создать новый каталог (он должен потерпеть неудачу). Как пользователь **contractor3**, вы должны иметь возможность читать из файла **shortlist.txt** в каталоге **cases**, и вы должны иметь возможность читать из «**test**» файлов, записанных в любом из новых каталогов, созданных пользователями **manager1** и **contractor1**.



ПРИМЕЧАНИЕ

Приведенный выше набор тестов - это некоторые из тестов, которые вы можете выполнить, чтобы проверить правильность разрешений на доступ. Вам следует разработать соответствующие тесты проверки доступа для своей среды.

Оценка

На рабочей станции запустите команду **lab acl-review grade**, чтобы подтвердить успех выполнения упражнения.

Завершение

На рабочей станции **workstation**, выполните команду **lab acl-review finish**, чтобы завершить данное упражнение.

```
[student@workstation ~]$ lab acl-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

УПРАВЛЕНИЕ ДОСТУПОМ К ФАЙЛАМ С ПОМОЩЬЮ ACL

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы создадите общий каталог для пользователей в двух группах, объединив разрешение **set-GID** и записи **ACL** по умолчанию, чтобы обеспечить правильные разрешения доступа.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

Настроить разрешение **set-GID** для папки, чтобы наследовать групповое владение файлами и папками внутри.

Настроить записи **ACL**, чтобы разрешить или запретить пользователям и группам доступ на **чтение/запись/выполнение** к файлам и каталогам.

Настроить **ACL** по умолчанию, чтобы автоматически получать правильные ACL и права доступа к новым файлам и каталогам.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab acl-review start**. Команда запускает сценарий, который определяет, доступен ли сервер **serverb** в сети. Он также создает пользователей, группы, каталоги и файлы, используемые в этом упражнении.

```
[student@workstation ~]$ lab acl-review start
```

Агентство по финансированию акций создает общий каталог для хранения файлов дел, для которых члены группы **managers** будут иметь разрешения на чтение и запись.

Соучредитель агентства **manager1** решил, что члены группы подрядчиков (**contractors**) также должны иметь возможность читать и писать в общий каталог. Однако **manager1** не доверяет пользователю **contractor3** (члену группы **contractors**), и поэтому **contractor3** должен иметь доступ к каталогу только для чтения.

manager1 создал пользователей и группы и начал процесс настройки общего каталога, копируя некоторые файлы шаблонов. Поскольку **manager1** был слишком занят, вам придется закончить работу.

Ваша задача - завершить настройку общего каталога. Каталог и все его содержимое должны принадлежать группе менеджеров (**managers**), а файлы обновляются для чтения и записи для владельца и группы (**managers**). У других пользователей не должно быть разрешений. Вам также

необходимо предоставить разрешения на чтение и запись для группы **contractors**, за исключением **contractor3**, который получает разрешения только на чтение. Убедитесь, что ваши настройки применимы к существующим и будущим файлам.

Важная информация:

- Общий каталог: **/shares/cases** на **serverb**.
- Пользователи **manager1** и **manager2** входят в группу **managers**.
- Пользователи «**contractor1**», «**contractor2**» и «**contractor3**» входят в группу **contractors**.
- В каталоге есть два файла: **shortlist.txt** и **backlog.txt**.
- Все пять паролей пользователей - **redhat**.
- Все изменения должны происходить в каталоге **/shares/cases** и его файлах; не изменяйте каталог **/share**.

1. Директория **cases** и его содержимое должны принадлежать группе **managers**. Новые файлы, добавленные в каталог **cases**, должны автоматически принадлежать группе **managers**. Владельцы пользователей и групп для существующих файлов должны иметь права на чтение и запись, а другие пользователи не должны иметь никаких разрешений вообще.



ПРИМЕЧАНИЕ

Подсказка: не используйте **setfacl**.

- 1.1. Войдите на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 1.2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Пароль для пользователя - **student** будет слово **student**.

```
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

- 1.3. Используйте команду **chgrp** для рекурсивного обновления группового владения каталогом и его содержимым.

```
[root@serverb ~]# chgrp -R managers /shares/cases
```

- 1.4. Используйте команду **chmod**, чтобы обновить флаг **set-GID** в каталоге.

```
[root@serverb ~]# chmod g+s /shares/cases
```

- 1.5.** Используйте **chmod**, чтобы обновить все существующие права доступа к файлам до **rw** для владельца и группы.

```
[root@serverb ~]# chmod 660 /shares/cases/*
```

- 2.** Добавьте записи **ACL** в каталог **cases** (и его содержимое), которые позволяют членам группы **contractors** иметь доступ на чтение/запись к файлам и выполнение разрешений в каталоге. Ограничьте пользователя **contractor3** для чтения файлов и разрешения на выполнение в каталоге.

- 2.1.** Используйте **setfacl** для рекурсивного обновления существующего каталога **cases** и его содержимого. Предоставьте группе **contractors** разрешения на чтение, запись и условное выполнение.

```
[root@serverb ~]# setfacl -Rm g:contractors:rwx /shares/cases
```

- 2.2.** Используйте **setfacl** для рекурсивного обновления существующего каталога **cases** и его содержимого. Предоставьте пользователю **contractor3** разрешения на чтение и условное выполнение.

```
[root@serverb ~]# setfacl -Rm u:contractor3:rX /shares/cases
```

- 3.** Добавьте записи **ACL**, которые гарантируют, что все новые файлы или каталоги в каталоге **cases** имеют правильные разрешения, применяемые для всех авторизованных пользователей и групп.

- 3.1.** Используйте **setfacl** для обновления разрешений по умолчанию для членов группы **contractors**. Разрешения по умолчанию - это чтение, запись и выполнение (необходимы для правильного создания подкаталогов и доступа к ним).

```
[root@serverb ~]# setfacl -m d:g:contractors:rwx /shares/cases
```

- 3.2.** Используйте **setfacl** для обновления разрешений по умолчанию для пользователя **contractor3**. Разрешения по умолчанию - чтение и выполнение (необходимо для правильного доступа к подкаталогам).

```
[root@serverb ~]# setfacl -m d:u:contractor3:rx /shares/cases
```

4. Убедитесь, что вы правильно внесли изменения в **ACL** и файловую систему.

Используйте команды **ls** и **getfacl**, чтобы просмотреть свои настройки в каталоге **/share/cases**. В качестве пользователя **student** используйте **su -user**, чтобы переключиться сначала на пользователя **manager1**, а затем на пользователя **contractor1**. Убедитесь, что вы можете записывать в файл, читать из файла, создавать каталог и записывать в файл в новом каталоге. Используйте команду **ls**, чтобы проверить новые права доступа к каталогу, и **getfacl**, чтобы просмотреть новый **ACL** каталога.

В качестве пользователя **student** используйте **su -contractor3** для переключения пользователя. Попробуйте записать в файл (он должен потерпеть неудачу) и попытаться создать новый каталог (он должен потерпеть неудачу). Как пользователь **contractor3**, вы должны иметь возможность читать из файла **shortlist.txt** в каталоге **cases**, и вы должны иметь возможность читать из «**test**» файлов, записанных в любом из новых каталогов, созданных пользователями **manager1** и **contractor1**.

4.1. Как пользователь **root** используйте команду **ls** для проверки каталога **cases** и его содержимого. Ищите групповое владение, права доступа к каталогам и файлам. Символ «**S**» в разрешениях группового файла указывает, что установлен флаг **set-GID**, а «**+**» указывает, что записи **ACL** существуют. В конце выйдите из сеанса пользователя **root**.

```
[root@serverb ~]# ls -ld /shares/cases
drwxrws---+ 2 root managers 46 Mar 29 00:40 /shares/cases
[root@serverb ~]# ls -l /shares/cases
total 8
-rw-rw----+ 1 root managers 44 Mar 29 00:33 backlog.txt
-rw-rw----+ 1 root managers 46 Mar 29 00:33 shortlist.txt
```

4.2. Используйте **getfacl** и просмотрите его вывод. Ищите записи именованного пользователя и именованной группы как в стандартном **ACL**, так и в **ACL** по умолчанию.

```
[root@serverb ~]# getfacl /shares/cases
# file: shares/cases
# owner: root
# group: managers
# flags: -suser::
rwx
user:contractor3:r-x
group::rwx
group:contractors:rwx
mask::rwx
other::---
default:user::rwx
default:user:contractor3:r-x
default:group::rwx
default:group:contractors:rwx
default:mask::rwx
default:other::---

[root@serverb ~]# exit
logout
```

- 4.3.** Переключитесь на пользователя **manager1** и выполните следующие операции. Убедитесь, что вы получаете ожидаемое поведение доступа.

```
[student@serverb ~]$ su - manager1
Password: redhat
[manager1@serverb ~]$ cd /shares/cases
[manager1@serverb cases]$ echo hello > manager1.txt
[manager1@serverb cases]$ cat shortlist.txt
###Shortlist of Clients to call###TEMPLATE###
[manager1@serverb cases]$ mkdir manager1.dir
[manager1@serverb cases]$ echo hello > manager1.dir/test.txt
[manager1@serverb cases]$ ls -ld manager1.dir
drwxrws---+ 2 manager1 managers 22 Mar 29 00:59 manager1.dir
[manager1@serverb cases]$ ls -l manager1.dir
total 4
-rw-rw----+ 1 manager1 managers 6 Mar 29 00:59 test.txt
[manager1@serverb cases]$ getfacl manager1.dir
# file: manager1.dir/
# owner: manager1
# group: managers
# flags: -suser::
rwx
user:contractor3:r-x
group::rwx
group:contractors:rwx
mask::rwx
other::---
default:user::rwx
default:user:contractor3:r-x
default:group::rwx
default:group:contractors:rwx
default:mask::rwx
default:other::---

[manager1@serverb cases]$ exit
Logout
```

- 4.4.** Переключитесь на пользователя **contractor1** и выполните следующие операции. Убедитесь, что вы получаете ожидаемое поведение доступа.

```
[student@serverb ~]$ su - contractor1
Password: redhat
[contractor1@serverb ~]$ cd /shares/cases
[contractor1@serverb cases]$ echo hello > manager1.txt
[contractor1@serverb cases]$ cat shortlist.txt
###Shortlist of Clients to call###TEMPLATE###
[contractor1@serverb cases]$ mkdir contractor1.dir
[contractor1@serverb cases]$ echo hello > contractor1.dir/test.txt
[contractor1@serverb cases]$ ls -ld contractor1.dir
drwxrws---+ 2 contractor1 managers 22 Mar 29 01:05 contractor1.dir
```

```
[contractor1@serverb cases]$ ls -l contractor1.dir
total 4
-rw-rw----+ 1 contractor1 managers 6 Mar 29 01:07 test.txt
[manager1@serverb cases]$ getfacl contractor1.dir
# file: contractor1.dir/
# owner: contractor1
# group: managers
# flags: -suser::
rwx
user:contractor3:r-x
group::rwx
group:contractors:rwx
mask::rwx
other::---
default:user::rwx
default:user:contractor3:r-x
default:group::rwx
default:group:contractors:rwx
default:mask::rwx
default:other::---

[contractor1@serverb cases]$ exit
Logout
```

- 4.5. Переключитесь на пользователя **contractor3** и выполните следующие операции. Убедитесь, что вы получаете ожидаемое поведение доступа.

```
[student@serverb ~]# su - contractor3
Password: redhat
[contractor3@serverb ~]# cd /shares/cases
[contractor3@serverb cases]# echo hello > contractor3.txt
-bash: contractor3.txt: Permission denied
[contractor3@serverb cases]# cat shortlist.txt
###Shortlist of Clients to call###TEMPLATE###
[contractor3@serverb cases]# mkdir contractor3.dir
mkdir: cannot create directory ‘contractor3.dir’ : Permission denied
[contractor3@serverb cases]# cat manager1.dir/test.txt
hello
[contractor3@serverb cases]# cat contractor1.dir/test.txt
hello
[contractor3@serverb cases]# exit
logout
```

- 4.6. Выйти с сервера **serverb**.

```
[student@serverb ~]# exit
logout
Connection to serverb closed.
[student@workstation ~]$
```



ПРИМЕЧАНИЕ

Приведенный выше набор тестов - это некоторые из тестов, которые вы можете выполнить, чтобы проверить правильность разрешений на доступ. Вам следует разработать соответствующие тесты проверки доступа для своей среды.

Оценка

На рабочей станции запустите команду **lab acl-review grade**, чтобы подтвердить успех выполнения упражнения.

Завершение

На рабочей станции **workstation**, выполните команду **lab acl-review finish**, чтобы завершить данное упражнение.

```
[student@workstation ~]$ lab acl-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- **ACL** обеспечивают детальный контроль доступа к файлам и каталогам.
- Команда **getfacl** отображает списки **ACL** для файла или каталога.
- Команда **setfacl** устанавливает, изменяет и удаляет стандартные и стандартные ACL для файлов и каталогов.
- Используйте списки управления доступом **ACLs** по умолчанию для управления разрешениями новых файлов и каталогов.
- Red Hat Enterprise Linux использует **systemd** и **udev** для применения предопределенных списков контроля доступа **ACLs** к устройствам, папкам и файлам.

ГЛАВА 5

УПРАВЛЕНИЕ ДОСТУПОМ К ФАЙЛАМ С ПОМОЩЬЮ ACL

ЦЕЛЬ

Защищайте и управляйте безопасностью сервера с помощью SELinux.

ЗАДАЧИ

- Опишите, как SELinux защищает ресурсы и как выбрать режим принудительного применения.
- Настройте контекст SELinux файла, чтобы контролировать взаимодействие процессов с этим файлом.
- Настройте логические значения SELinux, чтобы разрешить изменение политики времени выполнения для различных потребностей доступа.
- Исследуйте сообщения журнала SELinux и устраняйте отказы SELinux AVC.

РАЗДЕЛЫ

- Изменение режима принудительного применения SELinux (и упражнения с пошаговыми инструкциями)
- Управление контекстами файлов SELinux (и упражнения с пошаговыми инструкциями)
- Настройка политики SELinux с помощью с помощью логических параметров (флагов) (и упражнения с пошаговыми инструкциями)
- Исследование и решение проблем SELinux (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Управление безопасностью SELinux.

ИЗМЕНЕНИЕ РЕЖИМА ЗАЩИТЫ SELINUX

ЗАДАЧИ

После завершения этого раздела вы сможете:

- Объяснить, как **SELinux** защищает ресурсы.
- Изменить текущий режим **SELinux** в системе.
- Установить режим **SELinux** по умолчанию для системы.

КАК SELINUX ЗАЩИЩАЕТ РЕСУРСЫ

SELinux обеспечивает критическую цель безопасности в Linux, разрешая или запрещая доступ к файлам и другим ресурсам, которые являются значительно более точными, чем разрешения пользователей.

Права доступа к файлам определяют, какие пользователи или группы пользователей могут получить доступ к каким конкретным файлам. Однако пользователь, которому предоставлен доступ для чтения или записи к любому конкретному файлу, может использовать этот файл любым способом, который выберет пользователь, даже если это использование не является тем, как файл должен использоваться.

Например, с доступом на запись в файл, следует ли разрешить открывать и изменять файл структурированных данных, предназначенный для записи с использованием только определенной программы, другими редакторами, что может привести к повреждению?

Права доступа к файлам не могут остановить такой нежелательный доступ. Они никогда не были предназначены для управления использованием файла, а только для того, кому разрешено читать, писать или запускать файл.

SELinux состоит из наборов политик, определенных разработчиками приложения, которые точно декларируют, какие действия и доступы являются правильными и разрешенными для каждого исполняемого двоичного файла, файла конфигурации и файла данных, используемых приложением. Это называется целевой политикой (*targeted policy*), потому что одна политика написана для охвата действий отдельного приложения. Политики объявляют предопределенные метки (*labels*), которые размещаются на отдельных программах, файлах и сетевых портах.

ЗАЧЕМ ИСПОЛЬЗОВАТЬ УЛУЧШЕННУЮ БЕЗОПАСНОСТЬ LINUX?

Не все проблемы безопасности можно предсказать заранее. **SELinux** применяет набор правил доступа, не позволяющих уязвимости одного приложения влиять на другие приложения или базовую систему. **SELinux** обеспечивает дополнительный уровень безопасности; это также добавляет уровень сложности, который может оттолкнуть людей, плохо знакомых с этой подсистемой. Обучение работе с **SELinux** может занять время, но политика принудительного применения означает, что слабость в одной части системы не распространяется на другие части. Если **SELinux** плохо работает с определенной подсистемой, вы можете отключить принудительное применение для этой конкретной службы, пока не найдете решение основной проблемы.

SELinux имеет три режима:

- **Enforcing** (Принудительное исполнение): **SELinux** применяет правила контроля доступа. Компьютеры обычно работают в этом режиме.
- **Permissive** (Разрешающий): **SELinux** активен, но вместо того, чтобы применять правила контроля доступа, он записывает предупреждения о правилах, которые были нарушены. Этот режим используется в основном для тестирования и устранения неполадок.
- **Disabled** (Отключено): **SELinux** полностью отключен: нарушения **SELinux** не отвергаются и даже не записываются.

ОСНОВНЫЕ КОНЦЕПЦИИ БЕЗОПАСНОСТИ SELINUX

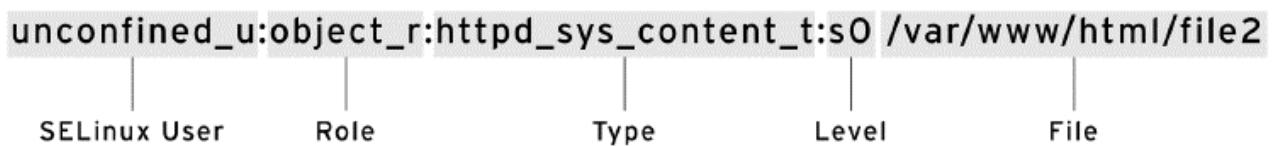
Security Enhanced Linux (SELinux) - это дополнительный уровень безопасности системы. Основная цель SELinux - защитить данные пользователя от скомпрометированных системных служб. Большинство администраторов Linux знакомы со стандартной моделью безопасности разрешений для пользователей/групп/других пользователей. Это модель на основе пользователей и групп, известная как дискреционный контроль доступа. SELinux обеспечивает дополнительный уровень безопасности, основанный на объектах и контролируемый более сложными правилами, известными как обязательный контроль доступа.

Чтобы разрешить удаленный анонимный доступ к веб-серверу, необходимо открыть порты брандмауэра. Однако это дает злоумышленникам возможность взломать систему с помощью эксплойта безопасности. Если им удастся скомпрометировать процесс веб-сервера, они получат его разрешения. В частности, разрешения **пользователя apache** и **группы apache**. Этот пользователь и группа имеют доступ на чтение к документам **root**, **/var/www/html**. Он также имеет доступ к **/tmp** и **/var/tmp**, а также к любым другим файлам и каталогам, которые доступны для записи всем.

SELinux - это набор правил безопасности, которые определяют, какой процесс может получить доступ к каким файлам, каталогам и портам. Каждый файл, процесс, каталог и порт имеет специальную метку защиты, называемую **контекстом SELinux**. **Контекст** - это имя, используемое политикой **SELinux** для определения того, может ли процесс получить доступ к файлу, каталогу или порту. По умолчанию политика не разрешает никакого взаимодействия, если явное правило не предоставляет доступ. Если разрешающего правила нет, доступ запрещен.

Метки **SELinux** имеют несколько контекстов: **пользователь**, **роль**, **тип** и **конфиденциальность**. Целевая политика, которая является политикой по умолчанию, включенной в Red Hat Enterprise Linux, основывает свои правила на третьем контексте: контексте типа. Имена контекста типов обычно заканчиваются на **_t**.

Рисунок 5.1: Контекст файла SELinux

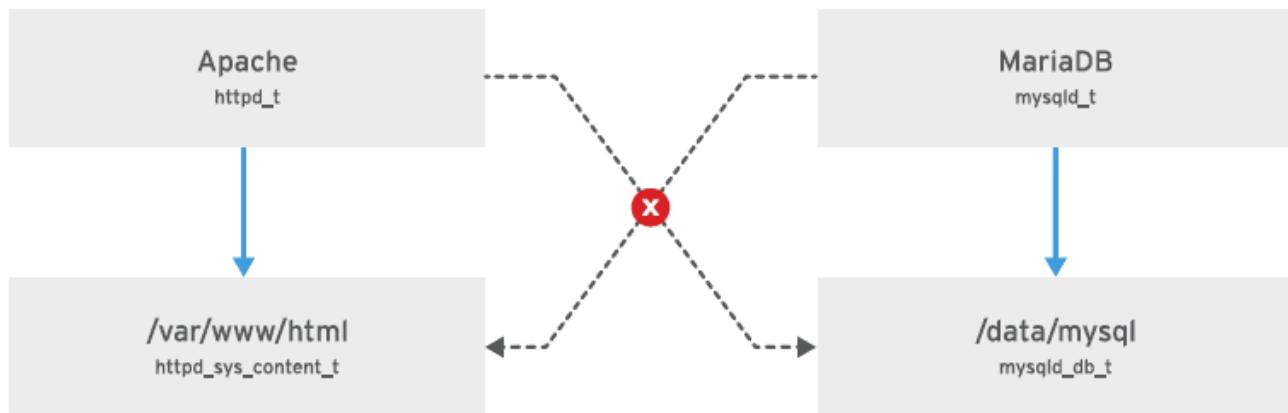


Контекст типа для веб-сервера - **httpd_t**. Контекст типа для файлов и каталогов, обычно находящихся в **/var/www/html**, - **httpd_sys_content_t**. Контекст для файлов и каталогов, обычно находящихся в **/tmp** и **/var/tmp**, - это **tmp_t**. Контекст типа для портов веб-сервера - **http_port_t**.

Apache имеет контекст типа **httpd_t**. Существует правило политики, разрешающее **Apache** доступ к файлам и каталогам с контекстом типа **httpd_sys_content_t**. По умолчанию файлы,

находящиеся в **/var/www/html** и других каталогах веб-сервера, имеют контекст типа **httpd_sys_content_t**. В политике нет разрешающего правила для файлов, обычно находящихся в **/tmp** и **/var/tmp**, поэтому доступ не разрешен. При включенном **SELinux** злоумышленник, взломавший процесс веб-сервера, не мог получить доступ к каталогу **/tmp**.

Рисунок 5.2: Доступ SELinux



Многие команды, работающие с файлами, используют параметр **-Z** для отображения или установки контекстов **SELinux**. Например, **ps**, **ls**, **cp** и **mkdir** используют параметр **-Z** для отображения или установки контекстов **SELinux**.

```
[root@host ~]# ps axZ
LABEL PID TTY STAT TIME COMMAND
system_u:system_r:init_t:s0 1 ? Ss 0:09 /usr/lib/systemd/...
system_u:system_r:kernel_t:s0 2 ? S 0:00 [kthreadd]
system_u:system_r:kernel_t:s0 3 ? S 0:00 [ksoftirqd/0]
...output omitted...
[root@host ~]# systemctl start httpd
[root@host ~]# ps -ZC httpd
LABEL PID TTY TIME CMD
system_u:system_r:httpd_t:s0 1608 ? 00:00:05 httpd
system_u:system_r:httpd_t:s0 1609 ? 00:00:00 httpd
...output omitted...
[root@host ~]# ls -Z /home
drwx-----. root root system_u:object_r:lost_found_t:s0 lost+found
drwx-----. student student unconfined_u:object_r:user_home_dir_t:s0 student
drwx-----. visitor visitor unconfined_u:object_r:user_home_dir_t:s0 visitor
[root@host ~]# ls -Z /var/www
drwxr-xr-x. root root system_u:object_r:httpd_sys_script_exec_t:s0 cgi-bin
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 error
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 icons
```

Подсистема **SELinux** предоставляет инструменты для отображения и изменения режимов. Чтобы определить текущий режим **SELinux**, запустите команду **getenforce**. Чтобы установить **SELinux** в другой режим, используйте команду **setenforce**:

```
[user@host ~]# getenforce
Enforcing
[user@host ~]# setenforce
usage: setenforce [ Enforcing | Permissive | 1 | 0 ]
[user@host ~]# setenforce 0
[user@host ~]# getenforce
Permissive
[user@host ~]# setenforce Enforcing
[user@host ~]# getenforce
Enforcing
```

В качестве альтернативы вы можете установить режим **SELinux** во время загрузки, передав параметр ядра: аргумент ядра **enforcing=0** загружает систему в разрешающий (**permissive**) режим; значение **enforcing=1** устанавливает принудительный (**enforcing**) режим. Вы также можете полностью отключить **SELinux**, передав параметр ядра **selinux=0**. Значение **selinux=1** включает **SELinux**.

НАСТРОЙКА РЕЖИМА SELINUX ПО УМОЛЧАНИЮ

Вы также можете настраивать значение **SELinux** по умолчанию с помощью файла **/etc/selinux/config**. В приведенном ниже примере (конфигурация по умолчанию) файл конфигурации устанавливает для **SELinux** принудительное (**enforcing**) исполнение. Комментарии также показывают другие допустимые значения: **permissive** и **disabled**.

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.

SELINUX=enforcing
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       minimum - Modification of targeted policy. Only selected processes
#                  are protected.
#       mls - Multi Level Security protection.

SELINUXTYPE=targeted
```

Система считывает этот файл во время загрузки и настраивает **SELinux**, как показано. Аргументы ядра (**selinux=0|1** и **enforcing=0|1**) переопределяют эту конфигурацию.



РЕКОМЕНДАЦИИ

Справочные страницы **man getenforce(8)**, **setenforce(8)**, и **selinux_config(5)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ИЗМЕНЕНИЕ РЕЖИМА ЗАЩИТЫ SELINUX

В этой лабораторной работе вы будете изменять режимы **SELinux** как временно, так и постоянно.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность просматривать и устанавливать текущий режим **SELinux**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab selinux-opsmode start**. Команда запускает сценарий, который определяет, доступна ли серверная машина **servera** в сети.

```
[student@workstation ~]$ lab selinux-opsmode start
```

1. Используйте команду **ssh** для входа на сервер **servera** как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Пароль для пользователя **student** - **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

3. Измените режим **SELinux** по умолчанию на разрешающий и перезагрузитесь.

- 3.1. Используйте команду **getenforce**, чтобы убедиться, что сервер находится в принудительном режиме.

```
[root@servera ~]# getenforce  
Enforcing
```

- 3.2.** Используйте команду **vim**, чтобы открыть файл конфигурации **/etc/selinux/config**. Измените параметр **SELINUX** с принудительного (**enforcing**) на разрешающий (**permissive**).

```
[root@servera ~]# vim /etc/selinux/config
```

- 3.3.** Используйте команду **grep**, чтобы убедиться, что для параметра **SELINUX** установлено значение **permissive**.

```
[root@servera ~]# grep '^SELINUX' /etc/selinux/config
SELINUX=permissive
SELINUXTYPE=targeted
```

- 3.4.** Используйте команду **systemctl reboot** для перезагрузки сервера.

```
[root@servera ~]# systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@workstation ~]$
```

- 4.** Серверу требуется несколько минут для перезагрузки. Через несколько минут войдите в систему как пользователь - **student**. Используйте команду **sudo -i**, чтобы стать пользователем **root**. Отобразите текущий режим **SELinux** с помощью команды **getenforce**.

- 4.1.** С рабочей станции **workstation** с помощью команды **ssh** войдите в систему как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 4.2.** Используйте команду **sudo -i**, чтобы стать пользователем **root**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

- 4.3.** Отобразите текущий режим **SELinux** с помощью команды **getenforce**.

```
[root@servera ~]# getenforce
Permissive
```

5. В файле **/etc/selinux/config** измените режим **SELinux** по умолчанию на принудительный (**enforcing**). Это изменение вступит в силу только при следующей перезагрузке.

5.1. Используйте команду **vim**, чтобы открыть файл конфигурации **/etc/selinux/config**. Измените **SELINUX** обратно на принудительное исполнение.

```
[root@servera ~]# vim /etc/selinux/config
```

5.2. Используйте команду **grep**, чтобы убедиться, что для параметра **SELINUX** установлено значение **enforcing**.

```
[root@servera ~]# grep '^SELINUX' /etc/selinux/config
SELINUX=enforcing
SELINUXTYPE=targeted
```

6. Используйте команду **setenforce**, чтобы установить текущий режим **SELinux** на принудительное (**enforcing**) без перезагрузки. Подтвердите, что установлен принудительный режим, с помощью команды **getenforce**.

```
[root@servera ~]# setenforce 1
[root@servera ~]# getenforce
Enforcing
```

7. Выход из сервера **servera**.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, выполните скрипт **lab selinux-opsmode finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab selinux-opsmode finish
```

На этом пошаговое упражнение завершено.

УПРАВЛЕНИЕ КОНТЕКСТАМИ ФАЙЛОВ SELINUX

ЗАДАЧИ

После изучения этого раздела вы сможете:

- Управляйте правилами политики **SELinux**, которые определяют контекст по умолчанию для файлов и каталогов, с помощью команды **semanage fcontext**.
- Примените контекст, определенный политикой **SELinux**, к файлам и каталогам с помощью команды **restorecon**.

НАЧАЛЬНЫЙ КОНТЕКСТ SELINUX

В системах под управлением **SELinux** все процессы и файлы помечены (labeled). Метка (**label**) представляет информацию, относящуюся к безопасности, известную, как контекст **SELinux**. Новые файлы обычно наследуют свой контекст **SELinux** от родительского каталога, тем самым гарантируя, что они имеют правильный контекст.

Но эту процедуру наследования можно подорвать двумя разными способами. Во-первых, если вы создаете файл в месте, отличном от предполагаемого местоположения, а затем перемещаете файл, файл по-прежнему будет иметь контекст **SELinux** для каталога, в котором он был создан, а не в целевом каталоге. Во-вторых, если вы копируете файл с сохранением контекста **SELinux**, как с командой **cp -a**, контекст **SELinux** отражает местоположение исходного файла.

Следующий пример демонстрирует наследование и его подводные камни. Рассмотрим два файла, созданные в **/tmp**, один перемещен в **/var/www/html**, а второй скопирован в тот же каталог. Обратите внимание на контексты **SELinux** в файлах. Файл, перемещенный в каталог **/var/www/html**, сохраняет контекст файла для каталога **/tmp**. Файл, скопированный в каталог **/var/www/html**, унаследовал контекст **SELinux** от каталога **/var/www/html**.

Команда **ls -Z** отображает контекст **SELinux** файла. Обратите внимание на метку файла (**label**).

```
[root@host ~]# ls -Z /var/www/html/index.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/
index.html
```

Обратите внимание, что **/var/www/html/index.html** имеет ту же метку, что и родительский каталог **/var/www/html/**. Теперь создайте файлы вне каталога **/var/www/html** и обратите внимание на их файловый контекст:

```
[root@host ~]# touch /tmp/file1 /tmp/file2
[root@host ~]# ls -Z /tmp/file*
unconfined_u:object_r:user_tmp_t:s0 /tmp/file1
unconfined_u:object_r:user_tmp_t:s0 /tmp/file2
```

Переместите один из этих файлов в каталог **/var/www/html**, скопируйте другой и обратите внимание на метку каждого:

```
[root@host ~]# mv /tmp/file1 /var/www/html/
[root@host ~]# cp /tmp/file2 /var/www/html/
```

```
[root@host ~]# ls -Z /var/www/html/*
unconfined_u:object_r:user_tmp_t:s0 /var/www/html/file1
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

Перемещенный файл сохраняет свою исходную метку, в то время как скопированный файл наследует метку из каталога `/var/www/html`. Пользователь **unlimited_u:**, **object_r:** обозначает роль, а **s0** - уровень. Уровень чувствительности 0 - это минимально возможный уровень чувствительности.

ИЗМЕНЕНИЕ КОНТЕКСТА ФАЙЛА SELINUX

Команды для изменения контекста SELinux в файлах включают **semanage fcontext**, **restorecon** и **chcon**.

Предпочтительный метод установки контекста SELinux для файла - объявить метку по умолчанию для файла с помощью команды **semanage fcontext**, а затем применить этот контекст к файлу с помощью команды **restorecon**. Это гарантирует, что маркировка будет применена даже после полной пере маркировки файловой системы.

Команда **chcon** изменяет контексты SELinux. **chcon** устанавливает контекст безопасности для файла, хранящегося в файловой системе. Это полезно для тестирования и экспериментов. Однако он не сохраняет изменения контекста в базе данных контекста SELinux. Когда запускается команда **restorecon**, изменения, сделанные командой **chcon**, не сохраняются. Кроме того, если вся файловая система пере маркирована, контекст SELinux для файлов, измененных с помощью **chcon**, возвращается в исходное состояние.

В следующем выводе показан создаваемый каталог. Каталог имеет значение типа **default_t**.

```
root@host ~]# mkdir /virtual
[root@host ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual
```

Команда **chcon** изменяет файловый контекст каталога `/virtual` на `httpd_sys_content_t`.

```
[root@host ~]# chcon -t httpd_sys_content_t /virtual
[root@host ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:httpd_sys_content_t:s0 /virtual
```

Команда **restorecon** выполняется, и значение типа возвращается к значению **default_t**. Обратите внимание на сообщение о переименование (**Relabeled**).

```
[root@host ~]# restorecon -v /virtual
Relabeled /virtual from unconfined_u:object_r:httpd_sys_content_t:s0 to
unconfined_u:object_r:default_t:s0
[root@host ~]# ls -Zd /virtual
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual
```

ОПРЕДЕЛЕНИЕ ПРАВИЛ КОНТЕКСТА ФАЙЛА ПО УМОЛЧАНИЮ SELINUX

Команда **semanage fcontext** отображает и изменяет правила, которые **restorecon** использует для установки контекстов файлов по умолчанию. Команда использует расширенные регулярные выражения для указания пути и имен файлов. Наиболее распространенным расширенным регулярным выражением, используемым в правилах **fcontext**, является **(/.*)?**, Что означает «необязательно сопоставить /, за которым следует любое количество символов». Команда рекурсивно соответствует каталогу, указанному перед выражением, и всему в этом каталоге.

Основные операции с файловым контекстом. В следующей таблице приведены параметры **semanage fcontext** для добавления, удаления или перечисления контекстов файлов **SELinux**.

команды **semanage fcontext**

ВАРИАНТ	ОПИСАНИЕ
-a, --add	Добавить запись указанного типа объекта
-d, --delete	Удалить запись указанного типа объекта
-l, --list	Список записей указанного типа объекта

Чтобы убедиться, что у вас есть инструменты для управления контекстами **SELinux**, установите пакет **policycoreutil** и пакет **policycoreutil-python**, если это необходимо. Они содержат, соответственно, команду **restorecon** и команду **semanage**.

Чтобы гарантировать, что все файлы в каталоге имеют правильный контекст файла, запустите **semanage fcontext -l**, а затем команду **restorecon**. В следующем примере обратите внимание на контекст каждого файла до и после запуска команд **semanage** и **restorecon**.

```
[root@host ~]# ls -Z /var/www/html/file*
unconfined_u:object_r:user_tmp_t:s0 /var/www/html/file1
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

```
[root@host ~]# semanage fcontext -l
...output omitted...
/var/www(/.*)?    all files    system_u:object_r:httpd_sys_content_t:s0
```

...output omitted...

```
[root@host: ~]# restorecon -Rv /var/www/  
Relabeled /var/www/html/file1 from unconfined_u:object_r:user_tmp_t:s0 to  
unconfined_u:object_r:httpd_sys_content_t:s0  
[root@host ~]# ls -Z /var/www/html/*  
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file1  
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/file2
```

В следующем примере показано, как использовать **semanage** для добавления контекста для нового каталога.

```
[root@host ~]# mkdir /virtual  
[root@host ~]# touch /virtual/index.html  
[root@host ~]# ls -Zd /virtual/  
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /virtual/
```

```
[root@host ~]# ls -Z /virtual/  
-rw-r--r--. root root unconfined_u:object_r:default_t:s0 index.html  
[root@host ~]# semanage fcontext -a -t httpd_sys_content_t '/virtual(.*)?'  
[root@host ~]# restorecon -RFvv /virtual  
[root@host ~]# ls -Zd /virtual/  
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 /virtual/  
[root@host ~]# ls -Z /virtual/  
-rw-r--r--. root root system_u:object_r:httpd_sys_content_t:s0 index.html
```



РЕКОМЕНДАЦИИ

Справочные страницы **man chcon(1)**, **restorecon(8)**, **semanage(8)**, и **semanage-fcontext(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ КОНТЕКСТАМИ ФАЙЛОВ SELINUX

В этой лабораторной работе вы внесете постоянные изменения в контекст SELinux каталога и его содержимое.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность настроить **HTTP-сервер Apache** для публикации веб-контента из нестандартного root документа.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab selinux-filecontexts start**. Эта команда запускает сценарий, который определяет, доступна ли машина **servera** в сети. Она также устанавливает службу **httpd** и настраивает брандмауэр на сервере, чтобы разрешить HTTP-соединения.

```
[student@workstation ~]$ lab selinux-filecontexts start
```

1. Используйте команду **ssh** для входа на сервер **servera** как пользователь **student**. Системы настроены на использование ключей SSH для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Пароль для пользователя **student** - **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

3. Настройте **Apache** на использование корня документа в нестандартном месте.

- 3.1. Создайте новый корень документа **/custom** с помощью команды **mkdir**.

```
[root@servera ~]# mkdir /custom
```

3.2. Создайте файл **index.html** в корне документа **/custom** с помощью команды **echo**.

```
[root@servera ~]# echo 'This is SERVERA.' > /custom/index.html
```

3.3. Настройте **Apache** для использования нового корневого расположения документа. Вам необходимо заменить два экземпляра **/var/www/html** на **/custom** в файле конфигурации **Apache**, **/etc/httpd/conf/httpd.conf**.

```
[root@servera ~]# vim /etc/httpd/conf/httpd.conf
[root@servera ~]# grep custom /etc/httpd/conf/httpd.conf
DocumentRoot "/custom"
<Directory "/custom">
```

4. Запустите и включите веб-службу **Apache** и убедитесь, что служба работает.

4.1. Запустите и включите веб-службу Apache с помощью команды **systemctl**.

```
[root@servera ~]# systemctl enable --now httpd
```

4.2. Используйте команду **systemctl**, чтобы убедиться, что служба запущена.

```
[root@servera ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled;
  vendor preset: disabled)
    Active: active (running) since Mon 2019-03-25 19:16:48 CET; 15h ago
      Docs: man:httpd.service(8)
   Main PID: 6565 (httpd)
     Status: "Total requests: 16; Idle/Busy workers 100/0;Requests/sec:
 0.000285; Bytes served/sec: 0 B/sec"
    Tasks: 213 (limit: 11406)
   Memory: 37.3M
    CGroup: /system.slice/httpd.service
            ├─6565 /usr/sbin/httpd -DFOREGROUND
            ├─6566 /usr/sbin/httpd -DFOREGROUND
            ├─6567 /usr/sbin/httpd -DFOREGROUND
            ├─6568 /usr/sbin/httpd -DFOREGROUND
            └─6569 /usr/sbin/httpd -DFOREGROUND

Mar 25 19:16:48 servera.lab.example.com systemd[1]: Starting The
Apache HTTP Server...
Mar 25 19:16:48 servera.lab.example.com httpd[6565]: Server
configured, listening on: port 80
```

- Откройте веб-браузер на рабочей станции **workstation** и попробуйте просмотреть **http://servera/index.html**. Вы получите сообщение об ошибке, в котором говорится, что у вас нет разрешения на доступ к файлу.
- Чтобы разрешить доступ к файлу **index.html** на сервере, необходимо настроить **SELinux**. Определите правило контекста файла **SELinux**, которое устанавливает тип контекста **httpd_sys_content_t** для каталога **/custom** и всех файлов в нём.

```
[root@servera ~]# semanage fcontext -a -t httpd_sys_content_t '/  
custom(/.*)?'
```

- Используйте команду **restorecon**, чтобы изменить контексты файлов.

```
[root@servera ~]# restorecon -Rv /custom  
Relabeled /custom from unconfined_u:object_r:default_t:s0 to  
unconfined_u:object_r:httpd_sys_content_t:s0  
Relabeled /custom/index.html from unconfined_u:object_r:default_t:s0 to  
unconfined_u:object_r:httpd_sys_content_t:s0
```

- Попробуйте снова просмотреть **http://servera/index.html**. Вы должны увидеть сообщение: **This is SERVERA**.
- Выход из сервера **servera**.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab selinux-filecontexts finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab selinux-filecontexts finish
```

На этом пошаговое упражнение завершено.

НАСТРОЙКА ПОЛИТИКИ SELINUX С ПОМОЩЬЮ BOOLEANS

ЗАДАЧИ

После завершения этого раздела вы сможете:

- Активируйте и деактивируйте правила политики **SELinux** с помощью **setsebool**.
- Управляйте постоянным значением логических значений **SELinux** с помощью команды **semanage boolean -l**.
- Обратитесь к страницам руководства, которые заканчиваются на **_selinux**, чтобы найти полезную информацию о логических значениях **SELinux**.

SELINUX BOOLEANS

Логические значения **SELinux** - это переключатели, которые изменяют поведение политики **SELinux**. Логические значения **SELinux** - это правила, которые можно включить или отключить. Они могут использоваться администраторами безопасности для настройки политики для внесения выборочных корректировок.

Страницы руководства **SELinux**, поставляемые с пакетом **selinux-policy-doc**, описывают назначение доступных логических значений. Команда **man -k '_selinux'** выводит список этих страниц руководства.

Команды, полезные для управления логическими значениями **SELinux**, включают команда **getsebool**, который перечисляет логические значения и их состояние, и команда **setsebool**, который изменяет логические значения. **setsebool -P** изменяет политику **SELinux**, чтобы сделать изменение постоянным. А **semanage boolean -l** сообщает о том, является ли логическое значение постоянным, вместе с кратким описанием логического значения.

Непrivилегированные пользователи могут запускать команду **getsebool**, но вы должны быть суперпользователем, чтобы запускать **semanage boolean -l** и **setsebool -P**.

```
[user@host ~]$ getsebool -a
abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
antivirus_can_scan_system --> off
antivirus_use_jit --> off
...output omitted...
[user@host ~]$ getsebool httpd_enable_homedirs
httpd_enable_homedirs --> off
```

```
[user@host ~]$ setsebool httpd_enable_homedirs on
Could not change active booleans. Please try as root: Permission denied
[user@host ~]$ sudo setsebool httpd_enable_homedirs on
[user@host ~]$ sudo semanage boolean -l | grep httpd_enable_homedirs
httpd_enable_homedirs      (on , off)  Allow httpd to enable homedirs
```

```
[user@host ~]$ getsebool httpd_enable_homedirs  
httpd_enable_homedirs --> on
```

Параметр **-P** записывает все ожидающие значения в политику, делая их постоянными при перезагрузках. В следующем примере обратите внимание на значения в скобках: оба теперь включены (**on**).

```
[user@host ~]$ setsebool -P httpd_enable_homedirs on  
[user@host ~]$ sudo semanage boolean -l | grep httpd_enable_homedirs  
httpd_enable_homedirs      (on , on)      Allow httpd to enable homedirs
```

Чтобы вывести логические значения (**booleans**), в которых текущее состояние отличается от состояния по умолчанию, выполните команду **semanage boolean -l -C**.

```
[user@host ~]$ sudo semanage boolean -l -C  
SELinux boolean      State          Default Description  
cron_can_relabel    (off , on)     Allow cron to can relabel
```



РЕКОМЕНДАЦИИ

Справочные страницы **man booleans(8)**, **getsebool(8)**, **setsebool(8)**, **semanage(8)**, **semanage-boolean(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

НАСТРОЙКА ПОЛИТИКИ SELINUX С ПОМОЩЬЮ BOOLEANS

Apache может публиковать веб-контент, размещенный в домашних каталогах пользователей, но **SELinux** предотвращает это по умолчанию. В этом упражнении вы определите и измените логическое значение **SELinux**, которое разрешает Apache доступ к домашним каталогам пользователей.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность настроить Apache для публикации веб-контента из домашних каталогов пользователей.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab selinux-booleans start**. Данная команда запускает сценарий, который определяет, доступна ли серверная машина **servera** в сети. Он также устанавливает службу **httpd** и настраивает брандмауэр на сервере, чтобы разрешить HTTP-соединения.

```
[student@workstation ~]$ lab selinux-booleans start
```

1. Используйте команду **ssh** для входа на сервер **servera** как пользователь **student**. Системы настроены на использование ключей SSH для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Пароль для пользователя **student** - **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

3. Чтобы включить функцию **Apache**, которая позволяет пользователям публиковать веб-контент из своих домашних каталогов, вы должны отредактировать файл конфигурации **/etc/httpd/conf.d/userdir.conf**. Закомментируйте строку, которая устанавливает **UserDir** в состояние **disabled**, и раскомментируйте строку, которая устанавливает **UserDir** в **public_html**.

```
[root@servera ~]# vim /etc/httpd/conf.d/userdir.conf
#UserDir disabled
UserDir public_html
```

4. Используйте команду **grep**, чтобы подтвердить изменения.

```
[root@servera ~]# grep '#UserDir' /etc/httpd/conf.d/userdir.conf
#UserDir disabled
[root@servera ~]# grep '^ *UserDir' /etc/httpd/conf.d/userdir.conf
UserDir public_html
```

5. Запустите и включите веб-службу Apache, чтобы изменения вступили в силу.

```
[root@servera ~]# systemctl enable --now httpd
```

6. В другом окне терминала войдите, как пользователь **student**. SSH на сервер **servera**. Создайте веб-контент, который публикуется из домашнего каталога пользователя.

- 6.1. В другом окне терминала войдите, как пользователь **student**. Используйте команду **ssh** для входа на сервер **servera** в качестве пользователя **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 6.2. Используйте команду **mkdir**, чтобы создать каталог с именем **~/public_html**.

```
[student@servera ~]$ mkdir ~/public_html
```

- 6.3. Создайте файл **index.html** со следующим содержанием:

```
[student@servera ~]$ echo 'This is student content on SERVERA.' > \
~/public_html/index.html
```

- 6.4. Используйте команду **chmod**, чтобы изменить права доступа к домашнему каталогу учащегося, чтобы **Apache** мог получить доступ к подкаталогу **public_html**.

```
[student@servera ~]$ chmod 711 ~
```

7. Откройте веб-браузер на рабочей станции **workstation** и попробуйте просмотреть следующий URL-адрес: **http://servera/~student/index.html**. Вы получаете сообщение об ошибке, в котором говорится, что у вас нет разрешения на доступ к файлу.
8. В окне терминала с доступом **root** используйте команду **getsebool**, чтобы увидеть, есть ли какие-либо логические значения, ограничивающие доступ к домашним каталогам.

```
[root@servera ~]# getsebool -a | grep home  
...output omitted...  
httpd_enable_homedirs --> off  
...output omitted...
```

9. В окне терминала с доступом **root** используйте команду **setsebool**, чтобы разрешать доступ к домашнему каталогу постоянно.

```
[root@servera ~]# setsebool -P httpd_enable_homedirs on
```

10. Попробуйте снова просмотреть **http://servera/~student/index.html**. Вы должны увидеть сообщение: **This is student content on SERVERA**.

11. Выход с сервера **servera**.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите скрипт **lab selinux-booleans finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab selinux-booleans finish
```

На этом пошаговое упражнение завершено.

ИССЛЕДОВАНИЕ И РЕШЕНИЕ ПРОБЛЕМ SELINUX

ЦЕЛИ

После рассмотрения этого раздела вы должны быть способны:

- Используйте инструменты анализа журнала **SELinux**.
- Отображение полезной информации во время устранения неполадок **SELinux** с помощью команды **sealert**.

УСТРАНЕНИЕ НЕИСПРАВНОСТЕЙ SELINUX

Важно понимать, какие действия вы должны предпринять, когда **SELinux** запрещает доступ к файлам на сервере, который, как вы знаете, должен быть доступен. Используйте следующие шаги в качестве руководства по устранению этих проблем:

1. Прежде чем думать о каких-либо корректировках, учтите, что SELinux может правильно выполнять свою работу, запрещая попытки доступа. Если веб-сервер пытается получить доступ к файлам в каталоге **/home**, данная ситуация может сигнализировать о компрометации службы, если веб-контент не публикуется пользователями. Если доступ должен был быть предоставлен, то необходимо предпринять дополнительные шаги для решения проблемы.
2. Самая распространенная проблема **SELinux** — неверный контекст файла. Это может произойти, когда файл создается в месте с одним контекстом файла и перемещается в место, где ожидается другой контекст. В большинстве случаев запуск **restorecon** решит проблему. Исправление проблем таким образом оказывает очень незначительное влияние на безопасность остальной части системы.
3. Другим средством от чрезмерно ограниченного доступа может быть корректировка логического значения. Например, логическое значение **ftpd_anon_write** определяет, могут ли анонимные пользователи **FTP** загружать файлы. Вы должны включить это логическое значение, чтобы разрешить анонимным пользователям **FTP** загружать файлы на сервер. Настройка логических значений требует большей осторожности, потому что они могут иметь большое влияние на безопасность системы.
4. Возможно, в политике **SELinux** есть ошибка, препятствующая легитимному доступу. Поскольку SELinux довольно зрелая система, это довольно редкое явление. Когда становится ясно, что обнаружена ошибка политики, обратитесь в службу поддержки Red Hat, чтобы сообщить об этом, чтобы ее можно было устраниć.

МОНИТОРИНГ НАРУШЕНИЙ SELINUX

Установите пакет **setroubleshoot-server** для отправки сообщений **SELinux** в **/var/log/messages**. **setroubleshoot-server** прослушивает сообщения аудита в **/var/log/audit/audit.log** и отправляет краткую сводку в **/var/log/messages**. Данная сводка включает уникальные идентификаторы (**UUID**) нарушений **SELinux**, которые можно использовать для сбора дополнительной информации. Команда **sealert -I UUID** используется для создания отчета по конкретному инциденту. Используйте **sealert -a /var/log/audit/audit.log** для создания отчетов обо всех инцидентах в этом файле.

Рассмотрим следующую примерную последовательность команд на стандартном веб-сервере **Apache**:

```
[root@host ~]# touch /root/file3
[root@host ~]# mv /root/file3 /var/www/html
[root@host ~]# systemctl start httpd
[root@host ~]# curl http://localhost/file3
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /file3
on this server.</p>
</body></html>
```

Вы ожидаете, что веб-сервер доставит содержимое **file3**, но вместо этого он вернет ошибку **permission denied** (отказ в разрешении). Проверка файлов **/var/log/audit/audit.log** и **/var/log/messages** позволяет получить дополнительную информацию об этой ошибке.

```
[root@host ~]# tail /var/log/audit/audit.log
...output omitted...
type=AVC msg=audit(1392944135.482:429): avc: denied { getattr } for
pid=1609 comm="httpd" path="/var/www/html/file3" dev="vda1" ino=8980981
scontext=system_u:system_r:httpd_t:s0
tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file
...output omitted...
[root@host ~]# tail /var/log/messages
...output omitted...
Feb 20 19:55:42 host setroubleshoot: SELinux is preventing /usr/sbin/httpd
from getattr access on the file . For complete SELinux messages. run
sealert -l 613ca624-248d-48a2-a7d9-d28f5bbe2763
```

Оба файла журнала указывают на то, что причиной является отказ **SELinux**. Команда **sealert**, которая является частью вывода в **/var/log/messages**, предоставляет дополнительную информацию, включая возможное исправление.

```
[root@host ~]# sealert -l 613ca624-248d-48a2-a7d9-d28f5bbe2763
SELinux is preventing /usr/sbin/httpd from getattr access on the file .

*****          Plugin catchall (100. confidence) suggests          *****
If you believe that httpd should be allowed getattr access on the
file by default.
Then you should report this as a bug.
```

You can generate a local policy module to allow this access.

Do

allow this access for now by executing:

```
# grep httpd /var/log/audit/audit.log | audit2allow -M mypol  
# semodule -i mypol.pp
```

Additional Information:

Source Context	system_u:system_r:httpd_t:s0
Target Context	unconfined_u:object_r:admin_home_t:s0
Target Objects	[file]
Source	httpd
Source Path	/usr/sbin/httpd
Port	<Unknown>
Host	servera
Source RPM Packages	httpd-2.4.6-14.el7.x86_64
Target RPM Packages	
Policy RPM	selinux-policy-3.12.1-124.el7.noarch
Selinux Enabled	True
Policy Type	targeted
Enforcing Mode	Enforcing
Host Name	servera
Platform	Linux servera 3.10.0-84.el7.x86_64 #1 SMP Tue Feb 4 16:28:19 EST 2014 x86_64 x86_64
Alert Count	2
First Seen	2014-02-20 19:55:35 EST
Last Seen	2014-02-20 19:55:35 EST
Local ID	613ca624-248d-48a2-a7d9-d28f5bbe2763

Raw Audit Messages

```
type=AVC msg=audit(1392944135.482:429): avc: denied { setattr } for  
pid=1609 comm="httpd" path="/var/www/html/file3" dev="vda1" ino=8980981  
scontext=system_u:system_r:httpd_t:s0  
tcontext=unconfined_u:object_r:admin_home_t:s0 tclass=file  
  
type=SYSCALL msg=audit(1392944135.482:429): arch=x86_64 syscall=lstat  
success=no exit=EACCES a0=7f9fed0edea8 a1=7fff7bffc770 a2=7fff7bffc770  
a3=0 items=0 ppid=1608 pid=1609 auid=4294967295 uid=48 gid=48 euid=48  
suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295  
comm=httpd exe=/usr/sbin/httpd subj=system_u:system_r:httpd_t:s0 key=(null)
```

Hash: httpd,httpd_t,admin_home_t,file,getattr



ПРИМЕЧАНИЕ

В разделе «Необработанные сообщения аудита (**Raw Audit Messages**)» указан целевой файл, в котором возникла проблема, **/var/www/html/file3**. Кроме того, целевой контекст, **tcontext**, не выглядит так, как будто он принадлежит веб-серверу. Используйте команду **restorecon /var/www/html/file3**, чтобы исправить контекст файла. Если есть другие файлы, которые необходимо изменить, **restorecon** может рекурсивно сбросить контекст: **restorecon -R /var/www/**.

Раздел **Raw Audit Messages** команды **sealert** содержит информацию из **/var/log/audit.log**. Для поиска в файле **/var/log/audit.log** используйте команду **ausearch**. Опция **-m** указывает выполнить поиск по типу сообщения. Опция **-ts** выполняет поиск по времени.

```
[root@host ~]# ausearch -m AVC -ts recent
-----
time->Tue Apr  9 13:13:07 2019
type=PROCTITLE msg=audit(1554808387.778:4002):
  proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1554808387.778:4002): arch=c000003e syscall=49
  success=no exit=-13 a0=3 a1=55620b8c9280 a2=10 a3=7ffed967661c items=0
  ppid=1 pid=9340 auid=4294967295 uid=0 gid=0 euid=0 suid=0 fsuid=0 egid=0
  sgid=0 fsgid=0 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd"
  subj=system_u:system_r:httpd_t:s0 key=(null)
type=AVC msg=audit(1554808387.778:4002): avc:  denied  { name_bind }
  for  pid=9340 comm="httpd" src=82 scontext=system_u:system_r:httpd_t:s0
  tcontext=system_u:object_r:reserved_port_t:s0 tclass=tcp_socket permissive=0
```

ВЕБ-КОНСОЛЬ

Если установлена веб-консоль, ее также можно использовать для устранения неполадок **SELinux**. Войдите в веб-консоль и выберите **SELinux** в меню слева. Окно политики **SELinux** информирует вас о текущей политике применения. Любые проблемы подробно описаны в разделе «Ошибки контроля доступа SELinux (**SELinux Access Control Errors**)».

Рисунок 5.3: Политика SELinux в веб-консоли

The screenshot shows the SELinux Policy configuration page. At the top, it says "SELinux Policy". Below that, there is a "Enforce policy:" section with a switch labeled "ON". Under "SELinux Access Control Errors", there is a single error message: "SELinux is preventing /usr/sbin/httpd from open access on the file /lab-content/lab.html." A backslash character is shown before the message.

```
> | SELinux is preventing /usr/sbin/httpd from open access on the file /lab-content/lab.html.
```

Щелкните символ >, чтобы отобразить сведения об ошибке. Нажмите на сведения о решении (*solution details*), чтобы просмотреть все сведения и возможное решение.

Рисунок 5.4: Решение политики SELinux в веб-консоли

The screenshot shows the SELinux Policy interface. At the top, there is a header 'SELinux Policy' with an 'Enforce policy: ON' switch. Below it is a section titled 'SELinux Access Control Errors'. A single error message is listed: 'SELinux is preventing /usr/sbin/httpd from open access on the file /lab-content/lab.html.' This message includes a note: 'If you believe that httpd should be allowed open access on the lab.html file by default. You should report this as a bug. You can generate a local policy module to allow this access.' There are two tabs at the bottom of this section: 'Solutions' (selected) and 'Audit log'. A timestamp indicates the errors occurred between 'Last Thursday at 11:22 AM and Last Thursday at 3:08 PM'. To the right, a note says 'Unable to apply this solution automatically'. A small trash bin icon is in the top right corner of the error list.

После того, как проблема будет решена, в разделе «Ошибки контроля доступа SELinux (*SELinux Access Control Errors*)» ошибка больше не должна отображаться. Если появляется сообщение. Нет предупреждений **SELinux**, значит, все проблемы устранены.

Рисунок 5.5: Нет предупреждений SELinux в веб-консоли

The screenshot shows the SELinux Policy interface. At the top, there is a header 'SELinux Policy' with an 'Enforce policy: ON' switch. Below it is a section titled 'SELinux Access Control Errors'. A message 'No SELinux alerts.' is displayed. At the bottom left, there is a blue sidebar containing a document icon and the word 'РЕКОМЕНДАЦИИ'. Below this, a link reads 'Справочные страницы man sealert(8)'. The main content area is mostly empty.

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ИССЛЕДОВАНИЕ И РЕШЕНИЕ ПРОБЛЕМ SELINUX

В этом лабораторном занятии вы узнаете, как устранять неполадки, связанные с отказами безопасности SELinux.

В РЕЗУЛЬТАТЕ

Вы получите некоторый опыт использования инструментов устранения неполадок SELinux.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab selinux-issues start**. Эта команда запускает сценарий, который определяет, доступен ли компьютер **servera** в сети. Так же устанавливает службу **httpd**, настраивает брандмауэр на сервере для разрешения HTTP-соединений и удаляет контекст SELinux для каталога **/custom**.

```
[student@workstation ~]$ lab selinux-issues start
```

1. Используйте команду **ssh**, чтобы войти на сервер **servera**, как студент. Системы настроены на использование ключей SSH для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Пароль для студенческого пользователя **student** - **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

3. Откройте веб-браузер на рабочей станции **workstation** и попробуйте просмотреть **http://servera/index.html**. Вы получите сообщение об ошибке, в котором говорится, что у вас нет прав доступа к файлу (not have permission to access).

4. С помощью команды **less** просмотрите содержимое файла **/var/log/messages**. Используйте ключ **/** для поиска слова **sealert**. Скопируйте предложенную команду **sealert**, чтобы ее можно было использовать на следующем шаге. Используйте клавишу **q**, чтобы выйти из команды **less**.

```
[root@servera ~]# less /var/log/messages
...output omitted...
Mar 28 06:07:03 servera setroubleshoot[15326]: SELinux is preventing /usr/
sbin/httpd from setattr access on the file /custom/index.html. For complete
SELinux messages run: sealert -l b1c9cc8f-a953-4625-b79b-82c4f4f1fee3
Mar 28 06:07:03 servera platform-python[15326]: SELinux is
preventing /usr/sbin/httpd from setattr access on the file /custom/
index.html.#012***** Plugin catchall (100. confidence) suggests
*****#012If you believe that httpd should be
allowed setattr access on the index.html file by default.#012Then you
should report this as a bug.#012You can generate a local policy module to
allow this access.#012Do#012allow this access for now by executing:#012#
ausearch -c 'httpd' --raw | audit2allow -M my-httpd#012# semodule -X 300 -i
my-httpd.pp#012
Mar 28 06:07:04 servera setroubleshoot[15326]: failed to retrieve rpm info
for /custom/index.html
...output omitted...
```

5. Запустите предложенную команду **sealert**. Обратите внимание на исходный контекст, целевые объекты, политику и принудительный (**enforcing**) режим.

```
[root@servera ~]# sealert -l b1c9cc8f-a953-4625-b79b-82c4f4f1fee3
SELinux is preventing /usr/sbin/httpd from setattr access on the file /
custom/index.html.
***** Plugin catchall (100. confidence) suggests
*****
If you believe that httpd should be allowed setattr access on the index.html
file by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# ausearch -c 'httpd' --raw | audit2allow -M my-httpd
# semodule -X 300 -i my-httpd.pp
```

Additional Information:

Source Context	system_u:system_r:httpd_t:s0
Target Context	unconfined_u:object_r:default_t:s0
Target Objects	/custom/index.html [file]
Source	httpd
Source Path	/usr/sbin/httpd
Port	<Unknown>
Host	servera.lab.example.com

```

Source RPM Packages
Target RPM Packages
Policy RPM           selinux-policy-3.14.1-59.el8.noarch
Selinux Enabled       True
Policy Type          targeted
Enforcing Mode       Enforcing

Host Name            servera.lab.example.com
Platform             Linux servera.lab.example.com
4.18.0-67.el8.x86_64
                      #1 SMP Sat Feb 9 12:44:00 UTC 2019 x86_64
x86_64
                      Alert Count      18
                      First Seen       2019-03-25
19:25:28 CET
                      Last Seen        2019-03-28
11:07:00 CET
                      Local ID         b1c9cc8f-
a953-4625-b79b-82c4f4f1fee3

Raw Audit Messages
type=AVC msg=audit(1553767620.970:16958):
  avc: denied { getattr } for pid=15067 comm="httpd" path="/custom/
index.html" dev="vda1" ino=4208311 scontext=system_u:system_r:httpd_t:s0
  tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0

Hash: httpd,httpd_t,default_t,file,getattr

```

6. Раздел «Необработанные сообщения аудита (**Raw Audit Messages**)» команды **sealert** содержит информацию из файла **/var/log/audit/audit.log**. Используйте команду **ausearch** для поиска в файле **/var/log/audit/audit.log**. Параметр **-m** выполняет поиск по типу сообщения. Опция **-ts** выполняет поиск по времени. Эта запись идентифицирует соответствующий процесс и файл, вызвавший предупреждение. Процесс **httpd** это веб-сервер **Apache**, файл **/custom/index.html**, а контекст **system_r:httpd_t**.

```

[root@servera ~]# ausearch -m AVC -ts recent
-----
time->Thu Mar 28 13:39:30 2019
type=PROCTITLE msg=audit(1553776770.651:17000):
proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1553776770.651:17000): arch=c000003e syscall=257
success=no exit=-13 a0=fffffff9c a1=7f8db803f598 a2=80000 a3=0 items=0
ppid=15063 pid=15065 auid=4294967295 uid=48 gid=48 euid=48 suid=48 fsuid=48
egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="httpd" exe="/usr/
sbin/httpd" subj=system_u:system_r:httpd_t:s0 key=(null)
type=AVC msg=audit(1553776770.651:17000): avc:      denied
{ open } for pid=15065 comm="httpd" path="/custom/index.html"
dev="vda1" ino=4208311 scontext=system_u:system_r:httpd_t:s0

```

```
tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0
```

7. Для решения проблемы используйте команды **semanage** и **restorecon**. Для управления контекстом **httpd_sys_content_t**.

```
[root@servera ~]# semanage fcontext -a -t httpd_sys_content_t '/  
custom(.*?)"  
[root@servera ~]# restorecon -Rv /custom  
Relabeled /custom from unconfined_u:object_r:default_t:s0 to  
unconfined_u:object_r:httpd_sys_content_t:s0  
Relabeled /custom/index.html from unconfined_u:object_r:default_t:s0 to  
unconfined_u:object_r:httpd_sys_content_t:s0
```

8. Попробуйте еще раз просмотреть <http://servera/index.html>. Вы должны увидеть сообщение **This is SERVERA**.
9. Выход из сервера **servera**.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Оканчание

На рабочей станции **workstation**, запустите сценарий **lab selinux-issues finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab selinux-issues finish
```

На этом упражнение заканчивается.

ЛАБОРАТОРНАЯ РАБОТА

УПРАВЛЕНИЕ SELINUX SECURITY

КОНТРОЛЬНЫЙ СПИСОК

В этой лабораторной работе вы решите проблему отказа в доступе **SELinux**. У системных администраторов возникают проблемы с получением нового веб-сервера для доставки контента клиентам, когда **SELinux** находится в режиме **enforcing**.

В РЕЗУЛЬТАТЕ

Вы должен быть способны:

- Выявлять проблемы в файлах системного журнала.
- Настройте конфигурацию **SELinux**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab selinux-review start**. Эта команда запускает сценарий, который определяет, доступен ли сервер **serverb** в сети. Он также устанавливает сервер **httpd Apache**, создает новый **DocumentRoot** для **Apache** и обновляет файл конфигурации.

```
[student@workstation ~]$ lab selinux-review start
```

1. Войдите на **serverb** как пользователь **root**.
2. Запустите веб-браузер на рабочей станции и перейдите по адресу **http://server/lab.html**. Вы увидите сообщение об ошибке: **You do not have permission to access /lab.html on this server**.
3. Исследуйте и определите проблему **SELinux**, которая не позволяет **Apache** обслуживать веб-контент.
4. Отобразите контекст **SELinux** для нового корня документа **HTTP** и исходного корня документа **HTTP**. Решите проблему **SELinux**, не позволяющую **Apache** обслуживать веб-контент.
5. Убедитесь, что проблема с **SELinux** устранена и **Apache** может обслуживать веб-контент.
6. Выйти с сервера **serverb**.

Оценка

На рабочей станции **workstation**, запустите сценарий **lab selinux-review grade**, чтобы подтвердить успешное выполнение этого упражнения.

```
[student@workstation ~]$ lab selinux-review grade
```

Окончание

На рабочей станции **workstation** запустите сценарий **lab selinux-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab selinux-review finish
```

На этом лабораторная работа заканчивается.

РЕШЕНИЕ

УПРАВЛЕНИЕ SELINUX SECURITY

КОНТРОЛЬНЫЙ СПИСОК

В этой лабораторной работе вы решите проблему отказа в доступе **SELinux**. У системных администраторов возникают проблемы с получением нового веб-сервера для доставки контента клиентам, когда **SELinux** находится в режиме **enforcing**.

В РЕЗУЛЬТАТЕ

Вы должен быть способны:

- Выявлять проблемы в файлах системного журнала.
- Настройте конфигурацию **SELinux**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab selinux-review start**. Эта команда запускает сценарий, который определяет, доступен ли сервер **serverb** в сети. Он также устанавливает сервер **httpd Apache**, создает новый **DocumentRoot** для **Apache** и обновляет файл конфигурации.

```
[student@workstation ~]$ lab selinux-review start
```

1. Войдите на **serverb** как пользователь **root**.

1.1. Используйте команду **ssh** для входа на **serverb** в качестве пользователя **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

1.2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Пароль для пользователя – **student - student**.

```
[student@serverb ~]$ sudo -i  
[sudo] password for student: student
```

```
[root@serverb ~]#
```

2. Запустите веб-браузер на рабочей станции и перейдите по адресу <http://server/lab.html>. Вы увидите сообщение об ошибке: **You do not have permission to access /lab.html on this server.**
3. Исследуйте и определите проблему **SELinux**, которая не позволяет **Apache** обслуживать веб-контент.
 - 3.1. С помощью команды **less** просмотрите содержимое **/var/log/messages**. Используйте ключ **/** и найдите текст **sealert**. Используйте клавишу **q**, чтобы выйти из команды **less**.

```
[root@serverb ~]# less /var/log/messages
Mar 28 10:19:51 serverb setroubleshoot[27387]: SELinux is
  preventing /usr/sbin/httpd from getattr access on the file /lab-
content/lab.html. For complete SELinux messages run: sealert -l
8824e73d-3ab0-4caf-8258-86e8792fee2d
Mar 28 10:19:51 serverb platform-python[27387]: SELinux is preventing /
  usr/sbin/httpd from getattr access on the file /lab-content/
lab.html.#012#012***** Plugin catchall (100. confidence) suggests
  *****#012#012If you believe that httpd should
  be allowed getattr access on the lab.html file by default.#012Then
  you should report this as a bug.#012You can generate a local policy
  module to allow this access.#012Do#012allow this access for now
  by executing:#012# ausearch -c 'httpd' --raw | audit2allow -M my-
httpd#012# semodule -X 300 -i my-httpd.pp#012
```

- 3.2. Запустите предложенную команду **sealert**. Обратите внимание на исходный контекст (**context**), целевые объекты (**target objects**), политику (**policy**) и режим **enforcing**.

```
[root@serverb ~]# sealert -l 8824e73d-3ab0-4caf-8258-86e8792fee2d
SELinux is preventing /usr/sbin/httpd from getattr access on the file /
lab-content/lab.html.
```

```
***** Plugin catchall (100. confidence) suggests
  *****

If you believe that httpd should be allowed getattr access on the
lab.html file by default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# ausearch -c 'httpd' --raw | audit2allow -M my-httpd
# semodule -X 300 -i my-httpd.pp
```

```

Additional Information:
Source Context           system_u:system_r:httpd_t:s0
Target Context          unconfined_u:object_r:default_t:s0
Target Objects          /lab-content/lab.html [ file ]
Source                  httpd
Source Path             /usr/sbin/httpd
Port                   <Unknown>
Host                   serverb.lab.example.com

Source RPM Packages
Target RPM Packages
Policy RPM              selinux-policy-3.14.1-59.el8.noarch
Selinux Enabled          True
Policy Type              targeted
Enforcing Mode           Enforcing
Host Name                serverb.lab.example.com
Platform                 Linux serverb.lab.example.com
                         4.18.0-67.el8.x86_64
                         #1 SMP Sat Feb 9 12:44:00 UTC 2019 x86_64
                         x86_64
                         Alert Count          2
                         First Seen           2019-03-28
                         15:19:46 CET
                         Last Seen            2019-03-28
                         15:19:46 CET
                         Local ID              8824e73d-3ab0-4caf-8258-86e8792fee2d

                         Raw Audit Messages
                         type=AVC msg=audit(1553782786.213:864):
                         avc: denied { getattr } for pid=15606
                         comm="httpd" path="/lab-content/lab.html" dev="vda1"
                         ino=8763212 scontext=system_u:system_r:httpd_t:s0
                         tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0

Hash: httpd,httpd_t,default_t,file,getattr

```

3.3. Раздел **Raw Audit Messages** команды **sealert** содержит информацию из файла **/var/log/audit/audit.log**. Используйте команду **ausearch** для поиска в файле **/var/log/audit/audit.log**. Параметр **-m** выполняет поиск по типу сообщения. Опция **ts** выполняет поиск по времени. Эта запись идентифицирует соответствующий процесс и файл, вызвавший предупреждение. Процесс — это веб-сервер **httpd Apache**, файл — **/lab-content/lab.html**, а контекст — **system_r:httpd_t**.

```

[root@serverb ~]# ausearch -m AVC -ts recent
time->Thu Mar 28 15:19:46 2019
type=PROCTITLE msg=audit(1553782786.213:864):
proctitle=2F7573722F7362696E2F6874747064002D44464F524547524F554E44
type=SYSCALL msg=audit(1553782786.213:864): arch=c000003e
syscall=6 success=no exit=-13 a0=7fb900004930 a1=7fb92dfca8e0

```

```
a2=7fb92dfca8e0 a3=1 items=0 ppid=15491 pid=15606 auid=4294967295  
uid=48 gid=48 euid=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48  
tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbin/httpd"  
subj=system_u:system_r:httpd_t:s0 key=(null)  
type=AVC msg=audit(1553782786.213:864): avc: denied { getattr }  
for pid=15606 comm="httpd" path="/lab-content/lab.html"  
dev="vda1" ino=8763212 scontext=system_u:system_r:httpd_t:s0  
tcontext=unconfined_u:object_r:default_t:s0 tclass=file permissive=0
```

4. Отобразите контекст **SELinux** для нового корня документа **HTTP** и исходного корня документа **HTTP**. Решите проблему **SELinux**, не позволяющую **Apache** обслуживать веб-контент.

4.1. Используйте **ls -dZ** для сравнения корня документа **/lab-content** и **/var/www/html**.

```
[root@serverb ~]# ls -dZ /lab-content /var/www/html  
unconfined_u:object_r:default_t:s0 /lab-content/  
system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
```

4.2. Создайте правило контекста файла, которое задает тип по умолчанию **httpd_sys_content_** для **/lab-content** и всех файлов ниже него.

```
[root@serverb ~]# semanage fcontext -a -t httpd_sys_content_t '/labcontent(/.*)?'
```

4.3. Используйте команду **restorecon**, чтобы установить контекст **SELinux** для файлов в **/lab-content**.

```
[root@serverb ~]# restorecon -R /lab-content/
```

5. Убедитесь, что проблема с **SELinux** устранена и **Apache** может обслуживать веб-контент.

Воспользуйтесь веб-браузером, чтобы обновить ссылку **http://server/lab.html**. Теперь вы должны увидеть веб-контент.

6. Выйти с сервера **serverb**.

```
[root@serverb ~]# exit  
logout  
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

Оценка

На рабочей станции **workstation**, запустите сценарий **lab selinux-review grade**, чтобы подтвердить успешное выполнение этого упражнения.

```
[student@workstation ~]$ lab selinux-review grade
```

Окончание

На рабочей станции **workstation** запустите сценарий **lab selinux-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab selinux-review finish
```

На этом лабораторная работа заканчивается.

РЕЗЮМЕ

В этой главе вы узнали:

- Команды **getenforce** и **setenforce** используются для управления режимом SELinux системы.
- Команда **semanage** используется для управления правилами политики SELinux. Команда **restorecon** применяет контекст, определенный политикой.
- Логические значения — это переключатели, которые изменяют поведение политики SELinux. Их можно включить или отключить, и они используются для настройки политики.
- **sealert** отображает полезную информацию, помогающую в устранении неполадок SELinux.

ГЛАВА 6

УПРАВЛЕНИЕ ОСНОВНЫМИ УСТРОЙСТВАМИ ХРАНЕНИЯ

ЦЕЛЬ

Создавайте и управляйте устройствами хранения, разделами, файловыми системами и пространствами подкачки из командной строки.

ЗАДАЧИ

- Создайте разделов на устройствах хранения, форматирование их в файловых системах и монтирования для дальнейшего использования.
- Создание и управление пространствами подкачки (swop) для дополнения физической памяти.

РАЗДЕЛЫ

- Добавление разделов, файловых систем и постоянных подключений (и упражнения с пошаговыми инструкциями)
- Управление пространством подкачки (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Управление основными устройствами хранения.

ДОБАВЛЕНИЕ РАЗДЕЛОВ, ФАЙЛОВЫХ СИСТЕМ И ПОСТОЯННЫХ ПОДКЛЮЧЕНИЙ

ЦЕЛИ

После завершения этого раздела вы сможете создавать разделы хранилища, форматировать их в файловых системах и монтировать для использования.

РАЗДЕЛЕНИЕ ДИСКА

Разделение диска позволяет системным администраторам разделить жесткий диск на несколько логических единиц хранения, называемых разделами. Разделив диск на разделы, системные администраторы могут использовать разные разделы для выполнения разных функций. Например, разбиение диска необходимо или полезно в следующих ситуациях:

- Ограничивать доступное пространство приложениями или пользователями.
- Отделять файлы операционной системы и программы от пользовательских файлов.
- Создавать отдельную область для подкачки памяти.
- Ограничить использование дискового пространства для повышения производительности инструментов диагностики и резервного копирования.

Схема разбиения MBR

С 1982 года схема разбиения основной загрузочной записи (**Master Boot Record (MBR)**) определяет, как разбиваются диски в системах с прошивкой BIOS. Эта схема поддерживает максимум четыре первичных раздела. В системах Linux с использованием расширенных и логических разделов администраторы могут создать максимум 15 разделов. Поскольку данные о размере раздела хранятся в виде 32-разрядных значений, диски, разбитые по схеме MBR, имеют максимальный размер диска и раздела 2 ТиБ.

Рисунок 6.1: Разделение MBR устройства хранения данных /dev/vdb



Поскольку физические диски становятся все больше, а тома на основе SAN еще больше, ограничение размера диска и раздела в 2 ТиБ в схеме разбиения MBR больше не является теоретическим ограничением, а скорее реальной проблемой, с которой системные администраторы сталкиваются все чаще и чаще в производственных условиях. В результате устаревшая схема MBR находится в процессе замены новой таблицей разделов GUID (GPT) для разбиения диска.

Схема разделов GPT

Для систем с прошивкой **Unified Extensible Firmware Interface (UEFI)** **GPT** является стандартом для размещения таблиц разделов на физических жестких дисках. **GPT** является частью стандарта **UEFI** и устраняет многие ограничения, налагаемые старой схемой на основе **MBR**.

GPT обеспечивает максимум 128 разделов. В отличие от **MBR**, который использует 32 бита для хранения адресов логических блоков и информации о размере, **GPT** выделяет 64 бита для адресов логических блоков. Это позволяет **GPT** размещать разделы и диски размером до восьми зебибайт (ZiB) или восьми миллиардов тебибайт.

Помимо устранения ограничений схемы разбиения **MBR**, **GPT** также предлагает некоторые дополнительные функции и преимущества. **GPT** использует глобальный уникальный идентификатор (**GUID**) для идентификации каждого диска и раздела. В отличие от **MBR**, которая имеет единую точку отказа, **GPT** предлагает избыточность информации таблицы разделов.

Основной **GPT** находится в начале диска, а резервная копия, вторичный **GPT**, находится в конце диска. **GPT** использует контрольную сумму для обнаружения ошибок и повреждений в заголовке **GPT** и таблице разделов.

Рисунок 6.2: Разделение GPT устройства хранения /dev/vdb



УПРАВЛЕНИЕ РАЗДЕЛАМИ С ПОМОЩЬЮ УТИЛИТЫ PARTED

Редакторы разделов — это программы, которые позволяют администраторам вносить изменения в разделы диска, например, создавать разделы, удалять разделы и изменять типы разделов. Для выполнения этих операций администраторы могут использовать различные утилиты, например, редактор разделов **parted** как для схемы разделов **MBR**, так и для схемы разделов **GPT**.

Команда **parted** принимает имя устройства всего диска в качестве первого аргумента и одной или нескольких подкоманд. В следующем примере подкоманда **print** используется для отображения таблицы разделов на диске **/dev/vda**.

```
[root@host ~]# parted /dev/vda print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 53.7GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
```

```
Number  Start   End     Size    Type      File system  Flags
 1      1049kB  10.7GB  10.7GB  primary   xfs          boot
 2      10.7GB  53.7GB  42.9GB  primary   xfs
```

Если вы не укажете подкоманду, **parted** откроет интерактивный сеанс для ввода команд.

```
[root@host ~]# parted /dev/vda
GNU Parted 3.2
Using /dev/vda
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 53.7GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system  Flags
 1      1049kB  10.7GB  10.7GB  primary   xfs          boot
 2      10.7GB  53.7GB  42.9GB  primary   xfs

(parted) quit
```

По умолчанию **parted** отображает все размеры в степени 10 (КБ, МБ, ГБ). Вы можете изменить это значение по умолчанию с помощью подкоманды **unit**, которая принимает следующие параметры:

- **s** для сектора
- **B** для байта
- **MiB**, **GiB** или **TiB** (степень 2)
- **MB**, **GB** или **TB** (степень 10)

```
[root@host ~]# parted /dev/vda unit s print
Model: Virtio Block Device (virtblk)
Disk /dev/vda: 104857600s
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:
```

```
Number  Start       End        Size       Type      File system  Flags
 1      2048s       20971486s  20969439s  primary   xfs          boot
 2      20971520s   104857535s  83886016s  primary   xfs
```

Как показано в приведенном выше примере, вы также можете указать несколько подкоманд (такие как, **unit** и **print**) в одной строке.

Запись таблицы разделов на новый диск

Чтобы разбить новый диск, сначала нужно записать на него метку диска. Метка диска указывает, какую схему разбиения использовать.



ПРИМЕЧАНИЕ

Имейте в виду, что команда **parted** вносит изменения немедленно. Ошибка с **parted** определенно может привести к потере данных.

В качестве пользователя **root** используйте следующую команду, чтобы записать на диск метку **MBR**-диска.

```
[root@host ~]# parted /dev/vdb mklabel msdos
```

Чтобы записать метку диска **GPT**, используйте следующую команду.

```
[root@host ~]# parted /dev/vdb mklabel gpt
```



ПРЕДУПРЕЖДЕНИЕ

Подкоманда **mklabel** очищает существующую таблицу разделов. Используйте **mklabel** только в том случае, если предполагается повторное использование диска без учета существующих данных. Если новая метка изменит границы раздела, все данные в существующих файловых системах станут недоступны.

Создание разделов MBR

Создание раздела диска MBR включает несколько шагов:

1. Укажите дисковое устройство, на котором будет создан раздел.

Как пользователь **root**, выполните команду **parted** и укажите имя дискового устройства в качестве аргумента. Будет выполнена команда **parted** в интерактивном режиме и отображает командную строку.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

2. Используйте подкоманду **mkpart** для создания нового основного или расширенного раздела.

```
(parted) mkpart
Partition type? primary/extended? primary
```



ПРИМЕЧАНИЕ

В ситуациях, когда вам нужно более четырех разделов на диске с разделами **MBR**, создайте три основных раздела и один расширенный раздел. Этот расширенный раздел служит контейнером, внутри которого вы можете создать несколько логических разделов.

3. Укажите тип файловой системы, которую вы хотите создать на разделе, например, **xfs** или **ext4**. Данное действие не создает файловую систему на разделе; это только указание типа раздела.

```
File system type? [ext2]? Xfs
```

Чтобы получить список поддерживаемых типов файловых систем, используйте следующую команду:

```
[root@host ~]# parted /dev/vdb help mkpart
  mkpart PART-TYPE [FS-TYPE] START END      make a partition

  PART-TYPE is one of: primary, logical, extended
  FS-TYPE is one of: btrfs, nilfs2, ext4, ext3, ext2, fat32, fat16, hfsx,
  hfs+, hfs, jfs, swsusp, linux-swap(v1), linux-swap(v0), ntfs, reiserfs,
  hp-ufs, sun-ufs, xfs, apfs2, apfs1, asfs, amufs5, amufs4, amufs3,
  amufs2, amufs1, amufs0, amufs, affs7, affs6, affs5, affs4, affs3, affs2,
  affs1, affs0, linux-swap, linux-swap(new), linux-swap(old)
  START and END are disk locations, such as 4GB or 10%. Negative values
  count from the end of the disk. For example, -1s specifies exactly the
  last sector.

  'mkpart' makes a partition without creating a new file system on the
  partition. FS-TYPE may be specified to set an appropriate partition
  ID.
```

- Укажите сектор на диске, с которого начинается новый раздел.

Start? **2048s**

Обратите внимание на суффикс **s** для указания значения в секторах. Вы также можете использовать суффиксы **MiB**, **GiB**, **TiB**, **MB**, **GB** или **TB**. Если вы не укажете суффикс, по умолчанию будет **МБ**. **parted** может округлить значение, которое вы предоставляемые, чтобы удовлетворить дисковые ограничения.

Когда **parted** запускается, он получает топологию диска с устройства. Например, размер физического блока диска обычно является параметром, который собирает утилита **parted**. Используя эту информацию, **parted** гарантирует, что указанная вами начальная позиция правильно выровняет раздел со структурой диска. Правильное выравнивание разделов важно для оптимальной производительности. Если начальная позиция приводит к смещению раздела, **parted** отображает предупреждение. Для большинства дисков начальный сектор, кратный **2048**, является безопасным предположением.

- Укажите сектор диска, где должен заканчиваться новый раздел.

End? **1000MB**

С **parted** вы не можете напрямую указать размер вашего раздела, но вы можете быстро вычислить его по следующей формуле:

Size = End – Start

Как только вы укажете конечную позицию, **parted** обновит таблицу разделов на диске новыми сведениями о разделе.

- Выход из утилиты **parted**.

(parted) **quit**

Information: You may need to update /etc/fstab.

[root@host ~]#

- Запустите команду **udevadm settle**. Эта команда ожидает, пока система обнаружит новый раздел и создаст связанный с ним файл устройства в каталоге **/dev**. Команда завершить выполнение только тогда, когда это сделано.

[root@host ~]# **udevadm settle**
[root@host ~]#

В качестве альтернативы интерактивному режиму вы также можете создать раздел следующим образом:

```
[root@host ~]# parted /dev/vdb mkpart primary xfs 2048s 1000MB
```

Создание разделов GPT

Схема GPT также использует команду **parted** для создания новых разделов:

1. Укажите дисковое устройство, на котором будет создан раздел.

Как пользователь **root**, выполните команду **parted** с указанием дискового устройства в качестве единственного аргумента, чтобы запустить **parted** в интерактивном режиме с помощью командной строки.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

2. Используйте подкоманду **mkpart**, чтобы начать создание нового раздела.

В схеме **GPT** каждому разделу дается имя.

```
(parted) mkpart
Partition name? []? usersdata
```

3. Укажите тип файловой системы, которую вы хотите создать на разделе, например, **xfs** или **ext4**. Данное действие не создает файловую систему на разделе; это только указание типа раздела.

```
File system type? [ext2]? xfs
```

4. Укажите сектор на диске, с которого начинается новый раздел.

```
Start? 2048s
```

5. Укажите сектор диска, где должен заканчиваться новый раздел.

```
End? 1000MB
```

Как только вы укажете конечную позицию, **parted** обновит таблицу разделов на диске новыми сведениями о разделе.

6. Выход из **parted**.

```
(parted) quit  
Information: You may need to update /etc/fstab.  
[root@host ~]#
```

7. Запустите команду **udevadm settle**. Данная команда ожидает, пока система обнаружит новый раздел и создаст связанный с ним файл устройства в каталоге **/dev**. Команда завершить выполнение только тогда, когда это сделано.

```
[root@host ~]# udevadm settle  
[root@host ~]#
```

В качестве альтернативы интерактивному режиму вы также можете создать раздел следующим образом:

```
[root@host ~]# parted /dev/vdb mkpart usersdata xfs 2048s 1000MB
```

Удаление разделов

Следующие шаги применимы как для схем разбиения **MBR**, так и для схем разделов **GPT**.

1. Укажите диск, на котором находится удаляемый раздел.

Как пользователь **root**, выполните команду **parted** с именем дискового устройства в качестве единственного аргумента, чтобы запустить **parted** в интерактивном режиме с помощью командной строки.

```
[root@host ~]# parted /dev/vdb  
GNU Parted 3.2  
Using /dev/vdb  
Welcome to GNU Parted! Type 'help' to view a list of commands.  
(parted)
```

2. Определите номер удаляемого раздела.

```
(parted) print
```

```
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	1000MB	999MB	xfs		usersdata

3. Удалите раздел.

```
(parted) rm 1
```

Подкоманда **rm** немедленно удаляет раздел из таблицы разделов на диске.

4. Выход **parted**.

```
(parted) quit
Information: You may need to update /etc/fstab.

[root@host ~]#
```

СОЗДАНИЕ ФАЙЛОВЫХ СИСТЕМ

После создания блочного устройства следующим шагом будет добавление к нему файловой системы. Red Hat Enterprise Linux поддерживает множество различных типов файловых систем, но наиболее распространенными являются **XFS** и **ext4**. **Anaconda**, установщик Red Hat Enterprise Linux, по умолчанию использует **XFS**.

В качестве пользователя **root** используйте команду **mkfs.xfs**, чтобы применить файловую систему **XFS** к блочному устройству. Для **ext4** используйте **mkfs.ext4**.

```
[root@host ~]# mkfs.xfs /dev/vdb1
meta-data=/dev/vdb1              isize=512    agcount=4, agsize=60992 blks
                                =                      sectsz=512  attr=2, projid32bit=1
                                =                      crc=1      finobt=1, sparse=1, rmapbt=0
                                =                      reflink=1
data     =                      bsize=4096   blocks=243968, imaxpct=25
                                =                      sunit=0    swidth=0 blks
naming   =version 2             bsize=4096   ascii-ci=0, ftype=1
log      =internal log          bsize=4096   blocks=1566, version=2
                                =                      sectsz=512  sunit=0 blks, lazy-count=1
realtime =none                  extsz=4096   blocks=0, rtextents=0
```

ПОДМОНТИРОВАНИЕ ФАЙЛОВЫХ СИСТЕМ

После того, как вы добавили файловую систему, последний шаг — смонтировать файловую систему в каталог в структуре каталогов. Когда вы монтируете файловую систему в иерархию каталогов, утилиты пользовательского пространства могут получать доступ к файлам на устройстве или записывать их.

Ручное монтирование файловых систем

Администраторы используют команду **mount** для ручного подключения устройства к каталогу или точке монтирования. Команда **mount** принимает в качестве аргументов устройство, точку монтирования и, возможно, параметры файловой системы. Параметры файловой системы настраивают поведение файловой системы.

```
[root@host ~]# mount /dev/vdb1 /mnt
```

Вы также используете команду **mount** для просмотра текущих смонтированных файловых систем, точек монтирования и параметров.

```
[root@host ~]# mount | grep vdb1
/dev/vdb1 on /mnt type xfs (rw,relatime,seclabel,attr2,inode64,noquota)
```

Постоянное монтирование файловых систем

Ручное монтирование файловой системы — хороший способ убедиться, что отформатированное устройство доступно и работает должным образом. Однако, когда сервер перезагружается, система не монтирует автоматически файловую систему в дерево каталогов; хотя данные не повреждены в файловой системе, но пользователи не могут получить к ним доступ.

Чтобы быть уверенным, что система автоматически монтирует файловую систему при загрузке системы, добавьте запись в файл **/etc/fstab**. В этом файле конфигурации перечислены файловые системы, которые необходимо монтировать при загрузке системы. **/etc/fstab** — это файл, разделенный пробелами, с шестью полями в строке.

```
[root@host ~]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Wed Feb 13 16:39:59 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
```

```
# units generated from this file.  
#  
UUID=a8063676-44dd-409a-b584-68be2c9f5570 / xfs defaults 0 0  
UUID=7a20315d-ed8b-4e75-a5b6-24ff9e1f9838 /dbdata xfs defaults 0 0
```

Когда вы добавляете или удаляете запись в файле `/etc/fstab`, выполните команду `systemctl daemon-reload` или перезагрузите сервер, чтобы `systemd` зарегистрировала новую конфигурацию.

```
[root@host ~]# systemctl daemon-reload
```

Первое поле определяет устройство. В этом примере для указания устройства используется **UUID**. Файловые системы создают и сохраняют **UUID** в своем суперблоке во время создания. В качестве альтернативы вы можете использовать файл устройства, например, `/dev/vdb1`.



ПРИМЕЧАНИЕ

Использование **UUID** предпочтительнее, поскольку идентификаторы блочных устройств могут меняться в определенных сценариях, например, когда облачный провайдер меняет базовый уровень хранения виртуальной машины или диски обнаруживаются в другом порядке при каждой загрузке системы. Имя файла блочного устройства может измениться, но **UUID** остается постоянным в суперблоке файловой системы.

Используйте команду `lsblk --fs` для сканирования блочных устройств, подключенных к машине, и получения **UUID** файловой системы.

```
[root@host ~]# lsblk --fs  
NAME   FSTYPE LABEL UUID                                     MOUNTPOINT  
sr0  
vda  
└─vda1 xfs   a8063676-44dd-409a-b584-68be2c9f5570 /  
vdb  
└─vdb1 xfs   7a20315d-ed8b-4e75-a5b6-24ff9e1f9838 /dbdata
```

Второе поле — это точка монтирования каталога, из которой блочное устройство будет доступно в структуре каталогов. Точка монтирования должна существовать; если нет, создайте его с помощью команды `mkdir`.

Третье поле содержит тип файловой системы, такой как `xfs` или `ext4`.

Четвертое поле представляет собой список параметров, разделенных запятыми, которые можно применить к устройству. **defaults** — это набор часто используемых опций. **Man**-страница `mount(8)` документирует другие доступные параметры.

Пятое поле используется командой **dump** для резервного копирования устройства. Другие приложения резервного копирования обычно не используют это поле.

Последнее поле, поле порядка **fsck**, определяет, следует ли запускать команду **fsck** при загрузке системы для проверки чистоты файловых систем. Значение в этом поле указывает порядок запуска **fsck**. Для файловых систем **XFS** установите в этом поле значение **0**, поскольку **XFS** не использует **fsck** для проверки состояния своей файловой системы. Для файловых систем **ext4** установите значение **1** для корневой файловой системы и **2** для остальных файловых систем **ext4**. Таким образом, **fsck** сначала обрабатывает корневую файловую систему, а затем одновременно проверяет файловые системы на отдельных дисках и последовательно файловые системы на одном диске.



ПРИМЕЧАНИЕ

Наличие неправильной записи в **/etc/fstab** может привести к тому, что машина не загружается. Администраторы должны убедиться, что запись действительна, размонтировав новую файловую систему и воспользовавшись командой **mount** /**mountpoint**, которая читается как **/etc/fstab**, для повторного монтирования файловой системы. Если команда **mount** возвращает ошибку, исправьте ее перед перезагрузкой машины.

В качестве альтернативы вы можете использовать команду **findmnt --verify** для управления файлом **/etc/fstab**.



РЕКОМЕНДАЦИИ

info parted (GNU Parted User Manual)

parted(8), mkfs(8), mount(8), lsblk(8), and fstab(5) man pages

Дополнительные сведения см. в руководстве по настройке файловых систем и управлению ими по адресу

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/htmlsingle/configuring_and_managing_file_systems/

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ДОБАВЛЕНИЕ РАЗДЕЛОВ, ФАЙЛОВЫХ СИСТЕМ И ПОСТОЯННОГО МОНТИРОВАНИЯ ФАЙЛОВЫХ СИСТЕМ

В этом упражнении вы создадите раздел на новом устройстве хранения, отформатируете его в файловой системе **XFS**, настроите его для монтирования при загрузке и смонтируете для использования.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность использовать утилиты **parted**, **mkfs.xfs** и другие команды для создания раздела на новом диске, его форматирования и постоянного монтирования.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab storage-partitions start**. Эта команда запускает сценарий, который определяет, доступен ли сервер **servera** в сети. Он также подготавливает второй диск на сервере к упражнению.

```
[student@workstation ~]$ lab storage-partitions start
```

1. Используйте команду **ssh**, чтобы войти на сервер как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Если будет предложено, используйте слово **student** в качестве пароля.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

3. Используйте **parted** для создания новой метки диска типа **msdos** на диске **/dev/vdb**, чтобы подготовить этот новый диск к схеме разбиения **MBR**.

```
[root@servera ~]# parted /dev/vdb mklabel msdos  
Information: You may need to update /etc/fstab.
```

4. Добавьте новый основной раздел размером **1 ГБ**. Для правильного выравнивания начните раздел с сектора **2048**. Установите тип файловой системы раздела на **XFS**.

4.1. Используйте разделенный интерактивный режим, чтобы помочь вам создать раздел.

```
[root@servera ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mkpart
Partition type? primary/extended? primary
File system type? [ext2]? xfs
Start? 2048s
End? 1001MB
(parted) quit
Information: You may need to update /etc/fstab.
```

Поскольку раздел начинается с сектора 2048, предыдущая команда задает конечную позицию 1001 МБ, чтобы получить размер раздела 1000 МБ (1 ГБ).

В качестве альтернативы вы можете выполнить ту же операцию с помощью следующей не интерактивной команды: **parted /dev/vdb mkpart primary xfs 2048s 1001MB**

4.2. Проверьте свою работу, перечислив разделы в **/dev/vdb**.

```
[root@servera ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start   End     Size    Type      File system  Flags
 1       1049kB  1001MB  1000MB  primary
```

- 4.3.** Запустите команду **udevadm settle**. Эта команда ожидает, пока система зарегистрирует новый раздел, и завершает выполнение, когда это будет сделано.

```
[root@servera ~]# udevadm settle
[root@servera ~]#
```

- 5.** Отформатируйте новый раздел в файловой системе **XFS**.

```
[root@servera ~]# mkfs.xfs /dev/vdb1
meta-data=/dev/vdb1              isize=512    agcount=4, agsize=61056 blks
                                =          sectsz=512  attr=2, projid32bit=1
                                =          crc=1     finobt=1, sparse=1, rmapbt=0
                                =          reflink=1
data     =           bsize=4096   blocks=244224, imaxpct=25
          =           sunit=0    swidth=0 blks
naming   =version 2             bsize=4096   ascii-ci=0, ftype=1
log      =internal log         bsize=4096   blocks=1566, version=2
          =           sectsz=512  sunit=0 blks, lazy-count=1
realtime =none                  extsz=4096   blocks=0, rtextents=0
```

6. Настройте новую файловую систему для постоянного монтирования в **/archive**.

6.1. Используйте **mkdir** для создания точки монтирования каталога **/archive**.

```
[root@servera ~]# mkdir /archive
[root@servera ~]#
```

6.2. Используйте команду **lsblk** с параметром **--fs**, чтобы узнать **UUID** устройства **/dev/vdb1**.

```
[root@servera ~]# lsblk --fs /dev/vdb
NAME   FSTYPE LABEL UUID                                     MOUNTPOINT
vdb
└─vdb1  xfs   e3db1abe-6d96-4faa-a213-b96a6f85dcc1
```

UUID в предыдущем выводе, вероятно, отличается в вашей системе.

6.3. Добавьте запись в **/etc/fstab**. В следующей команде замените **UUID** на тот, который вы обнаружили на предыдущем шаге.

```
[root@servera ~]# vim /etc/fstab
...output omitted...
UUID=e3db1abe-6d96-4faa-a213-b96a6f85dcc1  /archive  xfs  defaults
  0  0
```

6.4. Обновите **systemd**, чтобы система зарегистрировала новую конфигурацию **/etc/fstab**.

```
[root@servera ~]# systemctl daemon-reload
```

6.5. Выполните команду **mount /archive**, чтобы смонтировать новую файловую систему, используя новую запись, добавленную в **/etc/fstab**.

```
[root@servera ~]# mount /archive  
[root@servera ~]#
```

6.6. Убедитесь, что новая файловая система смонтирована в **/archive**.

```
[root@servera ~]# mount | grep /archive  
/dev/vdb1 on /archive type xfs  
(rw,relatime,seclabel,attr2,inode64,noquota)
```

7. Перезагружаем сервер **servera**. После перезагрузки сервера войдите в систему и убедитесь, что **/dev/vdb1** смонтирован в **/archive**. Когда закончите, выйдите из сервера **servera**.

7.1. Перезагрузите сервер.

```
[root@servera ~]# systemctl reboot  
Connection to servera closed by remote host.  
Connection to servera closed.  
[student@workstation ~]$
```

7.2. Подождите несколько минут, пока **servera** перезагрузится, и войдите в систему как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

7.3. Убедитесь, что **/dev/vdb1** смонтирован в **/archive**.

```
[student@servera ~]$ mount | grep /archive  
/dev/vdb1 on /archive type xfs  
(rw,relatime,seclabel,attr2,inode64,noquota)
```

7.4. Выйдите из сервера **servera**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab storage-partitions finish**, чтобы завершить данное упражнение.

```
[student@workstation ~]$ lab storage-partitions finish
```

На этом упражнения с пошаговыми инструкциями заканчивается.

УПРАВЛЕНИЕ ПРОСТРАНСТВОМ ПОДКАЧКИ SWAP

ЦЕЛИ

После завершения этого раздела вы сможете создавать пространства подкачки и управлять ими в дополнение к физической памяти.

ПРЕДСТАВЛЯЕМ КОНЦЕПЦИИ SWAP SPACE

Пространство подкачки (*swap space*) — это область диска, находящаяся под контролем подсистемы управления памятью ядра Linux. Ядро использует пространство подкачки для пополнения системной оперативной памяти, удерживая неактивные страницы памяти. Объединенная системная оперативная память плюс пространство подкачки называется виртуальной памятью (*virtual memory*).

Когда использование памяти в системе превышает определенный предел, ядро просматривает ОЗУ в поисках неиспользуемых страниц памяти, назначенных процессам. Ядро записывает незанятые страницы в область подкачки и переназначает страницы ОЗУ другим процессам. Если программе требуется доступ к странице на диске, ядро находит другую незанятую страницу памяти, записывает ее на диск, а затем вызывает нужную страницу из области подкачки.

Поскольку области подкачки находятся на диске, подкачка выполняется медленнее по сравнению с оперативной памятью. Хотя он используется для увеличения системной оперативной памяти, вы не должны рассматривать пространство подкачки как надежное решение при недостаточном объеме оперативной памяти для вашей рабочей нагрузки.

Размер пространства подкачки

Администраторы должны определять размер пространства подкачки в зависимости от рабочей нагрузки на память в системе. Поставщики приложений иногда дают рекомендации по этому вопросу. В следующей таблице приведены некоторые рекомендации, основанные на общем объеме физической памяти.

Рекомендации по оперативной памяти и пространству подкачки.

ОЗУ	SWAP SPACE	ПРОСТРАНСТВО ПОДКАЧКИ, ЕСЛИ ДОПУСКАЕТСЯ СПЯЩИЙ РЕЖИМ
2 ГиБ или меньше	В два раза больше оперативной памяти	В три раза больше оперативной памяти
От 2 ГиБ до 8 ГиБ	То же, что ОЗУ	В два раза больше оперативной памяти
От 8 ГиБ до 64 ГиБ	Не менее 4 ГиБ	в 1,5 раза больше оперативной памяти
Более 64 ГиБ	Не менее 4 ГиБ	Спящий режим не рекомендуется

Функция гибернации (Спящий режим) ноутбука и настольного компьютера использует пространство подкачки для сохранения содержимого ОЗУ перед выключением системы. При повторном включении системы ядро восстанавливает содержимое ОЗУ из области подкачки и не требует полной загрузки. Для этих систем пространство подкачки должно быть больше, чем объем оперативной памяти.

Статья базы знаний в разделе «Справочник» в конце этого раздела содержит дополнительные рекомендации по определению размера пространства подкачки.

СОЗДАНИЕ ПРОСТРАНСТВА ПОДКАЧКИ

Чтобы создать пространство подкачки, вам необходимо выполнить следующее:

- Создайте раздел с файловой системой типа **linux-swap**.
- Разместить сигнатуры подкачки на устройстве.

Создание раздела подкачки

Используйте команду **parted** для создания раздела нужного размера и установите для него тип файловой системы **linux-swap**. В прошлом инструменты смотрели на тип файловой системы раздела, чтобы определить, следует ли активировать устройство; однако это уже не так. Несмотря на то, что утилиты больше не используют тип файловой системы раздела, установка этого типа позволяет администраторам быстро определить назначение раздела.

В следующем примере создается раздел размером 256 МБ.

```
[root@host ~]# parted /dev/vdb
GNU Parted 3.2
Using /dev/vdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
 1      1049kB  1001MB  1000MB          data
```

```
(parted) mkpart
Partition name? []? swap1
File system type? [ext2]? linux-swap
Start? 1001MB
End? 1257MB
(parted) print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

```
Number  Start   End     Size   File system   Name   Flags
 1      1049kB  1001MB  1000MB          data
 2      1001MB  1257MB  256MB   linux-swap(v1)  swap1
```

```
(parted) quit
```

```
Information: You may need to update /etc/fstab.
```

После создания раздела запустите команду **udevadm settle**. Эта команда ожидает, пока система обнаружит новый раздел и создаст соответствующий файл устройства в **/dev**. Выполнение команды завершается только тогда, когда это сделано.

```
[root@host ~]# udevadm settle
[root@host ~]#
```

Форматирование устройства

Команда **mkswap** применяется к устройству сигнатуру подкачки. В отличие от других утилит форматирования, **mkswap** записывает один блок данных в начало устройства, оставляя остальную часть устройства неформатированной, чтобы ядро могло использовать ее для хранения страниц памяти.

```
[root@host ~]# mkswap /dev/vdb2
Setting up swapspace version 1, size = 244 MiB (255848448 bytes)
no label, UUID=39e2667a-9458-42fe-9665-c5c854605881
```

АКТИВАЦИЯ ПРОСТРАНСТВА ПОДКАЧКИ

Вы можете использовать команду **swapon** для активации отформатированного пространства подкачки. Используйте **swapon** с устройством в качестве параметра или используйте **swapon -a**, чтобы активировать все области подкачки, перечисленные в файле **/etc/fstab**. Используйте команды **swapon --show** и **free** для проверки доступных пространств подкачки.

```
[root@host ~]# free
              total        used        free      shared  buff/cache   available
Mem:       1873036     134688     1536436      16748      201912     1576044
Swap:          0          0          0
[root@host ~]# swapon /dev/vdb2
[root@host ~]# free
              total        used        free      shared  buff/cache   available
Mem:       1873036     135044     1536040      16748      201952     1575680
Swap:    249852          0     249852
```

Вы можете деактивировать пространство подкачки с помощью команды **swapoff**. Если в пространство подкачки записаны страницы, **swapoff** пытается переместить эти страницы в другие активные пространства подкачки или обратно в память. Если он не может записывать данные в другие места, команда **swapoff** завершается с ошибкой, и пространство подкачки остается активным.

Постоянная активация пространства подкачки

Чтобы активировать пространство подкачки при каждой загрузке, поместите запись в файл **/etc/fstab**. В приведенном ниже примере показана типичная строка в файле **/etc/fstab** на основе пространства подкачки, созданного выше.

```
UUID=39e2667a-9458-42fe-9665-c5c854605881 swap swap defaults 0 0
```

В примере в качестве первого поля используется **UUID** устройства. Когда вы форматируете устройство, команда **mkswap** отображает этот **UUID**. Если вы потеряли вывод **mkswap**, используйте команду **lsblk --fs**. В качестве альтернативы вы также можете использовать имя устройства в первом поле.

Второе поле обычно зарезервировано для точки монтирования. Однако для устройств подкачки, которые недоступны через структуру каталогов, это поле принимает замещающее значение **swap**.

Третье поле — тип файловой системы. Тип файловой системы для пространства подкачки — **swap**.

Четвертое поле для опций. В примере используется параметр **defaults**. Параметр по умолчанию включает параметр монтирования **auto**, что означает автоматическую активацию пространства подкачки при загрузке системы.

Последние два поля — это флаг **dump** и порядок **fsck**. Пространства подкачки не требуют ни резервного копирования, ни проверки файловой системы, поэтому эти поля должны быть установлены равными нулю.

Когда вы добавляете или удаляете запись в файле **/etc/fstab**, выполните команду **systemctl daemon-reload** или перезагрузите сервер, чтобы **systemd** зарегистрировала новую конфигурацию.

```
[root@host ~]# systemctl daemon-reload
```

Установка приоритета пространства подкачки

По умолчанию система последовательно использует пространства подкачки, то есть ядро использует первое активированное пространство подкачки, пока оно не заполнится, а затем начинает использовать второе пространство подкачки. Однако вы можете определить приоритет для каждого пространства подкачки, чтобы принудительно установить этот порядок.

Чтобы установить приоритет, используйте параметр **pri** в файле **/etc/fstab**. Ядро сначала использует пространство подкачки с наивысшим приоритетом. Приоритет по умолчанию равен **-2**.

В следующем примере показаны три пространства подкачки, определенные в файле **/etc/fstab**. Ядро сначала использует последнюю запись с **pri=10**. Когда это пространство заполнено, используется вторая запись с **pri=4**. Наконец, он использует первую запись, которая имеет приоритет по умолчанию **-2**.

```
UUID=af30cbb0-3866-466a-825a-58889a49ef33    swap      swap      defaults  0 0
UUID=39e2667a-9458-42fe-9665-c5c854605881    swap      swap      pri=4     0 0
UUID=fbcd7fa60-b781-44a8-961b-37ac3ef572bf    swap      swap      pri=10    0 0
```

Используйте команду **swapon --show** для отображения приоритетов пространства подкачки.

Когда области подкачки имеют одинаковый приоритет, ядро записывает в них по кругу.



РЕКОМЕНДАЦИИ

Справочные страницы **man mkswap(8)**, **swapon(8)**, **swapoff(8)**, **mount(8)**, и **parted(8)**

База знаний: Каков рекомендуемый размер подкачки для платформ Red Hat?

<https://access.redhat.com/solutions/15244>

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ ПРОСТРАНСТВОМ ПОДКАЧКИ

В этом упражнении вы создадите и отформатируете раздел для использования в качестве пространства подкачки, отформатируете его как подкачку и активируете на постоянной основе.

В РЕЗУЛЬТАТЕ

Вы должны быть в состоянии создать раздел и область подкачки (**swop**) на диске, используя схему разделов **GPT**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab storage-swap start**. Эта команда запускает сценарий, который определяет, доступен ли сервер **servera** в сети. Он также подготавливает второй диск на сервере к упражнению.

```
[student@workstation ~]$ lab storage-swap start
```

1. Используйте команду **ssh**, чтобы войти на сервер **servera** как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Если будет предложено, используйте слово **student** в качестве пароля.

```
[student@servera ~]$ sudo -i  
[sudo] password for student:  
[root@servera ~]#
```

3. Используйте команду **parted** для проверки диска **/dev/vdb**.

```
[root@servera ~]# parted /dev/vdb print  
Model: Virtio Block Device (virtblk)  
Disk /dev/vdb: 5369MB  
Sector size (logical/physical): 512B/512B
```

```
Partition Table: gpt
```

```
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	1001MB	1000MB		data	

Обратите внимание, что на диске уже есть таблица разделов и используется схема разделов **GPT**. Кроме того, раздел размером 1 ГБ уже существует.

- Добавьте новый раздел размером 500 МБ для использования в качестве пространства подкачки. Установите тип раздела на **linux-swap**.

4.1. Используйте команду **parted** для создания нового раздела. Поскольку на диске используется схема разбиения **GPT**, вам необходимо дать имя разделу. Назовите это **myswap**.

```
[root@servera ~]# parted /dev/vdb mkpart myswap linux-swap 1001MB  
1501MB
```

```
Information: You may need to update /etc/fstab.
```

Обратите внимание, что в предыдущей команде начальная позиция, 1001 МБ, является концом существующего первого раздела. Таким образом, **parted** гарантирует, что новый раздел сразу же следует за предыдущим, без какого-либо промежутка.

Поскольку раздел начинается с позиции 1001 МБ, команда устанавливает конечную позицию на 1501 МБ, чтобы получить размер раздела 500 МБ.

- Проверьте свою работу, перечислив разделы в **/dev/vdb**.

```
[root@servera ~]# parted /dev/vdb print
```

```
Model: Virtio Block Device (virtblk)
```

```
Disk /dev/vdb: 5369MB
```

```
Sector size (logical/physical): 512B/512B
```

```
Partition Table: gpt
```

```
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	1001MB	1000MB		data	

Размер нового раздела не соответствует точно 500 МБ. Это связано с тем, что **parted** должен выровнять раздел с разметкой диска.

- Запустите команду **udevadm settle**. Даная команда ожидает, пока система зарегистрирует новый раздел, и завершается, когда это будет сделано.

```
[root@servera ~]# udevadm settle  
[root@servera ~]#
```

5. Инициализируйте только что созданный раздел как область подкачки.

```
[root@servera ~]# mkswap /dev/vdb2  
Setting up swap space version 1, size = 476 MiB (499118080 bytes)  
no label, UUID=cb7f71ca-ee82-430e-ad4b-7dda12632328
```

6. Включите только что созданное пространство подкачки.

6.1. Используйте команду **swapon --show**, чтобы увидеть, что действия по созданию и инициализации пространства подкачки еще не позволяет его использовать.

```
[root@servera ~]# swapon --show  
[root@servera ~]#
```

6.2. Включите только что созданное пространство подкачки.

```
[root@servera ~]# swapon /dev/vdb2  
[root@servera ~]#
```

6.3. Убедитесь, что только что созданное пространство подкачки теперь доступно.

```
[root@servera ~]# swapon -show  
NAME      TYPE      SIZE USED PRIO  
/dev/vdb2 partition 476M   0B    -2
```

6.4. Отключите пространство подкачки.

```
[root@servera ~]# swapoff /dev/vdb2  
[root@servera ~]#
```

6.5. Убедитесь, что пространство подкачки отключено.

```
[root@servera ~]# swapon --show  
[root@servera ~]#
```

7. Настройте новое пространство подкачки для включения при загрузке системы.

7.1. Используйте команду **lsblk** с параметром **--fs**, чтобы узнать **UUID** устройства **/dev/vdb2**.

```
[root@servera ~]# lsblk --fs /dev/vdb2
NAME FSTYPE LABEL UUID                                     MOUNTPOINT
vdb2 swap          cb7f71ca-ee82-430e-ad4b-7dda12632328
```

UUID в предыдущем выводе, вероятно, отличается в вашей системе.

7.2. Добавьте запись в **/etc/fstab**. В следующей команде замените **UUID** на тот, который вы обнаружили на предыдущем шаге.

```
[root@servera ~]# vim /etc/fstab
...output omitted...
UUID=cb7f71ca-ee82-430e-ad4b-7dda12632328    swap    swap    defaults    0
0
```

7.3. Обновите **systemd**, чтобы система зарегистрировала новую конфигурацию **/etc/fstab**.

```
[root@servera ~]# systemctl daemon-reload
[root@servera ~]#
```

7.4. Включите пространство подкачки, используя запись, только что добавленную в **/etc/fstab**.

```
[root@servera ~]# swapon -a
[root@servera ~]#
```

7.5. Убедитесь, что новое пространство подкачки включено.

```
[root@servera ~]# swapon -show
NAME      TYPE      SIZE USED PRI0
/dev/vdb2 partition 476M   0B   -2
```

8. Перезагружаем сервер **servera**. После перезагрузки сервера войдите в систему и убедитесь, что пространство подкачки включено. Когда закончите, выйдите из сервера.

8.1. Перезагрузите сервер **servera**.

```
[root@servera ~]# systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
```

```
[student@workstation ~]$
```

8.2. Подождите несколько минут, пока **servera** перезагрузится, и войдите в систему как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

8.3. Убедитесь, что пространство подкачки включено.

```
[root@servera ~]# swapon -show  
NAME      TYPE      SIZE USED PRIO  
/dev/vdb2  partition 476M   0B    -2
```

8.4. Выходите из сервера **servera**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab storage-swap finish**, чтобы завершить выполнение упражнения.

```
[student@workstation ~]$ lab storage-swap finish
```

На этом упражнения с пошаговыми инструкциями заканчивается.

ЛАБОРАТОРНАЯ РАБОТА

УПРАВЛЕНИЕ ОСНОВНЫМИ УСТРОЙСТВАМИ ХРАНЕНИЯ КОНТРОЛЬНЫЙ СПИСОК

В этой лабораторной работе вы создадите несколько разделов на новом диске, отформатируете некоторые из файловых систем и смонтируете их, а другие активируете как пространства подкачки.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- С помощью команды **parted** отобразить и создать разделы.
- Создавать новые файловые системы на разделах и постоянно их монтировать.
- Создавать области подкачки и активировать их при загрузке.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab storage-review start**. Эта команда запускает сценарий, который определяет, доступна ли хост **serverb** в сети. Он также готовит к упражнению второй диск на сервере **serverb**.

```
[student@workstation ~]$ lab storage-review start
```

1. Доступны новые диски на **serverb**. На первом новом диске создайте раздел **GPT** размером **2 ГБ** с именем **backup**. Поскольку установить точный размер может быть сложно, допускается размер от 1,8 ГБ до 2,2 ГБ. Установите правильный тип файловой системы в этом разделе для размещения файловой системы **XFS**.

Пароль для учетной записи **student** на **serverb** — **student**. Этот пользователь имеет полный **root**-доступ через **sudo**.

2. Отформатируйте раздел размером **2 ГБ** с файловой системой **XFS** и постоянно смонтируйте его в точке **/backup**.
3. На том же новом диске создайте два раздела **GPT** размером **512 МБ** с именами **swap1** и **swap2**. Допускается размер от 460 МБ до 564 МБ. Установите правильный тип файловой системы на этих разделах для размещения областей подкачки.
4. Инициализируйте два раздела по 512 МБ как области подкачки и настройте их активацию при загрузке. Установите пространство подкачки в разделе подкачки 2, чтобы оно было предпочтительным по сравнению с другим.

5. Для проверки работы перезагрузите **serverb**. Убедитесь, что система автоматически монтирует первый раздел в **/backup**. Также убедитесь, что система активирует два пространства подкачки.

Когда закончите, выйдите из **serverb**.

Оценка

На рабочей станции **workstation**, запустите сценарий **lab storage-review grade**, чтобы подтвердить успешное выполнение упражнения.

```
[student@workstation ~]$ lab storage-review grade
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab storage-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab storage-review finish
```

На этом лабораторная работа завершается.

РЕШЕНИЕ

УПРАВЛЕНИЕ ОСНОВНЫМИ УСТРОЙСТВАМИ ХРАНЕНИЯ

КОНТРОЛЬНЫЙ СПИСОК

В этой лабораторной работе вы создадите несколько разделов на новом диске, отформатируете некоторые из файловых систем и смонтируете их, а другие активируете как пространства подкачки.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- С помощью команды **parted** отобразить и создать разделы.
- Создавать новые файловые системы на разделах и постоянно их монтировать.
- Создавать области подкачки и активировать их при загрузке.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab storage-review start**. Эта команда запускает сценарий, который определяет, доступна ли хост **serverb** в сети. Он также готовит к упражнению второй диск на сервере **serverb**.

```
[student@workstation ~]$ lab storage-review start
```

1. Доступны новые диски на **serverb**. На первом новом диске создайте раздел **GPT** размером **2 ГБ** с именем **backup**. Поскольку установить точный размер может быть сложно, допускается размер от 1,8 ГБ до 2,2 ГБ. Установите правильный тип файловой системы в этом разделе для размещения файловой системы **XFS**.

Пароль для учетной записи **student** на **serverb** — **student**. Этот пользователь имеет полный **root**-доступ через **sudo**.

- 1.1. Используйте команду **ssh** для входа на **serverb** в качестве пользователя **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

- 1.2.** Поскольку для создания разделов и файловых систем требуется **root**-доступ, используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Если будет предложено, используйте слово **student** в качестве пароля.

```
[student@serverb ~]$ sudo -i  
[sudo] password for student: student  
[root@serverb ~]#
```

- 1.3.** Используйте команду **lsblk** для определения новых дисков. На этих дисках еще не должно быть разделов.

```
[root@serverb ~]# lsblk  
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT  
sr0     11:0    1 1024M  0 rom  
vda    252:0    0   10G  0 disk  
└─vda1 252:1    0   10G  0 part /  
vdb  252:16   0    5G  0 disk  
vdc    252:32   0    5G  0 disk  
vdd    252:48   0    5G  0 disk
```

Обратите внимание, что на первом новом диске **vdb** нет разделов.

- 1.4.** Убедитесь, что диск не размечен.

```
[root@serverb ~]# parted /dev/vdb print  
Error: /dev/vdb: unrecognised disk label  
Model: Virtio Block Device (virtblk)  
Disk /dev/vdb: 5369MB  
Sector size (logical/physical): 512B/512B  
Partition Table: unknown  
Disk Flags:
```

- 1.5.** Используйте **parted** и подкоманду **mklabel** для определения схемы разбиения **GPT**.

```
[root@serverb ~]# parted /dev/vdb mklabel gpt  
Information: You may need to update /etc/fstab.
```

- 1.6.** Создайте раздел размером 2 ГБ. Назовите его **backup** и установите для него тип файловой системы **xfs**. Начните раздел с сектора 2048.

```
[root@serverb ~]# parted /dev/vdb mkpart backup xfs 2048s 2GB  
Information: You may need to update /etc/fstab.
```

1.7. Подтвердите правильность создания нового раздела.

```
[root@serverb ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name     Flags
 1      1049kB  2000MB  1999MB          backup
```

1.8. Запустите команду установки **udevadm settle**. Эта команда ожидает, пока система обнаружит новый раздел и создаст файл устройства **/dev/vdb1**. Команда завершится только тогда, когда действие будет завершено.

```
[root@serverb ~]# udevadm settle
[root@serverb ~]#
```

2. Отформатируйте раздел размером **2 ГБ** с файловой системой **XFS** и постоянно смонтируйте его в точке **/backup**.

2.1. Используйте команду **mkfs.xfs** для форматирования раздела **/dev/vbd1**.

```
[root@serverb ~]# mkfs.xfs /dev/vdb1
meta-data=/dev/vdb1              isize=512    agcount=4, agsize=121984
  blks
        =                      sectsz=512  attr=2, projid32bit=1
        =                      crc=1       finobt=1, sparse=1,
rmapbt=0
        =
data     =                      reflink=1
        =
        bsize=4096   blocks=487936, imaxpct=25
naming   =version 2            sunit=0     swidth=0 blks
log      =internal log         bsize=4096   blocks=2560, version=2
        =
        sectsz=512  sunit=0 blks, lazy-
count=1
realtime =none                 extsz=4096  blocks=0, rtextents=0
```

2.2. Создайте точку монтирования **/backup**.

```
[root@serverb ~]# mkdir /backup
[root@serverb ~]#
```

2.3. Перед добавлением новой файловой системы в **/etc/fstab** получите ее **UUID**.

```
[root@serverb ~]# lsblk --fs /dev/vdb1
NAME FSTYPE LABEL UUID                                     MOUNTPOINT
vdb1 xfs          a3665c6b-4bfb-49b6-a528-74e268b058dd
```

UUID в вашей системе, вероятно, отличается.

2.4. Отредактируйте файл **/etc/fstab** и определите новую файловую систему.

```
[root@serverb ~]# vim /etc/fstab
...output omitted...
UUID=a3665c6b-4bfb-49b6-a528-74e268b058dd    /backup      xfs      defaults
0 0
```

2.5. Перегрузите **systemd** перечитать файл **/etc/fstab**.

```
[root@serverb ~]# systemctl daemon-reload
[root@serverb ~]#
```

2.6. Вручную смонтируйте **/backup**, чтобы проверить свою работу. Убедитесь, что монтирование прошло успешно.

```
[root@serverb ~]# mount /backup
[root@serverb ~]# mount | grep /backup
/dev/vdb1 on /backup type xfs
(rw,relatime,seclabel,attr2,inode64,noquota)
```

3. На том же новом диске создайте два раздела **GPT** размером **512 МБ** с именами **swap1** и **swap2**. Допускается размер от 460 МБ до 564 МБ. Установите правильный тип файловой системы на этих разделах для размещения областей подкачки.

3.1. Получите конечную позицию первого раздела, отобразив текущую таблицу разделов в **/dev/vdb**. На следующем шаге вы используете это значение в качестве начала раздела **swap1**.

```
[root@serverb ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	2000MB	1999MB	xfs		backup

- 3.2. Создайте первый раздел размером **512 МБ** с именем **swap1**. Установите его тип на **linux-swap**. Используйте конечную позицию первого раздела в качестве отправной точки. Конечная позиция **2000 МБ + 512 МБ = 2512 МБ**

```
[root@serverb ~]# parted /dev/vdb mkpart swap1 linux-swap 2000MB 2512M
Information: You may need to update /etc/fstab.
```

- 3.3. Создайте второй раздел размером **512 МБ** с именем **swap2**. Установите его тип на **linux-swap**. Используйте конечную позицию предыдущего раздела в качестве начальной точки: **2512М**. Конечная позиция **2512 МБ + 512 МБ = 3024 МБ**.

```
[root@serverb ~]# parted /dev/vdb mkpart swap2 linux-swap 2512M 3024M
Information: You may need to update /etc/fstab.
```

- 3.4. Отобразите таблицу разделов, чтобы проверить свою работу.

```
[root@serverb ~]# parted /dev/vdb print
Model: Virtio Block Device (virtblk)
Disk /dev/vdb: 5369MB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name     Flags
 1      1049kB  2000MB  1999MB  xfs        backup
 2      2000MB 2512MB 513MB          swap1  swap
```

- 3.5. Запустите команду установки **udevadm settle**. Команда ожидает, пока система зарегистрирует новые разделы и создаст файлы устройств.

```
[root@serverb ~]# udevadm settle
[root@serverb ~]#
```

4. Инициализируйте два раздела по **512 МБ** как области подкачки и настройте их активацию при загрузке. Установите пространство подкачки в **swap2**, чтобы оно было предпочтительным по сравнению с другим.

- 4.1. Используйте команду **mkswap** для инициализации разделов подкачки.

```
[root@serverb ~]# mkswap /dev/vdb2
Setting up swapspace version 1, size = 489 MiB (512749568 bytes)
no label, UUID=87976166-4697-47b7-86d1-73a02f0fc803
[root@serverb ~]# mkswap /dev/vdb3
Setting up swapspace version 1, size = 488 MiB (511700992 bytes)
no label, UUID=4d9b847b-98e0-4d4e-9ef7-dfaaf736b942
```

Обратите внимание на **UUID** двух пространств подкачки. Вы используете эту информацию на следующем шаге. Если вы больше не видите вывод **mkswap**, используйте команду **lsblk --fs** для получения **UUID**.

- 4.2.** Отредактируйте файл **/etc/fstab** и определите новые области подкачки. Чтобы сделать пространство подкачки в разделе **swap2** более предпочтительным, чем **swap1**, присвойте ему более высокий приоритет с помощью параметра **pri**.

```
[root@serverb ~]# vim /etc/fstab
...output omitted...
UUID=a3665c6b-4bfb-49b6-a528-74e268b058dd  /backup  xfs  defaults  0 0
UUID=87976166-4697-47b7-86d1-73a02f0fc803  swap      swap  pri=10    0 0
UUID=4d9b847b-98e0-4d4e-9ef7-dfaaf736b942  swap      swap  pri=20    0 0
```

- 4.3.** Заставьте **systemd** перечитать файл **/etc/fstab**.

```
[root@serverb ~]# systemctl daemon-reload
[root@serverb ~]#
```

- 4.4.** Используйте команду **swapon -a**, чтобы активировать новые области подкачки. Используйте команду **swapon --show** для подтверждения правильной активации областей подкачки.

```
[root@serverb ~]# swapon -a
[root@serverb ~]# swapon --show
NAME      TYPE      SIZE USED PRIO
/dev/vdb2  partition 489M   0B   10
/dev/vdb3  partition 488M   0B   20
```

- 5.** Для проверки работы перезагрузите **serverb**. Убедитесь, что система автоматически монтирует первый раздел в **/backup**. Также убедитесь, что система активирует два пространства подкачки. Когда закончите, выйдите из сервера **serverb**.

- 5.1.** Перезагрузите сервер **serverb**.

```
[root@serverb ~]# systemctl reboot
[root@serverb ~]#
Connection to serverb closed by remote host.
```

```
Connection to serverb closed.  
[student@workstation ~]$
```

5.2. Подождите несколько минут, пока **serverb** перезагрузится, а затем войдите в систему как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

5.3. Убедитесь, что система автоматически монтирует **/dev/vdb1** в **/backup**.

```
[student@serverb ~]$ mount | grep /backup  
/dev/vdb1 on /backup type xfs  
(rw,relatime,seclabel,attr2,inode64,noquota)
```

5.4. Используйте команду **swapon --show**, чтобы убедиться, что система активирует оба пространства подкачки.

```
[student@serverb ~]$ swapon --show  
NAME      TYPE      SIZE USED PRIO  
/dev/vdb2 partition 489M   0B   10  
/dev/vdb3 partition 488M   0B   20
```

5.5. Выходите из сервера **serverb**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

Оценка

На рабочей станции **workstation**, запустите сценарий **lab storage-review grade**, чтобы подтвердить успешное выполнение упражнения.

```
[student@workstation ~]$ lab storage-review grade
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab storage-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab storage-review finish
```

На этом лабораторная работа закончена.

РЕЗЮМЕ

В этой главе вы узнали:

- Вы используете команду **parted** для добавления, изменения и удаления разделов на дисках со схемой разделов **MBR** или **GPT**.
- Вы используете команду **mkfs.xfs** для создания файловых систем XFS на разделах диска.
- Вам нужно добавить команды монтирования файловой системы в файл **/etc/fstab**, чтобы сделать монтирования постоянными.
- Вы используете команду **mkswap** для инициализации областей подкачки.

ГЛАВА 7

УПРАВЛЕНИЕ ЛОГИЧЕСКИМИ ТОМАМИ

ЦЕЛЬ

Создавайте логические тома, содержащие файловые системы и области подкачки, и управляйте ими из командной строки.

ЗАДАЧИ

- Создавайте и управляйте логическими томами с устройств хранения, а также форматируйте их с помощью файловых систем или подготавливайте их с помощью пространств подкачки.
- Добавлять и удалять хранилище, назначенное группам томов, и неразрушающим образом увеличивать размер логического тома, отформатированного в файловой системе.

РАЗДЕЛЫ

- Создание логических томов (и упражнения с пошаговыми инструкциями)
- Расширение логических томов (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Управление логическими томами

СОЗДАНИЕ ЛОГИЧЕСКИХ ТОМОВ

ЦЕЛИ

После завершения этого раздела вы должны уметь:

- Описывать компоненты и концепции управления логическими томами (**Logical Volume Management**).
- Реализовать хранилище **LVM**.
- Отображение информации о компонентах **LVM**.

КОНЦЕПЦИИ ЛОГИЧЕСКОГО УПРАВЛЕНИЯ ТОМАМИ (LVM)

Логические тома и управление логическими томами упрощают управление дисковым пространством. Если файловой системе, на которой расположен логический том, требуется больше места, оно может быть выделено на ее логическом томе из свободного пространства в ее группе томов, а размер файловой системы может быть изменен. Если диск начинает выходить из строя, новый диск можно зарегистрировать как физический том в группе томов, а экстенты логического тома можно перенести на новый диск.

Определения LVM

Физические устройства (*Physical devices*)

Физические устройства (*Physical devices*) — это устройства хранения, используемые для хранения данных, хранящихся в логическом томе. Это блочные устройства, которые могут быть разделами диска, целыми дисками, массивами RAID или дисками SAN. Устройство должно быть инициализировано как физический том **LVM**, чтобы его можно было использовать с **LVM**. Все устройство будет использоваться как физический том.

Физические тома (*Physical volumes (PVs)*)

Вы должны инициализировать устройство как физический том, прежде чем использовать его в системе **LVM**. Инструменты **LVM** сегментируют физические тома на физические экстенты (*physical extents (PE)*), представляющие собой небольшие порции данных, выступающие в качестве наименьшего блока хранения на физическом томе.

Группы томов (*Volume groups (VGs)*)

Группы томов (*VG*) — это пулы хранения, состоящие из одного или нескольких физических томов. Это функциональный эквивалент целого диска в базовом хранилище. PV (*Physical volumes*) может быть назначен только одной **VG**. **VG** может состоять из неиспользуемого пространства и любого количества логических томов.

Логические тома (*LV*)

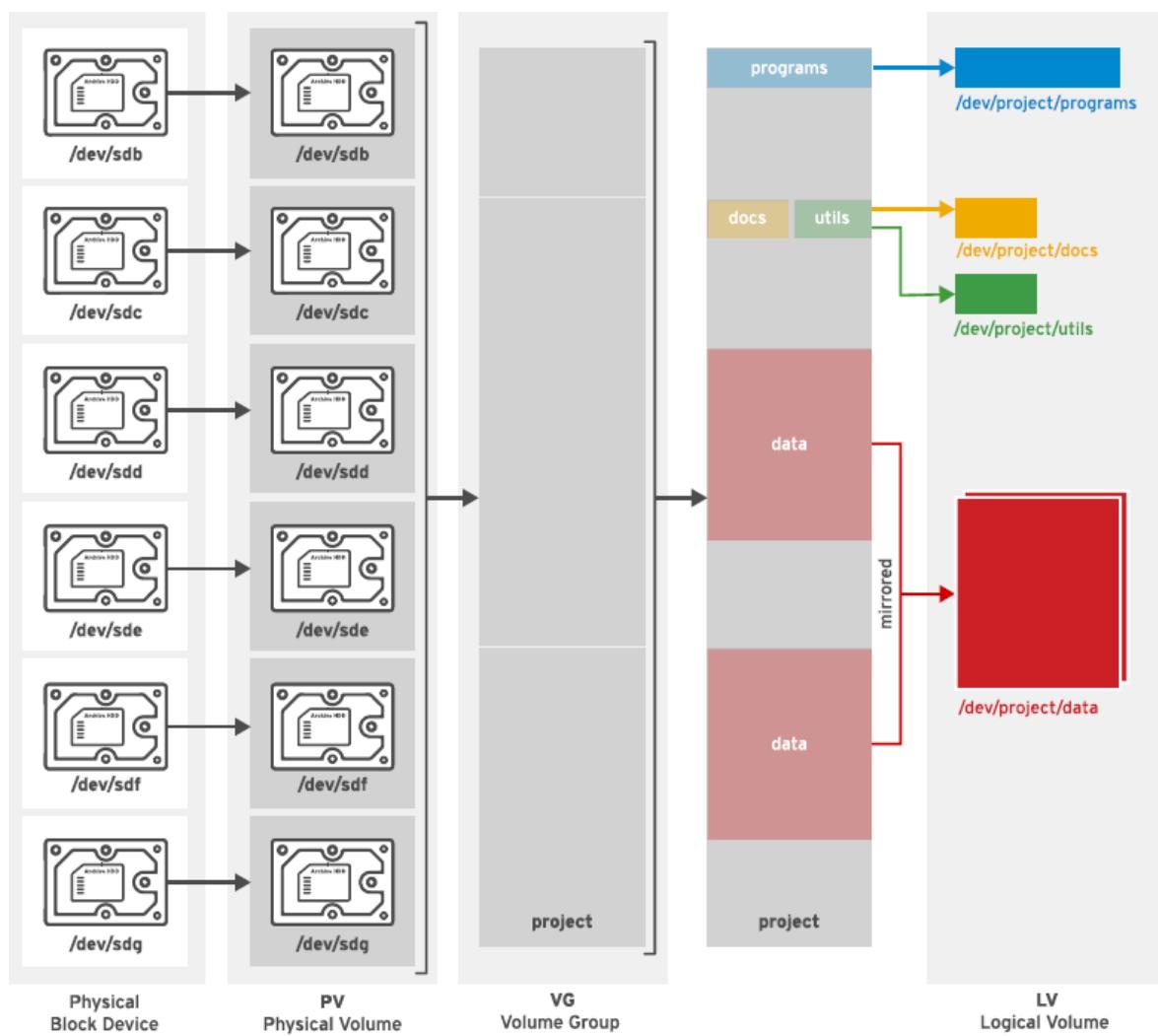
Логические тома создаются из свободных физических экстентов (physical extents) в группе томов и представляют собой «хранилище», используемое приложениями, пользователями и операционной системой. **LV** — это набор логических экстентов (**LE**), которые сопоставляются с физическими экстентами (physical extents **PE**), наименьшим фрагментом хранения **PV**. По умолчанию каждый **LE** сопоставляется с одним **PE**. Установка определенных параметров **LV** изменяет это сопоставление; например, зеркальное отображение заставляет каждый **LE** сопоставляться с двумя **PE**.

РЕАЛИЗАЦИЯ ХРАНИЛИЩА LVM

Создание хранилища **LVM** требует нескольких шагов. **Первый шаг** — определить, какие физические устройства использовать. После того, как набор подходящих устройств собран, они инициализируются как физические тома, чтобы они распознавались как принадлежащие **LVM**. **Затем** физические тома объединяются в группу томов. Это создает пул дискового пространства, из которого могут быть выделены логические тома.

Логические тома, созданные из доступного пространства в группе томов, можно отформатировать с помощью файловой системы, активировать как пространство подкачки и смонтировать или активировать на постоянной основе.

Рисунок 7.1: Компоненты управления логическими томами



LVM предоставляет исчерпывающий набор инструментов командной строки для реализации и управления хранилищем **LVM**. Эти инструменты командной строки можно использовать в сценариях, что делает их подходящими для автоматизации.



ВАЖНО

В следующих примерах **vdb** устройства и его разделы используются для иллюстрации команд **LVM**. На практике в этих примерах необходимо использовать правильные устройства для диска и разделов диска, используемых системой. Используйте команды **lsblk**, **blkid** или **cat /proc/partitions** для идентификации устройств в вашей системе.

Создание логического тома

Чтобы создать логический том, выполните следующие действия:

Подготовьте физическое устройство.

Используйте **parted**, **gdisk** или **fdisk**, чтобы создать новый раздел для использования с **LVM**. Всегда устанавливайте тип раздела **Linux LVM** на разделы **LVM**; используйте **0x8e** для разделов **MBR**. При необходимости используйте **partprobe** для регистрации нового раздела в ядре.

В качестве альтернативы можно использовать целый диск, массив **RAID** или диск **SAN**.

Физическое устройство необходимо подготовить только в том случае, если оно еще не подготовлено, а для создания или расширения группы томов требуется новый физический том.

```
[root@host ~]# parted -s /dev/vdb mkpart primary 1MiB 769MiB
[root@host ~]# parted -s /dev/vdb mkpart primary 770MiB 1026MiB
[root@host ~]# parted -s /dev/vdb set 1 lvm on
[root@host ~]# parted -s /dev/vdb set 2 lvm on
```

Создайте физический том (physical volume).

Используйте **pvcreate**, чтобы пометить раздел (или другое физическое устройство) как физический том. Команда **pvcreate** делит физический том на физические экстенты (**PE**) фиксированного размера, например, блоки по 4 МиБ. Вы можете пометить несколько устройств одновременно, используя имена устройств, разделенные пробелами, в качестве аргументов для **pvcreate**.

```
[root@host ~]# pvcreate /dev/vdb2 /dev/vdb1
```

Данная команда помечает устройства **/dev/vdb2** и **/dev/vdb1** как **PV**, готовые к размещению в группе томов.

PV нужно создавать только в том случае, если нет свободных **PV** для создания или расширения **VG**.

Создайте группу томов (volume group).

Используйте **vgcreate**, чтобы собрать один или несколько физических томов в группу томов. Группа томов (volume group) — это функциональный эквивалент жесткого диска; вы создадите логические тома из пула свободных физических экстентов в группе томов.

Командная строка **vgcreate** состоит из имени группы томов, за которым следует один или несколько физических томов, выделяемых этой группе томов.

```
[root@host ~]# vgcreate vg01 /dev/vdb2 /dev/vdb1
```

Команда создает виртуальную группу с именем **vg01**, которая представляет собой объединенный размер двух виртуальных томов **/dev/vdb2** и **/dev/vdb1** в единицах **PE**.

Виртуальную группу необходимо создать только в том случае, если она еще не существует. Дополнительные **VG** могут быть созданы по административным причинам для управления использованием **V** и **LV**. В противном случае существующие **VG** могут быть расширены для размещения новых **LV**, когда это необходимо.

Создайте логический том (logical volume).

Используйте **lvcreate** для создания нового логического тома из доступных физических экстентов в группе томов. Как минимум, команда **lvcreate** включает опцию **-n** для установки имени **LV**, либо опцию **-L** для установки размера **LV** в байтах, либо опцию **-l** для установки размера **LV** в экстентах, а также имя группы томов (volume group). размещения этого логического тома.

Также обратите внимание, что размер будет округлен до коэффициента физического размера экстента, если размер не может точно совпадать.

Вы можете указать размер с помощью параметра **-L**, который предполагает размеры в байтах, мегабайтах (двоичные мегабайты, 1048 576 байтов), гигабайтах (двоичные гигабайты) и т. п. В качестве альтернативы вы можете использовать параметр **-l**, который предполагает размеры, указанные как количество физических экстентов.

В следующем списке приведены некоторые примеры создания **LV**:

- **lvcreate -L 128M**: размер логического тома ровно 128 МБ.
- **lvcreate -l 128** : размер логического тома ровно до 128 экстентов. Общее количество байтов зависит от размера блока физического экстента на базовом физическом томе.



ВАЖНО

Различные инструменты отображают имя логического тома, используя либо традиционное имя, `/dev/vgname/lvname`, либо имя устройства отображения ядра, `/dev/mapper/vgname-lvname`.

Добавьте файловую систему.

Используйте `mkfs` для создания файловой системы **XFS** на новом логическом томе. В качестве альтернативы создайте файловую систему на основе предпочтаемой вами файловой системы, например, **ext4**.

```
[root@host ~]# mkfs -t xfs /dev/vg01/lv01
```

Чтобы сделать файловую систему доступной при перезагрузке, выполните следующие действия:

- Используйте `mkdir` для создания точки монтирования.

```
[root@host ~]# mkdir /mnt/data
```

- Добавьте запись в файл `/etc/fstab`:

```
/dev/vg01/lv01 /mnt/data xfs defaults 1 2
```



ПРИМЕЧАНИЕ

Подключение логического тома по имени эквивалентно подключению по **UUID**, потому что **LVM** находит свои физические тома на основе **UUID**, даже если вы изначально добавляете их в группу томов по имени.

- Выполните команду `mount /mnt/data`, чтобы смонтировать файловую систему, которую вы только что добавили в `/etc/fstab`.

```
[root@host ~]# mount /mnt/data
```

Удаление логического тома

Чтобы удалить все компоненты логического тома, выполните следующие действия:

Подготовьте файловую систему.

Переместите все данные, которые должны быть сохранены, в другую файловую систему. Используйте команду **umount**, чтобы размонтировать файловую систему, а затем удалить все записи **/etc/fstab**, связанные с этой файловой системой.

```
[root@host ~]# umount /mnt/data
```



ПРЕДУПРЕЖДЕНИЕ

Удаление логического тома уничтожает все данные, хранящиеся на логическом томе. Сделайте резервную копию или переместите данные перед удалением логического тома.

Удалите логический том.

Используйте **lvremove DEVICE_NAME**, чтобы удалить логический том, который больше не нужен.

```
[root@host ~]# lvremove /dev/vg01/lv01
```

Размонтируйте файловую систему **LV** перед выполнением этой команды. Команда запрашивает подтверждение перед удалением **LV**.

Физические экстенты **LV** освобождаются и становятся доступными для назначения существующим или новым **LV** в группе томов.

Удалите группу томов.

Используйте **vgremove VG_NAME**, чтобы удалить группу томов, которая больше не нужна.

```
[root@host ~]# vgremove vg01
```

Физические тома **VG** освобождаются и становятся доступными для назначения существующим или новым **VG** в системе.

Удалите физические тома.

Используйте **pvremove** для удаления физических томов, которые больше не нужны. Используйте список **PV**-устройств, разделенных пробелами, чтобы удалить более одного за раз. Эта команда удаляет метаданные **PV** из раздела (или диска). Теперь раздел свободен для перераспределения или переформатирования.

```
[root@host ~]# pvremove /dev/vdb2 /dev/vdb1
```

ПРОСМОТР ИНФОРМАЦИИ О СТАТУСЕ LVM

Физические тома

Используйте **pvdisplay** для отображения информации о физических томах. Чтобы вывести информацию обо всех физических томах, используйте команду без аргументов. Чтобы вывести информацию о конкретном физическом томе, передайте имя этого устройства в команду.

```
[root@host ~]# pvdisplay /dev/vdb1
```

```
--- Physical volume ---
```

PV Name	/dev/vdb1	1
VG Name	vg01	2
PV Size	768.00 MiB / not usable 4.00 MiB	3
Allocatable	yes	
PE Size	4.00 MiB	4
Total PE	191	
Free PE	16	5
Allocated PE	175	
PV UUID	JWzDpn-LG3e-n2oi-9EtD-VT2H-PMem-1ZXwP1	

1. **PV Name** сопоставляется с именем устройства.
2. **VG Name** показывает группу томов, в которой размещен физический том.
3. **PV Size** показывает физический размер PV, включая любое неиспользуемое пространство.
4. **PE Size** — это размер физического экстента, то есть наименьший размер, который может быть выделен логическому тому.
Это также умножающий коэффициент при расчете размера любого значения, выраженного в единицах **PE**, например **Free PE**; например: 26 **PE** x 4 МБ (размер **PE**) равно 104 МБ свободного места. Размер логического тома округляется до коэффициента единиц **PE**.
5. **LVM** автоматически устанавливает размер **PE**, хотя его можно указать.
5. **Free PE** показывает, сколько единиц **PE** доступно для выделения новым логическим томам.

Группы томов

Используйте команду **vgdisplay** для отображения информации о группах томов. Чтобы вывести информацию обо всех группах томов, используйте команду без аргументов. Чтобы просмотреть информацию об определенной группе томов, передайте это имя группы томов команде.

```
[root@host ~]# vgdisplay vg01
```

```
--- Volume group ---
```

VG Name	vg01	❶
System ID		
Format	lvm2	
Metadata Areas	2	
Metadata Sequence No	2	
VG Access	read/write	
VG Status	resizable	
MAX LV	0	
Cur LV	1	
Open LV	1	
Max PV	0	
Cur PV	2	
Act PV	2	
VG Size	1016.00 MiB	❷
PE Size	4.00 MiB	
Total PE	254	❸
Alloc PE / Size	175 / 700.00 MiB	
Free PE / Size	79 / 316.00 MiB	❹
VG UUID	3snNw3-CF71-CcYG-Llk1-p6EY-rHEv-xfUSez	

1. **VG Name** - это имя группы томов.

2. **VG Size** - это общий размер пула хранения, доступного для выделения логического тома.

3. **Total PE** - это общий размер, выраженный в единицах **PE**.

4. **Free PE/Size** показывает, сколько свободного места в **VG** для выделения новым **LV** или расширить существующие **LV**.

Логические тома

Используйте **lvdisplay** для отображения информации о логических томах. Если вы не укажете аргументы для команды, она отобразит информацию обо всех **LV**; если вы укажете имя устройства **LV** в качестве аргумента, команда отобразит информацию об этом конкретном устройстве.

```
[root@host ~]# lvdisplay /dev/vg01/lv01
```

```

--- Logical volume ---
LV Path          /dev/vg01/lv01      ①
LV Name          lv01
VG Name          vg01      ②
LV UUID          5IyRea-W8Zw-xLHk-3h2a-IuVN-YaeZ-i3IRrN
LV Write Access   read/write
LV Creation host, time host.lab.example.com, 2019-03-28 17:17:47 -0400
LV Status         available
# open            1
LV Size           700 MiB      ③
Current LE        175      ④
Segments          1
Allocation        inherit
Read ahead sectors auto
- current set to 256
Block device      252:0

```

1. **LV Path** показывает имя устройства логического тома.
2. Некоторые инструменты могут сообщать имя устройства как **/dev/mapper/vgname-lvname**; оба представляют один и тот же **LV**.
3. **VG Name** показывает группу томов, из которой выделен **LV**.
4. **LV Size** показывает общий размер **LV**. Используйте инструменты файловой системы, чтобы определить свободное пространство и используемое пространство для хранения данных.
5. **Current LE** показывает количество логических экстентов, используемых этим **LV**. **LE** обычно сопоставляется с физическим экстентом в **VG** и, следовательно, с физическим томом.



РЕКОМЕНДАЦИИ

Справочные страницы **man lvm(8)**, **pvcreate(8)**, **vgcreate(8)**, **lvcreate(8)**, **pvremove(8)**, **vgremove(8)**, **lvremove(8)**, **pvdisplay(8)**, **vgdisplay(8)**, **lvdisplay(8)**, **fdisk(8)**, **gdisk(8)**, **parted(8)**, **partprobe(8)**, и **mkfs(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

СОЗДАНИЕ ЛОГИЧЕСКИХ ТОМОВ

В этой лабораторной работе вы создадите физический том, группу томов, логический том и файловую систему **XFS**. Вы также будете постоянно монтировать файловую систему логического тома.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создавать физические тома, группы томов и логические тома с помощью инструментов LVM.
- Создавать новые файловые системы на логических томах и постоянно монтировать их.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab lvm-creating start**. Эта команда запускает сценарий, который определяет, доступен ли сервер **servera** в сети. Он также проверяет, доступно ли хранилище и установлены ли соответствующие программные пакеты.

```
[student@workstation ~]$ lab lvm-creating start
```

1. Используйте команду **ssh**, чтобы войти на сервер как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Пароль для пользователя **student** - **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

3. Создайте физические ресурсы.

- 3.1.** Используйте команду **parted** для создания двух разделов по **256 МБ** и установите для них тип **Linux LVM**.

```
[root@servera ~]# parted -s /dev/vdb mklabel gpt
[root@servera ~]# parted -s /dev/vdb mkpart primary 1MiB 257MiB
[root@servera ~]# parted -s /dev/vdb set 1 lvm on
[root@servera ~]# parted -s /dev/vdb mkpart primary 258MiB 514MiB
[root@servera ~]# parted -s /dev/vdb set 2 lvm on
```

- 3.2.** Используйте **udevadm settle**, чтобы система зарегистрировала новые разделы.

```
[root@servera ~]# udevadm settle
```

- 4.** Используйте команду **pvcreate**, чтобы добавить два новых раздела в качестве физических томов.

```
[root@servera ~]# pvcreate /dev/vdb1 /dev/vdb2
Physical volume "/dev/vdb1" successfully created.
Physical volume "/dev/vdb2" successfully created.
```

- 5.** Используйте команду **vgcreate** для создания новой группы томов с именем **servera_01_vg**, построенной из двух виртуальных томов.

```
[root@servera ~]# vgcreate servera_01_vg /dev/vdb1 /dev/vdb2
Volume group "servera_01_vg" successfully created
```

- 6.** Используйте команду **lvcreate** для создания **LV 400 МБ** с именем **servera_01_lv** из **servera_01_vg VG**.

```
[root@servera ~]# lvcreate -n servera_01_lv -L 400M servera_01_vg
Logical volume "servera_01_lv" created.
```

Команда создает устройство с именем **/dev/servera_01_vg/servera_01_lv**, но без файловой системы на нем.

- 7.** Добавьте постоянную файловую систему.

- 7.1.** Добавьте файловую систему **XFS** для **servera_01_lv LV** с помощью команды **mkfs**.

```
[root@servera ~]# mkfs -t xfs /dev/servera_01_vg/servera_01_lv
...output omitted...
```

7.2. Создайте точку монтирования в **/data**.

```
[root@servera ~]# mkdir /data
```

7.3. Добавьте следующую строку в конец **/etc/fstab** на сервере **servera**:

```
/dev/servera_01_vg/servera_01_lv      /data    xfs    defaults      1  2
```

7.4. Используйте команду **systemctl daemon-reload** для обновления **systemd** с новой конфигурацией **/etc/fstab**.

```
[root@servera ~]# systemctl daemon-reload
```

7.5. Проверьте запись в файле **/etc/fstab** и смонтируйте новое устройство **servera_01_lv LV** с помощью команды **mount**.

```
[root@servera ~]# mount /data
```

8. Тестируйте и анализируйте свою работу.

8.1. В качестве последнего теста скопируйте несколько файлов в каталог **/data** и проверьте, сколько из них было скопировано.

```
[root@servera ~]# cp -a /etc/*.conf /data  
[root@servera ~]# ls /data | wc -l  
34
```

В следующем управляемом упражнении вы убедитесь, что у вас все еще есть то же количество файлов.

8.2. Команда **parted /dev/vdb** выводит список разделов, существующих в **/dev/vdb**.

```
[root@servera ~]# parted /dev/vdb print  
Model: Virtio Block Device (virtblk)  
Disk /dev/vdb: 5369MB  
Sector size (logical/physical): 512B/512B  
Partition Table: gpt  
Disk Flags:
```

Number	Start	End	Size	File system	Name	Flags
1	1049kB	269MB	268MB		primary	lvm
2	271MB	539MB	268MB		primary	lvm

Обратите внимание на столбец **Number**, который содержит значения **1** и **2**. Они соответствуют **/dev/vdb1** и **/dev/vdb2** соответственно. Также обратите внимание на столбец **Flags**, в котором указан тип раздела.

8.3. Команда **pvdisplay** отображает информацию о каждом из физических томов. При необходимости укажите имя устройства, чтобы ограничить детали конкретным физическим **PV**.

```
[root@servera ~]# pvdisplay /dev/vdb2
--- Physical volume ---
PV Name          /dev/vdb2
VG Name          servera_01_vg
PV Size          256.00 MiB / not usable 4.00 MiB
Allocatable      yes
PE Size          4.00 MiB
Total PE         63
Free PE          26
Allocated PE     37
PV UUID          2z0Cf3-99YI-w9ny-a1EW-wWhL-S8RJ-M2rfZk
```

Вывод команды показывает, что **PV** выделен для **VG servera_01_vg**, имеет размер **256 МБ** (хотя 4 МБ нельзя использовать), а размер физического экстента (размер **PE**) составляет 4 МБ (наименьший выделяемый размер **LV**).

Имеется **63 PE**, из которых **26** свободны для распределения по **LV** в будущем, а **37** в настоящее время выделены для **LV**. Они преобразуются в значения **MiB** следующим образом:

- Всего 252 МБ (63 PE x 4 МБ); помните, 4 МБ непригодны для использования.
- Свободно 104 МБ (26 PE x 4 МБ)
- Выделено 148 МБ (37 PE x 4 МБ)

8.4. Команда **vgdisplay vgname** показывает информацию о группе томов с именем **vgname**. Проверьте следующие значения:

- **Размер VG (Size)** составляет 504,00 МБ.
- **Всего (Total) PE** составляет 126.
- **Выделено (Alloc) PE / Size:** 100 / 400,00 МБ.
- **Свободно (Free) PE / Size** 26 / 104,00 МБ.

8.5. Команда **lvdisplay /dev/vgname/lvname** отображает информацию о логическом томе с именем **lvname**.

```
[root@servera ~]# lvdisplay /dev/servera_01_vg/servera_01_lv
```

Просмотрите путь **LV**, имя **LV**, имя **VG**, статус **LV**, размер **LV** и текущий **LE** (логические экстенты, которые сопоставляются с физическими экстентами).

8.6. Команда **mount** показывает все смонтированные устройства и любые параметры монтирования. Он должен включать **/dev/servera_01_vg/servera_01_lv**.



ПРИМЕЧАНИЕ

Вместо этого многие инструменты сообщают имя устройства сопоставления устройств, **/dev/mapper/servera_01_vg-servera_01_lv**; это тот же логический том.

```
[root@servera ~]# mount
```

Вы должны увидеть (вероятно, в последней строке) **/dev/mapper/servera_01_vg-servera_01_lv**, смонтированный на **/data**, и соответствующую информацию о монтировании.

8.7. Команда **df -h** отображает свободное место на диске в удобочитаемом виде. При необходимости укажите точку монтирования, чтобы ограничить детали этой файловой системой.

```
[root@servera ~]# df -h /data
```

Filesystem	Size	Used	Avail	Use%	
Mounted on					
/dev/mapper/servera_01_vg-servera_01_lv	395M	24M	372M	6%	/data

Ожидается эти значения с учетом метаданных файловой системы.

9. Выйдите из сервера **servera**.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab lvm-creating finish**, чтобы завершить данное упражнение. Этот скрипт удаляет хранилище, настроенное на сервере **servera** во время работы.

```
[student@workstation ~]$ lab lvm-creating finish
```

На этом управляемое упражнение завершено.

РАСШИРЕНИЕ ЛОГИЧЕСКИХ ТОМОВ

ЦЕЛИ

После заполнения этого раздела вы должны быть способны:

- Расширить группу томов (**VG**) с помощью команд **pvcreate** и **vgextend** и используйте команду **vgdisplay** для просмотра результатов.
- Уменьшить виртуальную группу с помощью команд **pvmove** и **vgreduce**.
- Расширить логический том (**LV**) с помощью команды **lvextend**.
- Выполнить изменение размера файловых систем **XFS** с помощью команды **xfs_growfs**.
- Выполнить изменение размера файловых систем **ext4** с помощью команды **resize2fs**.

РАСШИРЕНИЕ И СОКРАЩЕНИЕ ГРУППЫ ТОМОВ (VOLUME GROUP)

Вы можете добавить больше дискового пространства в группу томов (**VG**), добавив дополнительные физические тома. Это называется расширением группы томов (**VG**). Затем вы можете назначить новые физические экстенты из дополнительных физических томов логическим томам.

Вы можете удалить неиспользуемые физические тома из группы томов. Это называется уменьшением группы томов. Сначала используйте команду **pvmove** для перемещения данных из экстентов одного физического тома в экстенты других физических томов в группе томов. Таким образом, новый диск может быть добавлен в существующую группу томов, данные могут быть перемещены со старого или более медленного диска на новый диск, а старый диск удален из группы томов. Вы можете выполнять эти действия, пока используются логические тома в группе томов.



ВАЖНО

В следующих примерах используется устройство **vdb** и его разделы для иллюстрации команд LVM. На практике используйте соответствующие устройства для диска и разделов диска в вашей собственной системе.

Расширение группы томов

Чтобы расширить группу томов, выполните следующие действия:

Подготовьте физическое устройство и создайте физический том.

Как и при создании новой группы томов, вы должны создать и подготовить новый раздел для использования в качестве физического тома, если ни один из них еще не подготовлен.

```
[root@host ~]# parted -s /dev/vdb mkpart primary 1027MiB 1539MiB
```

```
[root@host ~]# parted -s /dev/vdb set 3 lvm on  
[root@host ~]# pvcreate /dev/vdb3
```

PV нужно создавать только в том случае, если нет свободных **PV** для расширения **VG**.

Расширьте группу томов.

Используйте команду **vgextend**, чтобы добавить новый физический том в группу томов. Используйте имя **VG** и имя устройства **PV** в качестве аргументов для команды **vgextend**.

```
[root@host ~]# vgextend vg01 /dev/vdb3
```

Данное действие расширяет **VG vg01** на размер диска **/dev/vdb3 PV**.

Убедитесь, что новое пространство доступно.

Используйте команду **vgdisplay**, чтобы убедиться, что дополнительные физические экстенты доступны. Проверьте **Free PE/Size** в выходных данных. Он не должен быть равен нулю.

```
[root@host ~]# vgdisplay vg01  
--- Volume group ---  
VG Name          vg01  
...output omitted...  
Free  PE / Size    178 / 712.00 MiB  
...output omitted...
```

Уменьшение группы томов (VG)

Чтобы уменьшить группу томов, выполните следующие действия:

Переместите физические экстенты.

Используйте команду **pvmove PV_DEVICE_NAME** для перемещения любых физических экстентов с физического тома, который вы хотите удалить, на другие физические тома в группе томов. Другие физические тома должны иметь достаточно количество свободных экстентов для выполнения этого перемещения. Это возможно только в том случае, если в **VG** достаточно свободных экстентов и, если все они получены из других **PV**.

```
[root@host ~]# pvmove /dev/vdb3
```

Эта команда перемещает **PE** из **/dev/vdb3** в другие **PV** со свободными **PE** в той же **VG**.



ПРЕДУПРЕЖДЕНИЕ

Перед использованием команды **pvmove** сделайте резервную копию данных, хранящихся на всех логических томах в группе томов. Непредвиденная потеря питания во время операции может оставить группу томов в несогласованном состоянии. Это может привести к потере данных на логических томах в группе томов.

Уменьшение группы томов (VG).

Используйте команду **vgreduce VG_NAME PV_DEVICE_NAME**, чтобы удалить физический том из группы томов.

```
[root@host ~]# vgreduce vg01 /dev/vdb3
```

Команда удаляет **/dev/vdb3 PV** из **vg01 VG**, и теперь его можно добавить в другую **VG**. Кроме того, команда **pvremove** можно использовать для окончательного прекращения использования устройства в качестве **PV**.

РАСШИРЕНИЕ ЛОГИЧЕСКОГО ТОМА И ФАЙЛОВОЙ СИСТЕМЫ XFS

Одним из преимуществ логических томов является возможность увеличивать их размер без простоев. Свободные физические экстенты в группе томов могут быть добавлены к логическому тому для расширения его емкости, которую затем можно использовать для расширения содержащейся в нем файловой системы.

Расширение логического тома

Чтобы расширить логический том, выполните следующие действия:

Убедитесь, что в группе томов есть свободное место.

Используйте команду **vgdisplay**, чтобы убедиться, что доступно достаточно физических экстентов.

```
[root@host ~]# vgdisplay vg01
```

```
--- Volume group ---
VG Name          vg01
...output omitted...
Free PE / Size   178 / 712.00 MiB
...output omitted...
```

Проверьте **Free PE/Size** в выходных данных. Убедитесь, что в группе томов достаточно свободного места для расширения **LV**. Если недостаточно места, соответствующим образом расширьте группу томов. См. раздел «Расширение и уменьшение группы томов».

Расширьте логический том.

Используйте команду **lvextend LV_DEVICE_NAME**, чтобы расширить логический том до нового размера.

```
[root@host ~]# lvextend -L +300M /dev/vg01/lv01
```

Данное действие увеличивает размер логического тома **lv01** на **300 МБ**. Обратите внимание на знак плюс (+) перед размером, что означает добавление этого значения к существующему размеру; в противном случае значение определяет окончательный размер **LV**.

Как и в случае с командой **lvcreate**, существуют различные методы для указания размера: параметр **-l** принимает в качестве аргумента количество физических экстентов. Параметр **-L** предполагает размеры в байтах, мегабайтах, гигабайтах и т. д.

В следующем списке приведены некоторые примеры расширения **LV**.

Примеры расширения **LV**

КОМАНДА	РЕЗУЛЬТАТ
lvextend -l 128	Измените размер логического тома ровно до 128 экстентов.
lvextend -l +128	Добавьте 128 экстентов к текущему размеру логического тома.
Удлинитель -L 128М	Измените размер логического тома ровно на 128 МБ.
lvextend -L +128М	Добавьте 128 МБ к текущему размеру логического тома.
lvextend -l +50%FREE	Добавьте 50 процентов текущего свободного места в VG к LV.

Расширьте файловую систему.

Используйте команду **xfs_growfs mountpoint**, чтобы расширить файловую систему, чтобы занять расширенный **LV**. Целевая файловая система должна быть смонтирована при

использовании команды **xfs_growfs**. Вы можете продолжать использовать файловую систему во время изменения ее размера.

```
[root@host ~]# xfs_growfs /mnt/data
```



ПРИМЕЧАНИЕ

Распространенная ошибка — выполнить команду **lvextend**, но забыть запустить **xfs_growfs**. Альтернативой последовательному выполнению двух шагов является включение параметра **-r** в команду **lvextend**. Это изменяет размер файловой системы после расширения **LV**. Используете справочную страницу помощью **fsadm(8)**. Это работает с несколькими различными файловыми системами.

- Проверьте новый размер смонтированной файловой системы:
df -h /mountpoint.

РАСШИРЕНИЕ ЛОГИЧЕСКОГО ТОМА И ФАЙЛОВОЙ СИСТЕМЫ EXT4

Действия по расширению логического тома на основе **ext4** практически такие же, как и для **LV** на основе **XFS**, за исключением шага, который изменяет размер файловой системы. Просмотрите раздел «Расширение логического тома и файловой системы XFS».

Убедитесь, что в группе томов есть свободное место.

Используйте команду **vgdisplay VGNAME**, чтобы убедиться, что в группе томов имеется достаточное количество доступных физических экстентов.

Расширьте логический том.

Используйте команду **lvextend -l +extents /dev/vgname/lvname**, чтобы расширить логический том **/dev/vgname/lvname** на значение экстентов.

Расширьте файловую систему.

Используйте команду **resize2fs /dev/vgname/lvname**, для расширения файловой системы, чтобы занять новый расширенный **LV**. Файловая система может быть смонтирована и использоваться во время выполнения команды расширения. Вы можете включить параметр **-p**, чтобы отслеживать ход операции изменения размера.

```
[root@host ~]# resize2fs /dev/vg01/lv01
```



ПРИМЕЧАНИЕ

Основное различие между `xfs_growfs` и `resize2fs` заключается в аргументе, который передается для идентификации файловой системы. `xfs_growfs` берет точку монтирования, а `resize2fs` берет имя логического тома.

РАСШИРЕНИЕ ЛОГИЧЕСКОГО ТОМА И SWOP

Логические тома, отформатированные как пространство подкачки (**SWOP**), также могут быть расширены, однако этот процесс отличается от процесса расширения файловой системы, такой как `ext4` или `XFS`. Логические тома, отформатированные в файловой системе, можно динамически расширять без простоев. Логические тома, отформатированные для пространства подкачки, должны быть переведены в автономный режим для их расширения.

Убедитесь, что в группе томов есть свободное место.

Используйте команду `vgdisplay vgname`, чтобы убедиться, что доступно достаточно свободных физических экстентов.

Деактивируйте пространство подкачки.

Используйте команду `swapoff -v /dev/vgname/lvname`, чтобы деактивировать пространство подкачки на логическом томе.



ПРЕДУПРЕЖДЕНИЕ

В вашей системе должно быть достаточно свободной памяти или пространства подкачки, чтобы принять все, что необходимо для подкачки, когда пространство подкачки на логическом томе деактивировано.

Расширьте логический том.

Команда `lvextend -l +extents /dev/vgname/lvname` расширяет логический том `/dev/vgname/lvname` на значение экстентов.

Отформатируйте логический том, как пространство подкачки (swop).

Выполните команду `mkswap /dev/vgname/lvname` формирует весь логический том, как пространство подкачки.

Активируйте пространство подкачки.

Используйте команду `swapon -va /dev/vgname/lvname`, чтобы активировать пространство подкачки на логическом томе.



РЕКОМЕНДАЦИИ

Справочные страницы **man lvm(8)**, **pvcreate(8)**, **pvmove(8)**, **vgdisplay(8)**, **vgextend(8)**, **vgreduce(8)**, **vgdisplay(8)**, **vgextend(8)**, **vgreduce(8)**, **lvextend(8)**, **fdisk(8)**, **gdisk(8)**, **parted(8)**, **partprobe(8)**, **xfs_growfs(8)**, и **resize2fs(8)** **swapoff(8)**, **swapon(8)** **mkswap(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

РАСШИРЕНИЕ ЛОГИЧЕСКИХ ТОМОВ

В этой лабораторной работе вы расширите логический том, добавленный в предыдущем практическом упражнении.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

Расширить группу томов, включив в нее дополнительный физический том.

Изменять размер логического тома, пока файловая система еще смонтирована и используется.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation**, выполните скрипт **lab lvm-extending start**. Данная команда запускает сценарий, который определяет, доступен ли хост-сервер **servera** в сети, и обеспечивает доступность хранилища из предыдущего управляемого упражнения.

```
[student@workstation ~]$ lab lvm-extending start
```

1. Используйте команду **ssh**, чтобы войти на сервер **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Используйте команду **sudo -i**, чтобы переключиться на **root** в командной строке.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

3. Используйте команду **vgdisplay**, чтобы определить, достаточно ли свободного места в группе томов для расширения **LV** до общего размера **700 МБ**.

```
[root@servera ~]# vgdisplay servera_01_vg  
--- Volume group ---
```

```
VG Name           servera_01_vg
System ID
Format           lvm2
...output omitted...
VG Size          504.00 MiB
PE Size          4.00 MiB
Total PE         126
Alloc PE / Size 100 / 400.00 MiB
Free PE / Size   26 / 104.00 MiB
VG UUID          0BBATU-2nBS-4SW1-khmF-yJzi-z7bD-DpCrAV
```

Доступно только 104 МБ (26 PE x 4 МБ экстента), и вам нужно как минимум 300 МБ, чтобы иметь 700 МБ всего. Вам нужно расширить **VG**.

Для последующего сравнения используйте **df** для записи текущего свободного места на диске:

```
[root@servera ~]# df -h /data
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/servera_01_vg-servera_01_lv 395M  24M  372M   6% /data
```

4. Создайте физический ресурс.

4.1. Используйте команду **parted**, чтобы создать дополнительный раздел размером 512 МБ и установить для него тип **Linux LVM**.

```
[root@servera ~]# parted -s /dev/vdb mkpart primary 515MiB 1027MiB
[root@servera ~]# parted -s /dev/vdb set 3 lvm on
```

4.2. Используйте **udevadm settle**, чтобы система зарегистрировала новый раздел.

```
[root@servera ~]# udevadm settle
```

5. Используйте команду **pvcreate**, чтобы добавить новый раздел в качестве **PV**.

```
[root@servera ~]# pvcreate /dev/vdb3
Physical volume "/dev/vdb3" successfully created.
```

6. Расширьте группу томов.

6.1. Используйте команду **vgextend** для расширения виртуальной группы с именем **servera_01_vg**, используя новый **/dev/vdb3 PV**.

```
[root@servera ~]# vgextend servera_01_vg /dev/vdb3
Volume group "servera_01_vg" successfully extended
```

- 6.2.** Используйте команду **vgdisplay** для повторной проверки свободного места на **servera_01_vg VG**. Теперь должно быть много свободного места.

```
[root@servera ~]# vgdisplay servera_01_vg
--- Volume group ---
VG Name           servera_01_vg
System ID
Format           lvm2
...output omitted...
VG Size          1012.00 MiB
PE Size          4.00 MiB
Total PE         253
Alloc PE / Size 100 / 400.00 MiB
Free  PE / Size 153 / 612.00 MiB
VG UUID          0BBAtU-2nBS-4SW1-khmF-yJzi-z7bD-DpCrAV
```

Теперь доступно **612 МБ** свободного места (153 PE x 4 МБ экстента).

- 7.** Используйте команду **lvextend**, чтобы расширить существующий **LV** до **700 МБ**.

```
[root@servera ~]# lvextend -L 700M /dev/servera_01_vg/servera_01_lv
Size of logical volume servera_01_vg/servera_01_lv changed from 400.00 MiB
(100 extents) to 700.00 MiB (175 extents).
Logical volume servera_01_vg/servera_01_lv successfully resized.
```



ПРИМЕЧАНИЕ

В примере указан точный размер для создания окончательного тома **LV**, но вы могли бы указать желаемый объем дополнительного пространства:

- L +300M**, чтобы добавить новое пространство, используя размер в МиБ.
- l 175**, чтобы указать общее количество экстентов (175 PE x 4 МБ).
- l +75** для добавления необходимых дополнительных экстентов.

- 8.** Используйте команду **xfs_growfs**, чтобы расширить файловую систему **XFS** до оставшейся части свободного места на **LV**.

```
[root@servera ~]# xfs_growfs /data
meta-data=/dev/mapper/servera_01_vg-servera_01_lv isize=512    agcount=4,
agsize=25600 blks
...output omitted...
```

9. Используйте команды **df** и **ls | wc**, чтобы просмотреть новый размер файловой системы и убедиться, что ранее существовавшие файлы все еще присутствуют.

```
[root@servera ~]# df -h /data
Filesystem           Size  Used Avail Use% Mounted on
/dev/mapper/servera_01_vg-servera_01_lv  695M   26M  670M   4% /data
[root@servera ~]# ls /data | wc -l
34
```

Файлы все еще существуют, и файловая система приближается к указанному размеру.

10. Выйдите из сервера **servera**.

```
[root@servera ~]# exit
logout
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите команду **lab lvm-extending finish**, чтобы завершить это упражнение. Данный скрипт удаляет хранилище, настроенное на сервере **servera** во время упражнения.

```
[student@workstation ~]$ lab lvm-extending finish
```

На этом упражнения с пошаговыми инструкциями заканчивается.

ЛАБОРАТОРНАЯ РАБОТА

УПРАВЛЕНИЕ ЛОГИЧЕСКИМИ ТОМАМИ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы измените размер существующего логического тома, при необходимости добавите ресурсы **LVM**, а затем добавите новый логический том с постоянно смонтированной файловой системой **XFS**.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Изменить размер логического тома **serverb_01_lv** на **768 МБ**.
- Создать новый логический том размером **128 МБ** с именем **serverb_02_lv** с файловой системой **XFS**, постоянно смонтированный в **/storage/data2**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab lvm-review start**. Эта команда запускает сценарий, который определяет, доступна ли машина-сервер **serverb** в сети. Команда также подготавливает хранилище на **serverb** к упражнению.

```
[student@workstation ~]$ lab lvm-review start
```

На **serverb** логическому тому с именем **serverb_01_lv**, смонтированному в **/storage/data1**, не хватает места на диске, и вас попросили увеличить его размер до **768 МБ**. Вы должны убедиться, что **serverb_01_lv** постоянно смонтирован к **/storage/data1**.

Вас также попросили создать новый логический том размером **128 МБ** с именем **serverb_02_lv**, смонтированный в **/storage/data2**. Вам было предложено отформатировать новый логический том в файловой системе **XFS**.

Группа томов **serverb_01_vg** содержит логические тома. К сожалению, недостаточно места для расширения существующего логического тома и добавления нового. Ранее в **/dev/vdb** был создан раздел размером **512 МБ**. Вам дали инструкции использовать еще **512 МБ** в **/dev/vdb**. Вы должны создать новый раздел.

1. Создайте раздел размером **512 МБ** в **/dev/vdb**, инициализируйте его как физический том и расширите с его помощью группу томов **serverb_01_vg**.
2. Расширьте логический том **serverb_01_lv** до **768 МБ**, включая файловую систему.

3. В существующей группе томов создайте новый логический том с именем **serverb_02_lv** и размером **128 МБ**. Добавьте файловую систему **XFS** и смонтируйте ее для постоянного подключения к **/storage/data2**.
4. Когда вы закончите, перезагрузите **serverb**, затем запустите команду **lab lvmreview grade** на своей рабочей станции **workstation**, чтобы проверить свою работу.

Подождите, пока **serverb** полностью не заработает, а затем приступайте к выполнению оценки.

Оценка

На рабочей станции **workstation**, запустите скрипт **lab lvm-review grade**, чтобы подтвердить успешное выполнение данного упражнения.

```
[student@workstation ~]$ lab lvm-review grade
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab lvm-review finish**, для завершения лабораторной работы.

```
[student@workstation ~]$ lab lvm-review finish
```

На этом лабораторная работа завершена.

РЕШЕНИЕ

УПРАВЛЕНИЕ ЛОГИЧЕСКИМИ ТОМАМИ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы измените размер существующего логического тома, при необходимости добавите ресурсы **LVM**, а затем добавите новый логический том с постоянно смонтированной файловой системой **XFS**.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Изменить размер логического тома **serverb_01_lv** на **768 МБ**.
- Создать новый логический том размером **128 МБ** с именем **serverb_02_lv** с файловой системой **XFS**, постоянно смонтированный в **/storage/data2**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab lvm-review start**. Эта команда запускает сценарий, который определяет, доступна ли машина-сервер **serverb** в сети. Команда также подготавливает хранилище на **serverb** к упражнению.

```
[student@workstation ~]$ lab lvm-review start
```

На **serverb** логическому тому с именем **serverb_01_lv**, смонтированному в **/storage/data1**, не хватает места на диске, и вас попросили увеличить его размер до **768 МБ**. Вы должны убедиться, что **serverb_01_lv** постоянно смонтирован к **/storage/data1**.

Вас также попросили создать новый логический том размером **128 МБ** с именем **serverb_02_lv**, смонтированный в **/storage/data2**. Вам было предложено отформатировать новый логический том в файловой системе **XFS**.

Группа томов **serverb_01_vg** содержит логические тома. К сожалению, недостаточно места для расширения существующего логического тома и добавления нового. Ранее в **/dev/vdb** был создан раздел размером **512 МБ**. Вам дали инструкции использовать еще **512 МБ** в **/dev/vdb**. Вы должны создать новый раздел.

1. Создайте раздел размером **512 МБ** в **/dev/vdb**, инициализируйте его как физический том и расширите с его помощью группу томов **serverb_01_vg**.

1.1. Войдите на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

- 1.2.** Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Пароль для пользователя **student** - **student**.

```
[student@serverb ~]$ sudo -i  
[sudo] password for student: student
```

- 1.3.** Используйте команду **parted** для создания раздела размером **512 МБ** и установите для него тип **Linux LVM**.

```
[root@serverb ~]# parted -s /dev/vdb mkpart primary 514MiB 1026MiB  
[root@serverb ~]# parted -s /dev/vdb set 2 lvm on
```

- 1.4.** Используйте **udevadm settle**, чтобы система зарегистрировала новый раздел.

```
[root@servera ~]# udevadm settle
```

- 1.5.** Используйте **pvcreate** для инициализации раздела как **PV**.

```
[root@serverb ~]# pvcreate /dev/vdb2  
Physical volume "/dev/vdb2" successfully created.
```

- 1.6.** Используйте **vgextend** для расширения виртуальной группы с именем **serverb_01_vg**, используя новый **PV /dev/vdb2**.

```
[root@serverb ~]# vgextend serverb_01_vg /dev/vdb2  
Volume group "serverb_01_vg" successfully extended
```

- 2.** Расширьте логический том **serverb_01_lv** до **768 МБ**, включая файловую систему.

- 2.1.** Используйте **lvextend**, чтобы увеличить **serverb_01_lv LV** до **768 МБ**.

```
[root@serverb ~]# lvextend -L 768M /dev/serverb_01_vg/serverb_01_lv  
Size of logical volume serverb_01_vg/serverb_01_lv changed from  
256.00 MiB (64 extents) to 768.00 MiB (192 extents).  
Logical volume serverb_01_vg/serverb_01_lv successfully resized.
```



ПРИМЕЧАНИЕ

В качестве альтернативы вы могли бы использовать опцию **-L +512M** для изменения размера **LV**.

- 2.2.** Используйте **xfs_growfs**, чтобы расширить файловую систему **XFS** до оставшейся части свободного места на **LV**.

```
[root@serverb ~]# xfs_growfs /storage/data1
meta-data=/dev/mapper/serverb_01_vg-serverb_01_lv isize=512
    agcount=4, agsize=16384 blks
...output omitted...
```



ПРИМЕЧАНИЕ

В этом примере показан шаг **xfs_growfs** для расширения файловой системы. В качестве альтернативы можно было бы добавить параметр **-r** к команде **lvextend**.

- 3.** В существующей группе томов создайте новый логический том с именем **serverb_02_lv** и размером **128 МБ**. Добавьте файловую систему **XFS** и смонтируйте ее для постоянного подключения к **/storage/data2**.

- 3.1.** Используйте **lvcreate**, чтобы создать **128 МБ LV** с именем **serverb_02_lv** из **VG serverb_01_vg**.

```
[root@serverb ~]# lvcreate -n serverb_02_lv -L 128M serverb_01_vg
Logical volume "serverb_02_lv" created
```

- 3.2.** Используйте **mkfs** для размещения файловой системы **xfs** на **LV serverb_02_lv**. Используйте имя устройства **LV**.

```
[root@serverb ~]# mkfs -t xfs /dev/serverb_01_vg/serverb_02_lv
meta-data=/dev/serverb_01_vg/serverb_02_lv isize=512          agcount=4,
    agsize=8192 blks
...output omitted...
```

- 3.3.** Используйте **mkdir** для создания каталога точки монтирования в **/storage/data2**.

```
[root@serverb ~]# mkdir /storage/data2
```

3.4. Добавьте следующую строку в конец **/etc/fstab** на **serverb**:

```
/dev/serverb_01_vg/serverb_02_lv      /storage/data2  xfs  defaults  1 2
```

3.5. Используйте **systemctl daemon-reload** для обновления **systemd** с новой конфигурацией файла **/etc/fstab**.

```
[root@servera ~]# systemctl daemon-reload
```

3.6. Используйте команду **mount**, чтобы проверить запись в **/etc/fstab** и смонтировать новое устройство **LV serverb_02_lv**.

```
[root@serverb ~]# mount /storage/data2
```

4. Когда вы закончите, перезагрузите **serverb**, затем запустите команду **lab lvmreview grade** на своей рабочей станции **workstation**, чтобы проверить свою работу.

```
[root@serverb ~]# systemctl reboot
```

Подождите, пока **serverb** полностью не заработает, а затем приступайте к выполнению оценки.

Оценка

На рабочей станции **workstation**, запустите скрипт **lab lvm-review grade**, чтобы подтвердить успешное выполнение данного упражнения.

```
[student@workstation ~]$ lab lvm-review grade
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab lvm-review finish**, для завершения лабораторной работы.

```
[student@workstation ~]$ lab lvm-review finish
```

На этом лабораторная работа завершена.

РЕЗЮМЕ

В этой главе вы узнали:

- **LVM** позволяет создавать гибкие хранилища, выделяя место на нескольких устройствах хранения.
- Физические тома, группы томов и логические тома управляются различными инструментами, такими как **pvcreate**, **vgreduce** и **lvextend**.
- Логические тома можно форматировать с использованием файловой системы или пространства подкачки, а также их можно постоянно монтировать.
- К группам томов можно добавлять дополнительное хранилище, а логические тома можно динамически расширять.

ГЛАВА 8

УПРАВЛЕНИЕ ЛОГИЧЕСКИМИ ТОМАМИ

ЦЕЛЬ

Управляйте хранилищем с помощью системы управления локальным хранилищем **Stratis** и используйте тома **VDO** для оптимизации используемого пространства хранения..

ЗАДАЧИ

- Управление многоуровневым хранилищем с помощью **Stratis** (и упражнения с пошаговыми инструкциями)
- Сжатие и дедупликация хранилища с помощью VDO (и упражнения с пошаговыми инструкциями).

РАЗДЕЛЫ

- Создание логических томов (и упражнения с пошаговыми инструкциями)
- Расширение логических томов (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Внедрение расширенных функций хранения

МНОГОСТОРОННЕЕ УПРАВЛЕНИЕ ХРАНИЛИЩЕМ С ИСПОЛЬЗОВАНИЕМ STRATUS

ЦЕЛИ

После завершения этого раздела вы сможете управлять несколькими уровнями хранения с помощью управления локальным хранилищем **Stratis**.

ОПИСАНИЕ АРХИТЕКТУРЫ STRATIS

Текущее локальное хранилище в Red Hat Enterprise Linux (RHEL) включает в себя множество стабильных и зрелых технологий, включая средство сопоставления устройств ((device mapper) **dm**), диспетчер логических томов (**LVM**) и файловую систему **XFS**. Функции, предоставляемые этими компонентами, включают масштабируемые файловые системы, моментальные снимки (**snapshots**), избыточные (**RAID**) логические устройства, много путевое подключение, тонкое выделение ресурсов, кэширование, дедупликацию и поддержку виртуальных машин и контейнеров. Каждый уровень стека хранения (**dm**, **LVM** и **XFS**) управляет с помощью команд и утилит для конкретных уровней, что требует, чтобы системные администраторы управляли физическими устройствами, томами фиксированного размера и файловыми системами как отдельными компонентами хранилища.

В последние годы появилось новое поколение решений для управления хранением данных, именуемых файловыми системами управления томами, которые динамически и прозрачно управляют уровнем томов по мере создания и изменения размера файловых систем. Однако, хотя разработка этих файловых систем сообществом продолжалась в течение многих лет, ни одна из них не достигла уровня поддержки функций и стабильности, необходимого для того, чтобы стать основным локальным хранилищем для Red Hat Enterprise Linux.

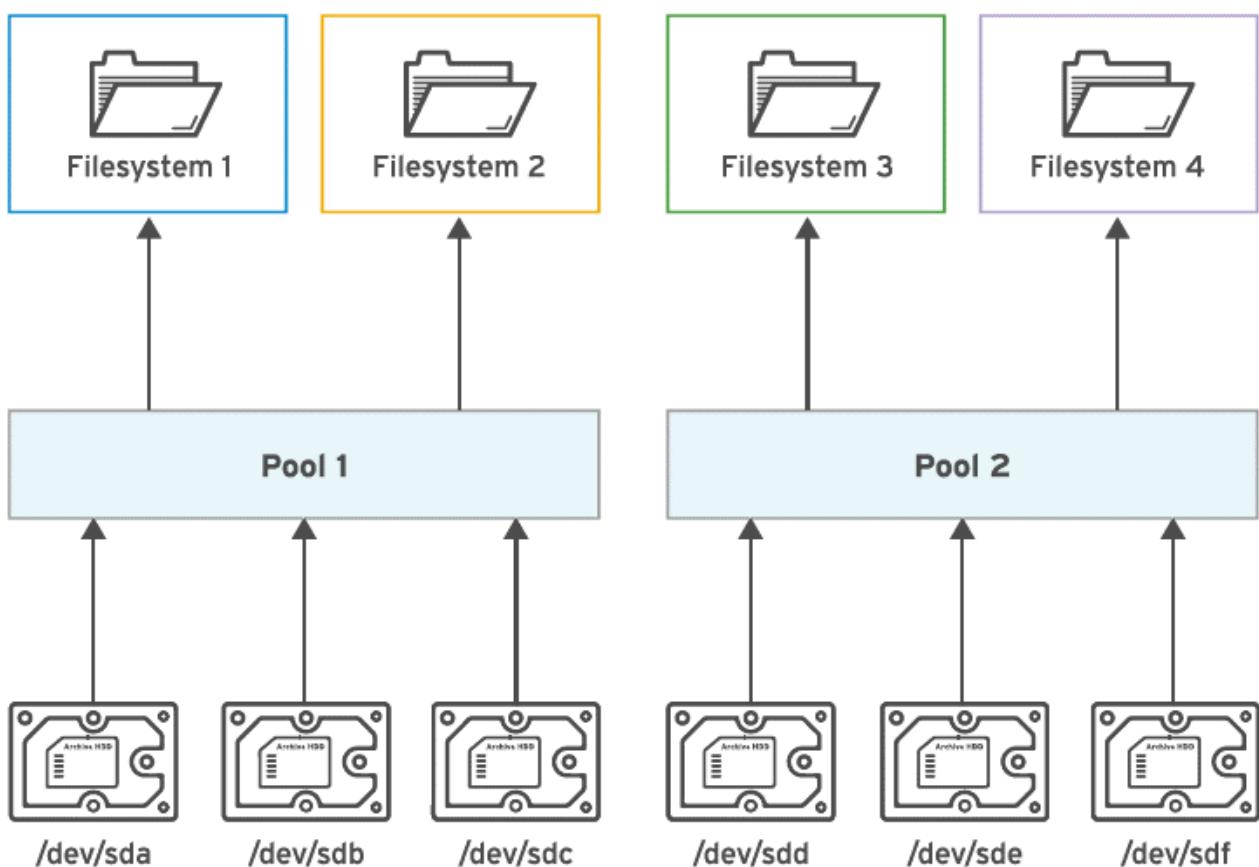
В RHEL 8 Red Hat представляет решение для управления хранилищем **Stratis**. Вместо разработки с нуля, как это пытались делать другие проекты хранения, **Stratis** работает с существующими компонентами хранилища RHEL. **Stratis** работает как служба, которая управляет пулами физических устройств хранения и прозрачно создает, и управляет томами для создаваемых файловых систем. Поскольку **Stratis** использует существующие драйверы и инструменты хранения, все расширенные функции хранения, которые вы в настоящее время используете в LVM, XFS и устройстве отображения, также поддерживаются **Stratis**.

В файловой системе с управлением томами файловые системы создаются внутри общих пулов дисковых устройств с использованием концепции, известной как тонкое выделение ресурсов. Файловые системы **Stratis** не имеют фиксированных размеров и больше не выделяют неиспользуемое блочное пространство заранее. Хотя файловая система по-прежнему построена на скрытом томе LVM, **Stratis** управляет базовым томом за вас и может расширить его при необходимости. Используемый размер файловой системы рассматривается как количество фактических блоков, используемых содержащимися файлами. Пространство, доступное для файловой системы, — это объем пространства, все еще неиспользованного на устройствах из пула, на которых она находится. Несколько файловых систем могут находиться в одном и том же пуле дисковых устройств, разделяя доступное пространство, но файловые системы также могут резервировать пространство пула, чтобы гарантировать доступность при необходимости.

Stratis использует сохраненные метаданные для распознавания управляемых пулов, томов и файловых систем. Поэтому файловые системы, созданные Stratis, никогда не следует переформатировать или переконфигурировать вручную; ими следует управлять только с помощью инструментов и команд **Stratis**. Ручная настройка файловых систем Stratis может привести к потере этих метаданных и помешать Stratis распознавать созданные им файловые системы.

Вы можете создать несколько пулов с разными наборами блочных устройств. Из каждого пула можно создать одну или несколько файловых систем. В настоящее время вы можете создать до 224 файловых систем на пул. На следующей диаграмме показано, как расположены элементы решения для управления хранением **Stratis**.

Рисунок 8.1: Элементы Stratis



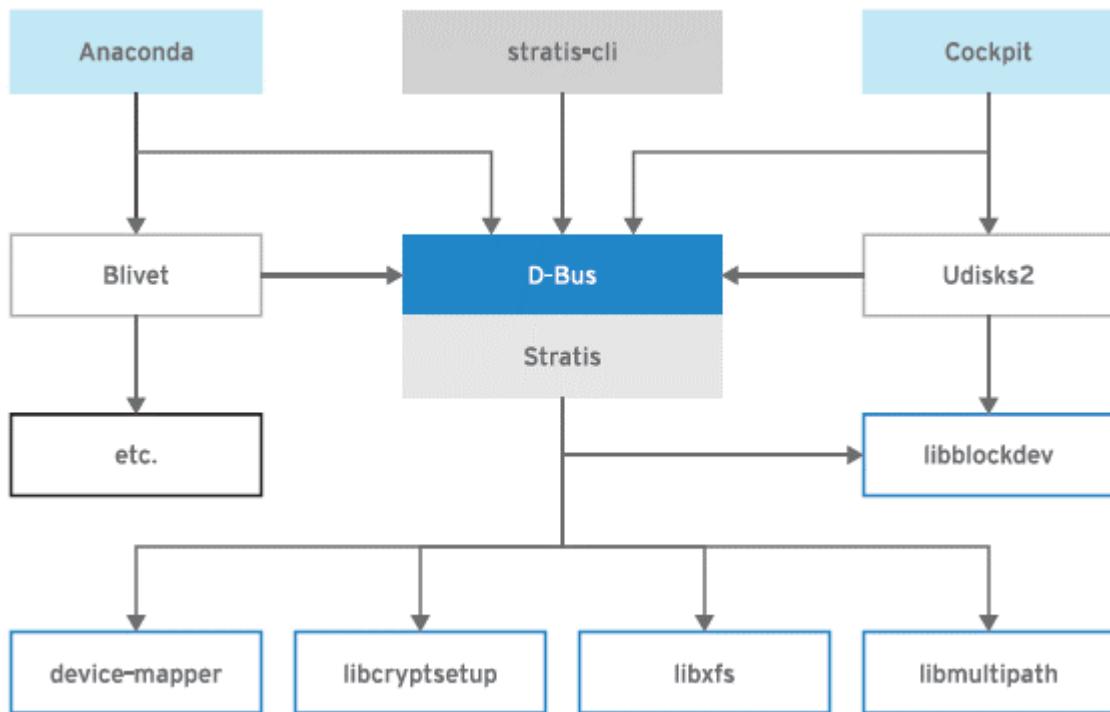
Пул группирует блочные устройства на уровне данных и, при необходимости, на уровне кэша. Уровень данных ориентирован на гибкость и целостность, а уровень кэша — на повышение производительности. Поскольку уровень кэша предназначен для повышения производительности, следует использовать блочные устройства с более высокой скоростью ввода-вывода в секунду (IOPS), например твердотельные накопители.

Описание упрощенного стека хранения

Stratis упрощает многие аспекты предоставления и настройки локального хранилища для целого ряда продуктов Red Hat. Например, в более ранних версиях установщика **Anaconda** системным администраторам приходилось накладывать один аспект управления дисками на другой. Теперь установщик использует **Stratis**, что упрощает настройку диска. Другие продукты,

использующие **Stratis**, включают **Cockpit**, **Red Hat Virtualization** и **Red Hat Enterprise Linux Atomic Host**. Для всех этих продуктов **Stratis** упрощает управление дисковым пространством и моментальными снимками и делает его менее подверженным ошибкам. **Stratis** обеспечивает более простую интеграцию с инструментами управления более высокого уровня, чем использование любого программного интерфейса командной строки.

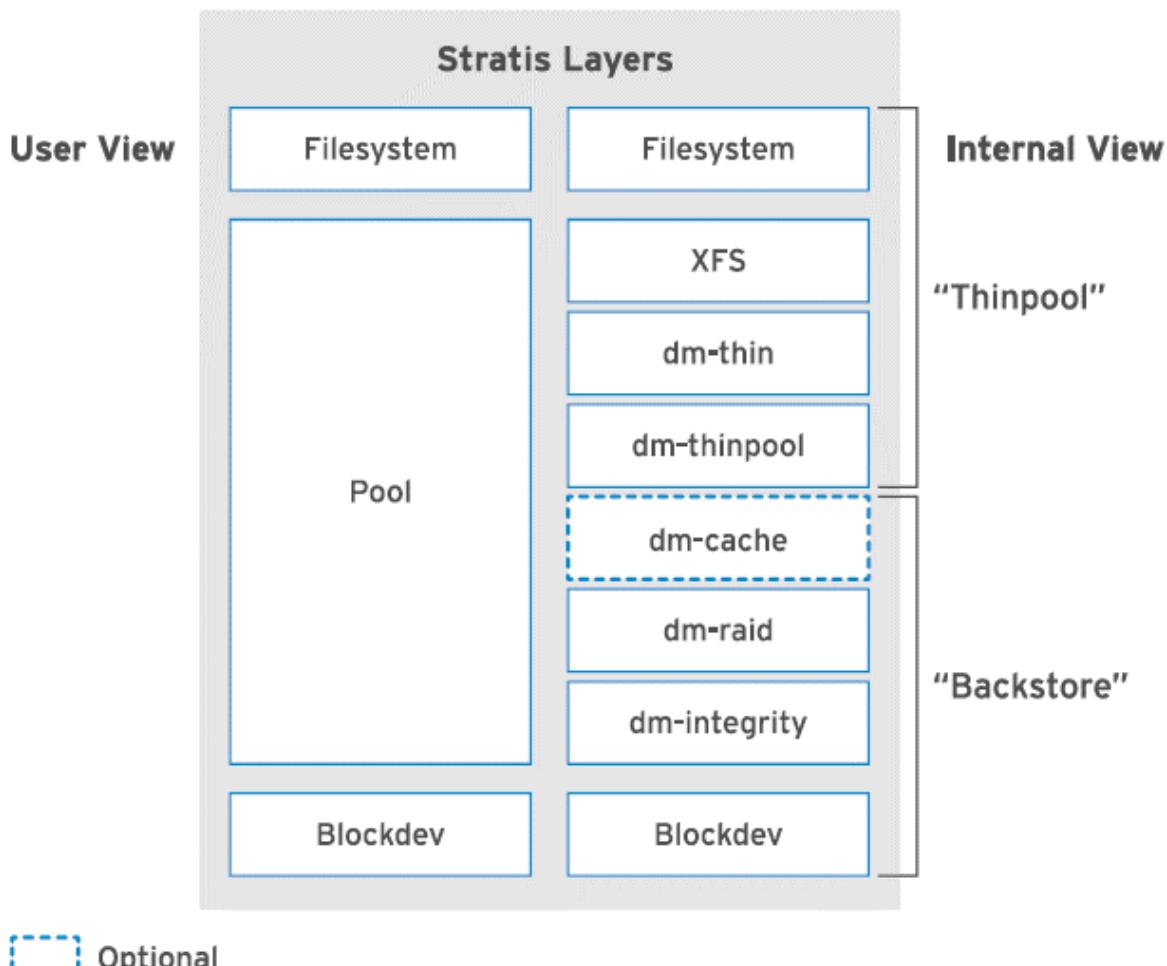
Рисунок 8.2: Stratis в стеке управления хранилищем Linux



Описание слоев Stratis

Внутри **Stratis** использует подсистему **Backstore** для управления блочными устройствами и подсистему **Thinpool** для управления пулами. Подсистема **Backstore** имеет уровень данных, который поддерживает метаданные на диске на блочных устройствах, а также обнаруживает и исправляет повреждение данных. Уровень кэша использует высокопроизводительные блочные устройства, которые действуют как кэш поверх уровня данных. Подсистема **Thinpool** управляет томами с тонким предоставлением, связанными с файловыми системами **Stratis**. Эта подсистема использует драйвер сопоставления устройств **dm-thin** для замены **LVM** при определении размера виртуального тома и управлении им. **dm-thin** создает тома большого виртуального размера, отформатированные с помощью **XFS**, но небольшого физического размера. Когда физический размер приближается к полному, **Stratis** автоматически увеличивает его.

Рисунок 8.3: Слои Stratis



Управление файловыми системами с тонким предоставлением

Для управления файловыми системами с тонким предоставлением с помощью решения для управления хранилищем **Stratis** установите пакеты **stratis-cli** и **stratisd**. Пакет **stratis-cli** предоставляет команду **stratis**, которая транслирует пользовательские запросы в службу **stratisd** через API D-Bus. Пакет **stratisd** предоставляет службу **stratisd**, реализующую интерфейс D-Bus, а также управляющую и отслеживающую элементы **Stratis**, такие как блочные устройства, пулы и файловые системы. API D-Bus доступен, если запущена служба **stratisd**.

Установите и активируйте **Stratis** с помощью обычных инструментов:

- Установите **stratis-cli** и **stratisd** с помощью команды **yum install**.

```
[root@host ~]# yum install stratis-cli stratisd
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

- Активируйте службу **stratisd** с помощью команды **systemctl**.

```
[root@host ~]# systemctl enable --now stratisd
```

Ниже приведены общие операции управления, выполняемые с помощью решения для управления хранением **Stratis**.

- Создайте пулы из одного или нескольких блочных устройств с помощью команды **stratis pool create**.

```
[root@host ~]# stratis pool create pool1 /dev/vdb
```

Каждый пул представляет собой подкаталог в каталоге **/stratis**.

- Используйте команду **stratis pool list** для просмотра списка доступных пулов.

```
[root@host ~]# stratis pool create pool1 /dev/vdb
Name      Total Physical Size  Total Physical Used
pool1          5 GiB            52 MiB
```

- Используйте команду **stratis pool add-data** для добавления в пул дополнительных блочных устройств.

```
[root@host ~]# stratis pool add-data pool1 /dev/vdc
```

- Используйте команду **stratis blockdev list** для просмотра блочных устройств пула.

```
[root@host ~]# stratis blockdev list pool1
Pool Name  Device Node    Physical Size   State  Tier
pool1      /dev/vdb        5 GiB     In-use  Data
pool1      /dev/vdc        5 GiB     In-use  Data
```

- Используйте команду **stratis filesystem create** для создания динамической и гибкой файловой системы из пула.

```
[root@host ~]# stratis filesystem create pool1 filesystem1
```

Ссылки на файловые системы **Stratis** находятся в каталоге **/stratis/pool1**.

- **Stratis** поддерживает создание моментальных снимков файловой системы с помощью команды **stratis filesystem snapshot**. Снимки не зависят от исходных файловых систем.

```
[root@host ~]# stratis filesystem snapshot pool1 filesystem1 snapshot1
```

- Используйте команду **stratis filesystem list** для просмотра списка доступных файловых систем.

```
[root@host ~]# stratis filesystem list  
...output omitted...
```

Чтобы обеспечить постоянное монтирование файловых систем **Stratis**, отредактируйте файл **/etc/fstab** и укажите сведения о файловой системе. Следующая команда отображает **UUID** файловой системы, который следует использовать в **/etc/fstab** для идентификации файловой системы.

```
[root@host ~]# lsblk --output=UUID /stratis/pool1/filesystem1  
UUID  
31b9363b-add8-4b46-a4bf-c199cd478c55
```

Ниже приведен пример записи в файле **/etc/fstab** для постоянного монтирования файловой системы **Stratis**.

```
UUID=31b9...8c55 /dir1 xfs defaults,x-systemd.requires=stratisd.service 0 0
```

Параметр монтирования **x-systemd.requires=stratisd.service** откладывает монтирование файловой системы до тех пор, пока **systemd** не запустит **stratisd.service** в процессе загрузки.



ПРИМЕЧАНИЕ

Неиспользование параметра монтирования **x-systemd.requires=stratisd.service** в **/etc/fstab** для файловой системы **Stratis** приведет к тому, что при следующей перезагрузке машина загрузится в **emergency.target**.



РЕКОМЕНДАЦИИ

Для получения дополнительной информации см. главу «Управление многоуровневым локальным хранилищем с помощью Stratis» в «Руководстве по настройке и управлению файловыми системами Red Hat Enterprise Linux 8» по адресу:

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/configuring_and_managing_file_systems/

Stratis Storage

<https://stratis-storage.github.io/>

Чему **Stratis** научился у **ZFS**, **Btrfs** и **Linux Volume Manager**

<https://opensource.com/article/18/4/stratis-lessons-learned>

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ МНОГОСТОРОННИМ ХРАНИЛИЩЕМ С ПОМОЩЬЮ STRATIS

В этом упражнении вы будете использовать решение для управления хранилищем Stratis для создания пулов, томов и файловых систем, которые работают совместно.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создавать файловую систему с тонким предоставлением, используя решение для управления хранилищем **Stratis**.
- Убедитесь, что тома **Stratis** динамично растут, чтобы поддерживать рост данных в реальном времени.
- Доступ к данным из моментального снимка файловой системы с тонким предоставлением.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab net-configure start**, чтобы начать упражнение. Этот сценарий правильно настраивает среду и гарантирует чистоту дополнительных дисках на сервере **servera**.

```
[student@workstation ~]$ lab advstorage-stratis start
```

1. С рабочей станции откройте сеанс **SSH** на сервер в качестве студента.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Переключитесь на пользователя **root**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

3. Установите пакеты **stratisd** и **stratis-cli** с помощью команды **yum**.

```
[root@servera ~]# yum install stratisd stratis-cli  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Complete!
```

4. Активируйте службу **stratisd** с помощью команды **systemctl**.

```
[root@servera ~]# systemctl enable --now stratisd
```

5. Убедитесь, что существует пул **stratispool1** с блочным устройством **/dev/vdb**.

5.1. Создайте пул **Stratis** с именем **stratispool1** с помощью команды создания пула **stratis**.

```
[root@servera ~]# stratis pool create stratispool1 /dev/vdb
```

5.2. Проверьте доступность **stratispool1** с помощью команды **stratis pool list**.

```
[root@servera ~]# stratis pool list  
Name          Total Physical Size  Total Physical Used  
stratispool1           5 GiB            52 MiB
```

Обратите внимание на размер пула в предыдущем выводе.

6. Расширьте емкость **stratispool1** с помощью блочного устройства **/dev/vdc**.

6.1. Добавьте блочное устройство **/dev/vdc** в **stratispool1** с помощью команды **stratis pool add-data**.

```
[root@servera ~]# stratis pool add-data stratispool1 /dev/vdc
```

6.2. Проверьте размер **stratispool1** с помощью команды **stratis pool list**.

```
[root@servera ~]# stratis pool list  
Name          Total Physical Size  Total Physical Used  
stratispool1           10 GiB           56 MiB
```

Как показано выше, размер пула **stratispool1** увеличился при добавлении блочного устройства.

- 6.3.** Проверьте блочные устройства, которые в настоящее время являются членами **stratispool1**, с помощью команды **stratis blockdev list**.

```
[root@servera ~]# stratis blockdev list stratispool1
Pool Name      Device Node    Physical Size   State   Tier
stratispool1   /dev/vdb           5 GiB   In-use   Data
stratispool1   /dev/vdc           5 GiB   In-use   Data
```

- 7.** Добавьте файловую систему с тонким предоставлением с именем **stratis-filesystem1** в пул **stratispool1**. Смонтируйте файловую систему в **/stratisvol**. Создайте в файловой системе **stratis-filesystem1** файл с именем **file1**, содержащий текст **Hello World!**.

- 7.1.** Создайте файловую систему с тонким предоставлением **stratis-filesystem1** в **stratispool1** с помощью команды создания файловой системы **stratis**. Выполнение команды может занять до минуты.

```
[root@servera ~]# stratis filesystem create stratispool1 stratis-filesystem1
```

- 7.2.** Проверьте доступность **stratis-filesystem1** с помощью команды **stratis filesystem list**.

```
[root@servera ~]# stratis filesystem list
Pool Name      Name          Used     Created           Device
                      UUID
stratispool1   stratis-filesystem1  546 MiB  Mar 29 2019 07:48  /
stratis/stratispool1/stratis-filesystem1  8714...e7db
```

Обратите внимание на текущее использование **stratis-filesystem1**. Это использование файловой системы увеличивается по требованию на следующих шагах.

- 7.3.** Создайте каталог с именем **/stratisvol** с помощью команды **mkdir**.

```
[root@servera ~]# mkdir /stratisvol
```

- 7.4.** Смонтируйте **stratis-filesystem1** в **/stratisvol** с помощью команды **mount**.

```
[root@servera ~]# mount /stratis/stratispool1/stratis-filesystem1 /stratisvol
```

- 7.5.** Убедитесь, что файловая система **Stratis stratis-filesystem1** смонтирована в **/stratisvol** с помощью команды **mount**.

```
[root@servera ~]# mount
```

```
...output omitted...
/dev/mapper/stratis-1-5c0e...12b9-thinfs-
8714...e7db on /stratisvol type xfs
  (rw,relatime,seclabel,attr2,inode64,sunit=2048,swidth=2048,noquota)
```

- 7.6. Создайте текстовый файл **/stratisvol/file1** с помощью команды **echo**.

```
[root@servera ~]# echo "Hello World!" > /stratisvol/file1
```

8. Убедитесь, что файловая система с тонким предоставлением **stratis-filesystem1** динамически растет по мере роста данных в файловой системе.

- 8.1. Просмотрите текущее использование **stratis-filesystem1** с помощью команды **stratis filesystem list**.

```
[root@servera ~]# stratis filesystem list
Pool Name      Name          Used     Created        Device
                  UUID
stratispool1  stratis-filesystem1  546 MiB  Mar 29 2019 07:48  /
stratis/stratispool1/stratis-filesystem1  8714...e7db
```

- 8.2. Создайте файл размером **2 ГБ** в **stratis-filesystem1** с помощью команды **dd**. Выполнение команды может занять до минуты.

```
[root@servera ~]# dd if=/dev/urandom of=/stratisvol/file2 bs=1M
count=2048
```

- 8.3. Проверьте использование **stratis-filesystem1** с помощью команды **stratis filesystem list**.

```
[root@servera ~]# stratis filesystem list
Pool Name      Name          Used     Created        Device
                  UUID
stratispool1  stratis-filesystem1  2.53 GiB  Mar 29 2019 07:48  /
stratis/stratispool1/stratis-filesystem1  8714...e7db
```

Предыдущий вывод показывает, что использование **stratis-filesystem1** увеличилось. Рост использования подтверждает, что файловая система с тонким предоставлением динамически расширилась, чтобы приспособиться к росту данных в реальном времени, вызванному созданием **/stratisvol/file2**.

9. Создайте моментальный снимок (*snapshot*) **stratis-filesystem1** с именем **stratis-filesystem1-snap**. Снимок предоставит вам доступ к любому файлу, удаленному из **stratis-filesystem1**.

9.1. Создайте моментальный снимок **stratis-filesystem1** с помощью команды **stratis filesystem snapshot**. Выполнение команды может занять до минуты.

```
[root@servera ~]# stratis filesystem snapshot stratispool1 \
stratis-filesystem1 stratis-filesystem1-snap
```

9.2. Проверьте доступность моментального снимка с помощью команды **stratis filesystem list**.

```
[root@servera ~]# stratis filesystem list
...output omitted...
stratispool1  stratis-filesystem1-snap  2.53 GiB   Mar 29 2019 10:28 /
stratis/stratispool1/stratis-filesystem1-snap  291d...8a16
```

9.3. Удалите файл **/stratisvol/file1**.

```
[root@servera ~]# rm /stratisvol/file1
rm: remove regular file '/stratisvol/file1'? y
```

9.4. Создайте каталог **/stratisvol-snap** с помощью команды **mkdir**.

```
[root@servera ~]# mkdir /stratisvol-snap
```

9.5. Смонтируйте моментальный снимок **stratis-filesystem1-snap** в **/stratisvol-snap** с помощью команды **mount**.

```
[root@servera ~]# mount /stratis/stratispool1/stratis-filesystem1-snap /stratisvol-snap
```

9.6. Убедитесь, что вы по-прежнему можете получить доступ к файлу, который вы удалили из **stratis-filesystem1**, с помощью моментального снимка (snapshot) **stratis-filesystem1-snap**.

```
[root@servera ~]# cat /stratisvol-snap/file1
Hello World!
```

10. Размонтируйте **/stratisvol** и **/stratisvol-snap** с помощью команды **umount**.

```
[root@servera ~]# umount /stratisvol-snap
[root@servera ~]# umount /stratisvol
```

11. Удалите из системы файловую систему с тонким предоставлением **stratis-filesystem1** и ее моментальный снимок **stratis-filesystem1-snap**.

11.1. Уничтожьте **stratis-filesystem1-snap** с помощью команды **stratis filesystem destroy**.

```
[root@servera ~]# stratis filesystem destroy stratispool1 stratisfilesystem1-snap
```

11.2. Уничтожьте **stratis-filesystem1** с помощью команды **stratis filesystem destroy**.

```
[root@servera ~]# stratis filesystem destroy stratispool1 stratisfilesystem1
```

11.3. Выйдите из оболочки пользователя **root**.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$
```

11.4. Выйдите из сервера **servera**.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$
```

Завершение

На рабочей станции **workstation**, запустите скрипт **lab advstorage-stratis finish**, чтобы завершить это упражнение. Данный сценарий удаляет разделы и файлы, созданные во время упражнения, и обеспечивает чистоту среды.

```
[student@workstation ~]$ lab advstorage-stratis finish
```

На этом управляемое упражнение заканчивается.

СЖАТИЕ И ДЕДУПЛИКАЦИЯ ХРАНИЛИЩА С ПОМОЩЬЮ VDO

Цели

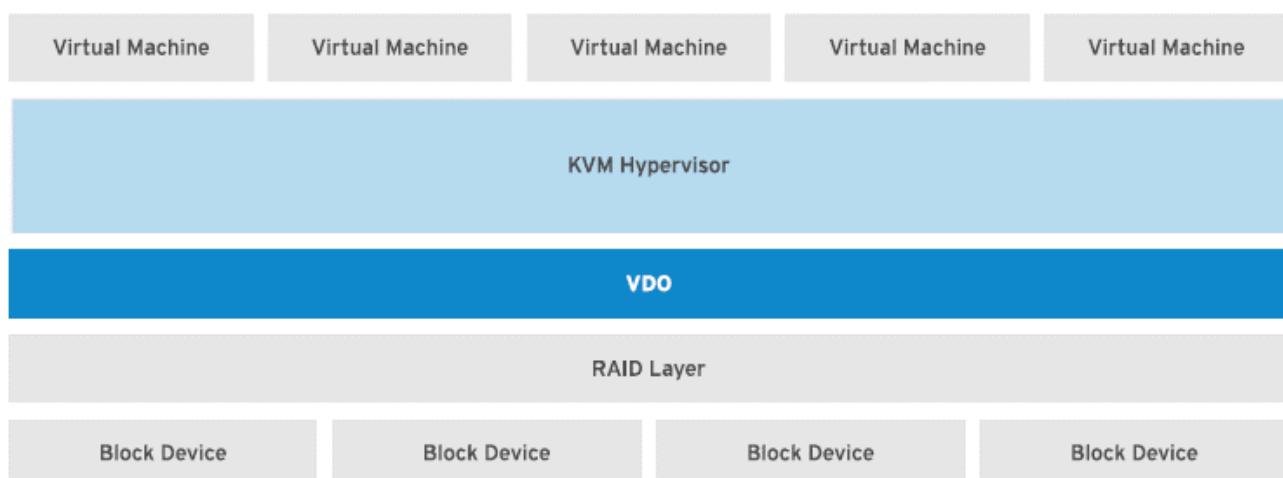
После завершения этого раздела вы должны быть в состоянии оптимизировать использование места для хранения с помощью VDO для сжатия и дедупликации данных на устройствах хранения.

Описание виртуального оптимизатора данных

Red Hat Enterprise Linux 8 включает в себя драйвер виртуального оптимизатора данных (Virtual Data Optimizer (**VDO**)), который оптимизирует след данных на блочных устройствах. **VDO** - это драйвер Mapper устройств Linux, который уменьшает использование дискового пространства на блочных устройствах и минимизирует репликацию даты, сохраняя дисковое пространство и даже увеличение пропускной способности данных. **VDO** включает в себя два модуля ядра: **модуль KVDO** для прозрачно контроля сжатия данных и **модуль UDS** для дедупликации.

Слой **VDO** размещен сверху существующего блока хранения блока, такого как устройство RAID или локальный диск. Эти блочные устройства также могут быть зашифрованы устройствами. Слои хранения, такие как логические объемы **LVM** и файловые системы, размещаются на вершине устройства **VDO**. Следующая диаграмма показывает размещение **VDO** в инфраструктуре, состоящей из виртуальных машин **KVM**, которые используют оптимизированные устройства хранения.

Рисунок 8.4: Виртуальные виртуальные машины



VDO применяет три фазы на сегодняшний день в следующем порядке, чтобы уменьшить след на устройствах хранения:

1. Удаление нулевого блока (**Zero-Block Elimination**) фильтрует блоки данных, которые содержат только **Zeroes** (0) и записывает информацию о этих блоках только в метаданных. Ненулевые блоки данных затем передаются на следующий этап обработки. Эта фаза позволяет тонкую функцию предоставления в устройствах **VDO**.

- Дедупликация (**Deduplication**) устраниет избыточные блоки данных. Когда вы создаете несколько копий некоторых данных, **VDO** обнаруживает блоки данных и обновляет метаданные для использования этих дубликатов блокирует ссылки на исходный блок данных без создания резервных блоков данных. Универсальный модуль сервиса **Deduplication** (UDS) проверяет избыточность даты через метаданные, которые она поддерживает. Этот модуль ядра отправляет часть **VDO**.
- Сжатие (**Compression**) - это последний этап. Модуль **KVDO Kernel** сжимает блоки данных с помощью сжатия **LZ4** и группы их на 4 КБ блоки.

Реализация виртуального оптимизатора данных (VDO)

Логические устройства, которые вы создаете с помощью **VDO**, называются **VDO Volums**. Объемы **VDO** похожи на разделы диска; Вы можете отформатировать тома с нужным типом файловой системы и монтировать его как обычная файловая система. Вы также можете использовать объем **VDO** в качестве физического объема **LVM**.

Чтобы создать том **VDO**, укажите блок-устройство и имя логического устройства, которое **VDO** представляет пользователю. Вам необязательно указать логический размер тома. Логический размер объема может быть больше, чем физический размер текущего блочного устройства.

Поскольку тома **VDO** имеют тонкое обеспечение, пользователи могут видеть только используемое логическое пространство и не знают о реальном доступном физическом пространстве. Если вы не укажете логический размер при создании тома, **VDO** примет фактический физический размер за логический размер тома. Такое соотношение 1:1 между логическим и физическим размером обеспечивает лучшую производительность, но менее эффективное использование пространства для хранения данных. Исходя из требований вашей инфраструктуры, вы должны отдать приоритет либо производительности, либо эффективности использования пространства. производительность или эффективность использования пространства.

Когда логический размер объема больше, чем текущий физический размер, вы должны предварительно следить за статистикой тома для просмотра текущего использования с использованием команды **vdostats --verbose**.

Включение (Enabling) VDO.

Установите пакеты **vdo** и **kmod-kvdo**, чтобы включить **VDO** в системе.

```
[root@host ~]# yum install vdo kmod-kvdo
...output omitted...
Is this ok [y/N]: y
...output omitted...
Complete!
```

Создание VDO Volume

Чтобы создать том **VDO**, выполните команду **vdo create**.

```
[root@host ~]# vdo create --name=vdo1 --device=/dev/vdd --vdoLogicalSize=50G  
...output omitted...
```

Если вы опустите логический размер, результирующий том **VDO** получит тот же размер, что и его физическое устройство.

Когда том **VDO** создан, вы можете отформатировать его с выбранным вами типом файловой системы и смонтировать его в иерархии файловых систем вашей системы.

Анализ тома VDO

Чтобы проанализировать том **VDO**, выполните команду **vdo status**. Данная команда выводит отчет о системе **VDO** и состоянии тома **VDO** в формате **YAML**. Она также отображает атрибуты тома **VDO**. Используйте параметр **--name=**, чтобы указать имя конкретного тома. Если вы опустите имя конкретного тома, вывод команды **vdo status** отобразит состояние всех томов **VDO**.

```
[root@host ~]# vdo status --name=vdo1  
...output omitted...
```

Команда **vdo list** отображает список томов **VDO**, которые запущены в данный момент. Вы можете запустить и остановить том **VDO** с помощью команд **vdo start** и **vdo stop** соответственно.



РЕКОМЕНДАЦИИ

Для получения дополнительной информации обратитесь к главе "Начало работы с VDO" в руководстве Red Hat Enterprise Linux 8 Deduplicating and Compressing Storage Guide на сайте:

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/deduplicating_and_compressing_storage/

Представляем Виртуальный Оптимизатор данных

<https://rhelblog.redhat.com/2018/04/11/introducing-virtual-data-optimizer-to-reduce-cloud-and-on-premise-storage-costs/>

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

Сжатие и дедупликация хранения с VDO

В этом упражнении вы создадите том **VDO**, отформатируйте его с файловой системой, установите его, сохраните данные на нем, и рассмотрите влияние сжатия и дедупликации на фактически используемое место для хранения.

В результате

Вы должны быть способны:

- Создать объем с использованием виртуального оптимизатора данных, отформатировать его с помощью файловой системы и установите на нём файловую систему.
- Исследовать влияние дедупликации и сжатия на объем оптимизатора виртуальных данных.

Прежде чем вы начнете

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab advstorage-vdo start**, чтобы начать упражнение. Данный скрипт гарантирует, что на диске **/dev/vdd** нет разделов и правильно устанавливает окружение.

```
[student@workstation ~]$ lab advstorage-vdo start
```

1. С рабочей станции **workstation**, откройте сеанс **SSH** на сервер в качестве пользователя **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Создайте том **VDO** с именем **vdo1**, используя устройство **/dev/vdd**. Установите его логический размер на **50 ГБ**.

2.1. Переключитесь на пользователя **root**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

2.2. Подтвердите, что пакет **vdo** установлен с помощью команды **rpm**.

```
[root@servera ~]# yum list installed vdo
vdo-6.2.0.293-10.el8.x86_64
```

2.3. Создайте том **vdo1** с помощью команды **vdo create**.

```
[root@servera ~]# vdo create --name=vdo1 --device=/dev/vdd --
vdoLogicalSize=50G
...output omitted...
```

2.4. Проверьте доступность тома **vdo1** с помощью команды **vdo list**.

```
[root@servera ~]# vdo list
vdo1
```

3. Убедитесь, что для тома **vdo1** включены функции сжатия и дедупликации.

3.1. Используйте команду **grep** для поиска строк, содержащих строку **Deduplication**, в выводе команды **vdo status --name=vdo1**.

```
[root@servera ~]# vdo status --name=vdo1 | grep Deduplication
Deduplication: enabled
```

3.2. Используйте команду **grep** для поиска строк, содержащих строку **Compression**, в выводе команды **vdo status --name=vdo1**.

```
[root@servera ~]# vdo status --name=vdo1 | grep Compression
Compression: enabled
```

4. Отформатируйте том **vdo1** с файловой системой **XFS** и смонтируйте его в **/mnt/vdo1**.

4.1. Отформатируйте том **vdo1** в файловой системе **XFS** с помощью команды **mkfs**.

```
[root@servera ~]# mkfs.xfs -K /dev/mapper/vdo1
...output omitted...
```

Параметр **-K** в предыдущей команде **mkfs.xfs** предотвращает немедленное отбрасывание неиспользуемых блоков в файловой системе, что позволяет команде вернуться быстрее.

4.2. Используйте команду **udevadm**, чтобы зарегистрировать новый узел устройства.

```
[root@servera ~]# udevadm settle
```

4.3. Создайте каталог **/mnt/vdo1** с помощью команды **mkdir**.

```
[root@servera ~]# mkdir /mnt/vdo1
```

4.4. Смонтируйте том **vdo1** в точке **/mnt/vdo1** с помощью команды **mount**.

```
[root@servera ~]# mount /dev/mapper/vdo1 /mnt/vdo1
```

4.5. Убедитесь, что том **vdo1** успешно смонтирован с помощью команды **mount**.

```
[root@servera ~]# mount  
...output omitted...  
/dev/mapper/vdo1 on /mnt/vdo1 type xfs  
(rw,relatime,seclabel,attr2,inode64,noquota)
```

5. Создайте три копии одного и того же файла с именем **/root/install.img** на томе **vdo1**. Сравните статистику тома, чтобы проверить дедупликацию и сжатие данных, происходящие на томе. Предыдущий вывод может отличаться в вашей системе.

5.1. Просмотрите начальную статистику и состояние тома с помощью команды **vdostats**.

```
[root@servera ~]# vdostats --human-readable  
Device          Size     Used Available Use% Space saving%  
/dev/mapper/vdo1    5.0G    3.0G      2.0G  60%        99%
```

Обратите внимание, что **Used** тома **3 ГБ** уже используются, потому что при создании том **VDO** резервирует **3–4 ГБ** для себя. Кроме того, обратите внимание, что значение **Space saving%** **99%** в поле % экономии места указывает на то, что вы еще не создали никакого содержимого в томе, занимающего все сохраненное пространство тома.

5.2. Скопируйте **/root/install.img** в **/mnt/vdo1/install.img.1** и проверьте статистику тома. Копирование файла может занять до минуты.

```
[root@servera ~]# cp /root/install.img /mnt/vdo1/install.img.1  
[root@servera ~]# vdostats --human-readable  
Device          Size     Used Available Use% Space saving%  
/dev/mapper/vdo1    5.0G    3.4G      1.6G  68%        5%
```

Обратите внимание, что значение поля «Used» увеличилось с **3,0 ГБ** до **3,4 ГБ**, потому что вы скопировали файл на том, а это занимает некоторое пространство. Кроме того, обратите внимание, что значение поля «**Space saving**» уменьшилось с **99 %** до **5 %**, поскольку изначально в томе не было содержимого, что способствовало низкому использованию пространства тома и большой экономии пространства тома, пока вы не создали файл. Экономия места на томе значительно ниже, потому что вы создали уникальную копию файла на томе и нечего дедуплицировать.

- 5.3.** Скопируйте **/root/install.img** в **/mnt/vdo1/install.img.2** и проверьте статистику тома.
Копирование файла может занять до минуты.

```
[root@servera ~]# cp /root/install.img /mnt/vdo1/install.img.2
[root@servera ~]# vdostats --human-readable
Device          Size     Used Available Use% Space saving%
/dev/mapper/vdo1    5.0G    3.4G      1.6G  68%           51%
```

Обратите внимание, что используемое пространство тома не изменилось, а процент сохраненного пространства тома увеличился, что доказывает, что дедупликация данных произошла для уменьшения потребления пространства для избыточных копий одного и того же файла. Значение **Space saving %** в предыдущем выводе может различаться в вашей системе.

- 5.4.** Выйдите из оболочки пользователя root.

```
[root@servera ~]# exit
logout
[student@servera ~]$
```

- 5.5.** Выйдите из сервера **servera**.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите **lab advstorage-vdo finish**, чтобы завершить упражнение. Этот сценарий удаляет файлы, созданные во время выполнения упражнения, и обеспечивает чистоту среды.

```
[student@workstation ~]$ lab advstorage-vdo finish
```

На этом управляемое упражнение заканчивается.

ЛАБОРАТОРНАЯ РАБОТА

РЕАЛИЗАЦИЯ РАСШИРЕННЫХ ФУНКЦИЙ ХРАНИЛИЩА

В этом упражнении вы будете использовать решение для управления хранилищем **Stratis** для создания файловых систем, которые будут расширяться в соответствии с растущими потребностями в данных, и **Virtual Data Optimizer** для создания томов для эффективного использования дискового пространства.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создать файловую систему с тонким выделением ресурсов с помощью решения для управления хранилищем **Stratis**.
- Убедитесь, что тома **Stratis** динамично растут, чтобы поддерживать рост данных в реальном времени.
- Доступ к данным из **snapshot** тонко подготовленной файловой системы.
- Создайте том с помощью **Virtual Data Optimizer** и смонтируйте его в файловой системе.
- Исследуйте влияние дедупликации и сжатия данных на том **Virtual Data Optimizer**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab advstorage-review start**. Этот сценарий правильно настраивает среду и гарантирует, что дополнительные диски на **serverb** чистые.

```
[student@workstation ~]$ lab advstorage-review start
```

1. С рабочей станции **workstation** откройте SSH-сессию на **serverb** в качестве студента.
2. Переключитесь на пользователя **root**.
3. Установите пакеты **stratisd** и **stratis-cli** с помощью команды **yum**.
4. Запустите и включите службу **stratisd** с помощью команды **systemctl**.
5. Создайте **Stratis** пул **labpool**, содержащий блочное устройство **/dev/vdb**.
6. Расширьте емкость **labpool**, используя доступный в системе диск **/dev/vdc**.
7. Создайте тонко подготовленную файловую систему с именем **labfs** в пуле с именем **labpool**. Смонтируйте эту файловую систему в **/labstratisvol**, чтобы она сохранялась при перезагрузке. Создайте файл с именем **labfile1**, содержащий текст **Hello World!** в файловой системе **labfs**. Не забудьте использовать параметр монтирования **x-systemd.requires=stratisd.service** в файле **/etc/fstab**.
8. Убедитесь, что **labfs** файловой системы с тонким предоставлением динамически увеличивается по мере роста данных в файловой системе.

9. Создайте моментальный снимок (*snapshot*) файловой системы **labfs** с именем **labfs-snap**. Снимок (*snapshot*) позволяет получить доступ к любому файлу, удаленному из **labfs**.
10. Создайте том **VDO** с именем **labvdo** с устройством **/dev/vdd**. Установите его логический размер на **50 ГБ**.
11. Смонтируйте том **labvdo** в **/labvdovol** с файловой системой **XFS**, чтобы он сохранялся при перезагрузке. Не забудьте использовать параметр монтирования **x-systemd.requires=vdo.service** в файле **/etc/fstab**.
12. Создайте три копии файла с именем **/root/install.img** на томе **labvdo**. Сравните статистику тома, чтобы проверить дедупликацию и сжатие данных, происходящие на томе.

Оценка

На рабочей станции **workstation**, запустите скрипт **lab advstorage-review grade**, чтобы подтвердить успешность выполнения этого упражнения.

```
[student@workstation ~]$ lab advstorage-review grade
```

Завершение

На рабочей станции **workstation**, выполните скрипт **lab advstorage-review finish**, чтобы завершить данное упражнение. Этот сценарий удаляет разделы и файлы, созданные во время упражнения, и обеспечивает чистоту среды.

```
[student@workstation ~]$ lab advstorage-review finish
```

На этом лабораторная работа заканчивается.

РЕШЕНИЕ

РЕАЛИЗАЦИЯ РАСШИРЕННЫХ ФУНКЦИЙ ХРАНИЛИЩА

В этом упражнении вы будете использовать решение для управления хранилищем **Stratis** для создания файловых систем, которые будут расширяться в соответствии с растущими потребностями в данных, и **Virtual Data Optimizer** для создания томов для эффективного использования дискового пространства.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создать файловую систему с тонким выделением ресурсов с помощью решения для управления хранилищем **Stratis**.
- Убедитесь, что тома **Stratis** динамично растут, чтобы поддерживать рост данных в реальном времени.
- Доступ к данным из **snapshot** тонко подготовленной файловой системы.
- Создайте том с помощью **Virtual Data Optimizer** и смонтируйте его в файловой системе.
- Исследуйте влияние дедупликации и сжатия данных на том **Virtual Data Optimizer**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab advstorage-review start**. Этот сценарий правильно настраивает среду и гарантирует, что дополнительные диски на **serverb** чистые.

```
[student@workstation ~]$ lab advstorage-review start
```

1. С рабочей станции **workstation** откройте SSH-сессию на **serverb** в качестве студента.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

2. Переключитесь на пользователя **root**.

```
[student@serverb ~]$ sudo -i  
[sudo] password for student: student  
[root@serverb ~]#
```

3. Установите пакеты **stratisd** и **stratis-cli** с помощью команды **yum**.

```
[root@serverb ~]# yum install stratisd stratis-cli  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Complete!
```

4. Запустите и включите службу **stratisd** с помощью команды **systemctl**.

```
[root@serverb ~]# systemctl enable --now stratisd
```

5. Создайте **Stratis** пул с именем **labpool**, содержащий блочное устройство **/dev/vdb**.

5.1. Создайте пул Stratis с именем **labpool** с помощью команды **stratis pool create**.

```
[root@serverb ~]# stratis pool create labpool /dev/vdb
```

5.2. Проверьте доступность **labpool** с помощью команды **stratis pool list**.

```
[root@serverb ~]# stratis pool list  
Name          Total Physical Size  Total Physical Used  
labpool        5 GiB                  52 MiB
```

Обратите внимание на размер пула в предыдущем выводе.

6. Расширьте емкость **labpool**, используя доступный в системе диск **/dev/vdc**.

6.1. Добавьте блочное устройство **/dev/vdc** в **labpool** с помощью команды **stratis pool add-data**.

```
[root@serverb ~]# stratis pool add-data labpool /dev/vdc
```

6.2. Проверьте размер пула **labpool** с помощью команды **stratis pool list**.

```
[root@serverb ~]# stratis pool list  
Name          Total Physical Size  Total Physical Used  
labpool        10 GiB                 56 MiB
```

Предыдущий вывод показывает, что размер **labpool** увеличился после добавления в пул нового диска.

6.3. Используйте команду **stratis blockdev list**, чтобы вывести список блочных устройств, которые теперь являются членами **labpool**.

```
[root@serverb ~]# stratis blockdev list labpool
Pool Name      Device Node   Physical Size   State   Tier
labpool        /dev/vdb          5 GiB  In-use  Data
labpool        /dev/vdc          5 GiB  In-use  Data
```

7. Создайте тонко подготовленную файловую систему с именем **labfs** в пуле с именем **labpool**. Смонтируйте эту файловую систему в **/labstratisvol**, чтобы она сохранялась при перезагрузке. Создайте файл с именем **labfile1**, содержащий текст **Hello World!** в файловой системе **labfs**. Не забудьте использовать параметр монтирования **x-systemd.requires=stratisd.service** в файле **/etc/fstab**.

7.1. Создайте тонко подготовленные файловые системы **labfs** в **labpool** с помощью команды **stratis filesystem create**. Выполнение команды может занять до минуты.

```
[root@serverb ~]# stratis filesystem create labpool labfs
```

7.2. Проверьте доступность **labfs** с помощью команды **stratis filesystem list**.

```
[root@serverb ~]# stratis filesystem list
Pool Name      Name           Used     Created           Device
                           UUID
labpool    labfs  546 MiB  Mar 29 2019 07:48  /stratis/labpool/labfs
9825...d6ca
```

Обратите внимание на текущее использование **labfs**. Это использование файловой системы увеличивается по требованию на следующих шагах.

7.3. Определите **UUID** **labfs** с помощью команды **lsblk**.

```
[root@serverb ~]# lsblk --output=UUID /stratis/labpool/labfs
UUID
9825e289-fb08-4852-8290-44d1b8f0d6ca
```

7.4. Отредактируйте файл **/etc/fstab** так, чтобы файловая система **labfs** с тонкой инициализацией монтировалась во время загрузки. Используйте **UUID**, который вы определили на предыдущем шаге. Ниже показана строка, которую вы должны добавить в файл **/etc/fstab**. Вы можете использовать команду **vi /etc/fstab** для редактирования файла.

```
UUID=9825...d6ca      /labstratisvol  xfs      defaults,xsystemd.requires=stratisd.service 0 0
```

- 7.5. Создайте каталог с именем **/labstratisvol** с помощью команды **mkdir**.

```
[root@serverb ~]# mkdir /labstratisvol
```

- 7.6. Смонтируйте **labfs** тонко подготовленной файловой системы с помощью команды **mount**, чтобы убедиться, что файл **/etc/fstab** содержит соответствующие записи.

```
[root@serverb ~]# mount /labstratisvol
```

Если предыдущая команда вызывает какие-либо ошибки, снова откройте файл **/etc/fstab** и убедитесь, что он содержит соответствующие записи.

- 7.7. Создайте текстовый файл с именем **/labstratisvol/labfile1** с помощью команды **echo**.

```
[root@serverb ~]# echo "Hello World!" > /labstratisvol/labfile1
```

8. Убедитесь, что **labfs** файловой системы с тонким предоставлением динамически увеличивается по мере роста данных в файловой системе.

- 8.1. Просмотрите текущее использование **labfs** с помощью команды **stratis filesystem list**.

```
[root@serverb ~]# stratis filesystem list
Pool Name      Name          Used      Created      Device
                  UUID
labpool  labfs  546 MiB  Mar 29 2019 07:48  /stratis/labpool/labfs
9825...d6ca
```

- 8.2. Создайте файл размером **2 ГБ** в **labfs** с помощью команды **dd**. Выполнение команды может занять до минуты.

```
[root@serverb ~]# dd if=/dev/urandom of=/labstratisvol/labfile2 bs=1M
count=2048
```

- 8.3. Убедитесь, что использование **labfs** увеличилось, используя команду **stratis filesystem list**.

```
[root@serverb ~]# stratis filesystem list
Pool Name      Name          Used      Created      Device
                  UUID
labpool  labfs  2.53 GiB  Mar 29 2019 07:48  /stratis/labpool/labfs
9825...d6ca
```

- 9.** Создайте моментальный снимок (*snapshot*) файловой системы **labfs** с именем **labfs-snap**. Снимок (*snapshot*) позволяет получить доступ к любому файлу, удаленному из **labfs**.
- 9.1.** Создайте снимок (*snapshot*) **labfs** с помощью команды **stratis filesystem snapshot**. Выполнение команды может занять до минуты.

```
[root@serverb ~]# stratis filesystem snapshot labpool \
labfs labfs-snap
```

- 9.2.** Проверьте доступность снимка (*snapshot*) с помощью команды **stratis filesystem list**.

```
[root@serverb ~]# stratis filesystem list
...output omitted...
labpool  labfs-snap  2.53 GiB  Mar 29 2019 10:28  /stratis/labpool/
labfs-snap  291d...8a16
```

- 9.3.** Удалите файл **/labstratisvol/labfile1**.

```
[root@serverb ~]# rm /labstratisvol/labfile1
rm: remove regular file '/labstratisvol/labfile1'? y
```

- 9.4.** Создайте каталог **/labstratisvol-snap** с помощью команды **mkdir**.

```
[root@serverb ~]# mkdir /labstratisvol-snap
```

- 9.5.** Смонтируйте моментальный снимок (*snapshot*) **labfs-snap** в **/labstratisvol-snap** с помощью команды **mount**.

```
[root@serverb ~]# mount /stratis/labpool/labfs-snap \
/labstratisvol-snap
```

- 9.6.** Убедитесь, что вы все еще можете получить доступ к файлу, который вы удалили из **labfs**, с помощью моментального снимка (*snapshot*) **labfs-snap**.

```
[root@serverb ~]# cat /labstratisvol-snap/labfile1
Hello World!
```

- 10.** Создайте том **VDO** с именем **labvdo** с устройством **/dev/vdd**. Установите его логический размер на **50 ГБ**.

- 10.1.** Создайте том **labvdo** с помощью команды **vdo create**.

```
[root@serverb ~]# vdo create --name=labvdo --device=/dev/vdd --
vdoLogicalSize=50G
...output omitted...
```

10.2. Проверьте доступность тома **labvdo** с помощью команды **vdo list**.

```
[root@serverb ~]# vdo list
labvdo
```

11. Смонтируйте том **labvdo** в **/labvdovol** с файловой системой **XFS**, чтобы он сохранялся при перезагрузке. Не забудьте использовать параметр монтирования **x-systemd.requires=vdo.service** в файле **/etc/fstab**.

11.1. Отформатируйте том **labvdo** в файловой системе **XFS** с помощью команды **mkfs**.

```
[root@serverb ~]# mkfs.xfs -K /dev/mapper/labvdo
...output omitted...
```

11.2. Используйте команду **udevadm**, чтобы зарегистрировать новый узел устройства.

```
[root@serverb ~]# udevadm settle
```

11.3. Создайте каталог **/labvdovol** с помощью команды **mkdir**.

```
[root@serverb ~]# mkdir /labvdovol
```

11.4. Определите **UUID** **labvdo** с помощью команды **lsblk**.

```
[root@serverb ~]# lsblk --output=UUID /dev/mapper/labvdo
UUID
ef8cce71-228a-478d-883d-5732176b39b1
```

11.5. Отредактируйте файл **/etc/fstab** так, чтобы **labvdo** монтировался во время загрузки. Используйте **UUID** тома, который вы определили на предыдущем шаге. Ниже показана строка, которую вы должны добавить в файл **/etc/fstab**. Вы можете использовать команду **vi /etc/fstab** для редактирования файла.

```
UUID=ef8c...39b1 /labvdovol xfs defaults,x-systemd.requires=vdo.service 0 0
```

11.6. Смонтируйте том **labvdo** с помощью команды **mount**, чтобы убедиться, что файл **/etc/fstab** содержит соответствующие записи.

```
[root@serverb ~]# mount /labvdovol
```

12. Создайте три копии файла с именем **/root/install.img** на томе **labvdo**. Сравните статистику тома, чтобы проверить дедупликацию и сжатие данных, происходящие на томе.

- 12.1. Просмотрите начальную статистику и состояние тома с помощью команды **vdostats**.

```
[root@serverb ~]# vdostats --human-readable
```

Device	Size	Used	Available	Use%	Space saving%
/dev/mapper/labvdo	5.0G	3.0G	2.0G	60%	99%

Обратите внимание, что **3 ГБ** тома уже используются, потому что при создании том **VDO** резервирует **3–4 ГБ** для себя. Также обратите внимание, что значение **99 %** в поле «**Space saving**» указывает на то, что вы еще не создали никакого содержимого в томе, внося вклад во все сохраненное пространство тома.

- 12.2. Скопируйте **/root/install.img** в **/labvdovol/install.img.1** и проверьте статистику тома. Копирование файла может занять до минуты.

```
[root@serverb ~]# cp /root/install.img /labvdovol/install.img.1
```

```
[root@serverb ~]# vdostats --human-readable
```

Device	Size	Used	Available	Use%	Space saving%
/dev/mapper/labvdo	5.0G	3.4G	1.6G	68%	5%

Обратите внимание, что значение поля «**Used**» увеличилось с **3,0 ГБ** до **3,4 ГБ**, потому что вы скопировали файл в томе, который занимает некоторое пространство. Кроме того, обратите внимание, что значение поля «**Экономия места**» уменьшилось с **99 %** до **5 %**, потому что изначально в томе не было содержимого, что способствовало низкому использованию пространства тома и большой экономии места до тех пор, пока вы не создали там файл. Экономия места на томе довольно низкая, потому что вы создали уникальную копию файла на томе и нечего дедуплицировать.

- 12.3. Скопируйте **/root/install.img** в **/labvdovol/install.img.2** и проверьте статистику тома. Копирование файла может занять до минуты.

```
[root@serverb ~]# cp /root/install.img /labvdovol/install.img.2
```

```
[root@serverb ~]# vdostats --human-readable
```

Device	Size	Used	Available	Use%	Space saving%
/dev/mapper/labvdo	5.0G	3.4G	1.6G	68%	51%

Обратите внимание, что используемое пространство тома не изменилось. Вместо этого увеличился процент сохраненного пространства тома, что доказывает, что дедупликация данных была произведена для уменьшения потребления пространства для избыточных

копий одного и того же файла. Значение **% Space saving** в предыдущем выводе может различаться в вашей системе.

12.4. Перезагрузите сервер **serverb**.

```
[root@serverb ~]# systemctl reboot
```



ПРИМЕЧАНИЕ

Примечание. Если при перезагрузке **serverb** не загружается с обычным запросом на вход, а вместо этого выдает «Дайте пароль root для обслуживания (Give root password for maintenance) (или нажмите **Control-D**, чтобы продолжить):», вы, вероятно, допустили ошибку в файле **/etc/fstab**. После предоставления пароля **root redhat** вам нужно будет перемонтировать корневую файловую систему для чтения и записи с помощью команды:

```
[root@server ~]# mount -o remount,rw /
```

Убедитесь, что в файле **/etc/fstab** настроен правильно, как указано в решениях. Обратите особое внимание на параметры монтирования строк, относящихся к **/labstratisvol** и **/labvdovol**.

Оценка

На рабочей станции **workstation**, запустите скрипт **lab advstorage-review grade**, чтобы подтвердить успешность выполнения этого упражнения.

```
[student@workstation ~]$ lab advstorage-review grade
```

Завершение

На рабочей станции **workstation**, выполните скрипт **lab advstorage-review finish**, чтобы завершить данное упражнение. Этот сценарий удаляет разделы и файлы, созданные во время упражнения, и обеспечивает чистоту среды.

```
[student@workstation ~]$ lab advstorage-review finish
```

На этом лабораторная работа заканчивается.

РЕЗЮМЕ

В этой главе вы узнали:

- Решение по управлению хранением **Stratis** реализует гибкие файловые системы, которые динамично растут вместе с данными.
- Решение для управления хранилищем **Stratis** поддерживает тонкое выделение ресурсов, моментальные снимки и мониторинг.
- **Virtual Data Optimizer (VDO)** предназначен для снижения стоимости хранения данных.
- **Virtual Data Optimizer** применяет устранение нулевых блоков (zero-block), дедупликацию данных и сжатие данных для оптимизации использования дискового пространства.

ГЛАВА 9

ДОСТУП К СЕТЕВЫМ ХРАНИЛИЩАМ

ЦЕЛЬ

Доступ к сетевому хранилищу по протоколу NFS.

ЗАДАЧИ

- Монтируйте, используйте и отключайте экспорт NFS из командной строки и во время загрузки.
- Настройте средство автоматического монтирования с прямыми и непрямыми сопоставлениями для автоматического монтирования файловой системы NFS по запросу и отключения ее, когда она больше не используется.
- Настройте клиент NFS для использования NFSv4 с помощью нового инструмента nfsconf.

РАЗДЕЛЫ

- Монтирование сетевого хранилища с помощью NFS (и упражнения с пошаговыми инструкциями)
- Автомонтирование сетевого хранилища (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Доступ к сетевому хранилищу.

МОНТАЖ СЕТЕВОГО ХРАНИЛИЩА С NFS

ЦЕЛИ

После завершения этого раздела вы должны уметь:

- Идентифицировать информацию об общем доступе **NFS**.
- Создайте каталог для использования в качестве точки подключения.
- Смонтировать общий ресурс **NFS** с помощью команды **mount** или путем настройки файла **/etc/fstab**.
- Размонтировать общий ресурс **NFS** с помощью команды **umount**.
- Настройте клиент **NFS** для использования **NFSv4** с помощью нового инструмента **nfsconf**.

МОНТАЖ И ДЕМОНТИРОВАНИЕ ОБЩЕГО ОБЕСПЕЧЕНИЯ NFS

NFS, сетевая файловая система (Network File System), представляет собой стандартный интернет-протокол, используемый Linux, UNIX и аналогичными операционными системами в качестве собственной сетевой файловой системы. Это открытый стандарт, который все еще активно совершенствуется и поддерживает собственные разрешения Linux и функции файловой системы.

Версия **NFS** по умолчанию в Red Hat Enterprise Linux 8 — 4.2. Поддерживаются основные версии **NFSv4** и **NFSv3**. **NFSv2** больше не поддерживается. **NFSv4** использует для связи с сервером только протокол **TCP**; более ранние версии **NFS** могли использовать либо **TCP**, либо **UDP**.

Серверы NFS экспортят общие ресурсы (каталоги). **Клиенты NFS** монтируют экспортенный общий ресурс в локальную точку монтирования (каталог), которая должна существовать. Общие ресурсы NFS можно подключить несколькими способами:

- Вручную с помощью команды **mount**.
- Автоматически во время загрузки с использованием записей в файле **/etc/fstab**.
- По запросу с помощью службы **autofs** или средства **systemd.automount**.

Монтирование общих ресурсов NFS

Чтобы смонтировать общий ресурс NFS, выполните следующие три шага:

1. **Идентифицировать:** администратор клиентской системы **NFS** может идентифицировать доступные общие ресурсы **NFS** различными способами:

Администратор сервера **NFS** может предоставить сведения об экспорте, включая требования безопасности.

Кроме того, администратор клиента может определить **общие ресурсы NFSv4**, подключив корневой каталог сервера **NFS** и изучив экспортированные каталоги. Сделайте это как пользователь **root**. Доступ к общим ресурсам, использующим безопасность **Kerberos**, будет запрещен, но имя общего ресурса (каталога) будет видно. Другие общие каталоги будут доступны для просмотра.

```
[user@host ~]$ sudo mkdir mountpoint  
[user@host ~]$ sudo mount server:/ mountpoint  
[user@host ~]$ sudo ls mountpoint
```

2. **Точка монтирования (Mount point):** Используйте **mkdir** для создания точки монтирования в подходящем месте.

```
[user@host ~]$ mkdir -p mountpoint
```

3. **Подключить (Mount):** как и в случае с файловыми системами на разделах, общие ресурсы NFS должны быть смонтированы, чтобы они были доступны. Чтобы смонтировать общий ресурс NFS, выберите один из следующих вариантов. В каждом случае вы должны запускать эти команды как суперпользователь, либо войдя в систему как **root**, либо с помощью команды **sudo**.

- **Временное подключение:** подключение общего ресурса NFS с помощью команды **mount**:

```
[user@host ~]$ sudo mount -t nfs -o rw,sync server:/share mountpoint
```

Параметр **-t nfs**, это тип файловой системы для общих ресурсов NFS (не обязательно, но показан для полноты). Параметр **-o sync** указывает команде **mount** немедленно синхронизировать операции записи с сервером NFS (по умолчанию асинхронно).

Эта команда монтирует общий ресурс немедленно, но не постоянно; при следующей загрузке системы этот общий ресурс NFS будет недоступен. Это полезно для одноразового доступа к данным. Это также полезно для тестового подключения общего ресурса NFS, прежде чем сделать его постоянно доступным.

- **Постоянное монтирование:** чтобы общий ресурс NFS монтировался во время загрузки, отредактируйте файл **/etc/fstab**, добавив запись монтирования.

```
[user@host ~]$ sudo vim /etc/fstab  
...  
server:/share /mountpoint nfs rw,soft 0 0
```

Затем смонтируйте общий ресурс NFS:

```
[user@host ~]$ sudo mount /mountpoint
```

Поскольку сервер NFS и параметры монтирования находятся в файле **/etc/fstab** службой клиента NFS, вам не нужно указывать их в командной строке.

Размонтирование общих ресурсов NFS

От имени пользователя **root** (или с помощью **sudo**) размонтируйте общий ресурс **NFS** с помощью команды **umount**.

```
[user@host ~]$ sudo umount mountpoint
```

ИНСТРУМЕНТ nfsconf

Red Hat Enterprise Linux 8 представляет инструмент **nfsconf** для управления файлами конфигурации клиента и сервера **NFS** в **NFSv4** и **NFSv3**. Настройте инструмент **nfsconf**, используя **/etc/nfs.conf** (файл **/etc/sysconfig/nfs** из более ранних версий операционной системы теперь устарел). Используйте инструмент **nfsconf** для получения, установки или отмены параметров конфигурации **NFS**.

Файл конфигурации **/etc/nfs.conf** состоит из нескольких разделов, начинающихся с ключевого слова в квадратных скобках (**[ключевое слово]**) со значениями, присвоенными внутри раздела. Для сервера NFS настройте раздел **[nfsd]**. Присвоение значения или ключ состоит из имени значения, знака равенства и настройки значения, например **vers4.2=y**. Строки, начинающиеся с **"#"** или **";"** игнорируются, как и любые пустые строки.

```
[user@host ~]$ sudo cat /etc/nfs.conf
```

...output omitted...

[nfsd]

```
# debug=0
# threads=8
# host=
# port=0
# grace-time=90
# lease-time=90
# tcp=y
# vers2=n
# vers3=y
# vers4=y
# vers4.0=y
# vers4.1=y
# vers4.2=y
# rdma=n
#
```

По умолчанию пары ключ-значение раздела **[nfsd]** закомментированы. Однако в комментариях показаны параметры по умолчанию, которые вступят в силу, если их не изменить. Это дает вам хорошую отправную точку для настройки **NFS**.

Используйте **nfsconf --set section key value**, чтобы установить значение ключа в указанном разделе.

```
[user@host ~]$ sudo nfsconf --set nfsd vers4.2 y
```

Эта команда обновляет файл конфигурации **/etc/nfs.conf**:

```
[user@host ~]$ sudo cat /etc/nfs.conf
...output omitted...
[nfsd]
vers4.2 = y
# debug=0
# threads=8
# host=
# port=0
# grace-time=90
# lease-time=90
# tcp=y
# vers2=n
# vers3=y
# vers4=y
# vers4.0=y
# vers4.1=y
# vers4.2=y
# rdma=n
#
```

Используйте ключ раздела **nfsconf --get section key**, чтобы получить значение ключа в указанном разделе:

```
[user@host ~]$ sudo nfsconf --get nfsd vers4.2
y
```

Используйте **nfsconf --unset section key**, чтобы отменить значение ключа в указанном разделе:

```
[user@host ~]$ sudo nfsconf --unset nfsd vers4.2
```

Настройка клиента только для NFSv4

Вы можете настроить клиента только для **NFSv4**, установив следующие значения в файле конфигурации **/etc/nfs.conf**.

Начните с отключения **UDP** и других ключей, связанных с **NFSv2** и **NFSv3**:

```
[user@host ~]$ sudo nfsconf --set nfsd udp n  
[user@host ~]$ sudo nfsconf --set nfsd vers2 n  
[user@host ~]$ sudo nfsconf --set nfsd vers3 n
```

Включите связанные ключи **TCP** и **NFSv4**.

```
[user@host ~]$ sudo nfsconf --set nfsd tcp y  
[user@host ~]$ sudo nfsconf --set nfsd vers4 y  
[user@host ~]$ sudo nfsconf --set nfsd vers4.0 y  
[user@host ~]$ sudo nfsconf --set nfsd vers4.1 y  
[user@host ~]$ sudo nfsconf --set nfsd vers4.2 y
```

Как и прежде, изменения появляются в конфигурационном файле **/etc/nfs.conf**:

```
[[user@host ~]$ cat /etc/nfs.conf  
[nfsd]  
udp = n  
vers2 = n  
vers3 = n  
tcp = y  
vers4 = y  
vers4.0 = y  
vers4.1 = y  
vers4.2 = y
```



РЕКОМЕНДАЦИИ

Справочные страницы **man mount(8)**, **umount(8)**, **fstab(5)**, **mount.nfs(8)**, **nfs.conf(8)** и **nfsconf(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ СЕТЕВЫМИ ХРАНИЛИЩАМИ С ПОМОЩЬЮ NFS

КОНТРОЛЬНЫЙ СПИСОК ПОКАЗАТЕЛЕЙ ЭФФЕКТИВНОСТИ

В этом упражнении вы измените файл **/etc/fstab** для постоянного монтирования экспорта **NFS** во время загрузки.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Протестировать сервер **NFS** с помощью команды **mount**.
- Настроить общие ресурсы **NFS** в файле конфигурации **/etc/fstab**, чтобы сохранить изменения даже после перезагрузки системы.
- Настройте клиенты **NFS** для использования **NFSv4** с помощью нового инструмента **nfsconf**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab netstorage-nfs start**. Эта команда запускает сценарий, который определяет, доступны ли компьютеры **servera** и **serverb** в сети. Скрипт предупредит вас, если это не так. Сценарий запуска настраивает **serverb** как сервер **NFSv4**, устанавливает разрешения и экспортирует каталоги. Он создает пользователей и группы, необходимые как на **servera**, так и на **serverb**.

```
[student@workstation ~]$ lab netstorage-nfs start
```

Транспортная компания использует центральный сервер **serverb** для размещения ряда общих документов и каталогов. Пользователям **servera**, которые все являются членами группы **admin**, необходим доступ к постоянно подключенному общему ресурсу **NFS**.

Важная информация:

- **serverb** разделяет каталог **/shares/public**, который содержит несколько текстовых файлов.
- Члены группы **admin** (**admin1**, **sysmanager1**) имеют доступ на чтение и запись к общему каталогу **/shares/public**.
- Принципиальная точка монтирования **servera** — **/public**.
- Все пароли пользователей установлены на **redhat**.

1. Войдите на **servera**, как пользователь **student** и переключитесь на пользователя **root**.

1.1. Войдите на сервер пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

- 1.2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Пароль пользователя **student** - **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

2. Используйте инструмент **nfsconf** для настройки **/etc/nfs.conf**, чтобы клиенты **NFS** могли работать только в версии **4.X** и чтобы убедиться, что режим **TCP** включен, а режим **UDP** отключен.

- 2.1. Используйте инструмент **nfsconf**, чтобы отключить ключи **udp**, **vers2**, **vers3**.

```
[root@servera ~]# nfsconf --set nfssd udp n  
[root@servera ~]# nfsconf --set nfssd vers2 n  
[root@servera ~]# nfsconf --set nfssd vers3 n
```

- 2.2. Используйте инструмент **nfsconf** для включения ключей **tcp**, **vers4**, **vers4.0**, **vers4.1**, **vers4.2**.

```
[root@servera ~]# nfsconf --set nfssd tcp y  
[root@servera ~]# nfsconf --set nfssd vers4 y  
[root@servera ~]# nfsconf --set nfssd vers4.0 y  
[root@servera ~]# nfsconf --set nfssd vers4.1 y  
[root@servera ~]# nfsconf --set nfssd vers4.2 y
```

3. Протестируйте сервер **NFS** на **serverb**, используя **servera** в качестве клиента **NFS**.

- 3.1. Создайте точку монтирования **/public** на сервере **servera**.

```
[root@servera ~]# mkdir /public
```

- 3.2. На **servera** используйте команду **mount**, чтобы убедиться, что общий ресурс **NFS** **/share/public**, экспортенный **serverb**, правильно монтируется в точке монтирования **/public**.

```
[root@servera ~]# mount -t nfs serverb.lab.example.com:/shares/  
public /public
```

3.3. Перечислите содержимое смонтированного общего ресурса **NFS**.

```
[root@servera ~]# ls -l /public  
total 16  
-rw-r--r--. 1 root admin 42 Apr  8 22:36 Delivered.txt  
-rw-r--r--. 1 root admin 46 Apr  8 22:36 NOTES.txt  
-rw-r--r--. 1 root admin 20 Apr  8 22:36 README.txt  
-rw-r--r--. 1 root admin 27 Apr  8 22:36 Trackings.txt
```

3.4. Изучите параметры подключения для подключенного общего ресурса NFS.

```
[root@servera ~]# mount | grep public  
serverb.lab.example.com:/shares/public on /public type nfs4  
(rw,relatime,vers=4.2,rsize=262144,wsize=262144,namlen=255,sync,proto=tcp,timeo=600,  
retrans=2,sec=sys,clientaddr=172.25.250.10,local_lock=none,addr=172.25.250.11)
```

3.5. Размонтируйте общий ресурс NFS.

```
[root@servera ~]# umount /public
```

4. Настройте **servera**, чтобы обеспечить постоянное подключение используемого выше общего ресурса.

4.1. Откройте файл **/etc/fstab** для редактирования.

```
[root@servera ~]# vim /etc/fstab
```

Добавьте следующую строку в конец файла:

```
serverb.lab.example.com:/shares/public /public nfs rw,sync 0 0
```

4.2. Используйте команду **mount** для монтирования общего каталога.

```
[root@servera ~]# mount /public
```

4.3. Перечислите содержимое общего каталога.

```
[root@servera ~]# ls -l /public  
total 16
```

4.4. Перезагрузите сервер **servera**.

```
[root@servera ~]# systemctl reboot
```

5. После того, как **servera** завершит перезагрузку, войдите в **servera** как пользователь **admin1** и проверьте постоянно подключенный общий ресурсc **NFS**.

5.1. Войдите на сервер **servera** как пользователь **admin1**.

```
[student@workstation ~]$ ssh admin1@servera  
[admin1@servera ~]$
```

5.2. Протестируйте общий ресурсc **NFS**, смонтированный на **/public**.

```
[admin1@servera ~]$ ls -l /public  
total 16  
-rw-r--r--. 1 root    admin 42 Apr  8 22:36 Delivered.txt  
-rw-r--r--. 1 root    admin 46 Apr  8 22:36 NOTES.txt  
-rw-r--r--. 1 root    admin 20 Apr  8 22:36 README.txt  
-rw-r--r--. 1 root    admin 27 Apr  8 22:36 Trackings.txt  
[admin1@servera ~]$ cat /public/NOTES.txt  
###In this file you can log all your notes###  
[admin1@servera ~]$ echo "This is a test" > /public/Test.txt  
[admin1@servera ~]$ cat /public/Test.txt  
This is a test
```

5.3. Выйдите из сервера **servera**.

```
[admin1@servera ~]$ exit  
logout  
Connection to servera closed.
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab netstorage-nfs finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab netstorage-nfs finish
```

На этом упражнения с пошаговыми инструкциями заканчивается.

АВТОМОНТАЖ СЕТЕВОГО ХРАНИЛИЩА

ЦЕЛИ

После заполнения этого раздела вы должны уметь:

- Описать преимущества использования авто монтирования.
- Автоматическое монтируние общих ресурсов **NFS** с использованием прямых и непрямых сопоставлений, включая подстановочные знаки.

МОНТАЖ СОВМЕСТНОГО ИСПОЛЬЗОВАНИЯ РЕСУРСА NFS С ПОМОЩЬЮ AUTOMOUNTER

Automounter — это служба (**autofs**), которая автоматически монтирует общие ресурсы **NFS** «по запросу (**on-demand**)» и автоматически отключает общие ресурсы **NFS**, когда они больше не используются.

Преимущества *Automounter*

- Пользователям не обязательно иметь привилегии **root** для выполнения команд **mount** и **umount**.
- Общие ресурсы **NFS**, настроенные в программе автоматического монтируния, доступны для всех пользователей на машине при наличии разрешений на доступ.
- Общие ресурсы **NFS** не связаны постоянно, как записи в файле **/etc/fstab**, освобождая сетевые и системные ресурсы.
- **Automounter** настраивается на стороне клиента; настройка на стороне сервера не требуется.
- Средство **Automounter** использует те же параметры, что и команда **mount**, включая параметры безопасности.
- Средство **Automounter** поддерживает как прямое, так и косвенное сопоставление точек монтируния, что обеспечивает гибкость в расположении точек монтируния.
- **autofs** создает и удаляет непрямые точки монтируния, исключая ручное управление.
- **NFS** является сетевой файловой системой автоматического монтируния по умолчанию, но другие сетевые файловые системы могут монтироваться автоматически.
- **autofs** — это служба, которая управляется так же, как и другие системные службы.

Создать авто монтирование

Настройка автоматического монтируния — это многоэтапный процесс:

1. Установите пакет **autofs**.

```
[user@host ~]$ sudo yum install autofs
```

Этот пакет содержит все необходимое для использования автоматического монтирования общих ресурсов **NFS**.

2. Добавьте файл *master map* в `/etc/auto.master.d`. Этот файл определяет базовый каталог, используемый для точек монтирования и определяет файл сопоставления, используемый для создания автомонтирований.

```
[user@host ~]$ sudo vim /etc/auto.master.d/demo.autofs
```

Имя файла master map может быть произвольным (хотя обычно осмысленным), но оно должно иметь расширение **.autofs**, чтобы подсистема могла его распознать. Вы можете разместить несколько записей в одном файле master map; в качестве альтернативы вы можете создать несколько файлов мастер-карт, каждый из которых будет иметь свои записи, логически сгруппированные.

Добавьте запись мастер-карты, в данном случае, для косвенно сопоставленных монтирований:

```
/shares    /etc/auto.demo
```

Эта запись использует каталог **/shares** в качестве основы для непрямого авто монтирования. Файл **/etc/auto.demo** содержит сведения о монтировании. Используйте абсолютное имя файла. Файл **auto.demo** необходимо создать перед запуском службы **autofs**.

3. Создайте файлы сопоставления (mapping). Каждый файл сопоставления определяет точку монтирования, параметры монтирования и исходное расположение для монтирования набора автомонтирований.

```
[user@host ~]$ sudo vim /etc/auto.demo
```

Соглашение об именовании файлов карты — **/etc/auto.name**, где *name* отражает содержимое карты.

```
work      -rw,sync      serverb:/shares/work
```

Формат записи: точка монтирования (*mount point*), параметры монтирования (*mount options*) и исходное местоположение (*source location*). В этом примере показана базовая запись косвенного сопоставления. Прямые сопоставления и косвенные сопоставления с использованием подстановочных знаков рассматриваются далее в этом разделе.

- Точка монтирования, известная как ключ (*key*) на справочных страницах, автоматически создается и удаляется службой **autofs**. В этом случае полной точкой монтирования является **/shares/work** (см. основной файл сопоставления). Каталоги **/shares** и **/shares/work** создаются и удаляются по мере необходимости службой **autofs**.

В этом примере локальная точка монтирования отражает структуру каталогов сервера, однако это не обязательно; локальная точка монтирования может называться, как угодно. Служба **autofs** не применяет к клиенту определенную структуру именования.

- Параметры монтирования начинаются с дефиса (-) и разделяются запятыми без пробелов. Параметры монтирования, доступные для ручного монтирования файловой системы, доступны при автоматическом монтировании. В этом примере авто монтиrovщик монтирует общий ресурс с доступом для чтения/записи (**read/write**) (параметр **rw**), а сервер синхронизируется сразу во время операций записи (параметр синхронизации (**sync**)).

Полезные параметры авто монтирования включают **-fstype=** и **-strict**. Используйте **fstype** для указания типа файловой системы, например, **nfs4** или **xfs**, и используйте **strict** для обработки ошибок при монтировании файловых систем как фатальных.

- Исходное расположение общих ресурсов **NFS** соответствует шаблону **host:/pathname**; в этом примере **serverb:/shares/work**. Для успешного автоматического монтирования сервер **NFS** должен экспортить каталог с доступом для чтения/записи, а пользователь, запрашивающий доступ, должен иметь стандартные права доступа к файлам Linux в каталоге. Если **serverb** экспортирует каталог с доступом только для чтения, то клиент получит доступ только для чтения, даже если он запросил доступ для чтения/записи.

4. Запустите и включите службу автоматического монтирования. Используйте **systemctl** для запуска и включения службы **autofs**.

```
[user@host ~]$ sudo systemctl enable --now autofs
Created symlink /etc/systemd/system/multi-user.target.wants/autofs.service → /
usr/lib/systemd/system/autofs.service.
```

Прямые сопоставления (Maps)

Прямые сопоставления используются для сопоставления общего ресурса **NFS** с существующей точкой монтирования с абсолютным путем.

Чтобы использовать непосредственно сопоставленные точки монтирования, файл **master map** может выглядеть следующим образом:

```
/-      /etc/auto.direct
```

Все записи прямого сопоставления используют **/-** в качестве базового каталога. В этом случае файл сопоставления, содержащий сведения о монтировании, называется **/etc/auto.direct**.

Содержимое файла **/etc/auto.direct** может выглядеть следующим образом:

```
/mnt/docs    -rw, sync      serverb:/shares/docs
```

Точка монтирования (или ключ (**key**)) всегда является абсолютным путем. Остальная часть файла сопоставления использует ту же структуру.

В этом примере каталог **/mnt** существует и не управляется **autofs**. Полный каталог **/mnt/docs** будет автоматически создан и удален службой **autofs**.

Использование подстановочных знаков в создание косвенных сопоставлений

Когда сервер **NFS** экспортирует несколько подкаталогов внутри каталога, то авто монтирование может быть настроено для доступа к любому из этих подкаталогов с помощью одной записи сопоставления.

Продолжая предыдущий пример, если **serverb:/shares** экспортирует два или более подкаталогов, и они доступны с использованием одних и тех же параметров монтирования, содержимое файла **/etc/auto.demo** может выглядеть следующим образом:

```
* -rw,sync serverb:/shares/&
```

Точка монтирования (или **key**) — это символ звездочки (*), а подкаталог в исходном расположении — это символ амперсанда (&). Все остальное в записи то же самое.

Когда пользователь пытается получить доступ к **/shares/work**, ключ * (в данном примере это **work**) заменяет амперсанд в исходном местоположении, и монтируется **serverb:/shares/work**. Как и в косвенном примере, рабочий каталог создается и удаляется автоматически с помощью **autofs**.



РЕКОМЕНДАЦИИ

Справочные страницы **man autofs(5)**, **automount(8)**, **auto.master(5)**, и **mount.nfs(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

АВТОМОНТАЖ СЕТЕВОГО ХРАНИЛИЩА

КОНТРОЛЬНЫЙ СПИСОК УПРАЖНЕНИЯ

В этом упражнении вы создадите точки монтирования с прямым и косвенным сопоставлением, управляемые автоматическим монтированием, для монтирования файловой системы **NFS**.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Установить необходимые пакеты для автомонтирования.
- Настроить сопоставление прямого и непрямого автоматического монтирования, получая ресурсы с предварительно настроенного сервера **NFSv4**.
- Понимать разницу между прямого и непрямого сопоставления автомонтирования.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab netstorage-autofs start**. Этот сценарий запуска определяет, доступны ли серверы **servera** и **serverb** в сети. Скрипт предупредит вас, если это не так. Сценарий запуска настраивает **serverb** как сервер **NFSv4**, устанавливает разрешения и экспортирует каталоги. Он также создает пользователей и группы, необходимые как на **servera**, так и на **serverb**.

```
[student@workstation ~]$ lab netstorage-autofs start
```

Интернет-провайдер использует центральный сервер **serverb** для размещения общих каталогов, содержащих важные документы, которые должны быть доступны по запросу. Когда пользователи входят на сервер **servera**, им требуется доступ к автоматически монтируемым общим каталогам.

Важная информация:

- **serverb** экспортирует как общий ресурс **NFS** каталог **/shares/indirect**, который, в свою очередь, содержит подкаталоги **west**, **center** и **east**.
- **serverb** также экспортирует как общий ресурс **NFS** каталог **/shares/direct/external**.
- Группа **operators** состоит из пользователей **operator1** и **operator2**. У них есть доступ для чтения и записи к общим каталогам **/shares/indirect/west**, **/shares/indirect/central** и **/shares/indirect/east**.

- Группа **contractors** состоит из пользователей **contractor1** и **contractor2**. У них есть доступ для чтения и записи к общему каталогу **/shares/direct/external**.
- Ожидаемые точки монтирования для сервера **servera**: **/external** и **/internal**.
- Общий каталог **/shares/direct/external** должен автоматически монтироваться на сервере **servera** с использованием прямого сопоставления на **/external**.
- Общий каталог **/shares/indirect/west** должен автоматически монтироваться на сервере **servera** с использованием непрямой сопоставления на **/internal/west**.
- Общий каталог **/shares/indirect/central** должен автоматически монтироваться на серверах с использованием непрямого сопоставления с **/internal/central**.
- Общий каталог **/shares/indirect/east** должен автоматически монтироваться на сервере **servera** с использованием непрямого сопоставления на **/internal/east**.
- Все пароли пользователей установлены как слово **redhat**.

1. Войдите на сервер **servera** и установите необходимые пакеты.

1.1. Войдите на сервер **servera**, как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

1.2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Пароль для пользователя **student** - **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

1.3. Установите пакет **autofs**.

```
[root@servera ~]# yum install autofs
...output omitted...
Is this ok [y/N]: y
...output omitted...
```

2. Настройте прямое сопоставление автоматического монтирования на **servera**, используя общие ресурсы с **serverb**. Создайте прямое сопоставление, используя файлы с именем **/etc/auto.master.d/direct.autofs** для **master map** и **/etc/auto.direct** для файла сопоставления. Используйте каталог **/external** в качестве основной точки монтирования на сервере **servera**.

2.1. Протестируйте сервер **NFS** и общий доступ, прежде чем приступать к настройке авто монтирования.

```
[root@servera ~]# mount -t nfs \
serverb.lab.example.com:/shares/direct/external /mnt
[root@servera ~]# ls -l /mnt
```

```
total 4  
-rw-r--r--. 1 root contractors 22 Apr 7 23:15 README.txt  
[root@servera ~]# umount /mnt
```

- 2.2.** Создайте файл **master map** с именем **/etc/auto.master.d/direct.autofs**, вставьте следующее содержимое и сохраните изменения.

```
[root@servera ~]# vim /etc/auto.master.d/direct.autofs  
/- /etc/auto.direct
```

- 2.3.** Создайте файл прямого сопоставления с именем **/etc/auto.direct**, вставьте следующее содержимое и сохраните изменения.

```
[root@servera ~]# vim /etc/auto.direct  
/external -rw,sync,fstype=nfs4 serverb.lab.example.com:/shares/direct/  
external
```

- 3.** Настройте непрямое сопоставление автоматического монтирования на **servera**, используя общие ресурсы с **serverb**. Создайте непрямое сопоставление, используя файлы с именем **/etc/auto.master.d/indirect.autofs** для master map и **/etc/auto.indirect** для файла сопоставления. Используйте каталог **/internal** в качестве основной точки монтирования на сервере **servera**.

- 3.1.** Протестируйте сервер NFS и общий доступ, прежде чем приступать к настройке авто монтирования.

```
[root@servera ~]# mount -t nfs serverb.lab.example.com:/shares/  
indirect /mnt  
[root@servera ~]# ls -l /mnt  
total 0  
drwxrws---. 2 root operators 24 Apr 7 23:34 central  
drwxrws---. 2 root operators 24 Apr 7 23:34 east  
drwxrws---. 2 root operators 24 Apr 7 23:34 west  
[root@servera ~]# umount /mnt
```

- 3.2.** Создайте файл master map с именем **/etc/auto.master.d/indirect.autofs**, вставьте следующее содержимое и сохраните изменения.

```
[root@servera ~]# vim /etc/auto.master.d/indirect.autofs  
/internal /etc/auto.indirect
```

- 3.3.** Создайте файл непрямого сопоставления с именем **/etc/auto.indirect**, вставьте следующее содержимое и сохраните изменения.

```
[root@servera ~]# vim /etc/auto.indirect
* -rw,sync,fstype=nfs4 serverb.lab.example.com:/shares/indirect/&
```

4. Запустите службу **autofs** на сервере **servera** и включите ее автоматический запуск во время загрузки. Перезагрузите сервер, чтобы определить, запускается ли служба autofs автоматически.

4.1. Запустите и включите службу **autofs** на сервере **servera**.

```
[root@servera ~]# systemctl enable --now autofs
Created symlink /etc/systemd/system/multi-user.target.wants/
autofs.service → /usr/lib/systemd/system/autofs.service.
```

4.2. Перезагрузите сервер **servera**.

```
[root@servera ~]# systemctl reboot
```

5. Протестируйте прямое сопоставление авто монтирования под пользователем **contract1**. Когда закончите, выйдите из сеанса пользователя **contract1** на сервере **servera**.

5.1. После того, как машина **servera** завершит загрузку, войдите на сервер как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

5.2. Переключитесь на пользователя **contract1**.

```
[student@servera ~]$ su - contractor1
Password: redhat
```

5.3. Укажите точку монтирования **/external**.

```
[contractor1@servera ~]$ ls -l /external
total 4
-rw-r--r--. 1 root contractors 22 Apr    7 23:34 README.txt
```

5.4. Просмотрите содержимое и проверьте доступ к точке подключения **/external**.

```
[contractor1@servera ~]$ cat /external/README.txt
###External Folder###
[contractor1@servera ~]$ echo testing-direct > /external/testing.txt
[contractor1@servera ~]$ cat /external/testing.txt
testing-direct
```

5.5. Выйдите из пользовательской сессии **contract1**.

```
[contractor1@servera ~]$ exit
Logout
```

6. Протестируйте непрямое сопоставление авто монтирования от имени пользователя **operator1**. Когда закончите, выйдите из сервера **servera**.

6.1. Переключитесь на пользователя **operator1**.

```
[student@servera ~]$ su - operator1
Password: redhat
```

6.2. Укажите точку монтирования **/internal**.

```
[operator1@servera ~]$ ls -l /internal
total 0
```



ПРИМЕЧАНИЕ

Вы заметите, что в непрямом сопоставлении автоматического монтирования, даже если вы находитесь в сопоставленной точке монтирования, вам нужно вызывать каждый из общих подкаталогов или файлов по запросу, чтобы получить к ним доступ. В прямой сопоставлении автоматического монтирования после открытия сопоставленной точки монтирования вы получаете доступ к каталогам и содержимому, настроенному в общем каталоге.

6.3. Проверьте доступ к общему каталогу **/internal/west automounter**.

```
[operator1@servera ~]$ ls -l /internal/west/
total 4
-rw-r--r--. 1      root    operators 18 Apr     7 23:34 README.txt
[operator1@servera ~]$ cat /internal/west/README.txt
###West Folder###
[operator1@servera ~]$ echo testing-1 > /internal/west/testing-1.txt
```

```
[operator1@servera ~]$ cat /internal/west/testing-1.txt  
testing-1  
[operator1@servera ~]$ ls -l /internal  
total 0  
drwxrws---. 2 root operators 24 Apr 7 23:34 west
```

6.4. Проверьте доступ к общему каталогу **/internal/central automounter**.

```
[operator1@servera ~]$ ls -l /internal/central  
total 4  
-rw-r--r--. 1 root operators 21 Apr 7 23:34 README.txt  
[operator1@servera ~]$ cat /internal/central/README.txt  
###Central Folder###  
[operator1@servera ~]$ echo testing-2 > /internal/central/  
testing-2.txt  
[operator1@servera ~]$ cat /internal/central/testing-2.txt  
testing-2  
[operator1@servera ~]$ ls -l /internal  
total 0  
drwxrws---. 2 root operators 24 Apr 7 23:34 central  
drwxrws---. 2 root operators 24 Apr 7 23:34 west
```

6.5. Проверьте доступ к общему каталогу авто монтирования **/internal/east**.

```
[operator1@servera ~]$ ls -l /internal/east  
total 4  
-rw-r--r--. 1 root operators 18 Apr 7 23:34 README.txt  
[operator1@servera ~]$ cat /internal/east/README.txt  
###East Folder###  
[operator1@servera ~]$ echo testing-3 > /internal/east/testing-3.txt  
[operator1@servera ~]$ cat /internal/east/testing-3.txt  
testing-3  
[operator1@servera ~]$ ls -l /internal  
total 0  
drwxrws---. 2 root operators 24 Apr 7 23:34 central  
drwxrws---. 2 root operators 24 Apr 7 23:34 east  
drwxrws---. 2 root operators 24 Apr 7 23:34 west
```

6.6. Проверьте доступ к общему каталогу **/external**.

```
[operator1@servera ~]$ ls -l /external  
ls: cannot open directory '/external': Permission denied
```

6.7. Выйдите из сервера **servera**.

```
[operator1@servera ~]$ exit  
logout  
[student@servera ~]$ exit  
logout  
Connection to servera closed.
```

Завершение

На рабочей станции **workstation** запустите сценарий **zlab netstorage-autofs finish**, чтобы завершить данное упражнение.

```
[student@workstation ~]$ lab netstorage-autofs finish
```

На этом упражнения с пошаговыми инструкциями заканчивается.

ЛАБОРАТОРНАЯ РАБОТА

ДОСТУП К СЕТЕВОМУ ХРАНИЛИЩУ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы настроите средство автоматического монтирования с непрямым сопоставлением, используя общие ресурсы с сервера **NFSv4**.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Установить необходимые пакеты, для настройки авто монтирования.
- Настроить непрямое сопоставления автоматического монтирования, получая ресурсы с предварительно настроенного сервера **NFSv4**.
- Настроить клиент **NFS** для использования **NFSv4** с помощью инструмента **nfsconf**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab netstorage-review start**. Этот сценарий определяет, доступны ли системы **servera** и **serverb** в сети. Сценарий запуска настраивает **serverb** как сервер **NFSv4**, устанавливает разрешения и экспортирует каталоги. Он также создает необходимых пользователей и группы как в системе **servera**, так и в системе **serverb**.

```
[student@workstation ~]$ lab netstorage-review start
```

Компания ИТ-поддержки использует центральный сервер **serverb** для размещения некоторых общих каталогов в **/remote/shares** для своих групп и пользователей. Пользователи должны иметь возможность войти в систему и иметь свои общие каталоги, смонтированные по запросу и готовые к использованию, в каталоге **/shares** на сервере.

Важная информация:

- На **serverb** открыт для совместно использует каталог **/shares**, который, в свою очередь, содержит подкаталоги **management**, **production** и **operation**.
- Группа **management** состоит из пользователей **manager1** и **manager2**. У них есть доступ для чтения и записи к общему каталогу **/shares/management**.
- Группа **production** состоит из пользователей **dbuser1** и **sysadmin1**. У них есть доступ для чтения и записи к общему каталогу **/shares/production**.
- Группа **operation** состоит из пользователей **contractor1** и **consultant1**. У них есть доступ для чтения и записи к общему каталогу **/shares/operation**.

- Основной точкой подключения сервера **servera** является каталог **/remote**.
- Общий каталог **/shares/management** должен быть автоматически подключен к **/remote/management** на сервере **servera**.
- Общий каталог **/shares/production** должен автоматически монтироваться на **/remote/production** на сервере **servera**.
- Общий каталог **/shares/operation** должен автоматически монтироваться на **/remote/operation** на сервере **servera**.
- Все пароли пользователей установлены как слово **redhat**.

1. Войдите на сервер **servera** и установите необходимые пакеты.
2. Используйте команду **nfsconf** для настройки **/etc/nfs.conf**. Включите работу клиента **NFS** только в версии **4.X** и убедитесь, что режим **TCP** включен, а режим **UDP** отключен.
3. Настройте косвенное сопоставление автоматического монтирования на **servera**, используя общие ресурсы с **serverb**. Создайте косвенное сопоставление, используя файлы с именем **/etc/auto.master.d/shares.autofs** для основной карты и **/etc/auto.shares** для файла сопоставления. Используйте каталог **/remote** в качестве основной точки монтирования на сервере **servera**. Перезагрузите сервер **servera**, чтобы определить, запускается ли служба **autofs** автоматически.
4. Протестируйте конфигурацию **autofs** с разными пользователями. Когда закончите, выйдите из сервера.

Оценка

На рабочей станции **workstation** запустите команду **lab netstorage-review grade**, чтобы подтвердить успешность выполнения данного упражнения.

```
[student@workstation ~]$ lab netstorage-review grade
```

Завершение

На рабочей станции **workstation** запустите команду **lab netstorage-review finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab netstorage-review finish
```

На этом лабораторная работа заканчивается.

РЕШЕНИЕ

ДОСТУП К СЕТЕВОМУ ХРАНИЛИЩУ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы настроите средство автоматического монтирования с непрямым сопоставлением, используя общие ресурсы с сервера **NFSv4**.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Установить необходимые пакеты, для настройки авто монтирования.
- Настроить непрямое сопоставления автоматического монтирования, получая ресурсы с предварительно настроенного сервера **NFSv4**.
- Настроить клиент **NFS** для использования **NFSv4** с помощью инструмента **nfsconf**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab netstorage-review start**. Этот сценарий определяет, доступны ли системы **servera** и **serverb** в сети. Сценарий запуска настраивает **serverb** как сервер **NFSv4**, устанавливает разрешения и экспортирует каталоги. Он также создает необходимых пользователей и группы как в системе **servera**, так и в системе **serverb**.

```
[student@workstation ~]$ lab netstorage-review start
```

Компания ИТ-поддержки использует центральный сервер **serverb** для размещения некоторых общих каталогов в **/remote/shares** для своих групп и пользователей. Пользователи должны иметь возможность войти в систему и иметь свои общие каталоги, смонтированные по запросу и готовые к использованию, в каталоге **/shares** на сервере.

Важная информация:

- На **serverb** открыт для совместно использует каталог **/shares**, который, в свою очередь, содержит подкаталоги **management**, **production** и **operation**.
- Группа **management** состоит из пользователей **manager1** и **manager2**. У них есть доступ для чтения и записи к общему каталогу **/shares/management**.
- Группа **production** состоит из пользователей **dbuser1** и **sysadmin1**. У них есть доступ для чтения и записи к общему каталогу **/shares/production**.
- Группа **operation** состоит из пользователей **contractor1** и **consultant1**. У них есть доступ для чтения и записи к общему каталогу **/shares/operation**.

- Основной точкой подключения сервера **servera** является каталог **/remote**.
- Общий каталог **/shares/management** должен быть автоматически подключен к **/remote/management** на сервере **servera**.
- Общий каталог **/shares/production** должен автоматически монтироваться на **/remote/production** на сервере **servera**.
- Общий каталог **/shares/operation** должен автоматически монтироваться на **/remote/operation** на сервере **servera**.
- Все пароли пользователей установлены как слово **redhat**.

1. Войдите на сервер **servera** и установите необходимые пакеты.

1.1. Войдите на сервер как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

1.2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Пароль для пользователя **student** - **student**.

```
[student@servera ~]$ sudo -i
[sudo] password for student: student
[root@servera ~]#
```

1.3. Установите пакет **autofs**.

```
[root@servera ~]# yum install autofs
...output omitted...
Is this ok [y/N]: y
...output omitted...
```

2. Используйте команду **nfsconf** для настройки **/etc/nfs.conf**. Включите работу клиента NFS только в версии **4.X** и убедитесь, что режим **TCP** включен, а режим **UDP** отключен.

2.1. Используйте инструмент **nfsconf**, чтобы отключить ключи **udp**, **vers2**, **vers3**.

```
[root@servera ~]# nfsconf --set nfsd udp n
[root@servera ~]# nfsconf --set nfsd vers2 n
[root@servera ~]# nfsconf --set nfsd vers3 n
```

2.2. Используйте инструмент **nfsconf** для включения ключей **tcp**, **vers4**, **vers4.0**, **vers4.1**, **vers4.2**.

```
[root@servera ~]# nfsconf --set nfsd tcp y
[root@servera ~]# nfsconf --set nfsd vers4 y
[root@servera ~]# nfsconf --set nfsd vers4.0 y
[root@servera ~]# nfsconf --set nfsd vers4.1 y
[root@servera ~]# nfsconf --set nfsd vers4.2 y
```

3. Настройте косвенное сопоставление автоматического монтирования на **servera**, используя общие ресурсы с **serverb**. Создайте косвенное сопоставление, используя файлы с именем **/etc/auto.master.d/shares.autofs** для основной карты и **/etc/auto.shares** для файла сопоставления. Используйте каталог **/remote** в качестве основной точки монтирования на сервере **servera**. Перезагрузите сервер **servera**, чтобы определить, запускается ли служба **autofs** автоматически.

3.1. Протестируйте сервер NFS перед тем, как приступить к настройке автомонтирования.

```
[root@servera ~]# mount -t nfs serverb.lab.example.com:/shares /mnt
[root@servera ~]# ls -l /mnt
total 0
drwxrwx---. 2 root managers      25 Apr    4 01:13 management
drwxrwx---. 2 root operators     25 Apr    4 01:13 operation
drwxrwx---. 2 root production   25 Apr    4 01:13 production
[root@servera ~]# umount /mnt
```

3.2. Создайте файл **master map** с именем **/etc/auto.master.d/shares.autofs**, вставьте следующее содержимое и сохраните изменения.

```
[root@servera ~]# vim /etc/auto.master.d/shares.autofs
/remote /etc/auto.shares
```

3.3. Создайте файл непрямого сопоставления с именем **/etc/auto.shares**, вставьте следующее содержимое и сохраните изменения.

```
[root@servera ~]# vim /etc/auto.shares
* -rw,sync,fstype=nfs4 serverb.lab.example.com:/shares/&
```

3.4. Запустите и включите службу **autofs** на сервере.

```
[root@servera ~]# systemctl enable --now autofs
Created symlink /etc/systemd/system/multi-user.target.wants/
autofs.service → /usr/lib/systemd/system/autofs.service.
```

3.5. Перезагрузите серверную машину **servera**.

```
[root@servera ~]# systemctl reboot
```

4. Протестируйте конфигурацию **autofs** с разными пользователями. Когда закончите, выйдите из сервера.

- 4.1. После завершения загрузки компьютера **servera** войдите в систему **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

- 4.2. Используйте команду **su - manager1**, чтобы переключиться на пользователя **manager1** и проверить доступ.

```
[student@servera ~]$ su - manager1  
Password: redhat  
[manager1@servera ~]$ ls -l /remote/management/  
total 4  
-rw-r--r--. 1 root managers 46 Apr 4 01:13 Welcome.txt  
[manager1@servera ~]$ cat /remote/management/Welcome.txt  
###Welcome to Management Folder on SERVERB###  
[manager1@servera ~]$ echo TEST1 > /remote/management/Test.txt  
[manager1@servera ~]$ cat /remote/management/Test.txt  
TEST1  
[manager1@servera ~]$ ls -l /remote/operation/  
ls: cannot open directory '/remote/operation/': Permission denied  
[manager1@servera ~]$ ls -l /remote/production/  
ls: cannot open directory '/remote/production/': Permission denied  
[manager1@servera ~]$ exit  
Logout
```

- 4.3. Переключитесь на пользователя **dbuser1** и проверьте доступ.

```
[student@servera ~]$ su - dbuser1  
Password: redhat  
[dbuser1@servera ~]$ ls -l /remote/production/  
total 4  
-rw-r--r--. 1 root production 46 Apr 4 01:13 Welcome.txt  
[dbuser1@servera ~]$ cat /remote/production/Welcome.txt  
###Welcome to Production Folder on SERVERB###  
[dbuser1@servera ~]$ echo TEST2 > /remote/production/Test.txt  
[dbuser1@servera ~]$ cat /remote/production/Test.txt  
TEST2  
[dbuser1@servera ~]$ ls -l /remote/operation/
```

```
ls: cannot open directory '/remote/operation/': Permission denied
[dbuser1@servera ~]$ ls -l /remote/management/
ls: cannot open directory '/remote/management/': Permission denied
[dbuser1@servera ~]$ exit
logout
```

4.4. Переключитесь на пользователя **contract1** и проверьте доступ.

```
[student@servera ~]$ su - contractor1
Password: redhat
[contractor1@servera ~]$ ls -l /remote/operation/
total 4
-rw-r--r--. 1 root operators 45 Apr 4 01:13 Welcome.txt
[contractor1@servera ~]$ cat /remote/operation/Welcome.txt
###Welcome to Operation Folder on SERVERB###
[contractor1@servera ~]$ echo TEST3 > /remote/operation/Test.txt
[contractor1@servera ~]$ cat /remote/operation/Test.txt
TEST3
[contractor1@servera ~]$ ls -l /remote/management/
ls: cannot open directory '/remote/management/': Permission denied
[contractor1@servera ~]$ ls -l /remote/production/
ls: cannot open directory '/remote/production/': Permission denied
[contractor1@servera ~]$ exit
logout
```

4.5. Изучите параметры подключения для автоматического подключения NFS.

```
[student@servera ~]$ mount | grep nfs
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
serverb.lab.example.com:/shares/management on /remote/management
type nfs4
(rw,relatime,vers=4.2,rsize=262144,wsize=262144,namlen=255,sync,proto=tcp,timeo=600,
retrans=2,sec=sys,clientaddr=172.25.250.10,local_lock=none,addr=172.25.250.11)
serverb.lab.example.com:/shares/operation on /remote/operation
type nfs4
(rw,relatime,vers=4.2,rsize=262144,wsize=262144,namlen=255,sync,proto=tcp,timeo=600,
retrans=2,sec=sys,clientaddr=172.25.250.10,local_lock=none,addr=172.25.250.11)
serverb.lab.example.com:/shares/production on /remote/production
type nfs4
(rw,relatime,vers=4.2,rsize=262144,wsize=262144,namlen=255,sync,proto=tcp,timeo=600,
retrans=2,sec=sys,clientaddr=172.25.250.10,local_lock=none,addr=172.25.250.11)
```

4.6. Выйдите из сервера **servera**.

```
[student@servera ~]$ exit
logout
```

Оценка

На рабочей станции **workstation** запустите команду **lab netstorage-review grade**, чтобы подтвердить успешность выполнения данного упражнения.

```
[student@workstation ~]$ lab netstorage-review grade
```

Завершение

На рабочей станции **workstation** запустите команду **lab netstorage-review finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab netstorage-review finish
```

На этом лабораторная работа заканчивается.

РЕЗЮМЕ

В этой главе вы узнали, как:

- Монтировать и отключать экспорт **NFS** из командной строки.
- Настройте экспорт **NFS** для автоматического монтирования при запуске.
- Настройте авто монтиrovщик с прямыми и непрямыми сопоставлениями и опишите их различия.
- Настройте клиенты **NFS** для использования **NFSv4** с помощью нового инструмента **nfsconf**.

ГЛАВА 10

УПРАВЛЕНИЕ ПРОЦЕССОМ ЗАГРУЗКИ

ЦЕЛЬ

Управляйте процессом загрузки, чтобы контролировать предлагаемые услуги, а также устранять неполадки и устранять их.

ЗАДАЧИ

- Опишите процесс загрузки Red Hat Enterprise Linux, установите **default target** по умолчанию, используемую при загрузке, и загрузите систему с целью, отличной от цели по умолчанию.
- Войти в систему и измените пароль **root**, если текущий пароль **root** был утерян.
- Вручную исправить конфигурацию файловой системы или проблемы с повреждением, которые останавливают процесс загрузки.

РАЗДЕЛЫ

- Выбор цели загрузки (**Boot Target**) (и упражнения с пошаговыми инструкциями)
- Сброс пароля **root** (и упражнения с пошаговыми инструкциями)
- Исправление проблем с файловой системой при загрузке (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Управление процессом загрузки.

ВЫБОР BOOT TARGET

ЦЕЛИ

После завершения этого раздела вы должны уметь:

- Описывать процесс загрузки Red Hat Enterprise Linux.
- Устанавливать **target** по умолчанию, используемую при загрузке.
- Загрузите систему с **target**, отличной от используемой по умолчанию.

ОПИСАНИЕ ПРОЦЕССА ЗАГРУЗКИ RED HAT ENTERPRISE LINUX 8

Современные компьютерные системы представляют собой сложные комбинации аппаратного и программного обеспечения. Для перехода от неопределенного состояния с отключенным питанием к работающей системе с запросом на вход в систему требуется большое количество аппаратных и программных средств для совместной работы. В следующем списке представлен общий обзор задач, связанных с физической системой x86_64, загружающей Red Hat Enterprise Linux 8. Список виртуальных машин x86_64 примерно такой же, но гипервизор обрабатывает некоторые этапы, связанные с аппаратным обеспечением, программно.

- Машина включена. Системная прошивка, современная **UEFI** или старая версия **BIOS**, запускает самотестирование при включении питания (**Power On Self Test POST**) и начинает инициализировать часть оборудования.
Настраивается с помощью системных экранов конфигурации **BIOS** или **UEFI**, которые обычно открываются нажатием определенной комбинации клавиш, например **F2**, в начале процесса загрузки.
- Системная прошивка ищет загрузочное устройство либо в загрузочной прошивке **UEFI**, либо путем поиска основной загрузочной записи (**Master Boot Record MBR**) на всех дисках в порядке, указанном в **BIOS**.
Настраивается с помощью системных экранов конфигурации **BIOS** или **UEFI**, которые обычно открываются нажатием определенной комбинации клавиш, например **F2**, в начале процесса загрузки.
- Микропрограмма системы считывает загрузчик с диска и затем передает управление системой загрузчику. В системе Red Hat Enterprise Linux 8 загрузчиком является **GRand Unified Bootloader** версии 2 (**GRUB2**).
Настраивается с помощью команды **grub2-install**, которая устанавливает **GRUB2** в качестве загрузчика на диск.
- **GRUB2** загружает свою конфигурацию из файла **/boot/grub2/grub.cfg** и отображает меню, в котором вы можете выбрать, какое ядро загружать.
Настраивается с использованием каталога **/etc/grub.d/**, файла **/etc/default/grub** и команды **grub2-mkconfig** для создания файла **/boot/grub2/grub.cfg**.
- После того, как вы выберете ядро или истечет время ожидания, загрузчик загружает ядро и файлы **initramfs** с диска и помещает их в память. **initramfs** — это архив, содержащий

модули ядра для всего оборудования, необходимого при загрузке, сценарии инициализации и многое другое. В Red Hat Enterprise Linux 8 **initramfs** содержит всю пригодную для использования систему.

Настраивается с помощью каталога **/etc/dracut.conf.d/**, команды **dracut** и команды **lsinitrd** для проверки файла **initramfs**.

- Загрузчик передает управление ядру, передавая любые параметры, указанные в командной строке ядра в загрузчике, и расположение файлов **initramfs** в памяти.
Настраивается с использованием каталога **/etc/grub.d/**, файла **/etc/default/grub** и команды **grub2-mkconfig** для создания файла **/boot/grub2/grub.cfg**.
- Ядро инициализирует все оборудование, для которого оно может найти драйвер в **initramfs**, затем выполняет **/sbin/init** из **initramfs** как **PID 1**. В Red Hat Enterprise Linux 8 **/sbin/init** является ссылкой на **systemd**.
Настраивается с помощью параметра командной строки **init= command-line**.
- Экземпляр **systemd** из **initramfs** выполняет все юниты для target **initrd.target**. Это включает в себя монтирование корневой файловой системы на диске в каталог **/sysroot**.
Настраивается с помощью **/etc/fstab**.
- Ядро переключает (разворачивает) корневую файловую систему с **initramfs** на корневую файловую систему в **/sysroot**. Затем **systemd** повторно запускается, используя копию **systemd**, установленную на диске.
Настраивается с помощью **/etc/systemd/system/default.target** и **/etc/systemd/system/**.

ПЕРЕЗАГРУЗКА И ВЫКЛЮЧЕНИЕ

Чтобы выключить или перезагрузить работающую систему из командной строки, вы можете использовать команду **systemctl**.

systemctl poweroff останавливает все запущенные службы, размонтирует все файловые системы (или перемонтирует их только для чтения, если их размонтировать нельзя), а затем выключает систему.

systemctl reboot останавливает все работающие службы, размонтирует все файловые системы, а затем перезагружает систему.

Вы также можете использовать более короткую версию этих команд, **poweroff** и **reboot**, которые являются символическими ссылками на их эквиваленты **systemctl**.



ПРИМЕЧАНИЕ

systemctl halt и **halt** также доступны для остановки системы, но, в отличие от **poweroff**, эти команды не выключают систему; они переводят систему в состояние, при котором ее можно безопасно отключить вручную.

ВЫБОР СИСТЕМНОГО TARGET

Tadrget **systemd** — это набор модулей (юнитов) **systemd**, которые система должна запустить для достижения желаемого состояния. В следующей таблице перечислены наиболее важные target (цели).

Часто используемые target

ЦЕЛЬ	НАЗНАЧЕНИЕ
graphical.target	Система поддерживает несколько пользователей, графический и текстовый вход в систему.
multi-user.target	Система поддерживает несколько пользователей, только текстовый вход.
rescue.target	sulogin приглашение, базовая инициализация системы завершена.
emergency.target	sulogin приглашение, разворачивание initramfs завершено, и системный root подключен / только для чтения.

Цель (target) загрузки может быть частью другого target. Например, **graphical.target** включает **multiuser.target**, который, в свою очередь, зависит от **basic.target** и других. Вы можете просмотреть эти зависимости с помощью следующей команды.

```
[user@host ~]$ systemctl list-dependencies graphical.target | grep target
graphical.target
* └─multi-user.target
*   ├─basic.target
*   | ├─paths.target
*   | ├─slices.target
*   | ├─sockets.target
*   | ├─sysinit.target
*   |   ├─cryptsetup.target
*   |   | ├─local-fs.target
*   |   | └─swap.target
```

Чтобы вывести список доступных целей, используйте следующую команду.

UNIT	LOAD	ACTIVE	SUB	DESCRIPTION
<hr/>				
basic.target	loaded	active	active	Basic System
cryptsetup.target	loaded	active	active	Local Encrypted Volumes
emergency.target	loaded	inactive	dead	Emergency Mode
getty-pre.target	loaded	inactive	dead	Login Prompts (Pre)
getty.target	loaded	active	active	Login Prompts
graphical.target	loaded	inactive	dead	Graphical Interface

Выбор target во время выполнения

В работающей системе администраторы могут переключаться на другую цель (target) с помощью команды **systemctl isolate**.

Tartget **isolate** останавливает все службы, которые не требуются для этой цели (и ее зависимостей), и запускает все необходимые службы, которые еще не запущены.

Не все цели могут быть изолированы. Вы можете изолировать только target, для которых в их файлах модулей (юнитов) установлено значение **AllowIsolate=yes**. Например, вы можете изолировать графическую цель, но не target **cryptsetup**.

```
[user@host ~]$ systemctl cat graphical.target
# /usr/lib/systemd/system/graphical.target
...output omitted...
[Unit]
Description=Graphical Interface
Documentation=man:systemd.special(7)
Requires=multi-user.target
Wants=display-manager.service
Conflicts=rescue.service rescue.target
After=multi-user.target rescue.service rescue.target display-manager.service
AllowIsolate=yes

[user@host ~]$ systemctl cat cryptsetup.target
# /usr/lib/systemd/system/cryptsetup.target
...output omitted...
[Unit]
Description=Local Encrypted Volumes
Documentation=man:systemd.special(7)
```

Установка target по умолчанию

Когда система запускается, **systemd** активирует target **default.target**. Обычно целью по умолчанию в **/etc/systemd/system/** является символьическая ссылка либо на **graphical.target**, либо на **multiuser.target**. Вместо того, чтобы редактировать эту символьическую ссылку вручную, команда **systemctl** предоставляет две подкоманды для управления этой ссылкой: **get-default** и **set-default**.

```
[root@host ~]# systemctl get-default
multi-user.target
[root@host ~]# systemctl set-default graphical.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target -> /usr/lib/systemd/system/
graphical.target.
[root@host ~]# systemctl get-default
graphical.target
```

Выбор другой target во время загрузки

Чтобы выбрать другую цель во время загрузки, добавьте параметр **systemd.unit=target.target** в командную строку ядра из загрузчика.

Например, чтобы загрузить систему в аварийную оболочку (**rescue shell**), где вы можете изменить конфигурацию системы практически без запуска служб, добавьте следующую опцию в командную строку ядра из загрузчика.

```
systemd.unit=rescue.target
```

Это изменение конфигурации влияет только на одну загрузку, что делает его полезным инструментом для устранения неполадок в процессе загрузки.

Чтобы использовать этот метод выбора другой цели (target), используйте следующую процедуру:

1. Загрузите или перезагрузите систему.
2. Прервите обратный отсчет меню загрузчика, нажав любую клавишу (кроме Enter, которая запустит обычную загрузку).
3. Переместите курсор на запись ядра, которую вы хотите запустить.
4. Нажмите **e**, чтобы отредактировать текущую запись.
5. Переместите курсор на строку, начинающуюся с **linux**. Это командная строка ядра.
6. Добавьте **systemd.unit=target.target**. Например, **systemd.unit=emergency.target**.
7. Нажмите **Ctrl+x**, чтобы загрузиться с этими изменениями.



РЕКОМЕНДАЦИИ

info grub2 (GNU GRUB manual)

Справочные страницы **man bootup(7)**, **dracut.bootup(7)**, **lsinitrd(1)**, **systemd.target(5)**, **systemd.special(7)**, **sulogin(8)**, and **systemctl(1)**

Для получения дополнительной информации см. главу «Управление службами с помощью **systemd**» в руководстве «Настройка основных параметров системы» по адресу:

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/configuring_basic_system_settings/#managing-services-with-systemd

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

ВЫБОР BOOT TARGET

В этом упражнении вы определите цель загрузки (target) по умолчанию, в которую загружается система, и выполните загрузку системы используя другие цели (target).

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность обновить target системы по умолчанию и использовать временную цель (target), выбранную во время загрузки.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab boot-selecting start**. Эта команда запускает сценарий, который готовит рабочую станцию **workstation** к упражнению.

```
[student@workstation ~]$ lab boot-selecting start
```

1. На рабочей станции **workstation** откройте терминал и убедитесь, что целью (target) по умолчанию является **graphical.target**.

```
[student@workstation ~]$ systemctl get-default  
graphical.target
```

2. На рабочей станции **workstation** переключитесь на multi-user target вручную без перезагрузки. Используйте команду **sudo** и, если будет предложено, используйте **student** в качестве пароля.

```
[student@workstation ~]$ sudo systemctl isolate multi-user.target  
[sudo] password for student: student
```

3. Получите доступ к текстовой консоли. Используйте последовательность клавиш **Ctrl+Alt+F1**, используя соответствующую кнопку или пункт меню. Войдите в систему как **root**, используя слово **redhat** в качестве пароля.

```
workstation login: root  
Password: redhat  
[root@workstation ~]#
```



ПРИМЕЧАНИЕ

Напоминание: если вы используете терминал через веб-страницу, вы можете щелкнуть значок «Показать клавиатуру» под строкой URL-адреса вашего веб-браузера, а затем справа от IP-адреса машины.

- Настройте рабочую станцию **workstation** на автоматическую загрузку в многопользовательском режиме (**multi-user target**), а затем перезагрузите рабочую станцию для проверки. Когда закончите, измените target systemd по умолчанию обратно на **graphical target**.

4.1. Используйте команду **systemctl set-default**, чтобы установить цель (target) по умолчанию.

```
[root@workstation ~]# systemctl set-default multi-user.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target -> /usr/lib/
systemd/system/multi-user.target.
```

4.2. Перезагрузите рабочую станцию **workstation**.

```
[root@workstation ~]# systemctl reboot
```

Обратите внимание, что после перезагрузки система представляет собой текстовую консоль, а не графический вход в систему.

4.3. Войдите в систему как пользователь **root**, используя слово **redhat** в качестве пароля.

```
workstation login: root
Password: redhat
Last login: Thu Mar 28 14:50:53 on tty1
[root@workstation ~]#
```

4.4. Установите target **systemd** по умолчанию обратно на **graphical target**.

```
[root@workstation ~]# systemctl set-default graphical.target
Removed /etc/systemd/system/default.target.
Created symlink /etc/systemd/system/default.target -> /usr/lib/
systemd/system/graphical.target.
```

На этом завершается первая часть упражнения, в которой вы тренировались устанавливать target **systemd** по умолчанию.

- 5.** Во второй части упражнения вы потренируетесь загружать систему в режиме **rescue** (спасения).

Получите доступ к загрузчику, снова перезагрузив рабочую станцию. Из меню загрузчика (boot loader menu) загрузитесь используя в цель восстановления.

5.1. Инициируйте перезагрузку.

```
[root@workstation ~]# systemctl reboot
```

5.2. Когда появится меню загрузчика, нажмите любую клавишу, чтобы прервать обратный отсчет (кроме Enter, который инициирует обычную загрузку).

5.3. Используйте клавиши курсора, чтобы выделить запись загрузчика по умолчанию.

5.4. Нажмите **e**, чтобы отредактировать текущую запись.

5.5. С помощью клавиш курсора перейдите к строке, начинающейся со слова **linux**.

5.6. Нажмите **End**, чтобы переместить курсор в конец строки.

5.7. Добавьте слова **systemd.unit=rescue.target** в конец строки.

5.8. Нажмите **Ctrl+x** для загрузки с измененной конфигурацией.

5.9. Войдите в режим **rescue**. Пароль **root** — **redhat**. Возможно, вам придется нажать **Enter**, чтобы получить приглашение.

```
Give root password for maintenance  
(or press Control-D to continue): redhat  
[root@workstation ~]#
```

- 6.** Убедитесь, что в режиме восстановления (**rescue**) корневая файловая система находится в режиме чтения/записи.

```
[root@workstation ~]# mount  
...output omitted...  
/dev/vda1 on / type xfs (rw,relatime,seclabel,attr2,inode64,noquota)  
...output omitted...
```

- 7.** Нажмите **Ctrl+d**, чтобы продолжить процесс загрузки.

Система представляет графический логин. Войдите в систему как пользователь **student**, используя слово **student** в качестве пароля.

Завершение

На рабочей станции **workstation**, запустите скрипт **lab boot-selecting finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab boot-selecting finish
```

На этом упражнения с пошаговыми инструкциями заканчивается.

СБРОС ПАРОЛЯ ROOT

ЦЕЛИ

После завершения этого раздела вы сможете войти в систему и изменить пароль **root**, если текущий пароль **root** был утерян.

СБРОС ПАРОЛЯ ROOT ПРИ ЗАГРУЗКЕ

Одной из задач, которую должен уметь выполнять каждый системный администратор, является сброс утерянного пароля **root**. Если администратор все еще может войти в систему либо как непривилегированный пользователь, но с полным доступом sudo, либо как **root**, эта задача тривиальна. Когда администратор не может войти в систему, эта задача усложняется.

Существует несколько способов установки нового пароля **root**. Системный администратор может, например, загрузить систему с **Live CD**, смонтировать оттуда корневую файловую систему и отредактировать файл **/etc/shadow**. В этом разделе мы исследуем метод, который не требует использования внешних носителей.



ПРИМЕЧАНИЕ

В Red Hat Enterprise Linux 6 и более ранних версиях администраторы могут загрузить систему на **уровне выполнения 1**, чтобы получить приглашение **root**. Ближайшими аналогами **уровня выполнения 1** на компьютере с Red Hat Enterprise Linux 8 являются **rescue** и **emergency targets**, оба из которых требуют пароля **root** для входа в систему.

В Red Hat Enterprise Linux 8 можно сделать так, чтобы сценарии, запускаемые из **initramfs**, приостанавливались в определенные моменты, предоставляя root оболочку (**root shell**), а затем продолжались при выходе из этой оболочки. В основном это предназначено для отладки, но вы также можете использовать этот метод для сброса утерянного пароля **root**.

Чтобы получить доступ к этой корневой оболочке, выполните следующие действия:

1. Перезагрузите систему.
2. Прервите обратный отсчет загрузчика, нажав любую клавишу, кроме Enter.
3. Переместите курсор на запись core для загрузки.
4. Нажмите **e**, чтобы отредактировать выбранную запись.
5. Переместите курсор в командную строку **core** (строка, начинающаяся с слова **linux**).
6. Добавьте в конец строки слово **rd.break**. С этой опцией система приревается во время загрузки непосредственно перед тем, как система передает управление от **initramfs** реальной системе.
7. Нажмите **Ctrl+x**, чтобы загрузиться с внесёнными изменениями.

На этом этапе система представляет собой корневую оболочку с фактической корневой файловой системой на диске, смонтированной **только для чтения** в каталоге **/sysroot**. Поскольку устранение неполадок часто требует изменения корневой файловой системы, вам необходимо

изменить корневую файловую систему на **чтение/запись** (**read/write**). Следующий шаг показывает, как параметр **remount,rw** команды **mount** перемонтирует файловую систему с установленным новым параметром (**rw (read/write)**).



ПРИМЕЧАНИЕ

Готовые образы могут помещать в ядро несколько аргументов **console=** для поддержки широкого спектра сценариев реализации. Эти аргументы **console=** указывают устройства, используемые для вывода на консоль. Предостережение относительно **rd.break** заключается в том, что даже несмотря на то, что система отправляет сообщения ядра на все консоли, приглашение в конечном итоге использует ту консоль, которая дана последней. Если вы не получили приглашение, вы можете временно изменить порядок аргументов **console=** при редактировании командной строки ядра из загрузчика.



ВАЖНО

Система еще **не включила SELinux**, поэтому любой создаваемый вами файл **не имеет контекста SELinux**. Некоторые инструменты, такие как команда **passwd**, сначала создают временный файл, а затем перемещают его вместо файла, который они предназначены редактировать, фактически создавая **новый файл без контекста SELinux**. По этой причине, когда вы используете команду **passwd** с **rd.break**, файл **/etc/shadow** не получает контекст **SELinux**.

Чтобы сбросить пароль **root** с этого момента, используйте следующую процедуру:

1. Перемонтируйте **/sysroot** как доступный для **чтения/записи**.

```
switch_root:/# mount -o remount,rw /sysroot
```

2. Переключитесь в **chroot-ограниченное пространство**, где **/sysroot** рассматривается как корень дерева файловой системы.

```
switch_root:/# chroot /sysroot
```

3. Установите новый пароль **root**.

```
sh-4.4# passwd root
```

4. Убедитесь, что все немаркованные файлы, включая **/etc/shadow** на данный момент, пере маркируют контекст во время загрузки.

```
sh-4.4# touch /.autorelabel
```

5. Дважды введите **exit**. Первая команда выходит из ограниченного пространства **chroot**, а вторая команда выходит из оболочки отладки **initramfs**.

В этот момент система продолжает загружаться, выполняет полную переназначение **SELinux**, а затем снова перезагружается.

ПРОВЕРКА ЖУРНАЛОВ ЛОГИРОВАНИЯ СИСТЕМЫ

Просмотр журналов ранее неудачных загрузок может быть полезен. Если системные журналы сохраняются после перезагрузки, вы можете использовать инструмент **journalctl** для проверки этих журналов.

Помните, что по умолчанию системные журналы хранятся в каталоге **/run/log/journal**, это означает, что журналы очищаются при перезагрузке системы. Чтобы журналы хранились в каталоге **/var/log/journal**, который сохраняется после перезагрузки, установите для параметра **Storage** значение **persistent** в файле **/etc/systemd/journald.conf**.

```
[root@host ~]# vim /etc/systemd/journald.conf
...output omitted...
[Journal]
Storage=persistent
...output omitted...
[root@host ~]# systemctl restart systemd-journald.service
```

Чтобы просмотреть журналы предыдущей загрузки, используйте параметр **-b** с командой **journalctl**. Без каких-либо аргументов опция **-b** отображает только сообщения с момента последней загрузки. С отрицательным числом в качестве аргумента он отображает журналы предыдущих загрузок.

```
[root@host ~]# journalctl -b -1 -p err
```

Эта команда показывает все сообщения, помеченные как ошибки (**error**) или хуже (**worse**), из предыдущей загрузки.

УСТРАНЕНИЕ ПРОБЛЕМ С ЗАГРУЗКОЙ SYSTEMD

Для устранения проблем с запуском служб во время загрузки Red Hat Enterprise Linux 8 предлагает следующие инструменты.

Включение оболочки ранней отладки (Early Debug Shell)

Включив службу отладочной оболочки с помощью **systemctl enable debug-shell.service**, система порождает корневую оболочку на **TTY9 (Ctrl+Alt+F9)** в начале последовательности загрузки. Эта оболочка автоматически регистрируется как **root**, поэтому администраторы могут отлаживать систему, пока операционная система еще загружается.



ПРЕДУПРЕЖДЕНИЕ

Не забудьте отключить службу **debug-shell.service** после завершения отладки, потому что она оставляет корневую оболочку без проверки подлинности открытой для всех, у кого есть доступ к локальной консоли.

Использование Emergency и Rescue Targets

При добавлении либо **systemd.unit=rescue.target**, либо **systemd.unit=emergency.target** в командную строку ядра из загрузчика система переходит в аварийную или аварийную оболочку вместо нормального запуска. Обе эти оболочки требуют пароля **root**.

Аварийная цель (**emergency.target**) сохраняет корневую файловую систему смонтированной только для чтения, в то время как **rescue.target** ожидает завершения **sysinit.target**, так что инициализируется дополнительная часть системы, например служба ведения журналов или файловые системы. В этот момент пользователь **root** не может вносить изменения в **/etc/fstab**, пока диск не будет перемонтирован в состоянии чтения и записи выполнить **mount -o remount,rw /**

Администраторы могут использовать эти оболочки для устранения любых проблем, препятствующих нормальной загрузке системы; например, петля зависимости между сервисами или неверная запись в **/etc/fstab**. Выход из этих оболочек продолжается обычным процессом загрузки.

Выявление зависших заданий

Во время запуска **systemd** порождает ряд заданий. Если некоторые из этих заданий не могут быть завершены, они блокируют выполнение других заданий. Чтобы проверить текущий список заданий, администраторы могут использовать команду **systemctl list-jobs**. Любые задания, указанные как выполняющиеся, должны быть завершены до того, как задания, указанные как ожидающие, смогут быть продолжены.



РЕКОМЕНДАЦИИ

Справочные станицы **man dracut.cmdline(7)**, **systemd-journald(8)**, **journald.conf(5)**, **journalctl(1)**, и **systemctl(1)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

СБРОС ПАРОЛЯ ROOT

В этом упражнении вы сбросите пароль root в системе.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность сбросить утерянный пароль root.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab boot-resetting start**. Эта команда запускает сценарий, который определяет, доступен ли сервер **servera** в сети. Он также сбрасывает пароль **root** на случайную строку и устанавливает более высокий тайм-аут для меню **GRUB2**.

```
[student@workstation ~]$ lab boot-resetting start
```

1. Перезагружаем сервер **servera**, и прерываем обратный отсчет в меню загрузчика.
 - 1.1. Найдите значок консоли **servera**, соответствующий среде вашего класса. Откройте консоль.
Отправьте **Ctrl+Alt+Del** в вашу систему, используя соответствующую кнопку или пункт меню.
 - 1.2. Когда появится меню загрузчика, нажмите любую клавишу, чтобы прервать обратный отсчет, кроме **Enter**.
2. Отредактируйте запись загрузчика по умолчанию в памяти, чтобы прервать процесс загрузки сразу после того, как ядро смонтирует все файловые системы, но до того, как оно передаст управление **systemd**.
 - 2.1. Используйте клавиши курсора, чтобы выделить запись загрузчика по умолчанию.
 - 2.2. Нажмите **e**, чтобы отредактировать текущую запись.
 - 2.3. Используйте клавиши курсора, чтобы перейти к строке, которая начинается со слова **linux**.
 - 2.4. Нажмите **End**, чтобы переместить курсор в конец строки.
 - 2.5. Добавьте **rd.break** в конец строки.
 - 2.6. Нажмите **Ctrl+x** для загрузки с измененной конфигурацией.

- По приглашению **switch_root** перемонтируйте файловую систему **/sysroot** для **чтения/записи**, затем используйте **chroot**, чтобы войти в «песочнице» **chroot** в **/sysroot**.

```
switch_root:/# mount -o remount,rw /sysroot  
switch_root:/# chroot /sysroot
```

- Измените пароль **root** обратно на **redhat**.

```
sh-4.4# passwd root  
Changing password for user root.  
New password: redhat  
BAD PASSWORD: The password is shorter than 8 characters  
Retype new password: redhat  
passwd: all authentication tokens updated successfully.
```

- Настройте систему для автоматического выполнения полного переименования **SELinux** после загрузки. Это необходимо, поскольку команда **passwd** воссоздает файл **/etc/shadow** без контекста **SELinux**.

```
sh-4.4# touch /.autorelabel
```

- Дважды введите **exit**, чтобы продолжить загрузку системы в обычном режиме. Система выполняет переназначение **SELinux**, а затем снова перезагружается сама по себе. Когда система заработает, проверьте свою работу, войдя в консоль как **root**. Используйте слово **redhat** в качестве пароля.

Завершение

На рабочей станции **workstation**, запустите сценарий **lab boot-resetting finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab boot-resetting finish
```

На этом упражнения с пошаговыми инструкциями заканчивается.

УСТРАНЕНИЕ ПРОБЛЕМ ФАЙЛОВОЙ СИСТЕМЫ ПРИ ЗАГРУЗКЕ

ЦЕЛИ

После завершения этого раздела вы сможете вручную исправить конфигурацию файловой системы или проблемы с повреждением, которые останавливают процесс загрузки.

ДИАГНОСТИКА И ИСПРАВЛЕНИЕ ПРОБЛЕМ С ФАЙЛОВОЙ СИСТЕМОЙ

Ошибки в **/etc/fstab** и поврежденные файловые системы могут помешать загрузке системы. В большинстве случаев **systemd** переходит в оболочку аварийного (**emergency**) восстановления, для которой требуется пароль **root**.

В следующей таблице перечислены некоторые распространенные ошибки и их результаты.

Распространенные проблемы с файловой системой

ПРОБЛЕМА	РЕЗУЛЬТАТ
Corrupt file system (Поврежденная файловая система)	systemd пытается восстановить файловую систему. Если проблема слишком серьезна для автоматического исправления, система переводит пользователя в аварийную оболочку.
Nonexistent device or UUID referenced in /etc/fstab (Ссылка на несуществующее устройство или UUID в /etc/fstab)	systemd ждет определенное время, ожидая, пока устройство станет доступным. Если устройство недоступно, система переводит пользователя в аварийную оболочку по истечении тайм-аута.
Nonexistent mount point in /etc/fstab (Несуществующая точка монтирования в /etc/fstab)	Система переводит пользователя в аварийную оболочку.
Incorrect mount option specified in /etc/fstab (В файле /etc/fstab указан неверный параметр монтирования.)	Система переводит пользователя в аварийную оболочку.

Во всех случаях администраторы также могут использовать аварийную цель (**emergency target**) для диагностики и устранения проблемы, поскольку перед отображением аварийной оболочки не монтируются файловые системы.



ПРИМЕЧАНИЕ

При использовании аварийной оболочки для решения проблем с файловой системой не забудьте запустить **systemctl daemon-reload** после редактирования **/etc/fstab**. Без этой перезагрузки **systemd** может продолжать использовать старую версию.



РЕКОМЕНДАЦИИ

Справочные страницы **man systemd-fsck(8)**, **systemd-fstab-generator(8)**, и **systemd.mount(5)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УСТРАНЕНИЕ ПРОБЛЕМ ФАЙЛОВОЙ СИСТЕМЫ ПРИ ЗАГРУЗКЕ

В этом упражнении вы восстановите систему после неправильной конфигурации в **/etc/fstab**, которая приводит к сбою процесса загрузки.

В РЕЗУЛЬТАТЕ

Вы должны уметь диагностировать проблемы **/etc/fstab** и использовать аварийный режим для восстановления системы.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab boot-repairing start**. Эта команда запускает стартовый сценарий, который определяет, доступен ли сервер **servera** в сети. Он также представляет проблему с файловой системой, устанавливает более высокий тайм-аут для меню GRUB2 и перезагружает сервер **servera**.

```
[student@workstation ~]$ lab boot-repairing start
```

1. Получите доступ к консоли сервера и обратите внимание, что процесс загрузки застрял на ранней стадии.

1.1. Найдите значок консоли **servera**, соответствующий среде вашего класса. Откройте консоль.

Обратите внимание, что начальное задание не кажется завершенным. Найдите минутку, чтобы подумать о возможной причине такого поведения.

1.2. Для перезагрузки отправьте **Ctrl+Alt+Del** вашей системе с помощью соответствующей кнопки или пункта меню. В этой конкретной проблеме с загрузкой эта последовательность клавиш не может немедленно прервать текущее задание, и вам, возможно, придется подождать, пока истечет время ожидания, прежде чем система перезагрузится.

Если вы дождитесь истечения времени выполнения задачи, не отправив **Ctrl+Alt+Del**, система в конечном итоге сама создаст аварийную оболочку.

1.3. Когда появится меню загрузчика, нажмите любую клавишу, чтобы прервать обратный отсчет, кроме **Enter**.

2. Глядя на ошибку предыдущей загрузки, кажется, что по крайней мере часть системы все еще работает. Поскольку вы знаете пароль **root**, слово **redhat**, попробуйте аварийную загрузку.

2.1. Используйте клавиши курсора, чтобы выделить запись загрузчика по умолчанию.

2.2. Нажмите **e**, чтобы отредактировать текущую запись.

- 2.3.** Используйте клавиши курсора, чтобы перейти к строке, которая начинается с слова **linux**.
- 2.4.** Нажмите **End**, чтобы переместить курсор в конец строки.
- 2.5.** Добавьте **systemd.unit=emergency.target** в конец строки.
- 2.6.** Нажмите **Ctrl+x** для загрузки с измененной конфигурацией.

3. Войдите в аварийный режим. Пароль root — redhat.

```
Give root password for maintenance  
(or press Control-D to continue): redhat  
[root@servera ~]#
```

4. Определите, какие файловые системы смонтированы в данный момент.

```
[root@servera ~]# mount  
...output omitted...  
/dev/vda1 on / type xfs (ro,relatime,seclabel,attr2,inode64,noquota)  
...output omitted...
```

Обратите внимание, что корневая файловая система монтируется только для чтения.

5. Перемонтируйте корневую файловую систему для чтения/записи.

```
[root@servera ~]# mount -o remount,rw /
```

6. Используйте команду **mount -a**, чтобы попытаться смонтировать все остальные файловые системы. С **--all (-a)**, команда монтирует все файловые системы, перечисленные в **/etc/fstab**, которые еще не смонтированы.

```
[root@servera ~]# mount -a  
mount: /RemoveMe: mount point does not exist.
```

7. Отредактируйте **/etc/fstab**, чтобы устранить проблему.

7.1. Удалите или закомментируйте неверную строку.

```
[root@servera ~]# vim /etc/fstab  
...output omitted...  
# /dev/sdz1      /RemoveMe  xfs      defaults      0 0
```

7.2. Обновите **systemd**, чтобы система зарегистрировала новую конфигурацию **/etc/fstab**.

```
[root@servera ~]# systemctl daemon-reload  
[root@servera ~]#
```

8. Убедитесь, что ваш **/etc/fstab** теперь корректен, попытавшись смонтировать все записи.

```
[root@servera ~]# mount -a  
[root@servera ~]#
```

9. Перезагрузите систему и дождитесь завершения загрузки. Теперь система должна загружаться нормально.

```
[root@servera ~]# systemctl reboot
```

Завершение

На рабочей станции запустите сценарий **lab boot-repairing finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab boot-repairing finish
```

На этом упражнения с пошаговыми инструкциями заканчивается.

ЛАБОРАТОРНАЯ РАБОТА

УПРАВЛЕНИЕ ПРОЦЕССОМ ЗАГРУЗКИ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы сбросите пароль **root** в системе, восстановите неправильную конфигурацию и установите цель загрузки по умолчанию.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Сбросить утерянный пароль **root**.
- Диагностировать и устранять проблемы с загрузкой.
- Установите **target systemd** по умолчанию.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab boot-review start**. Эта команда запускает сценарий, который определяет, доступна ли машина **serverb** в сети. Он также представляет проблему с файловой системой, сбрасывает пароль **root**, устанавливает более высокий тайм-аут для меню **GRUB2** и перезагружает **serverb**.

```
[student@workstation ~]$ lab boot-review start
```

1. На **serverb** сбросьте пароль **root** на слово **redhat**.

Найдите значок консоли **serverb** в соответствии со средой вашего класса. Работайте с этой консолью.

2. Система не загружается. Начальное задание, кажется, не завершено. Из консоли исправьте проблему.

3. Измените **target systemd** по умолчанию на **serverb**, чтобы система автоматически запускала графический интерфейс при загрузке.

На **serverb** еще не установлен графический интерфейс. Для этого упражнения установите только **target** по умолчанию и не устанавливайте пакеты.

Оценка

На рабочей станции **workstation**, запустите сценарий **lab boot-review grade**, чтобы подтвердить успешное выполнение этого упражнения.

```
[student@workstation ~]$ lab boot-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab boot-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab boot-review finish
```

На этом лабораторная работа заканчивается.

РЕШЕНИЕ

УПРАВЛЕНИЕ ПРОЦЕССОМ ЗАГРУЗКИ

КОНТРОЛЬНЫЙ СПИСОК РАБОТЫ

В этой лабораторной работе вы сбросите пароль **root** в системе, восстановите неправильную конфигурацию и установите цель загрузки по умолчанию.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Сбросить утерянный пароль **root**.
- Диагностировать и устранять проблемы с загрузкой.
- Установите **target systemd** по умолчанию.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab boot-review start**. Эта команда запускает сценарий, который определяет, доступна ли машина **serverb** в сети. Он также представляет проблему с файловой системой, сбрасывает пароль **root**, устанавливает более высокий тайм-аут для меню **GRUB2** и перезагружает **serverb**.

```
[student@workstation ~]$ lab boot-review start
```

1. На **serverb** сбросьте пароль **root** на слово **redhat**.

Найдите значок консоли **serverb** в соответствии со средой вашего класса. Работайте с этой консолью.

1.1. Отправьте **Ctrl+Alt+Del** в вашу систему, используя соответствующую кнопку или пункт меню.

1.2. Когда появится меню загрузчика, нажмите любую клавишу, чтобы прервать обратный отсчет, кроме **Enter**.

1.3. Используйте клавиши курсора, чтобы выделить запись загрузчика по умолчанию.

1.4. Нажмите **e**, чтобы отредактировать текущую запись.

1.5. Используйте клавиши курсора, чтобы перейти к строке, которая начинается с слова **linux**.

1.6. Нажмите **End**, чтобы переместить курсор в конец строки.

1.7. Добавьте **rd.break** в конец строки.

1.8. Нажмите **Ctrl+x** для загрузки с измененной конфигурацией.

1.9. По приглашению **switch_root** перемонтируйте файловую систему **/sysroot** для чтения/записи, затем используйте **chroot**, чтобы войти в ограниченную область **chroot** в **/sysroot**.

```
switch_root:/# mount -o remount,rw /sysroot
switch_root:/# chroot /sysroot
```

1.10. Установите пароль **root** слово **redhat**.

```
sh-4.4# passwd root
Changing password for user root.
New password: redhat
BAD PASSWORD: The password is shorter than 8 characters
Retype new password: redhat
passwd: all authentication tokens updated successfully.
```

1.11. Настройте систему для автоматического выполнения полного переименования SELinux после загрузки.

```
sh-4.4# touch /.autorelabel
```

1.12. Дважды введите **exit**, чтобы продолжить загрузку системы. Система не загружается из-за проблемы, которую вы решите на следующем шаге.

2. Система не загружается. Начальное задание, кажется, не завершено. Из консоли исправьте проблему.

- 2.1.** Загрузите систему в аварийном (**emergency**) режиме. Для этого перезагрузите **serverb**, отправив **Ctrl+Alt+Del** в вашу систему с помощью соответствующей кнопки или пункта меню.
- 2.2.** Когда появится меню загрузчика, нажмите любую клавишу, чтобы прервать обратный отсчет, кроме **Enter**.
- 2.3.** Используйте клавиши курсора, чтобы выделить запись загрузчика по умолчанию.
- 2.4.** Нажмите **e**, чтобы отредактировать текущую запись.
- 2.5.** Используйте клавиши курсора, чтобы перейти к строке, которая начинается с слова **linux**.
- 2.6.** Нажмите **End**, чтобы переместить курсор в конец строки.
- 2.7.** Добавьте слова **systemd.unit=emergency.target** в конец строки.
- 2.8.** Нажмите **Ctrl+x** для загрузки с измененной конфигурацией.
- 2.9.** Войдите в аварийный (**emergency**) режим. Пароль для **root** — слово **redhat**.

```
Give root password for maintenance
(or press Control-D to continue): redhat
[root@serverb ~]#
```

2.10. Перемонтируйте / файловую систему для чтения/записи (read/write).

```
[root@serverb ~]# mount -o remount,rw /
```

2.11. Используйте команду **mount -a**, чтобы попытаться смонтировать все остальные файловые системы.

```
[root@serverb ~]# mount -a
mount: /olddata: can't find UUID=4d5c85a5-8921-4a06-8aff-80567e9689bc.
```

2.12. Отредактируйте **/etc/fstab**, чтобы удалить или закомментировать неправильную строку.

```
[root@serverb ~]# vim /etc/fstab
...output omitted...
#UUID=4d5c85a5-8921-4a06-8aff-80567e9689bc /olddata xfs defaults 0 0
```

2.13. Обновите **systemd**, чтобы система зарегистрировала новую конфигурацию **/etc/fstab**.

```
[root@serverb ~]# systemctl daemon-reload
[root@serverb ~]#
```

2.14. Убедитесь, что содержание вашего файла **/etc/fstab** теперь правильный, попытавшись смонтировать все записи.

```
[root@serverb ~]# mount -a
[root@serverb ~]#
```

2.15. Перезагрузите систему и дождитесь завершения загрузки. Поскольку вы создали файл **/.autorelabel** на первом этапе, после установки пароля **root** система выполняет переназначение **SELinux**, а затем снова перезагружается сама. Теперь система должна загружаться нормально.

```
[root@serverb ~]# systemctl reboot
```

3. Измените **target systemd** по умолчанию на сервере **serverb**, чтобы система автоматически запускала графический интерфейс при загрузке.
На **serverb** еще не установлен графический интерфейс. Для этого упражнения установите только **target** по умолчанию и не устанавливайте пакеты.

- 3.1.** Войдите на **serverb** как пользователь **root**. Используйте слово **redhat** в качестве пароля.
- 3.2.** Используйте команду **systemctl set-default**, чтобы установить **graphical.target** в качестве **target** по умолчанию.

```
[root@serverb ~]# systemctl set-default graphical.target
```

3.3. Используйте команду **systemctl get-default**, чтобы проверить свою работу.

```
[root@serverb ~]# systemctl get-default  
graphical.target
```

3.4. Выйдите из сервера **serverb**.

```
[root@serverb ~]# exit
```

Оценка

На рабочей станции **workstation**, запустите сценарий **lab boot-review grade**, чтобы подтвердить успешное выполнение этого упражнения.

```
[student@workstation ~]$ lab boot-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab boot-review finish**, чтобы завершить лабораторную работу.

```
[student@workstation ~]$ lab boot-review finish
```

На этом лабораторная работа заканчивается.

РЕЗЮМЕ

В этой главе вы узнали:

- **systemctl reboot** и **systemctl poweroff** перезагружают и выключают систему соответственно.
- **systemctl isolate target-name.target** переключается на новую цель (**target**) во время выполнения.
- **systemctl get-default** и **systemctl set-default** могут использоваться для запроса и установки цели по умолчанию.
- Используйте **rd.break** в командной строке ядра, чтобы прервать процесс загрузки до того, как управление будет передано от **initramfs**. Корневая файловая система смонтирована только для чтения в каталоге **/sysroot**.
- Аварийную цель (**emergency target**) можно использовать для диагностики и устранения проблем с файловой системой.

ГЛАВА 11

УПРАВЛЕНИЕ СЕТЕВОЙ БЕЗОПАСНОСТЬЮ ЦЕЛЬ

ЦЕЛЬ

Контролируйте сетевые подключения к службам с помощью системного брандмауэра и правил **SELinux**.

ЗАДАЧИ

- Принимать или отклонять сетевые подключения к системным службам с помощью правил **firewalld**.
- Контролируйте, могут ли сетевые службы использовать определенные сетевые порты, управляя метками портов **SELinux**.

РАЗДЕЛЫ

- Управление серверными брандмауэрами (и упражнения с пошаговыми инструкциями)
- Управление маркировкой портов **SELinux** (и упражнения с пошаговыми инструкциями)

ЛАБОРОТОРНАЯ РАБОТА

Управление серверными брандмауэрами.

УПРАВЛЕНИЕ СЕРВЕРНЫМИ БРАНДМАУЕРАМИ

ЦЕЛИ

После заполнения этого раздела вы сможете принимать или отклонять сетевые подключения к системным службам с помощью правил **firewalld**.

КОНЦЕПЦИИ АРХИТЕКТУРЫ БРАНДМАУЭРА

Ядро **Linux** включает **netfilter**, инфраструктуру для операций с сетевым трафиком, таких как фильтрация пакетов, преобразование сетевых адресов и преобразование портов. Реализуя в ядре обработчики, перехватывающие вызовы функций и сообщения, **netfilter** позволяет другим модулям ядра напрямую взаимодействовать с сетевым стеком ядра. Программное обеспечение брандмауэра использует эти ловушки для регистрации правил фильтрации и функций модификации пакетов, позволяя обрабатывать каждый пакет, проходящий через сетевой стек. Любой входящий, исходящий или пересылаемый сетевой пакет может быть проверен, изменен, отброшен или маршрутизирован программно до того, как он достигнет компонентов или приложений пользовательского пространства. **Netfilter** является основным компонентом брандмауэров Red Hat Enterprise Linux 8.

Nftables улучшает сетевой фильтр netfilter

Ядро **Linux** также включает **nftables**, новую подсистему фильтрации и классификации пакетов, в которой улучшены части кода **netfilter**, но сохранена архитектура **netfilter**, такая как перехватчики сетевого стека, система отслеживания соединений и средство ведения журнала. Преимущества обновления **nftables** заключаются в более быстрой обработке пакетов, более быстром обновлении набора правил и одновременной обработке IPv4 и IPv6 по одним и тем же правилам. Еще одно существенное различие между **nftables** и исходным **netfilter** — это их интерфейсы. **Netfilter** настраивается с помощью нескольких служебных сред, включая **iptables**, **ip6tables**, **arptables** и **ebtables**, которые в настоящее время устарели. **Nftables** использует единую утилиту **nft** для пользовательского пространства, что позволяет управлять всеми протоколами через единый интерфейс, устранив историческую конкуренцию, вызванную различными внешними интерфейсами и несколькими интерфейсами **netfilter**.

Знакомство с firewalld

Firewalld — это динамический менеджер брандмауэра, внешний интерфейс к инфраструктуре **nftables**, использующий команду **nft**. До появления **nftables** **firewalld** использовал команду **iptables** для непосредственной настройки **netfilter** в качестве улучшенной альтернативы службе **iptables**. В RHEL 8 **firewalld** остается рекомендуемым внешним интерфейсом, управляющим наборами правил брандмауэра с помощью **nft**. **Firewalld** по-прежнему способен читать и управлять файлами конфигурации и наборами правил **iptables**, используя **xtables-nft-multi** для преобразования объектов **iptables** непосредственно в правила и объекты **nftables**. Хотя это настоятельно не рекомендуется, **firewalld** можно настроить для возврата к серверной части

iptables для сложных случаев использования, когда существующие наборы правил **iptables** не могут быть должным образом обработаны трансляциями **nft**.

Приложения опрашивают подсистему с помощью интерфейса **D-Bus**. Подсистема **firewalld**, доступная в RPM-пакете **firewalld**, не включена в минимальную установку, но включена в базовую установку. С помощью **firewalld** управление брандмаузом упрощается за счет классификации всего сетевого трафика по зонам. На основе таких критериев, как исходный IP-адрес пакета или входящий сетевой интерфейс, трафик направляется в правила брандмауэра для соответствующей зоны. Каждая зона имеет свой собственный список портов и служб, которые либо открыты, либо закрыты.



ПРИМЕЧАНИЕ

Для ноутбуков или других машин, которые регулярно меняют сети, **NetworkManager** можно использовать для автоматической установки зоны брандмауэра для соединения. Зоны настраиваются с помощью правил, подходящих для конкретных соединений.

Это особенно полезно при перемещении между домом, работой и общедоступными беспроводными сетями. Пользователь может захотеть, чтобы служба **sshd** его системы была доступна при подключении к домашней и корпоративной сетям, но не при подключении к общедоступной беспроводной сети в местной кофейне.

Firewalld проверяет исходный адрес для каждого пакета, поступающего в систему. Если этот исходный адрес назначен определенной зоне, применяются правила для этой зоны. Если исходный адрес не назначен зоне, **firewalld** связывает пакет с зоной для входящего сетевого интерфейса, и применяются правила для этой зоны. Если сетевой интерфейс по какой-либо причине не связан с зоной, то **firewalld** связывает пакет с зоной по умолчанию.

Зона по умолчанию (**default**) не является отдельной зоной, а является обозначением существующей зоны. Первоначально **firewalld** назначает общедоступную зону по умолчанию и сопоставляет интерфейс **lo loopback** с доверенной зоной.

Большинство зон разрешают трафик через брандмауэр, который соответствует списку определенных портов и протоколов, таких как **631/udp**, или предопределенных служб, таких как **ssh**. Если трафик не соответствует разрешенному порту и протоколу или службе, он обычно отклоняется. (Доверенная зона, которая по умолчанию разрешает весь трафик, является одним исключением из этого.)

Предопределенные зоны

Firewalld имеет предопределенные зоны, каждую из которых вы можете настроить. По умолчанию все зоны разрешают любой входящий трафик, являющийся частью связи, инициированной системой, и весь исходящий трафик. В следующей таблице подробно описана начальная конфигурация зоны.

Конфигурация зон Firewalld по умолчанию

НАЗВАНИЕ ЗОНЫ	КОНФИГУРАЦИЯ ПО УМОЛЧАНИЮ
trusted	Разрешить весь входящий трафик.
home	Отклонять входящий трафик, если он не связан с исходящим трафиком или не соответствует предопределенным службам ssh , mdns , ipp-client , samba-client или dhcpv6-client .
internal	Отклонять входящий трафик, если он не связан с исходящим трафиком или не соответствует предопределенным службам ssh , mdns , ipp-client , samba-client или dhcpv6-client (то же самое, что и домашняя зона для начала).
work	Отклонять входящий трафик, если он не связан с исходящим трафиком или не соответствует предопределенным службам ssh , ipp-client или dhcpv6-client .
public	Отклонять входящий трафик, если он не связан с исходящим трафиком или не соответствует предопределенным службам ssh или dhcpv6-client . Зона по умолчанию для недавно добавленных сетевых интерфейсов.
external	Отклонять входящий трафик, если он не связан с исходящим трафиком или не соответствует предопределенной службе ssh . Исходящий IPv4-трафик, направляемый через эту зону, маскируется, чтобы выглядеть так, как будто он исходит из IPv4-адреса исходящего сетевого интерфейса.
dmz	Отклонять входящий трафик, если он не связан с исходящим трафиком или не соответствует предопределенной службе ssh .
block	Отклонять весь входящий трафик, если он не связан с исходящим трафиком.
drop	Отбрасывать весь входящий трафик, если он не связан с исходящим трафиком (даже не отвечать ошибками ICMP).

Список доступных предопределенных зон и предполагаемого использования см. в **firewalld.zones(5)**.

Предустановленные услуги

Firewalld имеет ряд предустановленных сервисов. Определения службы помогают настроить прохождение пакетов для конкретных сетевых служб. Например, вместо того, чтобы искать соответствующие порты для службы **samba-client**, укажите предварительно созданную

службу **samba-client** для настройки правильных портов и протоколов. В следующей таблице перечислены предварительно определенные службы, используемые в начальной конфигурации зон брандмауэра.

Выбранные предварительно определенные службы Firewalld

НАИМЕНОВАНИЕ СЕРВИСА	КОНФИГУРАЦИЯ
ssh	Локальный SSH-сервер. Трафик на 22/tcp
dhcpv6-client	Локальный клиент DHCPv6 . Трафик на 546/udp в сети fe80::/64 IPv6
ipp-client	Локальная печать IPP . Трафик на 631/udp .
samba-client	Локальный клиент общего доступа к файлам и принтерам Windows. Трафик на 137/udp и 138/udp .
mdns	Разрешение имен локальных ссылок многоадресной DNS (mDNS). Трафик на 5353/udp на многоадресные адреса 224.0.0.251 (IPv4) или ff02::fb (IPv6) .



ПРИМЕЧАНИЕ

Многие предопределенные службы включены в пакет **firewalld**. Используйте команду **firewallcmd --get-services**, чтобы получить их список. Файлы конфигурации для предопределенных служб находятся в каталоге **/usr/lib/firewalld/services** в формате, определяемом **firewalld.zone(5)**.

Либо используйте предварительно определенные службы, либо напрямую укажите требуемый порт и протокол. Графический интерфейс веб-консоли используется для просмотра предопределенных служб и определения дополнительных служб.

НАСТРОЙКА БРАНДМАУЭРА

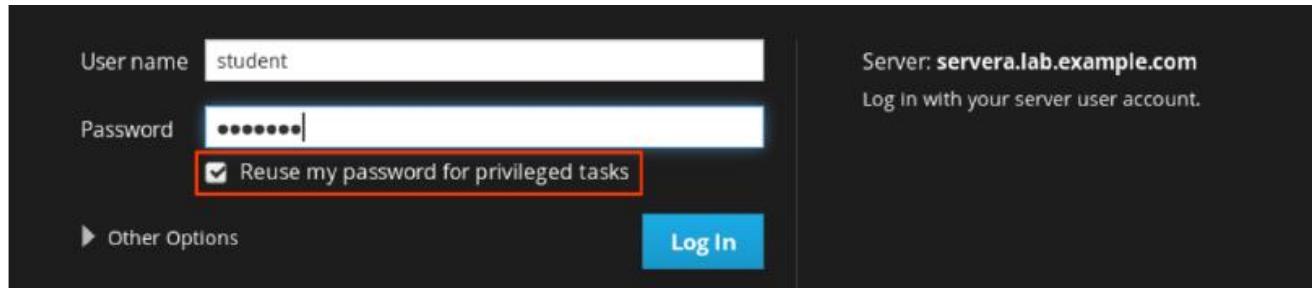
Системные администраторы взаимодействуют с **firewalld** тремя способами:

- Непосредственно редактировать файлы конфигурации в **/etc/firewalld/** (не обсуждается в этой главе)
- Графический интерфейс веб-консоли
- Инструмент командной строки **firewall-cmd**

Настройка служб брандмауэра с помощью веб-консоли

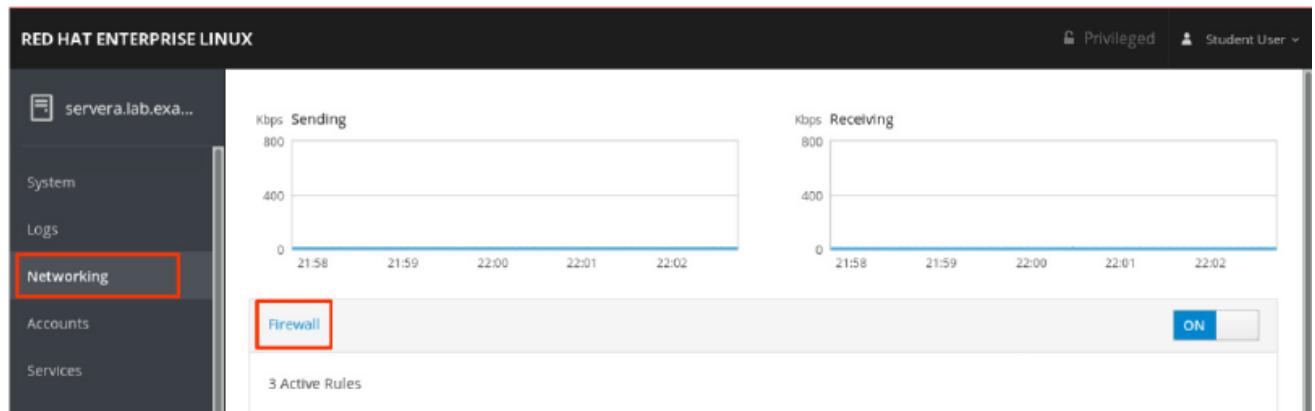
Чтобы настроить службы брандмауэра с помощью веб-консоли, войдите в систему с привилегированным доступом, щелкнув ***Reuse my password for privileged tasks***. Это позволяет пользователю выполнять команды с привилегиями ***sudo*** для изменения служб брандмауэра.

Рисунок 11.1: Привилегированный вход в веб-консоль



Щелкните параметр «Сеть (*Networking*)» в левом меню навигации, чтобы отобразить раздел «Брандмауэр (*Firewall*)» на главной странице сети. Щелкните ссылку Брандмауэр (*Firewall*), чтобы получить доступ к списку разрешенных служб.

Рисунок 11.2: Веб-консоль настройки NETWORKING



Перечисленные разрешенные службы — это те, которые в настоящее время разрешены брандмауэром. Щелкните стрелку (>) слева от имени службы, чтобы просмотреть сведения о службе. Чтобы добавить службу, нажмите кнопку «Добавить службы... (*Add Services...*)» в правом верхнем углу страницы «Разрешенные службы брандмауэра (*Firewall Allowed Services*)».

Рисунок 11.3: Список разрешенных служб брандмауэра в веб-консоли

The screenshot shows the 'Networking > Firewall' section of the Red Hat Enterprise Linux interface. On the left, a sidebar lists various system management options like System, Logs, Networking, Accounts, Services, Applications, Diagnostic Reports, Kernel Dump, SELinux, Software Updates, Subscriptions, and Terminal. The main panel displays the 'Firewall' status as 'ON'. Below it, the 'Allowed Services' table lists three services: Cockpit (TCP port 9090), DHCPv6 Client (TCP port 546), and SSH (TCP port 22). A red box highlights the 'Add Services...' button in the top right corner of the main panel. Another red box highlights the first service entry ('Cockpit') in the list.

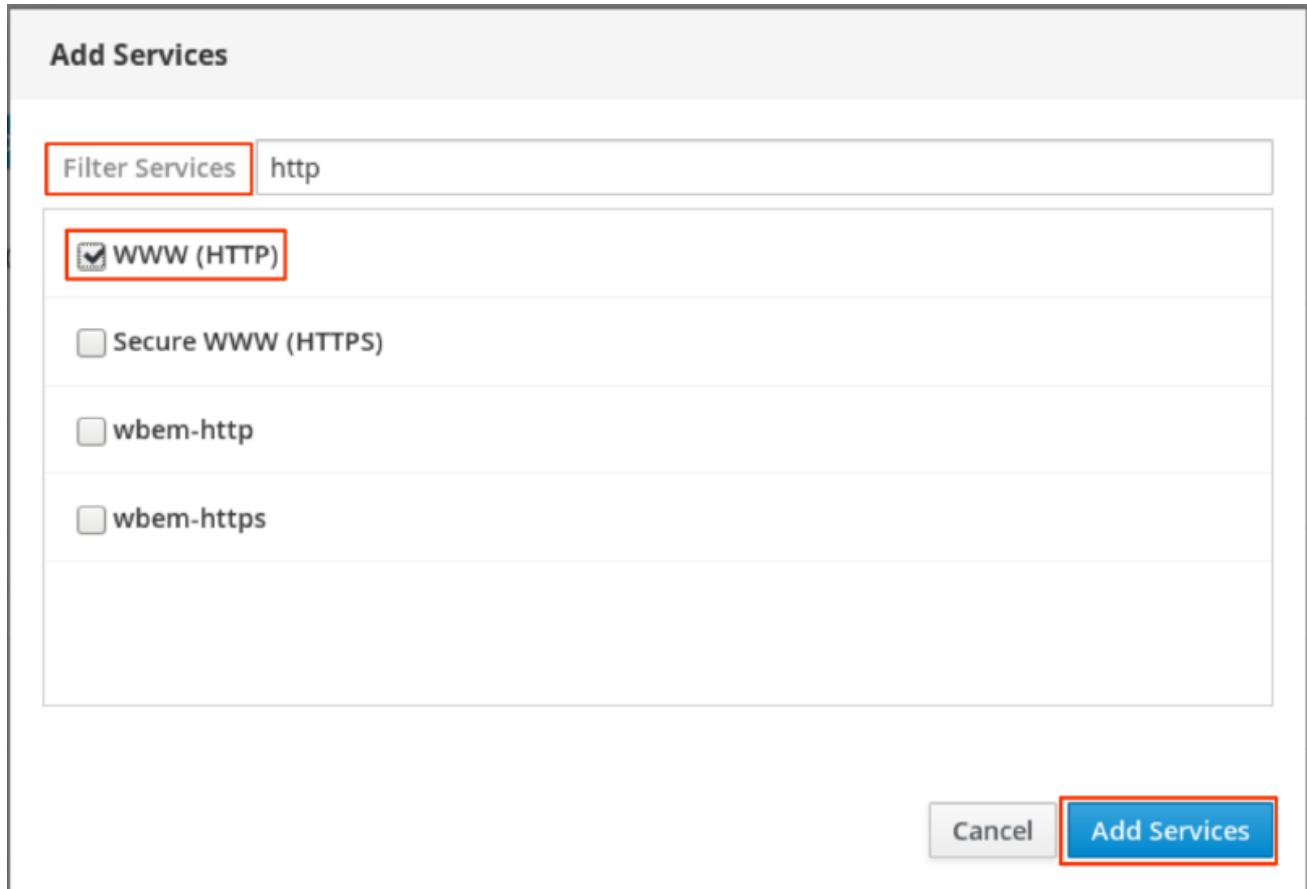
На странице «Добавить службы (*Add Services*)» отображаются доступные предварительно определенные службы.

Рисунок 11.4: Интерфейс добавления служб веб-консоли

The screenshot shows the 'Add Services' dialog box. At the top, a title bar contains the text 'Add Services' in a red-bordered box. Below it is a 'Filter Services' input field. The main area lists several services with checkboxes: Red Hat Satellite 6, Amanda Backup Client, Amanda Backup Client (kerberized), amqp, amqps, and apcupsd. At the bottom right of the dialog are 'Cancel' and 'Add Services' buttons, with 'Add Services' also in a red-bordered box.

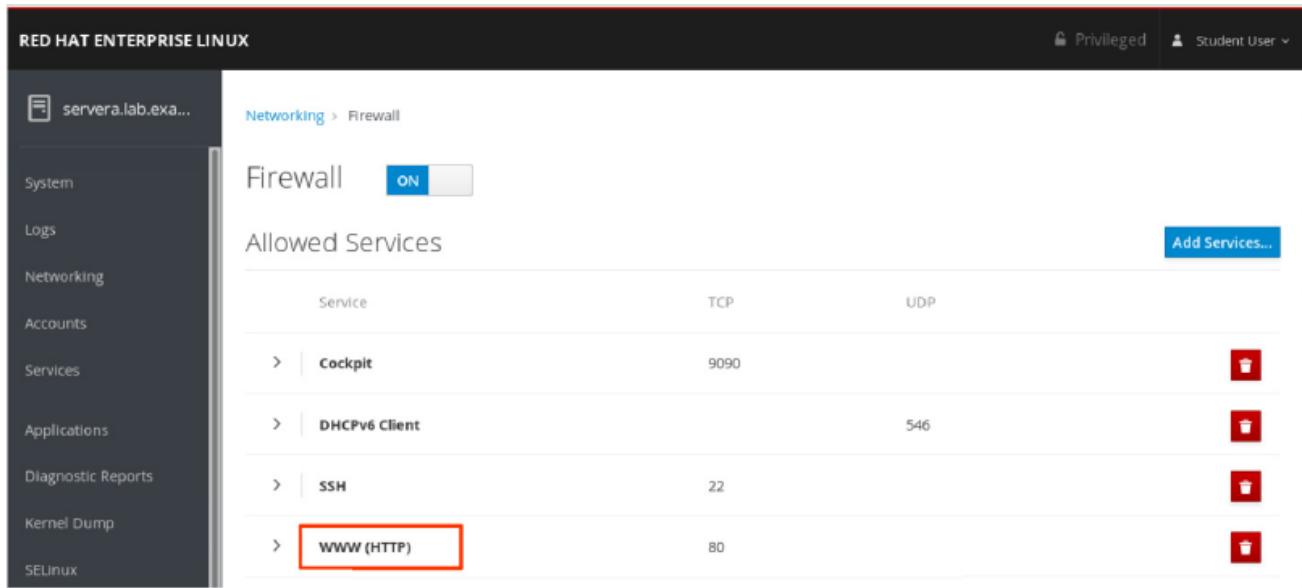
Чтобы выбрать службу, прокрутите список или введите значение в текстовое поле «Службы фильтра (*Filter Services*)». В следующем примере строка **http** вводится в текстовое поле поиска, чтобы найти службы, содержащие эту строку; то есть веб-сервис. Установите флажок слева от служб, чтобы разрешить доступ через брандмауэр. Нажмите кнопку «Добавить службы (*Add Services*)», чтобы завершить процесс.

Рисунок 11.5: Фильтр поиска служб веб-консоли



Интерфейс *Firewall Allowed Services* возвращается на страницу разрешенных служб брандмауэра, где вы можете просмотреть обновленный список разрешенных служб.

Рисунок 11.6: Список служб веб-консоли



Настройка брандмауэра из командной строки

Команда **firewall-cmd** взаимодействует с менеджером динамического брандмауэра **firewalld**. Он устанавливается как часть основного пакета **firewalld** и доступен для администраторов, которые предпочитают работать в командной строке, для работы в системах без графического окружения или для сценариев настройки брандмауэра.

В следующей таблице перечислены некоторые часто используемые **команды firewall-cmd** вместе с пояснениями. Обратите внимание, что если не указано иное, почти все команды будут работать в конфигурации среды (*runtime*) выполнения, если не указан параметр **--permanent**. Если указан параметр **--permanent**, вы должны активировать этот параметр, также выполнив команду **firewall-cmd --reload**, которая считывает текущую постоянную конфигурацию и применяет ее в качестве новой конфигурации среды выполнения. Многие из перечисленных команд используют параметр **--zone=ZONE**, чтобы определить, на какую зону они влияют. Если требуется маска сети, используйте **нотацию CIDR**, например **192.168.1/24**.

FIREWALL-CMD COMMANDS	ОБЪЯСНЕНИЕ КОМАНДЫ
--get-default-zone	Запросить текущую зону по умолчанию.
--set-default-zone=ZONE	Установите зону по умолчанию. Это изменяет как среду выполнения, так и постоянную конфигурацию.
--get-zones	Список всех доступных зон.
--get-active-zones	Перечислите все используемые в настоящее время зоны (имейте интерфейс или источник, привязанный к ним), а также информацию об их интерфейсе и источнике.
--add-source=CIDR [--zone=ZONE]	Направлять весь трафик, поступающий с IP-адреса или сети/маски сети, в указанную зону. Если опция --zone= не указана, используется зона по умолчанию.

--remove-source=CIDR [--zone=ZONE]	Удалите правило маршрутизации всего трафика из зоны, поступающего с IP-адреса или сети/сетевой маски. Если опция --zone= не указана, используется зона по умолчанию.
--add-interface=INTERFACE [--zone=ZONE]	Направить весь трафик, поступающий от INTERFACE , в указанную зону. Если опция --zone= не указана, используется зона по умолчанию.
--change-interface=INTERFACE [-- zone=ZONE]	Свяжите интерфейс с ZONE вместо его текущей зоны. Если опция --zone= не указана, используется зона по умолчанию.
--list-all [--zone=ZONE]	Перечислите все настроенные интерфейсы, источники, службы и порты для ZONE . Если опция --zone= не указана, используется зона по умолчанию.
--list-all-zones	Получить всю информацию для всех зон (интерфейсы, источники, порты, сервисы).
--add-service=SERVICE [--zone=ZONE]	Разрешить трафик на SERVICE . Если опция --zone= не указана, используется зона по умолчанию.
--add-port=PORT/PROTOCOL [--zone=ZONE]	Разрешить трафик на порт(ы) PORT/PROTOCOL . Если опция --zone= не указана, используется зона по умолчанию.
--remove-service=SERVICE [--zone=ZONE]	Удалите SERVICE из списка разрешенных для зоны. Если опция --zone= не указана, используется зона по умолчанию.
--remove-port=PORT/PROTOCOL [--zone=ZONE]	Удалите порт(ы) PORT/PROTOCOL из списка разрешенных для зоны. Если опция --zone= не указана, используется зона по умолчанию.
--reload	Отбросьте конфигурацию среды выполнения и примените постоянную конфигурацию.

Приведенные ниже примеры команд устанавливают зону по умолчанию в **dmz**, назначают весь трафик, поступающий из сети **192.168.0.0/24**, в зону **internal** и открывают сетевые порты для службы **mysql** в зоне **internal**.

```
[root@host ~]# firewall-cmd --set-default-zone=dmz
[root@host ~]# firewall-cmd --permanent --zone=internal \
--add-source=192.168.0.0/24
[root@host ~]# firewall-cmd --permanent --zone=internal --add-service=mysql
[root@host ~]# firewall-cmd --reload
```



ПРИМЕЧАНИЕ

В ситуациях, когда базового синтаксиса **firewalld** недостаточно, вы также можете добавить расширенные правила, более выразительный синтаксис, для написания сложных правил. Если даже синтаксиса расширенных правил недостаточно, вы также можете использовать правила прямой настройки, необработанный синтаксис **nft**, смешанный с правилами **firewalld**.

Эти расширенные режимы выходят за рамки этой главы.



РЕКОМЕНДАЦИИ

Справочные страницы **man firewall-cmd(1)**, **firewalld(1)**, **firewalld.zone(5)**, **firewalld.zones(5)**, и **nft(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ СЕРВЕРНЫМИ БРАНДМАУЕРАМИ

В этом упражнении вы будете контролировать доступ к системным службам, настраивая правила системного брандмауэра с помощью **firewalld**.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность настроить правила брандмауэра для управления доступом к службам.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab netsecurity-firewalls start**. Команда запускает сценарий, чтобы определить, доступен ли узел **servera** в сети.

```
[student@workstation ~]$ lab netsecurity-firewalls start
```

1. С рабочей станции **workstation**, используйте **SSH** для входа на сервер как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Убедитесь, что в системе **servera** установлены пакеты **httpd** и **mod_ssl**. Эти пакеты предоставляют веб-сервер **Apache**, который вы будете защищать с помощью брандмауэра, и необходимые расширения для веб-сервера, чтобы обслуживать контент через **SSL**.

```
[student@servera ~]$ sudo yum install httpd mod_ssl  
[sudo] password for student: student  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Complete!
```

3. Как пользователь **student** на **servera** создайте файл **/var/www/html/index.html**. Добавьте одну строку текста, которая гласит: **I am servera**.

```
[student@servera ~]$ sudo bash -c \  
I am servera
```

```
"echo 'I am servera.' > /var/www/html/index.html"
```

4. Запустите и включите службу **httpd** в вашей системе **servera**.

```
[student@servera ~]$ sudo systemctl enable --now httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service
→ /usr/lib/systemd/system/httpd.service.
```

5. Выйти из сервера **servera**.

```
[student@servera ~]$ exit
logout
Connection to servera closed.
[student@workstation ~]$
```

6. С рабочей станции **workstation** попытайтесь получить доступ к своему веб-серверу на **servera**, используя порт **80/TCP** и порт **443/TCP** с инкапсуляцией **SSL**. Обе попытки должны быть неудачными.

- 6.1. Эта команда должна завершиться ошибкой:

```
[student@workstation ~]$ curl -k http://servera.lab.example.com
curl: (7) Failed to connect to servera.lab.example.com port 80: No
route to host
```

- 6.2. Эта команда также должна завершиться ошибкой:

```
[student@workstation ~]$ curl -k https://servera.lab.example.com
curl: (7) Failed to connect to servera.lab.example.com port 443: No
route to host
```

7. Войдите на сервер **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

8. На сервере **servera**, убедитесь, что служба **nftables** замаскирована (**masked**), а служба **firewalld** включена и работает.

- 8.1. Определите, **masked** ли состояние службы **nftables**.

```
[student@servera ~]$ sudo systemctl status nftables
[sudo] password for student:
● nftables.service - Netfilter Tables
  Loaded: loaded (/usr/lib/systemd/system/nftables.service; disabled;
  vendor preset: disabled)
    Active: inactive (dead)
      Docs: man:nft(8)
```

Результаты показывают, что **nftables** отключен и неактивен, но не замаскирован(**masked**). Выполните следующую команду, чтобы это сделать.

```
[student@servera ~]$ sudo systemctl mask nftables
Created symlink /etc/systemd/system/nftables.service → /dev/null.
```

8.2. Убедитесь, что статус службы **nftables masked**.

```
[student@servera ~]$ sudo systemctl status nftables
● nftables.service
  Loaded: masked (Reason: Unit nftables.service is masked.)
  Active: inactive (dead)
```

8.3. Убедитесь, что статус службы **firewalld** включен (**enabled**) и работает (**running**).

```
[student@servera ~]$ sudo systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled;
  vendor preset: enabled)
    Active: active (running) since Wed 2019-05-22 15:36:02 CDT; 5min
      ago
        Docs: man:firewalld(1)
      Main PID: 703 (firewalld)
        Tasks: 2 (limit: 11405)
      Memory: 29.8M
      CGroup: /system.slice/firewalld.service
              └─703 /usr/libexec/platform-python -s /usr/sbin/firewalld
                  --nofork --nopid

May 22 15:36:01 jegui.ilt.example.com systemd[1]: Starting firewalld -
dynamic firewall daemon...
May 22 15:36:02 jegui.ilt.example.com systemd[1]: Started firewalld -
dynamic firewall daemon.
```

8.4. Выход из сервера **servera**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

- 9.** На рабочей станции **workstation**, откройте **Firefox** и войдите в веб-консоль, работающую на сервере **servera**, чтобы добавить службу **httpd** в сетевую зону **public**.
- 9.1.** Откройте **Firefox** и перейдите по адресу **https://servera.lab.example.com:9090**, чтобы получить доступ к веб-консоли. Примите само заверяющий сертификат, используемый **servera**, добавив исключение.
- 9.2.** Установите флажок «Повторно использовать мой пароль для привилегированных задач (*Reuse my password for privileged tasks*)», чтобы обеспечить административные привилегии.
Войдите в систему как пользователь **student**, использовать слово **student** в качестве пароля.
- 9.3.** Щелкните **Networking** на левой панели навигации.
- 9.4.** Щелкните ссылку «**Firewall**» на главной странице «**Networking**».
- 9.5.** Нажмите кнопку «**Add Services...**», расположенную в верхней правой части страницы брандмауэра (**Firewall**).
- 9.6.** В пользовательском интерфейсе **Add Services** прокрутите вниз или используйте **Filter Services**, чтобы найти и установить флажок рядом со службой **Secure WWW (HTTPS)**.
- 9.7.** Нажмите кнопку «**Add Services**», расположенную в нижней правой части пользовательского интерфейса «**Add Services**».

- 10.** Вернитесь к терминалу на рабочей станции **workstation** и проверьте свою работу, попытавшись просмотреть содержимое веб-сервера **servera**.

- 10.1.** Эта команда должна завершиться ошибкой:

```
[student@workstation ~]$ curl -k http://servera.lab.example.com  
curl: (7) Failed to connect to servera.lab.example.com port 80: No  
route to host
```

- 10.2.** Эта команда должна быть успешной:

```
[student@workstation ~]$ curl -k https://servera.lab.example.com  
I am servera.
```



ПРИМЕЧАНИЕ

Если вы используете **Firefox** для подключения к веб-серверу, он запросит проверку сертификата хоста, если он успешно пройдет через брандмауэр.

Завершение

На рабочей станции **workstation**, запустите сценарий **lab netsecurity-firewalls finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab netsecurity-firewalls finish
```

На этом управляемое упражнение завершенно.

УПРАВЛЕНИЕ МАРКИРОВКОЙ ПОРТОВ SELINUX

ЦЕЛИ

После завершения этого раздела вы должны уметь проверять, что сетевые порты имеют правильный тип **SELinux**, чтобы службы могли к ним привязываться.

МАРКИРОВКА ПОРТОВ SELINUX

SELinux делает больше, чем просто маркирует файлы и процессы. Сетевой трафик также строго контролируется политикой **SELinux**. Один из методов, который SELinux использует для управления сетевым трафиком, является маркировка сетевых портов; например, в целевой политике порт 22/TCP имеет связанную с ним метку **ssh_port_t**. Порты **HTTP** по умолчанию, **80/TCP** и **443/TCP**, имеют связанную с ними метку **http_port_t**.

Всякий раз, когда процесс хочет прослушивать порт, **SELinux** проверяет, разрешено ли метке, связанной с этим процессом (доменом), связывать эту метку порта. Это может помешать мошеннической службе захватить порты, используемые другими (законными) сетевыми службами.

УПРАВЛЕНИЕ МАРКИРОВКОЙ ПОРТОВ SELINUX

Если вы решите запустить службу на нестандартном порту, **SELinux** почти наверняка заблокирует трафик. В этом случае необходимо обновить **метки портов SELinux**. В некоторых случаях целевая политика уже пометила порт с типом, который можно использовать; например, поскольку порт **8008/TCP** часто используется для веб-приложений, этот порт уже помечен как **http_port_t** тип порта по умолчанию для веб-сервера.

Список меток портов

Чтобы получить обзор всех текущих назначений меток портов, выполните команду **semanage port -l**. Опция **-l** выводит список всех текущих назначений в такой форме:

<i>port_label_t</i>	<i>tcp/udp</i>	<i>comma-separated,list,of,ports</i>
---------------------	----------------	--------------------------------------

Пример вывода:

[root@host ~]# semanage port -l	<i>...output omitted...</i>
http_cache_port_t	tcp 8080, 8118, 8123, 10001-10010
http_cache_port_t	udp 3130
http_port_t	tcp 80, 81, 443, 488, 8008, 8009, 8443, 9000
<i>...output omitted...</i>	

Для уточнения поиска используйте команду **grep**:

```
[root@host ~]# semanage port -l | grep ftp
ftp_data_port_t          tcp      20
ftp_port_t                tcp      21, 989, 990
ftp_port_t                udp      989, 990
tftp_port_t               udp      69
```

Обратите внимание, что метка порта может появляться в выводе дважды: один раз для **TCP** и один раз для **UDP**.

Управление метками портов

Используйте команду **semanage**, чтобы назначить новые метки портов, удалить метки портов или изменить существующие.



ВАЖНО

Большинство стандартных служб, доступных в дистрибутиве Linux, предоставляют модуль политики **SELinux**, который устанавливает метки для портов. Вы не можете изменить метки этих портов с помощью **semanage**; чтобы изменить их, вам нужно заменить модуль политики. Написание и создание модулей политик выходит за рамки данного курса.

Чтобы добавить порт к существующей метке порта (типу (**type**)), используйте следующий синтаксис. **-a** добавляет новую метку порта, **-t** обозначает тип, **-p** обозначает протокол.

```
[root@host ~]# semanage port -a -t port_label -p tcp/udp PORTNUMBER
```

Например, чтобы разрешить сервису **gopher** прослушивать порт 71/TCP:

```
[root@host~]# semanage port -a -t gopher_port_t -p tcp 71
```

Чтобы просмотреть локальные изменения политики по умолчанию, администраторы могут добавить параметр **-C** к команде **semanage**.

```
[root@host~]# semanage port -l -C
SELinux Port Type      Proto      Port Number
gopher_port_t           tcp        71
```



ПРИМЕЧАНИЕ

Политика **targeted** поставляется с большим количеством типов портов.

Справочные страницы **SELinux** для конкретных служб, найденные в пакете *selinux-policy-doc*, включают документацию по типам **SELinux**, логическим значениям и типам портов. Если эти справочные страницы еще не установлены в вашей системе, выполните следующую процедуру:

```
[root@host ~]# yum -y install selinux-policy-doc  
[root@host ~]# man -k _selinux
```

Удаление меток портов

Синтаксис удаления пользовательской метки порта аналогичен синтаксису добавления метки порта, но вместо использования параметра **-a** (для добавления) используйте параметр **-d** (для удаления).

Например, чтобы убрать привязку порта **71/TCP** к **gopher_port_t**:

```
[root@host ~]# semanage port -d -t gopher_port_t -p tcp 71
```

Изменение привязки портов

Чтобы изменить привязку порта, возможно, из-за изменения требований, используйте параметр **-m** (изменить (**Modify**)). Это более эффективный процесс, чем удаление старой привязки и добавление новой.

Например, чтобы изменить порт **71/TCP** с **gopher_port_t** на **http_port_t**, администратор может использовать следующую команду:

```
[root@server ~]# semanage port -m -t http_port_t -p tcp 71
```

Как и раньше, просмотрите модификацию с помощью команды **semanage**.

```
[root@server ~]# semanage port -l -c
SELinux Port Type          Proto   Port Number

http_port_t                  tcp     71
[root@server ~]# semanage port -l | grep http
http_cache_port_t            tcp     8080, 8118, 8123, 10001-10010
http_cache_port_t            udp     3130
http_port_t                  tcp     71, 80, 81, 443, 488, 8008, 8009, 8443,
                                9000
pegasus_http_port_t          tcp     5988
pegasus_https_port_t         tcp     5989
```



РЕКОМЕНДАЦИИ

Справочные страницы **man semanage(8)**, **semanage-port(8)**, и ***_selinux(8)**

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УПРАВЛЕНИЕ МАРКИРОВКОЙ ПОРТОВ SELINUX

В этой лабораторной работе вы настроите свою систему **servera**, чтобы разрешить **HTTP**-доступ через нестандартный порт.

В РЕЗУЛЬТАТЕ:

Вы настроите веб-сервер, работающий на сервере **servera**, успешно обслуживающий контент с использованием нестандартного порта.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab netsecurity-ports start**. Эта команда запускает сценарий, который определяет, доступен ли сервер **servera** в сети. Он также устанавливает службу **httpd** и настраивает брандмауэр на сервере для разрешения соединений **http**.

```
[student@workstation ~]$ lab netsecurity-ports start
```

Ваша организация развертывает новое пользовательское веб-приложение. Веб-приложение работает на нестандартном порту; в данном случае **82/TCP**.

Один из ваших младших администраторов уже настроил приложение на вашем сервере. Однако содержимое веб-сервера недоступно.

1. Используйте команду **ssh**, чтобы войти на сервер **servera** как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Используйте команду **sudo -i**, чтобы переключиться на пользователя **root**. Пароль для пользователя **student** - **student**.

```
[student@servera ~]$ sudo -i  
[sudo] password for student: student  
[root@servera ~]#
```

3. Попытайтесь решить проблему с веб-контентом, перезапустив службу **httpd**.

3.1. Используйте команду `systemctl` для перезапуска `httpd.service`. Ожидается, что эта команда завершится ошибкой.

```
[root@servera ~]# systemctl restart httpd.service
Job for httpd.service failed because the control process exited with
error code.
See "systemctl status httpd.service" and "journalctl -xe" for details.
```

3.2. Используйте команду `systemctl status -l`, чтобы узнать статус службы `httpd`. Обратите внимание на ошибку `permission denied`.

```
[root@servera ~]# systemctl status -l httpd.service
● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled;
  vendor preset: disabled)
    Active: failed (Result: exit-code) since Mon 2019-04-08 14:23:29
  CEST; 3min 33s ago
      Docs: man:httpd.service(8)
    Process: 28078 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
   (code=exited, status=1/FAILURE)
   Main PID: 28078 (code=exited, status=1/FAILURE)
     Status: "Reading configuration..."

Apr 08 14:23:29 servera.lab.example.com systemd[1]: Starting The
Apache HTTP Server...
Apr 08 14:23:29 servera.lab.example.com httpd[28078]: (13)Permission
denied: AH00072: make_sock: could not bind to address [::]:82
Apr 08 14:23:29 servera.lab.example.com httpd[28078]: (13)Permission
denied: AH00072: make_sock: could not bind to address 0.0.0.0:82
Apr 08 14:23:29 servera.lab.example.com httpd[28078]: no listening
sockets available, shutting down
Apr 08 14:23:29 servera.lab.example.com httpd[28078]: AH00015: Unable
to open logs
Apr 08 14:23:29 servera.lab.example.com systemd[1]: httpd.service:
Main process exited, code=exited, status=1/FAILURE
Apr 08 14:23:29 servera.lab.example.com systemd[1]: httpd.service:
Failed with result 'exit-code'.
```

3.3. Используйте команду `sealert`, чтобы проверить, не блокирует ли **SELinux** привязку `httpd` к порту **82/TCP**.

```
[root@servera ~]# sudo sealert -a /var/log/audit/audit.log
100% done
```

```
found 1 alerts in /var/log/audit/audit.log
```

```
SELinux is preventing /usr/sbin/httpd from name_bind access on the
tcp_socket port 82.
```

```
***** Plugin bind_ports (99.5 confidence) suggests
*****
```

```
If you want to allow /usr/sbin/httpd to bind to network port 82
Then you need to modify the port type.
```

```
Do
```

```
# semanage port -a -t PORT_TYPE -p tcp 82
    where PORT_TYPE is one of the following: http_cache_port_t,
http_port_t, jboss_management_port_t, jboss.messaging_port_t,
ntop_port_t, puppet_port_t.
...output omitted...
```

```
Raw Audit Messages
```

```
type=AVC msg=audit(1554726569.188:852): avc:
denied { name_bind } for pid=28393 comm="httpd"
src=82 scontext=system_u:system_r:httpd_t:s0
tcontext=system_u:object_r:reserved_port_t:s0 tclass=tcp_socket
permissive=0
```

- Настройте SELinux, чтобы разрешить привязку **httpd** к порту **82/TCP**, затем перезапустите службу **httpd.service**.

- Используйте команду **semanage**, чтобы найти подходящий тип порта для **порта 82/TCP**. **http_port_t** содержит порты **HTTP** по умолчанию, **80/TCP** и **443/TCP**. Это правильный тип порта для веб-сервера.

```
[root@servera ~]# semanage port -l | grep http
http_cache_port_t          tcp      8080, 8118, 8123, 10001-10010
http_cache_port_t          udp      3130
http_port_t                tcp      80, 81, 443, 488, 8008, 8009,
                                8443, 9000
pegasus_http_port_t         tcp      5988
pegasus_https_port_t        tcp      5989
```

- Используйте команду **semanage**, чтобы назначить порту **82/TCP** тип **http_port_t**.

```
[root@servera ~]# semanage port -a -t http_port_t -p tcp 82
```

- Используйте команду **systemctl** для перезапуска службы **httpd.service**. Эта команда должна быть успешной.

```
[root@servera ~]# systemctl restart httpd.service
```

- Проверьте, можете ли вы теперь получить доступ к веб-серверу, работающему на порту **82/TCP**. Используйте команду **curl** для доступа к веб-службе с сервера.

```
[root@servera ~]# curl http://servera.lab.example.com:82
Hello
```

- В другом окне терминала проверьте, можете ли вы получить доступ к новой веб-службе с рабочей станции **workstation**. Используйте команду **curl** для доступа к веб-службе с рабочей станции.

```
[student@workstation ~]$ curl http://servera.lab.example.com:82
curl: (7) Failed to connect to servera.example.com:82; No route to host
```

Эта ошибка означает, что вы по-прежнему не можете подключиться к веб-службе с рабочей станции.

- На сервере **servera**, откройте порт **82/TCP** на брандмауэре.

7.1. Используйте команду **firewall-cmd**, чтобы открыть порт **82/TCP** в постоянной конфигурации для зоны **default** на брандмауэре на сервере.

```
[root@servera ~]# firewall-cmd --permanent --add-port=82/tcp
success
```

7.2. Активируйте изменения брандмауэра на сервере **servera**.

```
[root@servera ~]# firewall-cmd --reload
Success
```

- Используйте команду **curl** для доступа к веб-службе с рабочей станции **workstation**.

```
[student@workstation ~]$ curl http://servera.lab.example.com:82
Hello
```

- Выход из сервера **servera**.

```
[root@servera ~]# exit  
logout  
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab netsecurity-ports finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab netsecurity-ports finish
```

На этом упражнения с пошаговыми инструкциями заканчивается.

ЛАБОРАТОРНАЯ РАБОТА

УПРАВЛЕНИЕ СЕТЕВОЙ БЕЗОПАСНОСТЬЮ

СПИСОК РАБОТЫ

В этой лабораторной работе вы настроите параметры брандмауэра и **SELinux**, чтобы разрешить доступ к нескольким веб-серверам, работающим на **serverb**.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность настроить параметры брандмауэра и **SELinux** на хосте веб-сервера.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab netsecurity-review start**. Команда запускает сценарий, который определяет, доступен ли хост **serverb** в сети.

```
[student@workstation ~]$ lab netsecurity-review start
```

Ваша компания решила запустить новое веб-приложение. Это приложение прослушивает порты **80/TCP** и **1001/TCP**. Также должен быть доступен порт **22/TCP** для доступа по **ssh**. Все внесенные вами изменения должны сохраняться после перезагрузки.

При запросе **sudo** используйте слово **student** в качестве пароля.

Важно: Графический интерфейс, используемый в среде онлайн-обучения Red Hat, также должен оставаться доступным через порт **5900/TCP**. Этот порт также известен под именем службы **vncserver**. Если вы случайно заблокировали себя на своем **serverb**, вы можете либо попытаться восстановиться с помощью **ssh** на вашем сервере с вашей рабочей станции, либо перезагрузить свой **serverb**. Если вы решите перезагрузить машину **serverb**, вам придется снова запустить сценарии установки для этого практического занятия. Конфигурация на ваших машинах уже включает настраиваемую зону под названием **ROL**, которая открывает эти порты.

1. С рабочей станции **workstation**, проверьте доступ к веб-серверу по умолчанию по адресу **http://server.lab.example.com** и к виртуальному хосту по адресу **http://server.lab.example.com:1001**.
2. Войдите в систему **serverb**, чтобы определить, что препятствует доступу к веб-серверам.
3. Настройте **SELinux**, чтобы служба **httpd** прослушивала порт **1001/TCP**.
4. С рабочей станции проверьте доступ к веб-серверу по умолчанию по адресу **http://server.lab.example.com** и к виртуальному хосту по адресу **http://server.lab.example.com:1001**.
5. Войдите в систему **serverb**, чтобы определить, правильные ли порты назначены брандмауэру.

6. Добавьте порт **1001/TCP** в постоянную конфигурацию сетевой зоны **public**. Подтвердите вашу конфигурацию.
7. С рабочей станции **workstation** убедитесь, что веб-сервер по умолчанию на **serverb.lab.example.com** возвращает **SERVER B**, а виртуальный хост на **serverb.lab.example.com:1001** возвращает **VHOST 1**.

Оценка

На рабочей станции **workstation**, выполните команду **lab netsecurity-review grade**, чтобы подтвердить успешное выполнение лабораторного упражнения.

```
[student@workstation ~]$ lab netsecurity-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab netsecurity-review finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab netsecurity-review finish
```

На этом лаборатория заканчивается.

РЕШЕНИЕ

УПРАВЛЕНИЕ СЕТЕВОЙ БЕЗОПАСНОСТЬЮ

СПИСОК РАБОТЫ

В этой лабораторной работе вы настроите параметры брандмауэра и **SELinux**, чтобы разрешить доступ к нескольким веб-серверам, работающим на **serverb**.

В РЕЗУЛЬТАТЕ

Вы должны иметь возможность настроить параметры брандмауэра и **SELinux** на хосте веб-сервера.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab netsecurity-review start**. Команда запускает сценарий, который определяет, доступен ли хост **serverb** в сети.

```
[student@workstation ~]$ lab netsecurity-review start
```

Ваша компания решила запустить новое веб-приложение. Это приложение прослушивает порты **80/TCP** и **1001/TCP**. Также должен быть доступен порт **22/TCP** для доступа по **ssh**. Все внесенные вами изменения должны сохраняться после перезагрузки.

При запросе **sudo** используйте слово **student** в качестве пароля.

Важно: Графический интерфейс, используемый в среде онлайн-обучения Red Hat, также должен оставаться доступным через порт **5900/TCP**. Этот порт также известен под именем службы **vncserver**. Если вы случайно заблокировали себя на своем **serverb**, вы можете либо попытаться восстановиться с помощью **ssh** на вашем сервере с вашей рабочей станции, либо перезагрузить свой **serverb**. Если вы решите перезагрузить машину **serverb**, вам придется снова запустить сценарии установки для этого практического занятия. Конфигурация на ваших машинах уже включает настраиваемую зону под названием **ROL**, которая открывает эти порты.

1. С рабочей станции **workstation**, проверьте доступ к веб-серверу по умолчанию по адресу **http://server.lab.example.com** и к виртуальному хосту по адресу **http://server.lab.example.com:1001**.

- 1.1. Протестируйте доступ к веб-серверу **http://server.lab.example.com**. В настоящее время тест не проходит. В конечном итоге веб-сервер должен вернуть **SERVER B**.

```
[student@workstation ~]$ curl http://serverb.lab.example.com
```

```
curl: (7) Failed to connect to serverb.lab.example.com port 80:  
Connection refused
```

- 1.2.** Протестируйте доступ к виртуальному хосту **http://server.lab.example.com:1001**. В настоящее время тест не проходит. В конечном итоге виртуальный хост должен вернуть **VHOST 1**.

```
[student@workstation ~]$ curl http://serverb.lab.example.com:1001  
curl: (7) Failed to connect to serverb.lab.example.com port 1001: No  
route to host
```

- 2.** Войдите в систему **serverb**, чтобы определить, что препятствует доступу к веб-серверам.

- 2.1.** С рабочей станции откройте сеанс **SSH** на **serverb** в качестве пользователя **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

- 2.2.** Определите, активна ли служба **httpd**.

```
[student@serverb ~]$ systemctl is-active httpd  
Inactive
```

- 2.3.** Включите и запустите службу **httpd**. Служба **httpd** не запускается.

```
[student@serverb ~]$ sudo systemctl enable --now httpd  
[sudo] password for student: student  
Created symlink /etc/systemd/system/multi-user.target.wants/  
httpd.service → /usr/lib/systemd/system/httpd.service.  
Job for httpd.service failed because the control process exited with  
error code.  
See "systemctl status httpd.service" and "journalctl -xe" for details.
```

- 2.4.** Выясните причины, по которым не удалось запустить службу **httpd.service**.

```
[student@serverb ~]$ systemctl status httpd.service  
● httpd.service - The Apache HTTP Server  
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled;  
  vendor preset: disabled)
```

```
Active: failed (Result: exit-code) since Thu 2019-04-11 19:25:36
CDT; 19s ago
Docs: man:httpd.service(8)
Process: 9615 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
(code=exited, status=1/FAILURE)
Main PID: 9615 (code=exited, status=1/FAILURE)
Status: "Reading configuration..."
Apr 11 19:25:36 serverb.lab.example.com systemd[1]: Starting The Apache
HTTP Server...
Apr 11 19:25:36 serverb.lab.example.com httpd[9615]: (13)Permission
denied: AH00072: make_sock: could not bind to address [::]:1001
Apr 11 19:25:36 serverb.lab.example.com httpd[9615]: (13)Permission
denied: AH00072: make_sock: could not bind to address 0.0.0.0:1001
Apr 11 19:25:36 serverb.lab.example.com httpd[9615]: no listening
sockets available, shutting down
Apr 11 19:25:36 serverb.lab.example.com httpd[9615]: AH00015: Unable to
open logs
Apr 11 19:25:36 serverb.lab.example.com systemd[1]: httpd.service: Main
process exited, code=exited, status=1/FAILURE
Apr 11 19:25:36 serverb.lab.example.com systemd[1]: httpd.service:
Failed with result 'exit-code'.
Apr 11 19:25:36 serverb.lab.example.com systemd[1]: Failed to start The
Apache HTTP Server.
```

2.5. Используйте команду **sealert**, чтобы проверить, блокирует ли **SELinux** привязку службы **httpd** к порту **1001/TCP**.

```
[student@serverb ~]$ sudo sealert -a /var/log/audit/audit.log
100% done
found 1 alerts in /var/log/audit/audit.log
-----
SELinux is preventing /usr/sbin/httpd from name_bind access on the
tcp_socket port 1001.

***** Plugin bind_ports (99.5 confidence) suggests
*****
If you want to allow /usr/sbin/httpd to bind to network port 1001
Then you need to modify the port type.
Do
# semanage port -a -t PORT_TYPE -p tcp 1001
    where PORT_TYPE is one of the following: http_cache_port_t,
http_port_t, jboss_management_port_t, jboss.messaging_port_t,
ntop_port_t, puppet_port_t.

***** Plugin catchall (1.49 confidence) suggests
*****
...output omitted...
```

3. Настройте SELinux, чтобы служба **httpd** прослушивала порт **1001/TCP**.

3.1. Используйте команду **semanage**, чтобы найти правильный тип порта.

```
[student@serverb ~]$ sudo semanage port -l | grep 'http'  
http_cache_port_t      tcp  8080, 8118, 8123, 10001-10010  
http_cache_port_t      udp  3130  
http_port_t            tcp  80, 81, 443, 488, 8008, 8009, 8443, 9000  
pegasus_http_port_t    tcp  5988  
pegasus_https_port_t   tcp  5989
```

3.2. Используйте команду **semanage** для привязки порта **1001/TCP** к типу **http_port_t**.

```
[student@serverb ~]$ sudo semanage port -a -t http_port_t -p tcp 1001  
[student@serverb ~]$
```

3.3. Убедитесь, что порт **1001/TCP** привязан к типу порта **http_port_t**.

```
[student@serverb ~]$ sudo semanage port -l | grep '^http_port_t'  
http_port_t          tcp    1001, 80, 81, 443, 488, 8008, 8009, 8443, 9000
```

3.4. Включите и запустите службу **httpd**.

```
[student@serverb ~]$ sudo systemctl enable --now httpd
```

3.5. Проверьте рабочее состояние службы **httpd**.

```
[student@serverb ~]$ systemctl is-active httpd; systemctl is-enabled httpd  
active  
enabled
```

3.6. Выход с сервера **serverb**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

4. С рабочей станции проверьте доступ к веб-серверу по умолчанию по адресу **http://server.lab.example.com** и к виртуальному хосту по адресу **http://server.lab.example.com:1001**.

- 4.1. Протестируйте доступ к веб-серверу **http://server.lab.example.com**. Веб-сервер должен вернуть **SERVER B**.

```
[student@workstation ~]$ curl http://serverb.lab.example.com  
SERVER B
```

- 4.2. Протестируйте доступ к виртуальному хосту **http://server.lab.example.com:1001**. Тест продолжает давать сбой.

```
[student@workstation ~]$ curl http://serverb.lab.example.com:1001  
curl: (7) Failed to connect to serverb.lab.example.com port 1001: No  
route to host
```

5. Войдите в систему **serverb**, чтобы определить, правильные ли порты назначены брандмауэру.

- 5.1. С рабочей станции **workstation**, войдите на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

- 5.2. Убедитесь, что зона брандмауэра по умолчанию установлена как **public**.

```
[student@serverb ~]$ firewall-cmd --get-default-zone  
public
```

- 5.3. Если предыдущий шаг не вернул **public** в качестве зоны по умолчанию, исправьте это с помощью следующей команды:

```
[student@serverb ~]$ sudo firewall-cmd --set-default-zone public
```

- 5.4. Определите открытые порты, перечисленные в сетевой зоне **public**.

```
[student@serverb ~]$ sudo firewall-cmd --permanent --zone=public --list-all  
[sudo] password for student: student  
    public  
    target: default
```

```
icmp-block-inversion: no
interfaces:
sources:
services: cockpit dhcpcv6-client http ssh
ports:
protocols:
masquerade: no
forward-ports:
source-ports:
icmp-blocks:
rich rules:
```

6. Добавьте **порт 1001/TCP** в постоянную конфигурацию сетевой зоны **public**. Подтвердите вашу конфигурацию.

6.1. Добавьте порт **1001/TCP** в зону сети **public**.

```
[student@serverb ~]$ sudo firewall-cmd --permanent --zone=public --add-port=1001/tcp
success
```

6.2. Перезагрузите конфигурацию брандмауэра.

```
[student@serverb ~]$ sudo firewall-cmd --reload
success
```

6.3. Подтвердите вашу конфигурацию.

```
[student@serverb ~]$ sudo firewall-cmd --permanent --zone=public --list-all
public
target: default
icmp-block-inversion: no
interfaces:
sources:
services: cockpit dhcpcv6-client http ssh
ports: 1001/tcp
protocols:
masquerade: no
forward-ports:
source-ports:
icmp-blocks:
rich rules:
```

6.4. Выход с сервера **serverb**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.  
[student@workstation ~]$
```

7. С рабочей станции **workstation** убедитесь, что веб-сервер по умолчанию на **serverb.lab.example.com** возвращает **SERVER B**, а виртуальный хост на **serverb.lab.example.com:1001** возвращает **VHOST 1**.

7.1. Протестируйте доступ к веб-серверу <http://server.lab.example.com>.

```
[student@workstation ~]$ curl http://serverb.lab.example.com  
SERVER B
```

7.2. Протестируйте доступ к виртуальному хосту <http://server.lab.example.com:1001>.

```
[student@workstation ~]$ curl http://serverb.lab.example.com:1001  
VHOST 1
```

Оценка

На рабочей станции **workstation**, выполните команду **lab netsecurity-review grade**, чтобы подтвердить успешное выполнение лабораторного упражнения.

```
[student@workstation ~]$ lab netsecurity-review grade
```

Завершение

На рабочей станции **workstation** запустите сценарий **lab netsecurity-review finish**, чтобы завершить это упражнение.

```
[student@workstation ~]$ lab netsecurity-review finish
```

На этом лабораторная работа заканчивается.

РЕЗЮМЕ

В этой главе вы узнали:

- Подсистема **netfilter** позволяет модулям ядра проверять каждый пакет, проходящий через систему. Проверяются все входящие, исходящие или пересылаемые сетевые пакеты.
- Использование **firewalld** упростило управление за счет классификации всего сетевого трафика по зонам. Каждая зона имеет свой собственный список портов и сервисов. Общедоступная зона установлена как зона по умолчанию.
- Служба **firewalld** поставляется с рядом предопределенных служб. Их можно получить с помощью команды **firewall-cmd --get-services**.
- Сетевой трафик строго контролируется политикой **SELinux**. Сетевые порты помечены. Например, порт **22/TCP** имеет связанную с ним метку **ssh_port_t**. Когда процесс хочет прослушивать порт, **SELinux** проверяет, разрешено ли связанной с ним метке связывать эту метку порта.
- Команда **semanage** используется для добавления, удаления и изменения меток.

ГЛАВА 12

УСТАНОВКА RED HAT ENTERPRISE LINUX

ЦЕЛЬ

Установите **Red Hat Enterprise Linux** на серверы и виртуальные машины.

ЗАДАЧИ

- Установите **Red Hat Enterprise Linux** на сервер.
- Автоматизируйте процесс установки с помощью **Kickstart**.
- Установите виртуальную машину на свой сервер **Red Hat Enterprise Linux** с помощью **Cockpit**.

РАЗДЕЛЫ

- Установка **Red Hat Enterprise Linux** (и упражнения с пошаговыми инструкциями)
- Автоматизация установки с помощью **Kickstart** (и упражнения с пошаговыми инструкциями)
- Установка и настройка виртуальных машин (и контрольный опрос)

ЛАБОРОТОРНАЯ РАБОТА

Установка Red Hat Enterprise Linux.

УСТАНОВКА RED HAT ENTERPRISE LINUX

ЦЕЛИ

После завершения этого раздела вы сможете установить **Red Hat Enterprise Linux** на сервер.

ВЫБОР НОСИТЕЛЯ ДЛЯ УСТАНОВКИ

Red Hat предоставляет несколько вариантов носителей для установки, которые можно загрузить с веб-сайта Customer Portal, используя активную подписку.

Бинарный DVD-диск, содержащий **Anaconda**, программу установки **Red Hat Enterprise Linux** и репозитории пакетов **BaseOS** и **AppStream**. Эти репозитории содержат пакеты, необходимые для завершения установки без дополнительных материалов.

Загрузочный ISO-образ, содержащий **Anaconda**, но требующий настроенной сети для доступа к репозиториям пакетов, доступным через **HTTP**, **FTP** или **NFS**.

Образ **QCOW2**, содержащий готовый системный диск, готовый к развертыванию в качестве виртуальной машины в облачной или корпоративной виртуальной среде. **QCOW2 (QEMU Copy On Write)** — стандартный формат образа, используемый Red Hat.

Red Hat предоставляет носители для установки четырех поддерживаемых процессорных архитектур: x86 64-разрядная (AMD и Intel), IBM Power Systems (Little Endian), IBM Z и ARM 64-разрядная.

После загрузки запишите DVD-диск или загрузочный ISO-образ на физический носитель, скопируйте каждый на флэш-накопитель USB или аналогичный или опубликуйте каждый с сетевого сервера для автоматического использования **Kickstart**.

Создание образа с помощью Composer

Composer - это новый инструмент, доступный в **RHEL 8**. Для особых случаев использования **Composer** позволяет администраторам создавать собственные образы систем для развертывания на облачных платформах или в виртуальных средах.

Composer использует графическую веб-консоль **Cockpit**. Его также можно вызвать из командной строки с помощью команды **composer-cli**.

РУЧНАЯ УСТАНОВКА С ANACONDA

С помощью **бинарного DVD-диска** или **загрузочного ISO-образа** администраторы могут установить новую систему **RHEL** на «голый» сервер или виртуальную машину. Программа **Anaconda** поддерживает два метода установки:

- При ручной установке пользователь взаимодействует с запросом о том, как **Anaconda** должна установить и настроить систему.

- Автоматическая установка использует файл **Kickstart**, который сообщает **Anaconda**, как установить систему. В следующем разделе установка **Kickstart** обсуждается более подробно.

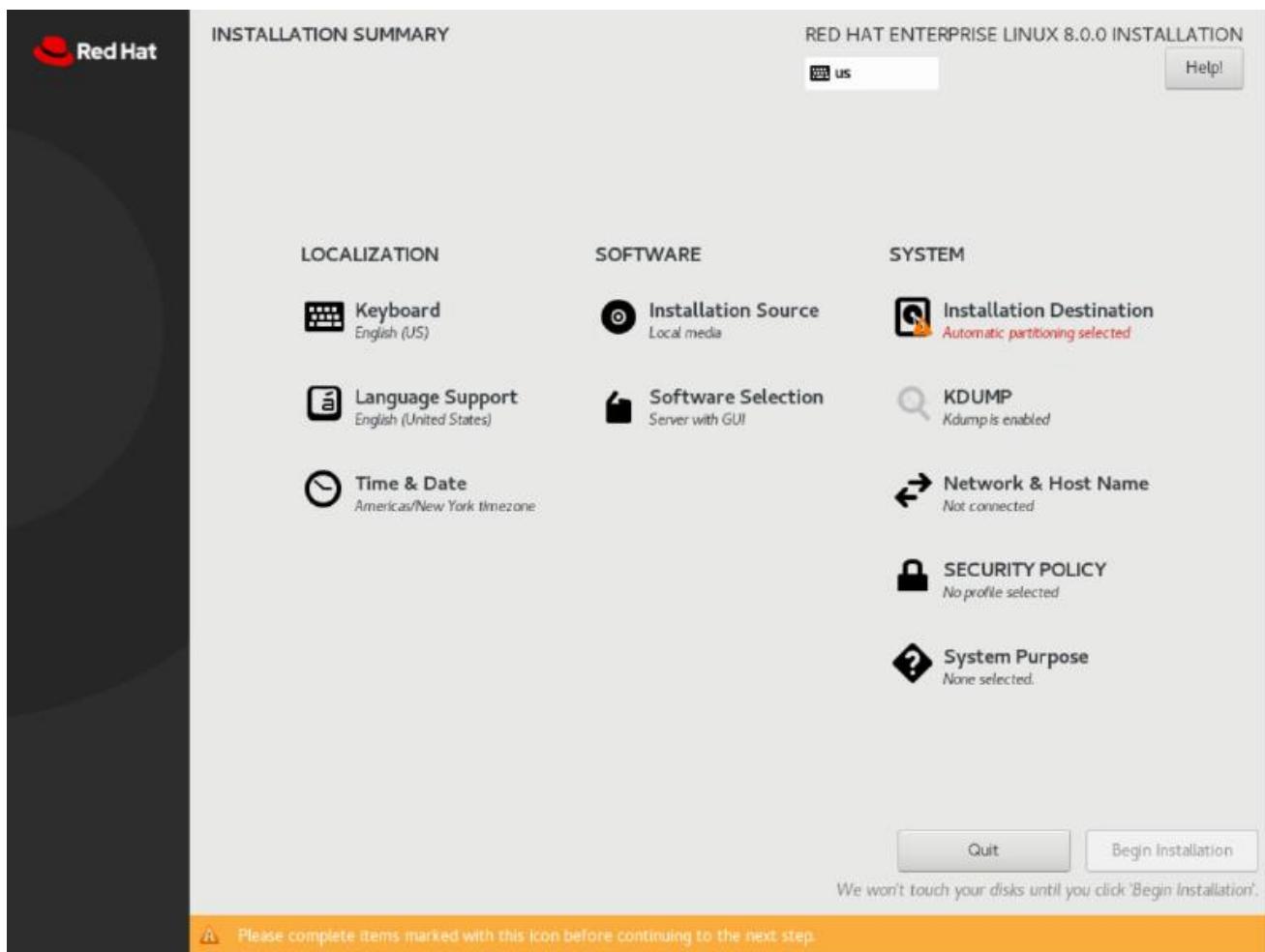
Установка RHEL с графическим интерфейсом

Когда вы загружаете систему с **бинарного DVD** или **загрузочного ISO**, **Anaconda** запускается как графическое приложение.

На экране «Добро пожаловать в Red Hat Enterprise Linux 8 (*Welcome to Red Hat Enterprise Linux 8*)» выберите язык, который будет использоваться во время установки. Это также устанавливает язык системы по умолчанию после установки. Отдельные пользователи могут выбрать предпочтительный язык своей учетной записи после установки.

Anaconda представляет окно «сводной информации (*Installation Summary*)» об установке, центральное место для настройки параметров перед началом установки.

Рисунок 12.1: Окно *Installation Summary*



В этом окне настройте параметры установки, выбирая значки в любом порядке. Выберите элемент для просмотра или редактирования. В любом элементе нажмите «Готово (**Done**)», чтобы вернуться к этому центральному экрану.

Anaconda помечает обязательные элементы треугольным предупреждающим символом и сообщением. Оранжевая строка состояния в нижней части экрана напоминает о том, что перед началом установки необходимо выполнить обязательные действия.

При необходимости заполните следующие пункты:

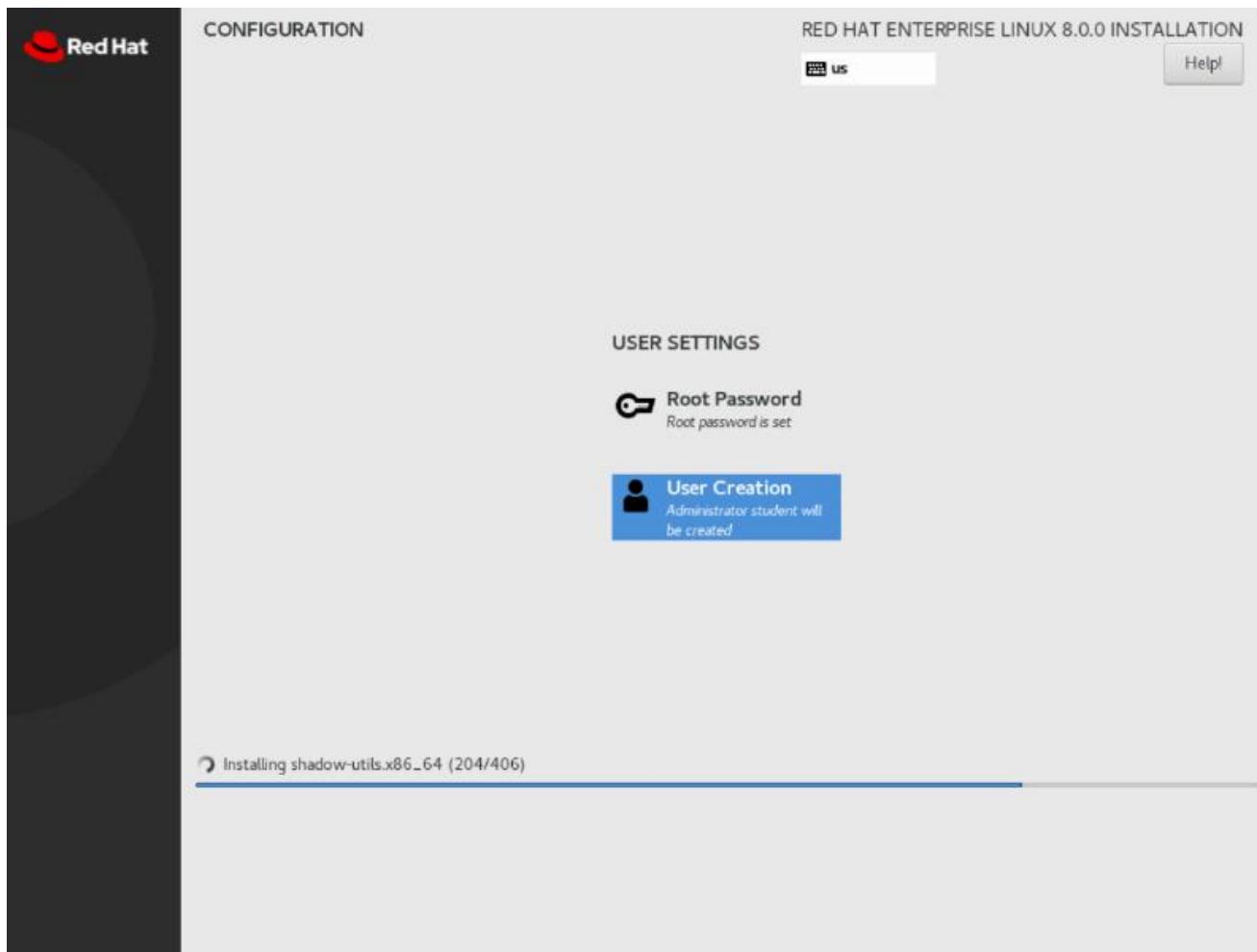
- Клавиатура (**Keyboard**) — добавьте дополнительные раскладки клавиатуры.
- Поддержка языков (**Language Support**) — выберите дополнительные языки для установки.
- Время и дата (**Time & Date**) — выберите город, в котором находится система, щелкнув интерактивную карту или выбрав его из раскрывающегося списка. Укажите местный часовой пояс даже при использовании протокола сетевого времени (**NTP**).
- Источник установки (**Installation Source**). Укажите местоположение исходного пакета, которое необходим **Anaconda** для установки. При использовании бинарного **DVD** поле источника установки уже относится к DVD.
- Выбор программного обеспечения (**Software Selection**) — выберите базовую среду для установки, а также любые дополнительные надстройки. Минимальная установка (**Minimal Install**) устанавливает только необходимые пакеты для запуска **Red Hat Enterprise Linux**.
- Место установки (**Installation Destination**) — выберите и разбейте на разделы диски, на которые будет установлена **Red Hat Enterprise Linux**. Этот элемент предполагает, что администратор понимает схемы разбиения и критерии выбора файловой системы. Радиокнопка по умолчанию для автоматического разделения выделяет выбранные устройства хранения, используя все доступное пространство.
- **KDUMP. Kdump** — это функция ядра, которая собирает содержимое системной памяти при сбое ядра. Инженеры Red Hat могут проанализировать **kdump**, чтобы определить причину сбоя. Используйте этот элемент **Anaconda**, чтобы включить или отключить **Kdump**.
- Имя сети и хоста (**Network & Host Name**) — список обнаруженных сетевых подключений слева. Выберите соединение, чтобы отобразить сведения о нем. Чтобы настроить выбранное сетевое подключение, нажмите «Настроить (**Configure**)».
- ПОЛИТИКА БЕЗОПАСНОСТИ (**SECURITY POLICY**). При активации профиля политики безопасности, такого как профиль стандарта безопасности данных индустрии платежных карт (**Payment Card Industry Data Security Standard**) (**PCI DSS**), **Anaconda** применяет ограничения и рекомендации, определенные выбранным профилем, во время установки.
- Назначение системы (**System Purpose**) — новая функция установки, которая распределяет активные права системы в соответствии с предполагаемым использованием системы.

После завершения настройки установки и устранения всех предупреждений нажмите «Начать установку (**Begin Installation**)». Нажатие кнопки «Выход (**Quit**)» прерывает установку без применения каких-либо изменений в системе.

Во время установки системы выполните следующие пункты, когда они появятся:

- Пароль root (**Root Password**) — программа установки предложит установить пароль **root**. Заключительный этап процесса установки не продолжится, пока вы не определите пароль **root**.
- Создание пользователя (**User Creation**) — создайте необязательную учетную запись без полномочий **root**. Рекомендуется поддерживать локальную учетную запись общего назначения. Учетные записи также можно создавать после завершения установки.

Рисунок 12.2: Установка пароля root и создание пользователя



Нажмите «Перезагрузить (**Reboot**)», когда установка будет завершена. **Anaconda** отображает экран **Initial Setup**, если был установлен графический рабочий стол. Примите информацию о лицензии и при необходимости зарегистрируйте систему в диспетчере подписки. Вы можете пропустить системную регистрацию и выполнить ее позже.

Устранение неполадок при установке

Во время установки **Red Hat Enterprise Linux 8 Anaconda** предоставляет две виртуальные консоли. Первый имеет пять окон, предоставляемых программным мультиплексором терминала **tmux**. Вы можете получить доступ к этой консоли с помощью **Ctrl+Alt+F1**. Вторая виртуальная консоль, которая отображается по умолчанию, показывает графический интерфейс **Anaconda**. Вы можете получить к нему доступ с помощью **Ctrl+Alt+F6**.

В первой виртуальной консоли **tmux** предоставляет приглашение оболочки во втором окне. Вы можете использовать его для ввода команд для проверки и устранения неполадок в системе, пока установка продолжается. Другие окна предоставляют диагностические сообщения, журналы и другую информацию.

В следующей таблице перечислены комбинации клавиш для доступа к виртуальным консолям и окнам **tmux**. Для **tmux** сочетания клавиш выполняются в два действия: нажмите и

отпустите **Ctrl+b**, затем нажмите цифровую клавишу окна, к которому вы хотите получить доступ. С **tmux** вы также можете использовать **Alt+Tab** для перемещения текущего фокуса между окнами.

КЛЮЧЕВАЯ ПОСЛЕДОВАТЕЛЬНОСТЬ	СОДЕРЖАНИЕ
Ctrl+Alt+F1	Получите доступ к мультиплексору терминала tmux .
Ctrl+b 1	Находясь в tmux , перейдите на главную страницу с информацией о процессе установки.
Ctrl+b 2	В tmux предоставьте корневую оболочку. Anaconda хранит файлы журнала установки в файле /tmp .
Ctrl+b 3	В tmux отображать содержимое файла /tmp/anaconda.log .
Ctrl+b 4	В tmux отображать содержимое файла /tmp/storage.log .
Ctrl+b 5	В tmux отображать содержимое файла /tmp/program.log .
Ctrl+Alt+F6	Получите доступ к графическому интерфейсу Anaconda .



ПРИМЕЧАНИЕ

Для совместимости с более ранними версиями **Red Hat Enterprise Linux** виртуальные консоли от **Ctrl+Alt+F2** до **Ctrl+Alt+F5** также представляют корневые оболочки во время установки.



РЕКОМЕНДАЦИИ

Дополнительные сведения см. в руководстве по установке и развертыванию RHEL по адресу

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/performing_a_standard_rhel_installation/index

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

УСТАНОВКА RED HAT ENTERPRISE LINUX

В этом упражнении вы переустановите один из ваших серверов с минимальной установкой Red Hat Enterprise Linux.

В РЕЗУЛЬТАТЕ

У вас должна быть возможность вручную установить Red Hat Enterprise Linux 8.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab installing-install start**. Эта команда запускает сценарий, определяющий, доступна ли машина **servera** в сети. Она также добавляет новую запись в меню **GRUB2** для загрузки **servera** с медиа носителя.

1. Войдите в консоль сервера и перезагрузите систему с помощью установочного носителя.
 - 1.1. Найдите значок консоли **servera**, соответствующий среде вашего класса. Откройте консоль.
 - 1.2. Для перезагрузки отправьте **Ctrl+Alt+Del** в свою систему, используя соответствующую клавиатуру, виртуальную или пункт меню.
 - 1.3. Когда появится меню загрузчика, выберите «Установить Red Hat Enterprise Linux 8 (*Install Red Hat Enterprise Linux 8*)».
 - 1.4. Дождитесь появления окна выбора языка.
2. Оставьте язык выбранным по умолчанию и нажмите «Продолжить (*Continue*)».
3. Используйте автоматическое создание разделов на диске **/dev/vda**.
 - 3.1. Щелкните «Место установки (*Installation Destination*)».
 - 3.2. Нажмите на первый диск, **vda**, чтобы выбрать его. Нажмите «Готово (*Done*)», чтобы использовать параметр автоматического разбиения по умолчанию.
 - 3.3. В окне «Параметры установки (*Installation Options*)» нажмите «Освободить место (*Reclaim space*)». Поскольку на диске **/dev/vda** уже есть разделы и файловые системы из предыдущей установки, этот выбор позволяет очистить диск для новой установки. В окне «Освобождение места на диске (*Reclaim Disk Space*)» нажмите «Удалить все (*Delete all*)», а затем «Освободить место (*Reclaim space*)».
4. Установите имя хоста сервера **servera.lab.example.com** и проверьте конфигурацию сетевого интерфейса.
 - 4.1. Щелкните «Сеть и имя хоста (*Network & Host Name*)».

- 4.2.** В поле «Имя хоста (*Host Name*)» введите **servera.lab.example.com** и нажмите кнопку «Применить (*Apply*)».
- 4.3.** Нажмите «Настроить (*Configure*)», а затем перейдите на вкладку «Параметры IPv4 (*IPv4 Settings*)».
- 4.4.** Подтвердите правильность параметров сети. IP-адрес — **172.25.250.10**, сетевая маска — **24**, а шлюз и сервер имен — **172.25.250.254**. Щелкните «Сохранить (*Save*)».
- 4.5.** Убедитесь, что сетевой интерфейс включен, установив для **ON/OFF** значение **ON**.
- 4.6.** Нажмите «Готово (*Done*)».
- 5.** В поле «Источник установки (*Installation Source*)» укажите http://content.example.com/rhel8.0/x86_64/dvd.
- 5.1.** Щелкните «Источник установки (*Installation Source*)».
- 5.2.** В поле **http://** введите **content.example.com/rhel8.0/x86_64/dvd**.
- 5.3.** Нажмите «Готово (*Done*)».
- 6.** Выберите программное обеспечение, необходимое для запуска минимальной установки.
- 6.1.** Щелкните «Выбор программного обеспечения (*Software Selection*)».
- 6.2.** Выберите «Минимальная установка (*Minimal Install*)» в списке «Базовая среда (*Base Environment*)».
- 6.3.** Нажмите «Готово (*Done*)».
- 7.** Настройте назначение системы.
- 7.1.** Щелкните «Цель системы (*System Purpose*)».
- 7.2.** Выберите роль **Red Hat Enterprise Linux Server**.
- 7.3.** Выберите уровень само поддержки (*Self-Support*) SLA.
- 7.4.** Выберите использование **Development/Test**.
- 7.5.** Нажмите «Готово (*Done*)».
- 8.** Щелкните «Начать установку (*Begin Installation*)».
- 9.** Пока идет установка, установите пароль для **root** слово **redhat**.
- 9.1.** Щелкните «Пароль root (*Root Password*)».
- 9.2.** Введите слово **redhat** в поле **Root Password**.
- 9.3.** Введите слово **redhat** в поле «Подтвердить (*Confirm*)».
- 9.4.** Пароль ненадежный, поэтому вам нужно дважды нажать «Готово (*Done*)».
- 10.** Пока идет установка, добавьте пользователя **student**.
- 10.1.** Щелкните «Создание пользователя (*User Creation*)».
- 10.2.** Введите **student** в поле «Полное имя (*Full Name*)».
- 10.3.** Установите флажок «Сделать этого пользователя администратором (*Make this user administrator*)», чтобы пользователь **student** мог использовать **sudo** для запуска команд от имени пользователя **root**.
- 10.4.** Введите слово **student** в поле «Пароль (*Password*)».
- 10.5.** Введите слово **student** в поле «Подтвердите пароль (*Confirm password*)».
- 10.6.** Пароль ненадежный, поэтому вам нужно дважды нажать «Готово (*Done*)».

11. По завершении установки нажмите «Перезагрузить (*Reboot*)».

12. Когда система отобразит запрос на вход в систему, войдите в систему как пользователь **student** с паролем **student**.

Завершение

Используйте метод, подходящий для среды вашего класса, для перезагрузки компьютера **servera**.

На этом упражнения с пошаговыми инструкциями заканчивается.

АВТОМАТИЗАЦИЯ УСТАНОВКИ С ПОМОЩЬЮ KICKSTART

ЦЕЛИ

После заполнения этого раздела вы должны уметь:

- Объяснить концепции и архитектуру **Kickstart**.
- Создайте файл **Kickstart** на веб-сайте **Kickstart Generator**.
- Измените существующий файл **Kickstart** с помощью текстового редактора и проверьте его синтаксис с помощью **ksvalidator**.
- Опубликовать файл **Kickstart** в программе установки.
- Выполните сетевую установку **Kickstart**.

СОЗДАНИЕ ПРОФИЛЯ Kickstart

Вы можете автоматизировать установку **Red Hat Enterprise Linux** с помощью функции **Kickstart**. Используя **Kickstart**, вы указываете все, что **Anaconda** необходимо для завершения установки, включая разметку диска, конфигурацию сетевого интерфейса, выбор пакета и другие параметры, в текстовом файле **Kickstart**. Ссылаясь на текстовый файл, **Anaconda** выполняет установку без дальнейшего взаимодействия с пользователем.



ПРИМЕЧАНИЕ

Kickstart в **Red Hat Enterprise Linux** похож на средство **Jumpstart** в **Oracle Solaris** или на использование файла ответов автоматической установки для **Microsoft Windows**.

Файлы **Kickstart** начинаются со списка команд, которые определяют, как установить целевую машину. Строки, начинающиеся с символов **#**, являются комментариями, которые программа установки игнорирует. Дополнительные разделы начинаются с директивы, распознаваемой первым символом **%**, и заканчиваются строкой с директивой **%end**.

Раздел **%packages** указывает программное обеспечение, которое будет установлено в целевой системе. Укажите отдельные пакеты по имени (без версий). Группы пакетов, указанные по имени или идентификатору, распознаются по началу с символа **@**. Группы окружения (группы групп пакетов) распознаются по началу с символов **@^**. Укажите модули, потоки и профили с помощью синтаксиса **@module:stream/profile**.

Группы имеют обязательные, стандартные и необязательные компоненты. Обычно **Kickstart** устанавливает обязательные компоненты и компоненты по умолчанию. Чтобы исключить пакет или группу пакетов из установки, поставьте перед ними символ **-** (минус). Однако исключенные пакеты или группы пакетов по-прежнему могут устанавливаться, если они являются обязательными зависимостями других запрошенных пакетов.

Конфигурация **Kickstart** обычно использует два дополнительных раздела, **%pre** и **%post**, которые содержат команды сценариев оболочки, которые дополнительно настраивают систему. Сценарий **%pre** выполняется до того, как будет выполнено любое разбиение диска. Обычно этот раздел используется только в том случае, если требуются действия для распознавания или

инициализации устройства перед разбиением диска на разделы. Сценарий **%post** выполняется после завершения установки.

Вы должны указать основные команды **Kickstart** перед разделами **%pre**, **%post** и **%packages**, но в противном случае вы можете разместить эти разделы в файле в любом порядке.

КОМАНДЫ ФАЙЛА Kickstart

Команды установки (Installation)

Определите источник установки и способ выполнения установки. Каждый сопровождается примером.

- **url:** указывает URL-адрес, указывающий на установочный носитель.

```
url --url=http://classroom.example.com/content/rhel8.0/x86_64/dvd/
```

- **repo:** указывает, где найти дополнительные пакеты для установки. Этот параметр должен указывать на действительный репозиторий **yum**.

```
repo --name="appstream" --baseurl=http://classroom.example.com/content/rhel8.0/x86_64/dvd/AppStream/
```

- **text:** установка в текстовом режиме.
- **vnc:** позволяет удаленно просматривать графическую установку через **VNC**.

```
vnc --password=redhat
```

Команды создания разделов

Определите устройства и схему разделения, которые будут использоваться.

- **clearpart:** удаляет разделы из системы перед созданием новых разделов. По умолчанию разделы не удаляются.

```
clearpart --all --drives=sda,sdb --initlabel
```

- **part:** определяет размер, формат и имя раздела.

```
part /home --fstype=ext4 --label=homes --size=4096 --maxsize=8192 --grow
```

- **autopart**: автоматически создает раздел **root**, раздел подкачки (**swap**) и соответствующий загрузочный (**boot**) раздел для данной архитектуры. На достаточно больших дисках также создается раздел **/home**.
- **ignoredisk**: контролирует доступ **Anaconda** к дискам, подключенными к системе.

```
ignoredisk --drives=sdc
```

- **bootloader**: определяет место для установки загрузчика.

```
bootloader --location=mbr --boot-drive=sda
```

- **volgroup, logvol**: создает группы томов **LVM** и логические тома.

```
part pv.01 --size=8192
volgroup myvg pv.01
logvol / --vgname=myvg --fstype=xfs --size=2048 --name=rootvol --grow
logvol /var --vgname=myvg --fstype=xfs --size=4096 --name=varvol
```

- **zerombr**: инициализировать диски, форматирование которых не распознано.

Команды определения Сети

Определите сетевые функции и имя хоста.

- **network**: настраивает сетевую информацию для целевой системы. Активирует сетевые устройства в среде установщика.

```
network --device=eth0 --bootproto=dhcp
```

- **firewall**: определяет конфигурацию брандмауэра для целевой системы.

```
firewall --enabled --service=ssh,http
```

Расположение и команды безопасности

Настройте параметры, связанные с безопасностью, языком и регионами.

- **lang**: устанавливает язык, который будет использоваться во время установки, и язык установленной системы по умолчанию. Необходимый.

```
lang en_US.UTF-8
```

- **keyboard:** устанавливает тип системной клавиатуры. Необходимый.

```
keyboard --vckeymap=us --xlayouts=""
```

- **timezone:** определяет часовой пояс, **NTP-серверы** и использование аппаратными часами **UTC**.

```
timezone --utc --ntpservers=time.example.com Europe/Amsterdam
```

- **authselect:** настраивает параметры аутентификации. Параметры, распознаваемые **authselect**, действительны для этой команды. См. **authselect(8)**.
- **rootpw:** определяет начальный пароль **root**.

```
rootpw --plaintext redhat
```

or

```
rootpw --iscrypted $6$KUnFfrTzO8jv.PiH$YIBbOtXBkWzoMuRfb0.SpbQ....XDR1UuchoMG1
```

- **selinux:** устанавливает режим **SELinux** для установленной системы.

```
selinux --enforcing
```

- **services:** изменяет набор служб по умолчанию для запуска под целевым объектом **systemd** по умолчанию.

```
services --disabled=network,iptables,ip6tables --
enabled=NetworkManager,firewalld
```

- **group, user:** создание локальной группы или пользователя в системе.

```
group --name=admins --gid=10001
user --name=jdoe --gecos="John Doe" --groups=admins --password=changeme --
plaintext
```

Разные команды

Настройте различные элементы, связанные с ведением журнала во время установки и состоянием питания хоста по завершении.

- **logging:** эта команда определяет, как **Anaconda** будет вести журнал во время установки.

```
logging --host=loghost.example.com --level=info
```

- **firstboot:** если этот параметр включен, агент установки запускается при первой загрузке системы. Пакет начальной установки должен быть установлен.

```
firstboot -disabled
```

- **reboot, poweroff, halt:** укажите последнее действие, которое необходимо выполнить после завершения установки.



ПРИМЕЧАНИЕ

Утилита **ksverendiff** из пакета **pykickstart** полезна для выявления изменений в синтаксисе файла **Kickstart** между двумя версиями **Red Hat Enterprise Linux** или **Fedor**a.

Например, **ksverendiff -f RHEL7 -t RHEL8** указывает на изменения в синтаксисе с **RHEL 7** на **RHEL 8**. Доступные версии перечислены в верхней части файла **/usr/lib/python3.6/site-packages/pykickstart/version.py**.

ПРИМЕР ФАЙЛА KICKSTART

Первая часть файла состоит из команд установки, таких как разметка диска и источник установки.

```
#version=RHEL8
ignoredisk --only-use=vda
# System bootloader configuration
bootloader --append="console=ttyS0 console=ttyS0,115200n8 no_timer_check
net.ifnames=0 crashkernel=auto" --location=mbr --timeout=1 --boot-drive=vda
# Clear the Master Boot Record
zerombr
# Partition clearing information
clearpart --all --initlabel
# Use text mode install
text
repo --name="appstream" --baseurl=http://classroom.example.com/content/rhel8.0/
x86_64/dvd/AppStream/
# Use network installation
```

```
url --url="http://classroom.example.com/content/rhel8.0/x86_64/dvd/"
# Keyboard layouts
# old format: keyboard us
# new format:
keyboard --vckeymap=us --xlayouts="
# System language
lang en_US.UTF-8
# Root password
rootpw --plaintext redhat
# System authorization information
auth --enablesshadow --passalgo=sha512
# SELinux configuration
selinux --enforcing
firstboot --disable
# Do not configure the X Window System
skipx
# System services
services --disabled="kdump,rhsmcertd" --enabled="sshd,rngd,chrony"
# System timezone
timezone America/New_York --isUtc
# Disk partitioning information
part / --fstype="xfs" --ondisk=vda --size=10000
```

Вторая часть содержит раздел **%packages**, подробно описывающий, какие пакеты и группы пакетов следует устанавливать, а какие пакеты устанавливать не следует.

```
%packages
@core
chrony
cloud-init
dracut-config-generic
dracut-norescue
firewalld
grub2
kernel
rsync
tar
-plymouth
%end
```

Последняя часть содержит любые сценарии установки **%pre** и **%post**.

```
%post --erroronfail
```

```
# For cloud images, 'eth0' _is_ the predictable device name, since
# we don't want to be tied to specific virtual (!) hardware
rm -f /etc/udev/rules.d/70*
ln -s /dev/null /etc/udev/rules.d/80-net-name-slot.rules

# simple eth0 config, again not hard-coded to the build hardware
cat > /etc/sysconfig/network-scripts/ifcfg-eth0 << EOF
DEVICE="eth0"
BOOTPROTO="dhcp"
ONBOOT="yes"
TYPE="Ethernet"
USERCTL="yes"
PEERDNS="yes"
IPV6INIT="no"
EOF

%end
```



ПРИМЕЧАНИЕ

В файле **Kickstart** отсутствующие обязательные значения приводят к тому, что установщик в интерактивном режиме запрашивает ответ или полностью прерывает установку.

ЭТАПЫ УСТАНОВКИ KICKSTART

Чтобы успешно автоматизировать установку **Red Hat Enterprise Linux**, выполните следующие действия:

1. Создайте файл **Kickstart**.
2. Опубликуйте файл **Kickstart** в программе установки.
3. Загрузите **Anaconda** и укажите файл **Kickstart**.

СОЗДАНИЕ ФАЙЛА KICKSTART

Используйте любой из этих методов для создания файла **Kickstart**:

- Используйте веб-сайт **Kickstart Generator**.
- Используйте текстовый редактор.

На веб-сайте **Kickstart Generator** по адресу <https://access.redhat.com/labs/kickstartconfig/> представлены диалоговые окна для ввода данных пользователем и создается текстовый файл директив **Kickstart** с вариантами выбора пользователя. Каждое диалоговое окно соответствует настраиваемым элементам в программе установки **Anaconda**.

Рисунок 12.3: Базовая конфигурация с Kickstart Generator

The screenshot shows the 'Kickstart Generator' interface on a web browser. At the top, there's a navigation bar with icons for back, forward, search, and a URL field showing <https://access.redhat.com/labs/kickstartconfig/>. Below the navigation is the title 'Kickstart Generator'. To the left of the main content area, there are two small icons: a flask and a speech bubble. The main content area has a heading 'Generate a custom kickstart file based on your configuration parameters.' and a link to the 'Red Hat Enterprise Linux Install guide'. A dropdown menu shows 'Red Hat Enterprise Linux 8 Beta' and a red 'DOWNLOAD' button. On the left, a sidebar lists 'Basic Configuration' sections: Installation, Partition, BootLoader, Packages, Authentication, Network, Security, Display, Pre-Installation Script, and Post-Installation Script. The right side shows a detailed 'Basic Configuration' form with fields for Default Language (English (USA)), Keyboard (U.S. English), Time Zone (America/New York), and a checked checkbox for 'Use UTC clock'. There's also a field for Root Password.



ПРИМЕЧАНИЕ

На момент написания статьи веб-сайт **Kickstart Generator** не предлагал **Red Hat Enterprise Linux 8** в качестве пункта меню. **Red Hat Enterprise Linux 8 Beta** была правильным выбором.

Создание файла **Kickstart** с нуля обычно слишком сложно, но редактирование существующего файла **Kickstart** распространено и полезно. При каждой установке создается файл **/root/anaconda-ks.cfg**, содержащий директивы **Kickstart**, используемые при установке. Этот файл является хорошей отправной точкой при создании файла **Kickstart** вручную.

ksvalidator — это утилита, которая проверяет синтаксические ошибки в файле **Kickstart**. Он гарантирует, что ключевые слова и параметры используются правильно, но не проверяет успешность **URL-адресов**, отдельных пакетов или групп, а также какой-либо части сценариев **%post** или **%pre**. Например, если директива **firewall --disabled** написана с ошибкой, **ksvalidator** может выдать одну из следующих ошибок:

```
[user@host ~]$ ksvalidator /tmp/anaconda-ks.cfg
The following problem occurred on line 10 of the kickstart file:
```

Unknown command: firewall

```
[user@host ~]$ ksvalidator /tmp/anaconda-ks.cfg
```

The following problem occurred on line 10 of the kickstart file:

no such option: --disabled

Пакет **pykickstart** предоставляет **ksvalidator**.

ПУБЛИКУЕМ ФАЙЛ KICKSTART В ANACONDA

ПУБЛИКУЕМ ФАЙЛ KICKSTART В ANACONDA

Сделайте файл **Kickstart** доступным для установщика, поместив его в одно из следующих мест:

- **Сетевой сервер**, доступный во время установки по **FTP**, **HTTP** или **NFS**.
- Доступный **USB-диск** или **компакт-диск**.
- **Локальный жесткий диск** в системе, которую необходимо установить.

Установщик должен получить доступ к файлу **Kickstart**, чтобы начать автоматическую установку. В наиболее распространенном методе автоматизации используется сетевой сервер, такой как **FTP**, **веб-сервер** или сервер **NFS**. Сетевые серверы облегчают обслуживание файлов **Kickstart**, поскольку изменения можно внести один раз, а затем сразу же использовать для нескольких будущих установок.

Предоставление файлов **Kickstart** на **USB** или **CD-ROM** также удобно. Файл **Kickstart** можно встроить в загрузочный носитель, используемый для запуска установки. Однако при изменении файла **Kickstart** необходимо создать новый установочный носитель.

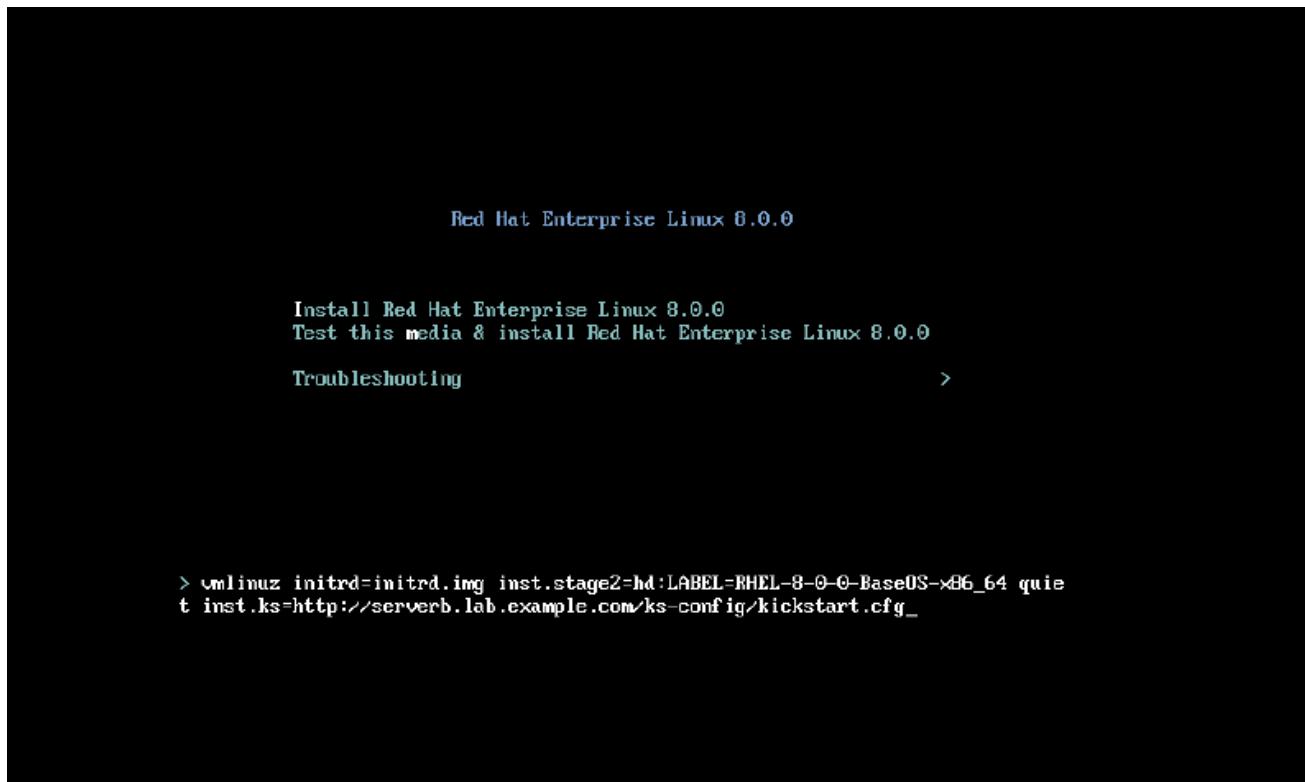
Предоставление файла **Kickstart** на локальном диске позволяет быстро восстановить систему.

ЗАГРУЗИТЕ ANACONDA И УКАЖИТЕ ЕЕ НА ФАЙЛ KICKSTART

После выбора метода **Kickstart** установщику сообщается, где найти файл **Kickstart**, путем передачи параметра **inst.ks=LOCATION** ядру установки. Далее приведены некоторые примеры:

- `inst.ks=http://server/dir/file`
- `inst.ks=ftp://server/dir/file`
- `inst.ks=nfs:server:/dir/file`
- `inst.ks=hd:device:/dir/file`
- `inst.ks=cdrom:device`

Рисунок 12.4: Указание местоположения файла Kickstart во время установки



Для установки виртуальной машины с помощью **Virtual Machine Manager** или **virt-manager** URL-адрес Kickstart можно указать в поле в разделе «Параметры URL-адреса (*URL Options*)». При установке физических машин загрузитесь с установочного носителя и нажмите клавишу **Tab**, чтобы прервать процесс загрузки. Добавьте параметр **inst.ks=LOCATION** в ядро установки.



ПРИМЕЧАНИЕ

Глава «Основы установки Kickstart» в разделе «Выполнение расширенной установки RHEL» по адресу

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/performing_an_advanced_rhel_installation/kickstart-installationbasics_installing-rhel-as-an-experienced-user#kickstart-installationbasics_installing-rhel-as-an-experienced-user.

Команды **Kickstart** для настройки программы установки и раздел управления потоком в Приложении В. Справочник по командам и параметрам Kickstart в Выполнение расширенной установки RHEL по адресу

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/performing_an_advanced_rhel_installation/kickstart-installationbasics_installing-rhel-as-an-experienced-user#kickstart-commands-forinstallation-program-configuration-and-flow-control_kickstart-commands-andoptions-reference

Глава «Параметры загрузки» в разделе «Выполнение расширенной установки RHEL» по адресу

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html-single/performing_an_advanced_rhel_installation/kickstart-installationbasics_installing-rhel-as-an-experienced-user#kickstart-and-advanced-bootoptions_installing-rhel-as-an-experienced-user

УПРАЖНЕНИЯ С ПОШАГОВЫМИ ИНСТРУКЦИЯМИ

АВТОМАТИЗАЦИЯ УСТАНОВКИ С ПОМОЩЬЮ KICKSTART

В этой лабораторной работе вы создадите файл kickstart и проверите синтаксис.

В РЕЗУЛЬТАТЕ

Ты должен быть способен:

- Создайте файл **kickstart**.
- Используйте **ksvalidator** для проверки синтаксиса файла **кикстарта**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab installing-kickstart start**. Эта команда запускает сценарий, который определяет, доступна ли машина **servera** в сети. Скрипт также проверяет, что **Apache** установлен и настроен на сервере **servera**.

```
[student@workstation ~]$ lab installing-kickstart start
```

1. Используйте команду **ssh**, чтобы войти на сервер как пользователь **student**. Системы настроены на использование ключей **SSH** для аутентификации, поэтому пароль не требуется.

```
[student@workstation ~]$ ssh student@servera  
...output omitted...  
[student@servera ~]$
```

2. Скопируйте файл **/root/anaconda-ks.cfg** на **servera** в файл с именем **/home/student/kickstart.cfg**, чтобы пользователь **student** мог его редактировать. Используйте команду **cat /root/anacondaks.cfg > ~/kickstart.cfg**, чтобы скопировать содержимое **root/anacondaks.cfg** в **/home/student/kickstart.cfg**. Если **sudo** запрашивает пароль пользователя **student**, используйте в качестве пароля слово **student**.

```
[student@servera ~]$ sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg  
[sudo] password for student: student
```

3. Внесите следующие изменения в файл **/home/student/kickstart.cfg**.
3.1. Закомментируйте директиву **reboot**:

```
#reboot
```

- 3.2. Закомментируйте команду **repo** для репозитория **BaseOS**. Измените команду **repo** для **AppStream**, чтобы она указывала на репозиторий **AppStream** в классе:

```
#repo --name="koji-override-0" --baseurl=http://downloadnode-02.eng.bos.redhat.com/rhel-8/devel/candidate-trees/RHEL-8/RHEL-8.0.0-20190213.0/compose/BaseOS/x86_64/os
repo --name="appstream" --baseurl=http://classroom.example.com/content/rhel8.0/x86_64/dvd/AppStream/
```

- 3.3. Измените команду **url**, чтобы указать источник установки класса по протоколу **HTTP** медиа:

```
url --url=http://classroom.example.com/content/rhel8.0/x86_64/dvd/
```

- 3.4. Закомментируйте команду **network**:

```
#network           --bootproto=dhcp      --device=link      --activate
```

- 3.5. Установите пароль **root** на как **redhat**. Измените строку, начинающуюся с **rootpw**, на:

```
rootpw           --plaintext    redhat
```

- 3.6. Удалите строку, в которой используется команда **auth**, и добавьте строку **authselect select sssd**, чтобы установить службу **sssd** в качестве источника удостоверения и аутентификации.

```
authselect  select  sssd
```

В Red Hat Enterprise Linux 8 команда **authselect** заменяет команду **authconfig**.

- 3.7. Упростите команду **services**, чтобы она выглядела точно так же, как показано ниже:

```
services --disabled="kdump,rhsmcertd" --enabled="sshd,rngd,chrony"
```

- 3.8. Закомментируйте команды **part** и **reqpart**. Добавьте команду **autopart**:

```
#reqpart
# Disk partitioning information
#part / --fstype="xfs" --ondisk=vda --size=8000
autopart
```

- 3.9.** Удалите все содержимое между разделом **%post** и его **%end**. Добавьте следующую строку:
echo "Kickstarted on \$(date)" >> /etc/issue

```
%post --erroronfail
echo "Kickstarted on $(date)" >> /etc/issue
%end
```

Весь раздел **%post** должен выглядеть так.

- 3.10.** Упростите спецификацию пакета, чтобы она выглядела точно так же, как показано ниже:

```
%packages
@core
chrony
dracut-config-generic
dracut-norescue
firewalld
grub2
kernel
rsync
tar
httpd
-plymouth
%end
```

Когда вы закончите редактирование файла, сохраните и выйдите.

- 4.** С помощью команды **ksvalidator** проверьте файл **Kickstart** на наличие синтаксических ошибок.

```
[student@servera ~]$ ksvalidator kickstart.cfg
```

- 5.** Скопируйте **kickstart.cfg** в каталог **/var/www/html/ks-config**.

```
[student@servera ~]$ sudo cp ~/kickstart.cfg /var/www/html/ks-config
```

6. Выход из сервера **servera**.

```
[student@servera ~]$ exit  
logout  
Connection to servera closed.  
[student@workstation ~]$
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab installing-kickstart finish**, чтобы завершить упражнение.

```
[student@workstation ~]$ lab installing-kickstart finish
```

На этом упражнения с пошаговыми инструкциями заканчивается.

УСТАНОВКА И НАСТРОЙКА ВИРТУАЛЬНЫХ МАШИН

ЦЕЛИ

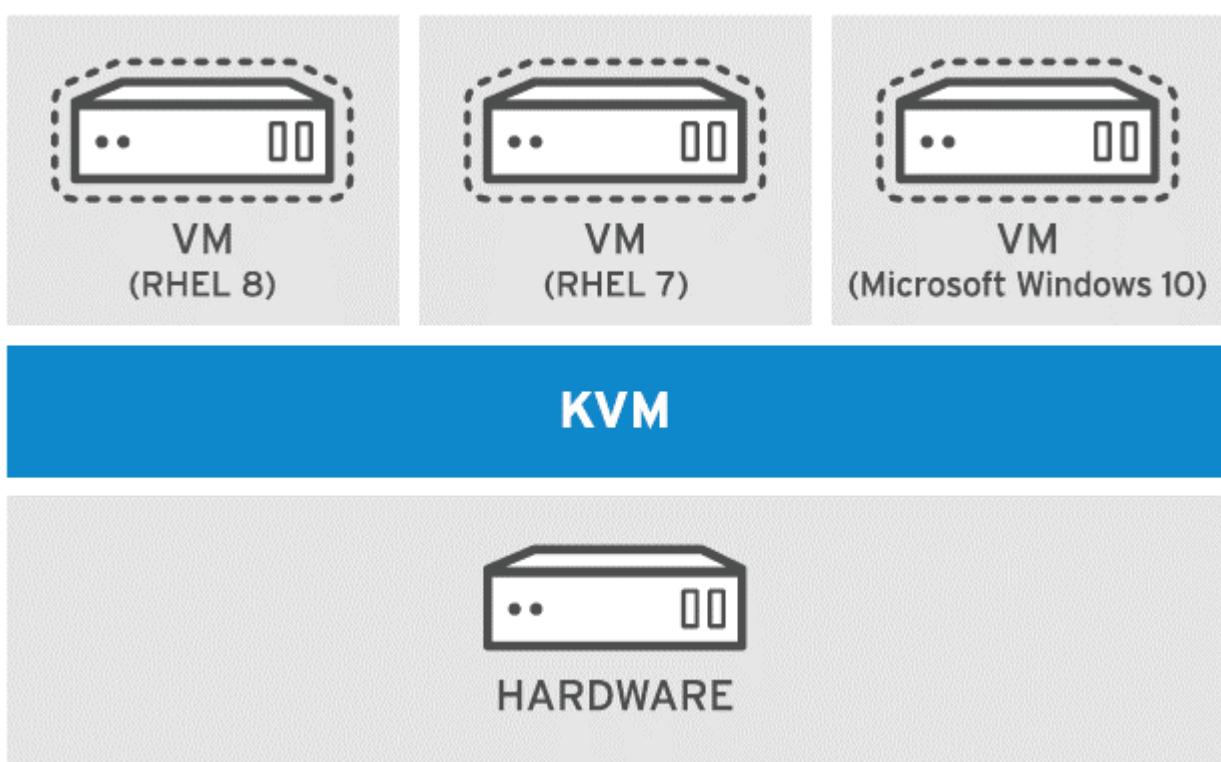
После изучения этого раздела вы сможете установить виртуальную машину на свой сервер Red Hat Enterprise Linux с помощью Cockpit.

ВВЕДЕНИЕ ВИРТУАЛИЗАЦИИ KVM

Виртуализация это функция, позволяющая разделить одну физическую машину на несколько виртуальных машин (ВМ), каждая из которых может запускать независимую операционную систему.

Red Hat Enterprise Linux 8 поддерживает **KVM** (*Kernel-based Virtual Machine*), полное решение для виртуализации, встроенное в стандартное ядро Linux. KVM может работать с несколькими гостевыми операционными системами Windows и Linux.

Рисунок 12.5: Виртуализация KVM



В Red Hat Enterprise Linux управляйте KVM с помощью команды `virsh` или инструмента виртуальных машин Cockpit.

Технология виртуальной машины KVM доступна во всех продуктах Red Hat, от автономных физических экземпляров Red Hat Enterprise Linux до платформы Red Hat OpenStack:

- Физические аппаратные системы работают под управлением Red Hat Enterprise Linux для обеспечения виртуализации KVM. Red Hat Enterprise Linux обычно представляет собой

thick хост, система, которая поддерживает виртуальные машины, а также предоставляет другие локальные и сетевые службы, приложения и функции управления.

- **Red Hat Virtualization (RHV)** предоставляет централизованный веб-интерфейс, который позволяет администраторам управлять всей виртуальной инфраструктурой. Он включает в себя расширенные функции, такие как миграция **KVM**, избыточность и высокая доступность. **Red Hat Virtualization Hypervisor** — это настроенная версия **Red Hat Enterprise Linux**, предназначенная для единственной цели предоставления и поддержки виртуальных машин.
- Платформа **Red Hat OpenStack (RHOSP)** обеспечивает основу для создания, развертывания и масштабирования публичного или частного облака.

Red Hat поддерживает виртуальные машины под управлением следующих операционных систем:

- Red Hat Enterprise Linux 6 и выше
- Microsoft Windows 10 и более поздние версии
- Microsoft Windows Server 2016 и более поздние версии

НАСТРОЙКА ФИЗИЧЕСКОЙ СИСТЕМЫ RED HAT ENTERPRISE LINUX В КАЧЕСТВЕ ХОСТА ВИРТУАЛИЗАЦИИ

Администраторы могут настроить систему Red Hat Enterprise Linux в качестве узла виртуализации, подходящего для разработки, тестирования, обучения или при необходимости одновременной работы в нескольких операционных системах.

Установка инструментов виртуализации

Установите модуль virt Yum, чтобы подготовить систему к тому, чтобы она стала хостом виртуализации.

```
[root@host ~]# yum module list virt
Name           Stream          Profiles        Summary
virt           rhel [d][e]      common [d]     Virtualization module

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
[root@host ~]# yum module install virt
...output omitted...
```

Проверка системных требований

Для KVM требуется либо процессор Intel с расширениями Intel VT-x и Intel 64 для систем на базе x86, либо процессор AMD с расширениями AMD-V и AMD64. Чтобы проверить ваше оборудование и проверить системные требования, используйте команду **virt-host-validate**.

```
[root@host ~]# virt-host-validate
QEMU: Checking for hardware virtualization : PASS
QEMU: Checking if device /dev/kvm exists : PASS
QEMU: Checking if device /dev/kvm is accessible : PASS
QEMU: Checking if device /dev/vhost-net exists : PASS
QEMU: Checking if device /dev/net/tun exists : PASS
QEMU: Checking for cgroup 'memory' controller support : PASS
QEMU: Checking for cgroup 'memory' controller mount-point : PASS
QEMU: Checking for cgroup 'cpu' controller support : PASS
QEMU: Checking for cgroup 'cpu' controller mount-point : PASS
QEMU: Checking for cgroup 'cpuacct' controller support : PASS
QEMU: Checking for cgroup 'cpuacct' controller mount-point : PASS
QEMU: Checking for cgroup 'cpuset' controller support : PASS
QEMU: Checking for cgroup 'cpuset' controller mount-point : PASS
QEMU: Checking for cgroup 'devices' controller support : PASS
QEMU: Checking for cgroup 'devices' controller mount-point : PASS
QEMU: Checking for cgroup 'blkio' controller support : PASS
QEMU: Checking for cgroup 'blkio' controller mount-point : PASS
QEMU: Checking for device assignment IOMMU support : PASS
```

Система должна пройти все элементы проверки, чтобы иметь возможность быть хостом **KVM**.

УПРАВЛЕНИЕ ВИРТУАЛЬНЫМИ МАШИНАМИ С ПОМОЩЬЮ COCKPIT

Модуль **virt Yum** предоставляет команду **virsh** для управления вашими виртуальными машинами. Инструмент **Cockpit** предоставляет интерфейс веб-консоли для управления **KVM** и создания виртуальных машин.

Установите пакет **cockpit-machines**, чтобы добавить меню «виртуальных машин (*Virtual Machines*)» в **Cockpit**.

```
[root@host ~]# yum install cockpit-machines
```

Если **Cockpit** еще не запущен, запустите и включите его.

```
[root@host ~]# systemctl enable --now cockpit.socket
```

Чтобы создать новую виртуальную машину с помощью **Cockpit**, откройте меню «Виртуальные машины (*Virtual Machines*)» в веб-интерфейсе **Cockpit**. Оттуда нажмите «Создать виртуальную машину (*Create VM*)» и введите конфигурацию виртуальной машины в окне «Создать новую виртуальную машину (*Create New Virtual Machine*)».

Рисунок 12.6: Управление виртуальными машинами в **Cockpit**

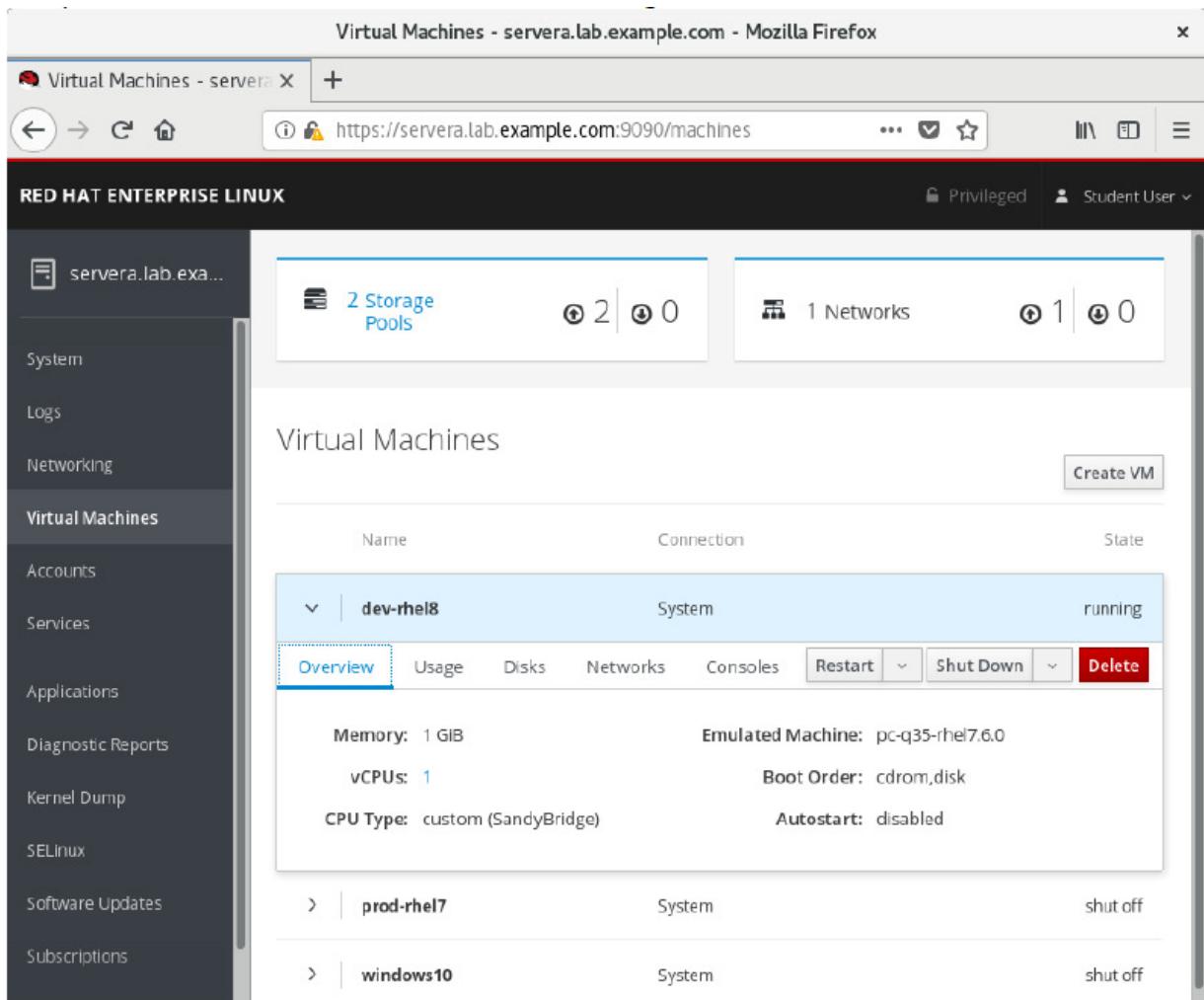
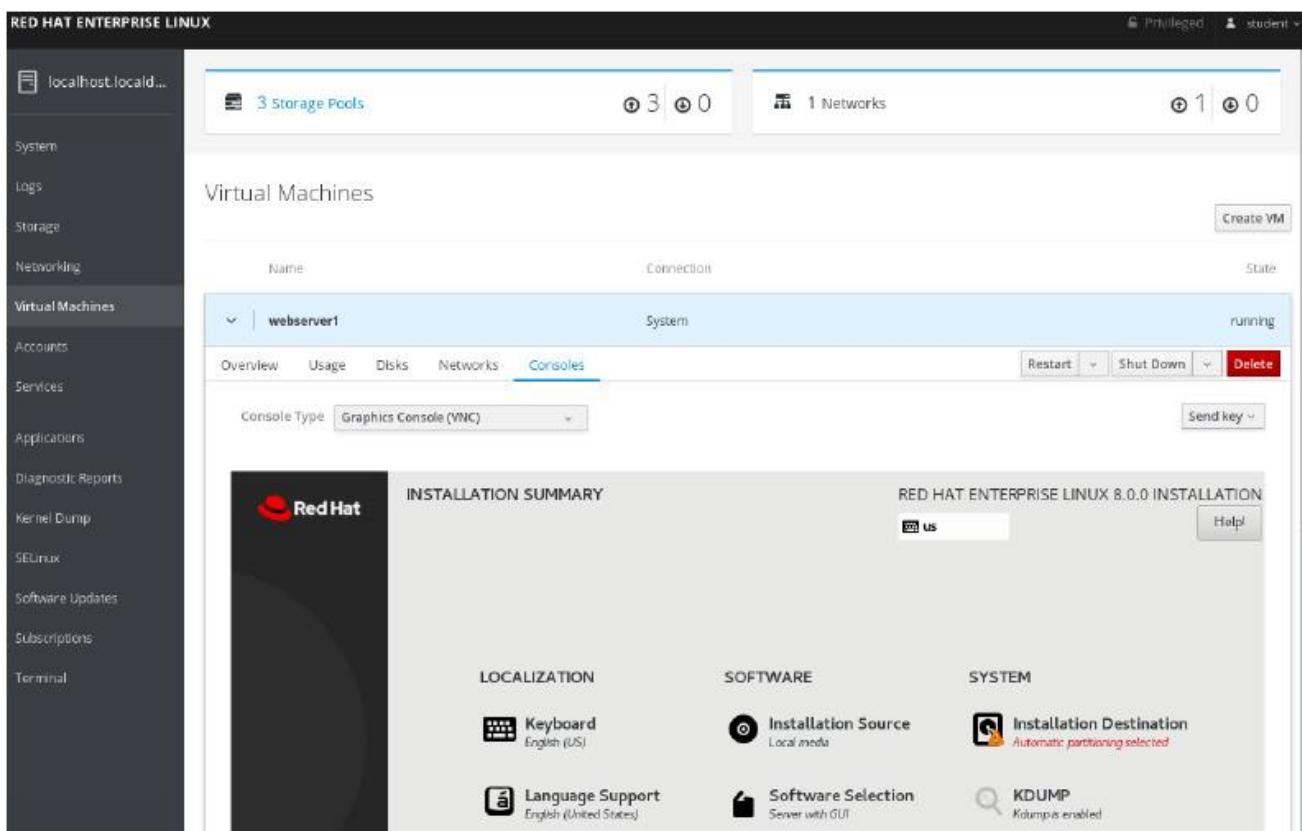


Рисунок 12.7: Создание виртуальной машины в Cockpit

- Имя (*Name*) задает доменное имя для конфигурации виртуальной машины. Это имя не связано с именем сетевого хоста, которое вы даете системе в установленной виртуальной машине.
- Тип источника установки (*Installation Source Type*) — это метод получения установочного файла ISO. Возможные варианты включают локальную файловую систему или URL-адрес HTTPS, FTP или NFS.
- Источник установки (*Installation Source*) указывает путь к источнику установки.
- Поставщик ОС (*OS Vendor*) и операционная система (*Operating System*) указывает операционную систему виртуальной машины. Уровень виртуализации представляет аппаратную эмуляцию, совместимую с выбранной операционной системой.
- Память (*Memory*) — это объем ОЗУ, доступный для новой виртуальной машины.
- Размер хранилища (*Storage Size*) — это размер диска для новой виртуальной машины. Свяжите дополнительные диски с виртуальной машиной после установки.
- Немедленно запустить виртуальную машину (*Immediately Start VM*) указывает, должна ли виртуальная машина запускаться немедленно после нажатия кнопки «Создать (*Create*)».

Нажмите «Создать (*Create*)», чтобы создать виртуальную машину, и «Установить (*Install*)», чтобы начать установку операционной системы. В кабине отображается консоль **ВМ**, из которой можно установить систему.

Рисунок 12.8: Установка ОС виртуальной машины



РЕКОМЕНДАЦИИ

Дополнительные сведения см. в руководстве по настройке и управлению виртуализацией по адресу:

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/htmlsingle/configuring_and_managing_virtualization/index

Что такое виртуализация?

<https://www.redhat.com/en/topics/virtualization/what-is-virtualization>

КОНТРОЛЬНЫЙ ОПРОС

УСТАНОВКА И НАСТРОЙКА ВИРТУАЛЬНЫХ МАШИН

Выберите правильные ответы на следующие вопросы:

1. Какие три гостевые операционные системы поддерживает Red Hat в качестве виртуальных машин KVM?

(Выберите три.)

- a. **Fedora 28** и выше
- b. **Red Hat Enterprise Linux 6** и выше
- c. **CoreOS Container Linux 2023** и более поздние версии
- d. **Microsoft Windows 7 SP1**
- e. **Microsoft Windows 10** и более поздние версии
- f. **Microsoft Windows Server 2016** и более поздние версии

2. Какие два компонента необходимы для настройки вашей системы в качестве узла виртуализации и управления виртуальными машинами с помощью веб-консоли?

(Выберите два.)

- a. Модуль *virt Yum*
- b. Группа пакетов **openstack**
- c. Пакет **cockpit-machines**
- d. Группа пакетов **Virtualization Platform**
- e. Модуль *kvm yum*
- f. Пакет **cockpit-virtualization**

3. Какая команда проверяет, поддерживает ли ваша система виртуализацию?

- a. **grep kvm /proc/cpuinfo**
- b. **virsh validate**
- c. **virt-host-validate**
- d. **rhv-validate**
- e. **cockpit-validate**

4. Какие два инструмента можно использовать для запуска и остановки виртуальных машин в системе **Red Hat Enterprise Linux**? (Выберите два.)

- a. **vmctl**
- b. **libvirtd**
- c. **virsh**
- d. **openstack**
- e. **Веб-консоль**

РЕШЕНИЕ

УСТАНОВКА И НАСТРОЙКА ВИРТУАЛЬНЫХ МАШИН

Выберите правильные ответы на следующие вопросы:

1. Какие три гостевые операционные системы поддерживает Red Hat в качестве виртуальных машин KVM?

(Выберите три.)

- a. **Fedora 28** и выше
- b. **Red Hat Enterprise Linux 6** и выше
- c. **CoreOS Container Linux 2023** и более поздние версии
- d. **Microsoft Windows 7 SP1**
- e. **Microsoft Windows 10** и более поздние версии
- f. **Microsoft Windows Server 2016** и более поздние версии

2. Какие два компонента необходимы для настройки вашей системы в качестве узла виртуализации и управления виртуальными машинами с помощью веб-консоли?

(Выберите два.)

- a. Модуль **virt Yum**
- b. Группа пакетов **openstack**
- c. Пакет **cockpit-machines**
- d. Группа пакетов **Virtualization Platform**
- e. Модуль **kvm yum**
- f. Пакет **cockpit-virtualization**

3. Какая команда проверяет, поддерживает ли ваша система виртуализацию?

- a. **grep kvm /proc/cpuinfo**
- b. **virsh validate**
- c. **virt-host-validate**
- d. **rhv-validate**
- e. **cockpit-validate**

4. Какие два инструмента можно использовать для запуска и остановки виртуальных машин в системе **Red Hat Enterprise Linux**? (Выберите два.)

- a. **vmctl**
- b. **libvirtd**
- c. **virsh**
- d. **openstack**
- e. **Веб-консоль**

ЛАБОРАТОРНАЯ РАБОТУ

УСТАНОВКА RED HAT ENTERPRISE LINUX

КОНТРОЛЬНЫЙ СПИСОК

В этой лабораторной работе вы создадите файл **kickstart** и выполните установку **kickstart** на **serverb**.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создать файл **kickstart**.
- Сделайте файл **kickstart** доступным для установщика.
- Выполните **kickstart** установку.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab installing-review start**. Эта команда запускает сценарий, для определения, доступны ли машины **servera** и **serverb** в сети, и настраивает **Apache** на **serverb**. Скрипт также настраивает загрузочное меню на **serverb** для выполнения установки **kickstart**.

```
[student@workstation ~]$ lab installing-review start
```

Подготовьте файл **kickstart** на **serverb**, как указано, и сделайте его доступным по адресу <http://server.lab.example.com/ks-config/kickstart.cfg>. Выполните установку **kickstart** на **servera**, используя подготовленный вами файл **кикстарта**.

1. На **serverb** скопируйте файл **/root/anaconda-ks.cfg** в **/home/student/kickstart.cfg**, чтобы пользователь **student** мог его редактировать.
2. Внесите следующие изменения в файл **/home/student/kickstart.cfg**.
 - Закомментируйте команду **reboot**.
 - Закомментируйте команду **repo** для репозитория **BaseOS**. Измените команду **repo** для репозитория **AppStream**, чтобы она указывала на http://classroom.example.com/content/rhel8.0/x86_64/dvd/AppStream/. Имя репозитория должно быть установлено как **appstream**.

- Измените команду URL, чтобы использовать http://classroom.example.com/content/rhel8.0/x86_64/dvd/ в качестве источника установки.
- Закомментируйте команду **network**.
- Измените команду **rootpw**, чтобы она использовала открытый текст, и установите пароль **root** на слово **redhat**.
- Удалите строку, в которой используется команда **auth**, и добавьте строку **authselect select sssd**, чтобы установить службу **sssd** в качестве источника удостоверения и аутентификации.
- Упростите команду **services**, чтобы отключались только службы **kdump** и **rhsmcertd**. Оставьте включенными только **sshd**, **rngd** и **chrony**.
- Добавьте команду **autopart**. Команды **part** и **repart** уже должны быть закомментированы.
- Упростите раздел **%post**, чтобы он запускал сценарий только для добавления текста **Kickstarted on DATE** в конец файла **/etc/issue**. **DATE** — это переменная информация, которая должна генерироваться сценарием с помощью команды **date** без дополнительных параметров.
- Упростите раздел **%package** следующим образом: включите пакеты **@core, chrony, dracut-configgeneric, dracut-norescue, firewalld, grub2, kernel, rsync, tar** и **httpd**. Убедитесь, что пакет **plymouth** не установлен.

3. Проверьте синтаксис файла **kickstart.cfg**.
4. Сделайте файл **/home/student/kickstart.cfg** доступным по адресу <http://server.lab.example.com/ks-config/kickstart.cfg>.
5. Вернитесь к системе **workstation**, чтобы проверить свою работу.

Оценка

На рабочей станции **workstation** запустите сценарий **lab installing-review grade**, чтобы оценить это упражнение. Перезагрузите сервер, чтобы выполнить установку **kickstart**.

```
[student@workstation ~]$ lab installing-review grade
```

Исправьте любые сбои в **kickstart.cfg**, передаваемом с веб-сервера **serverb**, либо изменив **/var/www/html/ks-config/kickstart.cfg** напрямую, либо изменив **~/kickstart.cfg** и скопировав его в **/var/www/html/ks-config/**.

Перезагрузите сервер **servera**, чтобы выполнить установку **kickstart**. В меню **GRUB** выберите **Kickstart Red Hat Enterprise Linux 8** и нажмите **Enter**.

Завершение

На рабочей станции **workstation**, запустите сценарий **lab installing-review finish**, чтобы завершить это упражнение. Этот сценарий удаляет **веб-сервер**, настроенный на **serverb** во время упражнения.

```
[student@workstation ~]$ lab installing-review finish
```

Сбросьте систему **servera**, чтобы вернуть ее в состояние по умолчанию.

На этом лабораторная работа заканчивается.

РЕШЕНИЕ

УСТАНОВКА RED HAT ENTERPRISE LINUX

КОНТРОЛЬНЫЙ СПИСОК

В этой лабораторной работе вы создадите файл **kickstart** и выполните установку **kickstart** на **serverb**.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создать файл **kickstart**.
- Сделайте файл **kickstart** доступным для установщика.
- Выполните **kickstart** установку.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab installing-review start**. Эта команда запускает сценарий, для определения, доступны ли машины **servera** и **serverb** в сети, и настраивает **Apache** на **serverb**. Скрипт также настраивает загрузочное меню на **serverb** для выполнения установки **kickstart**.

```
[student@workstation ~]$ lab installing-review start
```

Подготовьте файл **kickstart** на **serverb**, как указано, и сделайте его доступным по адресу <http://server.lab.example.com/ks-config/kickstart.cfg>. Выполните установку **kickstart** на **servera**, используя подготовленный вами файл **кикстарта**.

1. На **serverb** скопируйте файл **/root/anaconda-ks.cfg** в **/home/student/kickstart.cfg**, чтобы пользователь **student** мог его редактировать.

1.1. Используйте команду **ssh** для входа на **serverb** в качестве пользователя **student**.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
[student@serverb ~]$
```

- 1.2.** Скопируйте `/root/anaconda-ks.cfg` на `serverb` в файл с именем `/home/student/kickstart.cfg`, чтобы пользователь `student` мог его редактировать. Используйте команду `sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg`, чтобы скопировать содержимое `/root/anaconda-ks.cfg` в `/home/student/kickstart.cfg`. Если `sudo` запрашивает пароль пользователя `student`, используйте слово `student` в качестве пароля.

```
[student@serverb ~]$ sudo cat /root/anaconda-ks.cfg > ~/kickstart.cfg  
[sudo] password for student: student
```

- 2.** Внесите следующие изменения в файл `/home/student/kickstart.cfg`.

- Закомментируйте команду `reboot`.
- Закомментируйте команду `repo` для репозитория **BaseOS**. Измените команду `repo` для репозитория **AppStream**, чтобы она указывала на `http://classroom.example.com/content/rhel8.0/x86_64/dvd/AppStream/`. Имя репозитория должно быть установлено как `appstream`.
- Измените команду `URL`, чтобы использовать `http://classroom.example.com/content/rhel8.0/x86_64/dvd/` в качестве источника установки.
- Закомментируйте команду `network`.
- Измените команду `rootpw`, чтобы она использовала открытый текст, и установите пароль `root` на слово `redhat`.
- Удалите строку, в которой используется команда `auth`, и добавьте строку `authselect select sssd`, чтобы установить службу `sssd` в качестве источника удостоверения и аутентификации.
- Упростите команду `services`, чтобы отключались только службы `kdump` и `rhsmcertd`. Оставьте включенными только `sshd`, `rngd` и `chrony`.
- Добавьте команду `autopart`. Команды `part` и `repart` уже должны быть закомментированы.
- Упростите раздел `%post`, чтобы он запускал сценарий только для добавления текста **Kickstarted on DATE** в конец файла `/etc/issue`. **DATE** — это переменная информация, которая должна генерироваться сценарием с помощью команды `date` без дополнительных параметров.
- Упростите раздел `%package` следующим образом: включите пакеты `@core, chrony, dracut-configgeneric, dracut-norescue, firewalld, grub2, kernel, rsync, tar` и `httpd`. Убедитесь, что пакет `plymouth` не установлен.

- 2.1.** Закомментируйте директиву `reboot`:

```
#reboot
```

- 2.2.** Команда `repo` дважды встречается в `kickstart.cfg`. Закомментируйте команду `repo` для репозитория **BaseOS**. Измените команду `repo` для репозитория **AppStream**, чтобы она указывала на репозиторий **AppStream** в классе:

```
#repo --name="koji-override-0" --baseurl=http://downloadnode-02.eng.bos.redhat.com/rhel-8/devel/candidate-trees/RHEL-8/RHEL-8.0.0-20190213.0/compose/BaseOS/x86_64/os
repo --name="appstream" --baseurl=http://classroom.example.com/content/rhel8.0/x86_64/dvd/AppStream/
```

- 2.3.** Измените команду **url**, чтобы указать исходный установочный носитель **HTTP**, используемый в классе:

```
url --url=http://classroom.example.com/content/rhel8.0/x86_64/dvd/
```

- 2.4.** Закомментируйте команду **network**:

```
#network --bootproto=dhcp --device=link --activate
```

- 2.5.** Установите пароль **root** слово **redhat**. Измените строку, начинающуюся с **rootpw**, на:

```
rootpw --plaintext redhat
```

- 2.6.** Удалите строку, в которой используется команда **auth**, и добавьте строку **authselect select sssd**, чтобы установить службу **sssd** в качестве источника удостоверения и аутентификации.

```
authselect select sssd
```

- 2.7.** Упростите команду **services**, чтобы она выглядела точно так же, как показано ниже:

```
services --disabled="kdump,rhsmcertd" --enabled="sshd,rngd,chrony"
```

- 2.8.** Закомментируйте команды **part** и **reqpart**. Добавьте команду **autopart**:

```
#reqpart
# Disk partitioning information
#part / --fstype="xfs" --ondisk=vda --size=8000
autopart
```

- 2.9.** Удалите все содержимое между разделом **%post** и его **%end**. Добавьте следующую строку:
echo "Kickstarted on \$(date)" >> /etc/issue. Весь раздел **%post** должен выглядеть так.

```
%post --erroronfail  
echo "Kickstarted on $(date)" >> /etc/issue  
%end
```

- 2.10.** Упростите спецификацию пакета, чтобы она выглядела точно так же, как показано ниже:

```
%packages  
@core  
chrony  
dracut-config-generic  
dracut-norescue  
firewalld  
grub2  
kernel  
rsync  
tar  
httpd  
-plymouth  
%end
```

- 3.** Проверьте синтаксис файла **kickstart.cfg**.

- 3.1.** Используйте команду **ksvalidator**, чтобы проверить файл **Kickstart** на наличие синтаксических ошибок.

```
[student@serverb ~]$ ksvalidator kickstart.cfg
```

- 4.** Сделайте файл **/home/student/kickstart.cfg** доступным по адресу <http://server.lab.example.com/ks-config/kickstart.cfg>.

- 4.1.** Скопируйте **kickstart.cfg** в каталог **/var/www/html/ks-config/**.

```
[student@serverb ~]$ sudo cp ~/kickstart.cfg /var/www/html/ks-config
```

- 5.** Вернитесь к системе **workstation**, чтобы проверить свою работу.

- 5.1.** Выход с сервера **serverb**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.
```

```
[student@workstation ~]$
```

Оценка

На рабочей станции **workstation** запустите сценарий **lab installing-review grade**, чтобы оценить это упражнение. Перезагрузите сервер, чтобы выполнить установку **kickstart**.

```
[student@workstation ~]$ lab installing-review grade
```

Исправьте любые сбои в **kickstart.cfg**, передаваемом с веб-сервера **serverb**, либо изменив **/var/www/html/ks-config/kickstart.cfg** напрямую, либо изменив **~/kickstart.cfg** и скопировав его в **/var/www/html/ks-config/**.

Перезагрузите сервер **servera**, чтобы выполнить установку **kickstart**. В меню **GRUB** выберите **Kickstart Red Hat Enterprise Linux 8** и нажмите **Enter**.

Завершение

На рабочей станции **workstation**, запустите сценарий **lab installing-review finish**, чтобы завершить это упражнение. Этот сценарий удаляет **веб-сервер**, настроенный на **serverb** во время упражнения.

```
[student@workstation ~]$ lab installing-review finish
```

Сбросьте систему **servera**, чтобы вернуть ее в состояние по умолчанию.

На этом лабораторная работа заканчивается.

РЕЗЮМЕ

В этой главе вы узнали:

- Двоичный **DVD-диск RHEL 8** включает **Anaconda** и все репозитории, необходимые для установки.
- Загрузочный **ISO-образ RHEL 8** включает установщик **Anaconda**, обеспечивающий доступ к репозиториям по сети во время установки.
- Система **Kickstart** выполняет автоматическую установку.
- Файлы **Kickstart** можно создавать с помощью веб-сайта **Kickstart Generator** или путем копирования и редактирования файла `/root/anaconda-ks.cfg`.
- Модуль **virt Yum** предоставляет пакеты для того, чтобы система **RHEL** стала хостом виртуализации.
- Пакет **cockpit-machines** добавляет в **Cockpit** меню **Virtual Machines**.

ГЛАВА 13

ОБЩИЙ ОБЗОР ВСЕХ РАЗДЕЛОВ

ЦЕЛЬ

Обзор задач из *Red Hat System Administration II*.

ЗАДАЧИ

- Просмотрите задания из *Red Hat System Administration II*.

РАЗДЕЛЫ

- Всесторонний обзор

ЛАБОРОТОРНАЯ РАБОТА

- Лабораторная работа: Исправление проблем с загрузкой и обслуживание серверов
- Лабораторная работа: Настройка и управление файловыми системами и хранилищами.
- Лабораторная работа: Настройка и управление безопасностью сервера.

ВСЕСТОРОННИЙ ОБЗОР

ЦЕЛИ

После завершения этого раздела вы должны были просмотреть и освежить знания и навыки, полученные в **Red Hat System Administration II**.

ОБЗОР СИСТЕМНОГО АДМИНИСТРИРОВАНИЯ RED HAT II

Прежде чем приступить к всестороннему обзору этого курса, вам должно быть удобно знакомиться с темами, затронутыми в каждой главе.

Вы можете обратиться к предыдущим разделам учебника для дополнительного изучения.

Глава 1. Повышение производительности командной строки

Выполняйте команды более эффективно, используя расширенные функции оболочки Bash, сценарии оболочки и различные утилиты, предоставляемые Red Hat Enterprise Linux.

- Автоматизируйте последовательность команд, написав простой сценарий оболочки.
- Эффективно запускайте команды над списками элементов в скрипте или из командной строки, используя циклы **for** и условные операторы.
- Поиск текста, соответствующего шаблону, в файлах журналов и выходных данных команды **grep** и регулярных выражений.

Глава 2. Планирование будущих задач

Запланируйте задачи для автоматического выполнения в будущем.

- Настройте команду, которая запускается один раз в какой-то момент в будущем.
- Запланировать запуск команд по повторяющемуся расписанию с использованием пользовательского файла **crontab**.
- Расписание выполнения команд по повторяющемуся расписанию с использованием системного файла **crontab** и каталогов.
- Включать и отключать системные таймеры, а также настраивать таймер для управления временными файлами.

Глава 3. Настройка производительности системы

Улучшите производительность системы, установив параметры настройки и отрегулировав приоритет планирования процессов.

- Оптимизировать производительность системы, выбрав профиль настройки, управляемый настроенным демоном.

- Установите или снимите приоритет определенных процессов с помощью команд **nice** и **renice**.

Глава 4. Управление доступом к файлам с помощью ACL

Интерпретируйте и устанавливайте списки контроля доступа (**ACL**) для файлов, чтобы справляться с ситуациями, требующими сложных разрешений доступа пользователей и групп.

- Описывать варианты использования списков **ACL**, определять файлы, в которых установлены списки управления доступом, и интерпретировать влияние этих списков **ACL**.
- Устанавливать и удалять списки **ACL** для файлов и определять списки **ACL** по умолчанию, автоматически устанавливаемые каталогом для вновь создаваемых файлов.

Глава 5. Управление безопасностью SELinux

Заштитите и управляйте безопасностью сервера с помощью **SELinux**.

- Описать, как **SELinux** защищает ресурсы и как выбрать принудительный режим.
- Настройте контекст **SELinux** файла, чтобы управлять тем, как процессы взаимодействуют с этим файлом.
- Настройте логические значения **SELinux**, чтобы разрешить изменение политики во время выполнения для различных потребностей доступа.
- Изучите сообщения журнала **SELinux** и устранит неполадки, связанные с отказами **SELinux AVC**.

Глава 6. Управление основным хранилищем

Создавайте и управляйте устройствами хранения, разделами, файловыми системами и пространствами подкачки из командной строки.

- Создайте разделы хранилища, отформатируйте их в файловых системах и смонтируйте для использования.
- Создание и управление пространствами подкачки для дополнения физической памяти.

Глава 7. Управление логическими томами

Создавайте логические тома, содержащие файловые системы и области подкачки, и управляйте ими из командной строки.

- Создавайте и управляйте логическими томами с устройств хранения, форматируйте их с помощью файловых систем или подготавливайте их с помощью областей подкачки.
- Добавлять и удалять хранилище, назначенное группам томов, и неразрушающим образом увеличивать размер логического тома, отформатированного в файловой системе.

Глава 8. Реализация расширенных функций хранения

Управляйте хранилищем с помощью системы управления локальным хранилищем **Stratis** и используйте тома **VDO** для оптимизации используемого пространства хранения.

- Управление несколькими уровнями хранения с помощью управления локальным хранилищем **Stratis**.
- Оптимизируйте использование дискового пространства, используя **VDO** для сжатия и дедупликации данных на устройствах хранения.

Глава 9. Доступ к сетевому хранилищу

Доступ к сетевому хранилищу по протоколу **NFS**.

- Монтировать, использовать и демонтировать экспорт **NFS** из командной строки и во время загрузки.
- Настройте средство автоматического монтирования с прямыми и непрямыми сопоставлениями для автоматического монтирования файловой системы **NFS** по запросу и отключения ее, когда она больше не используется.
- Настройте клиент **NFS** для использования **NFSv4** с помощью нового инструмента **nfsconf**.

Глава 10. Управление процессом загрузки

Управляйте процессом загрузки, чтобы контролировать предлагаемые услуги, а также устранять неполадки и устранять их.

- Опишите процесс загрузки **Red Hat Enterprise Linux**, установите цель по умолчанию, используемую при загрузке, и загрузите систему с целью, отличной от цели по умолчанию.
- Войдите в систему и измените пароль **root**, если текущий пароль **root** был утерян.
- Вручную исправить конфигурацию файловой системы или проблемы с повреждением, которые останавливают процесс загрузки.

Глава 11. Управление сетевой безопасностью

Контролируйте сетевые подключения к службам с помощью системного брандмауэра и правил **SELinux**.

Принимать или отклонять сетевые подключения к системным службам с помощью правил **firewalld**.

Контролируйте, могут ли сетевые службы использовать определенные сетевые порты, управляя метками портов **SELinux**.

Глава 12. Установка Red Hat Enterprise Linux

Установите **Red Hat Enterprise Linux** на серверы и виртуальные машины.

- Установите **Red Hat Enterprise Linux** на сервер.
- Автоматизируйте процесс установки с помощью **Kickstart**.
- Установите виртуальную машину на свой сервер **Red Hat Enterprise Linux** с помощью **Cockpit**.

ЛАБОРАТОРНАЯ РАБОТА

РЕШЕНИЕ ПРОБЛЕМ ЗАГРУЗКИ И ОБСЛУЖИВАНИЕ СЕРВЕРОВ

В этом обзоре вы будете диагностировать и устранять проблемы с загрузкой, а также обновлять системы по умолчанию. Вы также запланируете выполнение задач по повторяющемуся расписанию как обычный пользователь.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Выполнить диагностику проблемы и восстановить систему из аварийного (**emergency**) режима.
- Измените **target** по умолчанию с **graphical.target** на **multi-user.target**.
- Запланировать повторяющиеся задания для запуска от имени обычного пользователя.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Скопируйте любые файлы или работу, которые вы хотите сохранить, в другие системы перед перезагрузкой. Сейчас перезагрузите системы **workstation**, **servera** и **serverb**.

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-compreview1 start**, чтобы начать всестороннюю проверку. Этот сценарий создает необходимые файлы для правильной настройки среды.

```
[student@workstation ~]$ lab rhcsa-compreview1 start
```

ИНСТРУКЦИИ

Выполните следующие задачи на **serverb**, чтобы завершить всестороннюю проверку:

- На рабочей станции выполните команду **lab rhcsa-compreview1 break1**. Этот сценарий прерывания приводит к сбою процесса загрузки на **serverb**. Он также устанавливает более длительный тайм-аут в меню **GRUB2**, чтобы прервать процесс загрузки, и перезагружает **serverb**.

Устранимте возможную причину и устранимте сбой загрузки. Исправление должно гарантировать, что **serverb** перезагрузится без вмешательства. При необходимости используйте слово **redhat** в качестве пароля суперпользователя.

- На рабочей станции **workstation**, выполните команду **lab rhcsa-compreview1 break2**. Этот сценарий прерывания заставляет **target** по умолчанию **multi-user target** переключаться на **graphical target** на **serverb**. Он также устанавливает более длительный тайм-аут для меню **GRUB2**, чтобы прервать процесс загрузки, и перезагружает **serverb**.

На **serverb** исправьте **target** по умолчанию, чтобы использовать **multi-user target**. Настройки **target** по умолчанию должны сохраняться после перезагрузки без ручного вмешательства.

Используйте команду **sudo** от имени пользователя **student** со словом **student** в качестве пароля для выполнения привилегированных команд.

- Запланируйте повторяющееся задание в качестве пользователя **student**, который выполняет сценарий **/home/student/backup-home.sh** каждый час с 19:00 до 20:00. и 9 вечера во все дни, кроме субботы и воскресенья.

Загрузите сценарий резервного копирования с <http://materials.example.com/labs/backup-home.sh>. Сценарий резервного копирования **backup-home.sh** создает резервную копию каталога **/home/student** с **serverb** на **servera** в каталоге **/home/student/server-backup**. Используйте сценарий **backup-home.sh**, чтобы запланировать повторяющееся задание от имени пользователя **student** на **serverb**.

- Перезагрузите систему и дождитесь завершения загрузки перед оценкой.

Оценка

На рабочей станции запустите сценарий **lab rhcsa-compreview1 grade**, чтобы подтвердить успешное выполнение этого упражнения. Исправьте все сообщения об ошибках и перезапустите сценарий до тех пор, пока он не будет завершён без сообщений об ошибках.

```
[student@workstation ~]$ lab rhcsa-compreview1 grade
```

Завершение

На рабочей станции **workstation**, выполните команду **lab rhcsa-compreview1 finish**, чтобы завершить это упражнение. Данный сценарий удаляет файлы и ресурсы, созданные во время упражнения, и обеспечивает чистоту среды.

```
[student@workstation ~]$ lab rhcsa-compreview1 finish
```

Сохраните любые файлы или работу, которые вы хотите сохранить, в других системах, а затем перезагрузите рабочую станцию **workstation**, серверы **servera** и **serverb** перед следующим упражнением.

На этом всеобъемлющий обзор завершается.

РЕШЕНИЕ

РЕШЕНИЕ ПРОБЛЕМ ЗАГРУЗКИ И ОБСЛУЖИВАНИЕ СЕРВЕРОВ

В этом обзоре вы будете диагностировать и устранять проблемы с загрузкой, а также обновлять системы по умолчанию. Вы также запланируете выполнение задач по повторяющемуся расписанию как обычный пользователь.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Выполнить диагностику проблемы и восстановить систему из аварийного (**emergency**) режима.
- Измените **target** по умолчанию с **graphical.target** на **multi-user.target**.
- Запланировать повторяющиеся задания для запуска от имени обычного пользователя.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Скопируйте любые файлы или работу, которые вы хотите сохранить, в другие системы перед перезагрузкой. Сейчас перезагрузите системы **workstation**, **servera** и **serverb**.

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-comprevew1 start**, чтобы начать всестороннюю проверку. Этот сценарий создает необходимые файлы для правильной настройки среды.

```
[student@workstation ~]$ lab rhcsa-comprevew1 start
```

ИНСТРУКЦИИ

Выполните следующие задачи на **serverb**, чтобы завершить всестороннюю проверку:

- На рабочей станции выполните команду **lab rhcsa-comprevew1 break1**. Этот сценарий прерывания приводит к сбою процесса загрузки на **serverb**. Он также устанавливает более длительный тайм-аут в меню **GRUB2**, чтобы прервать процесс загрузки, и перезагружает **serverb**.

Устранимте возможную причину и устранимте сбой загрузки. Исправление должно гарантировать, что **serverb** перезагрузится без вмешательства. При необходимости используйте слово **redhat** в качестве пароля суперпользователя.

- На рабочей станции **workstation**, выполните команду **lab rhcsa-compreview1 break2**. Этот сценарий прерывания заставляет **target** по умолчанию **multi-user target** переключаться на **graphical target** на **serverb**. Он также устанавливает более длительный тайм-аут для меню **GRUB2**, чтобы прервать процесс загрузки, и перезагружает **serverb**.

На **serverb** исправьте **target** по умолчанию, чтобы использовать **multi-user target**. Настройки **target** по умолчанию должны сохраняться после перезагрузки без ручного вмешательства.

Используйте команду **sudo** от имени пользователя **student** со словом **student** в качестве пароля для выполнения привилегированных команд.

- Запланируйте повторяющееся задание в качестве пользователя **student**, который выполняет сценарий **/home/student/backup-home.sh** каждый час с 19:00 до 20:00. и 9 вечера во все дни, кроме субботы и воскресенья.

Загрузите сценарий резервного копирования с <http://materials.example.com/labs/backup-home.sh>. Сценарий резервного копирования **backup-home.sh** создает резервную копию каталога **/home/student** с **serverb** на **servera** в каталоге **/home/student/server-backup**. Используйте сценарий **backup-home.sh**, чтобы запланировать повторяющееся задание от имени пользователя **student** на **serverb**.

- Перезагрузите систему и дождитесь завершения загрузки перед оценкой.

1. На рабочей станции выполните команду **lab rhcsa-compreview1 break1**.

```
[student@workstation ~]$ lab rhcsa-compreview1 break1
```

2. После того, как **serverb** загрузится, войдите в консоль и обратите внимание, что процесс загрузки остановился досрочно. Найдите минутку, чтобы подумать о возможной причине такого поведения.

2.1. Найдите значок консоли **serverb** в соответствии со средой вашего класса. Откройте консоль.

2.2. Глядя на ошибку, кажется, что по крайней мере части системы все еще работают

2.3. Нажмите **Ctrl+Alt+Del**, чтобы перезагрузить **серверb**.

Когда появится меню загрузчика, нажмите любую клавишу, кроме **Enter**, чтобы прервать обратный отсчет.

2.4. Отредактируйте запись загрузчика по умолчанию в памяти, чтобы войти в аварийный режим.

Нажмите **e**, чтобы отредактировать текущую запись.

2.5. Используйте клавиши курсора, чтобы перейти к строке, которая начинается со слова **linux**. Добавьте **systemd.unit=emergency.target** в конец строки.

2.6. Нажмите **Ctrl+x** для загрузки с измененной конфигурацией.

2.7. Войдите в аварийный (**emergency**) режим. Пароль **root** — **redhat**.

```
Give root password for maintenance  
(or press Control-D to continue): redhat  
[root@serverb ~]#
```

- 3.** Перемонтируйте **/** файловую систему для **чтения/записи**. Используйте команду **mount -a**, чтобы попытаться смонтировать все остальные файловые системы.

3.1. Перемонтируйте **/** файловую систему для **чтения/записи**, что бы получить возможность редактирования файловой системы.

```
[root@serverb ~]# mount -o remount,rw /
```

3.2. Используйте команду **mount -a**, чтобы попытаться смонтировать все остальные файловые системы. Обратите внимание, что одна из файловых систем не может быть смонтирована.

```
[root@serverb ~]# mount -a  
mount: /FakeMount: can't find UUID=fake.
```

3.3. Отредактируйте **/etc/fstab**, чтобы исправить проблему. Удалите или закомментируйте неверную строку.

3.4. Обновите **systemd**, чтобы система зарегистрировала новую конфигурацию **/etc/fstab**.

```
[root@serverb ~]# systemctl daemon-reload  
[ 206.828912] systemd[1]: Reloading.
```

3.5. Убедитесь, что файл **/etc/fstab** теперь корректен, попытавшись смонтировать все записи.

```
[root@serverb ~]# mount -a
```

3.6. Перезагрузите **serverb** и дождитесь завершения загрузки. Теперь система должна загружаться нормально.

```
[root@serverb ~]# systemctl reboot
```

- 4.** На рабочей станции выполните команду **lab rhcsa-compreview1 break2**.

```
[student@workstation ~]$ lab rhcsa-compreview1 break2
```

Дождитесь завершения перезагрузки, прежде чем продолжить.

5. На **serverb** переключитесь **multi-user target**. Установите **target** по умолчанию как **multi-user** режима. Используйте команду **sudo** для запуска любой необходимой административной команды и, если будет предложено, используйте слово **student** в качестве пароля.

- 5.1. С рабочей станции **workstation**, откройте сеанс **SSH** на **serverb** в качестве пользователя **student**.

```
[student@workstation ~]$ ssh student@serverb  
...output omitted...  
[student@serverb ~]$
```

- 5.2. В качестве пользователя **student** на **serverb** определите установленный **target** по умолчанию.

```
[student@serverb ~]$ systemctl get-default  
graphical.target
```

- 5.3. Переключитесь на **multi-user target**. Используйте команду **sudo** и, если будет предложено, используйте слово **student** в качестве пароля.

```
[student@serverb ~]$ sudo systemctl isolate multi-user.target  
[sudo] password for student: student
```

- 5.4. Установите на **serverb** использования **multi-user target** в качестве **target** по умолчанию.

```
[student@serverb ~]$ sudo systemctl set-default multi-user.target  
Removed /etc/systemd/system/default.target.  
Created symlink /etc/systemd/system/default.target -> /usr/lib/systemd/  
system/multi-user.target.
```

- 5.5. Перезагрузите **serverb**, чтобы убедиться, что **multi-user target** установлена в качестве **target** по умолчанию.

```
[student@serverb ~]$ sudo systemctl reboot  
Connection to serverb closed by remote host.  
Connection to serverb closed.  
[student@workstation ~]$
```

- 5.6. После перезагрузки откройте сеанс **SSH** для **serverb** как пользователь **student**. Убедитесь, что **multi-user target** установлена как **target** по умолчанию.

```
[student@workstation ~]$ ssh student@serverb
```

```
...output omitted...
[student@serverb ~]$ systemctl get-default
multi-user.target
```

6. Запланируйте повторяющееся задание в качестве пользователя `student`, которое выполняет сценарий `/home/student/backup-home.sh` каждый час с 19:00 до 20:00. и 9 вечера во все дни, кроме субботы и воскресенья.

Используйте сценарий `backup-home.sh`, чтобы запланировать повторяющееся задание. Загрузите сценарий резервного копирования с <http://materials.example.com/labs/backup-home.sh>.

- 6.1. На `serverb` загрузите сценарий резервного копирования с <http://materials.example.com/labs/backup-home.sh>. Используйте команду `chmod`, чтобы сделать исполняемый скрипт резервного копирования.

```
[student@serverb ~]$ wget http://materials.example.com/labs/backup-home.sh
...output omitted...
[student@serverb ~]$ chmod +x backup-home.sh
```

- 6.2. Используйте команду `crontab -e`, чтобы открыть файл `crontab` с помощью текстового редактора по умолчанию.

```
[student@serverb ~]$ crontab -e
```

- 6.3. Отредактируйте файл, добавив следующую строку:

```
0 19-21 * * Mon-Fri /home/student/backup-home.sh
```

Сохраните изменения и выйдите из редактора.

- 6.4. Используйте команду `crontab -l`, чтобы вывести список запланированных повторяющихся заданий.

```
[student@serverb ~]$ crontab -l
0 19-21 * * Mon-Fri /home/student/backup-home.sh
```

7. Перезагрузите `serverb` и дождитесь завершения загрузки перед оценкой выполнения лабораторной работы.

Оценка

На рабочей станции запустите сценарий **lab rhcsa-comprevew1 grade**, чтобы подтвердить успешное выполнение этого упражнения. Исправьте все сообщения об ошибках и перезапустите сценарий до тех пор, пока он не будет завершён без сообщений об ошибках.

```
[student@workstation ~]$ lab rhcsa-comprevew1 grade
```

Завершение

На рабочей станции **workstation**, выполните команду **lab rhcsa-comprevew1 finish**, чтобы завершить это упражнение. Данный сценарий удаляет файлы и ресурсы, созданные во время упражнения, и обеспечивает чистоту среды.

```
[student@workstation ~]$ lab rhcsa-comprevew1 finish
```

Сохраните любые файлы или работу, которые вы хотите сохранить, в других системах, а затем перезагрузите рабочую станцию **workstation**, серверы **servera** и **serverb** перед следующим упражнением.

На этом всеобъемлющий обзор завершается.

ЛАБОРАТОРНАЯ РАБОТА

КОНФИГУРИРОВАНИЕ И УПРАВЛЕНИЕ ФАЙЛОВЫМИ СИСТЕМАМИ И ХРАНИЛИЩАМИ

В этом обзоре вы создадите логический том **LVM**, смонтируете сетевую файловую систему, создадите раздел подкачки, который автоматически активируется при загрузке, настроите удаление временных неиспользуемых файлов из системы и используете **ACL** для защиты каталога.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создать логический том **LVM**.
- Монтировать сетевую файловую систему.
- Создайте раздел подкачки (**swop**), который автоматически активируется при загрузке.
- Настройте удаление временных неиспользуемых файлов из системы.
- Используйте **ACL** для защиты каталога.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Скопируйте любые файлы или работу, которые вы хотите сохранить, в другие системы перед сбросом. Перезагрузите системы рабочей станции **workstation**, **servera** и **serverb** сейчас, если только вы не завершили их сброс в конце последнего упражнения.

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-compreview2 start**, чтобы начать всестороннюю проверку. Этот сценарий создает необходимые файлы для правильной настройки среды.

```
[student@workstation ~]$ lab rhcsa-compreview2 start
```

ИНСТРУКЦИИ

Выполните следующие задачи на **serverb**, чтобы выполнить всестороннюю проверку.

- Настройте новый логический том размером **1 ГиБ** с именем **vol_home** в новой группе томов размером **2 ГиБ** с именем **extra_storage**. Используйте неразмеченный диск **/dev/vdb** для создания разделов.
- Логический том **vol_home** должен быть отформатирован с использованием файловой системы **XFS** и постоянно монтироваться в каталоги **/home**.
- Убедитесь, что сетевая файловая система с именем **/share** постоянно подключена к **/local-share** после перезагрузки. Сервер **NFS servera.lab.example.com** экспортирует сетевую файловую систему **/share**. Путь экспорта **NFS — servera.lab.example.com:/share**.
- Создайте новый раздел размером **512 МБ** на диске **/dev/vdc**, который будет использоваться в качестве пространства подкачки (**swop**). Это пространство подкачки должно автоматически активироваться при загрузке.

- Создайте новую группу под названием **production**. Создайте пользователей **production1**, **production2**, **production3** и **production4**. Убедитесь, что они используют новую группу под названием **production** в качестве дополнительной группы.
- Настройте свою систему так, чтобы она использовала новый каталог с именем **/run/volatile** для хранения временных файлов. Файлы в этом каталоге должны подлежать очистке на основе времени, если к ним не обращались более **30 секунд**. Восьмеричные разрешения для каталога должны быть **0700**. Убедитесь, что вы используете файл **/etc/tmpfiles.d/volatile.conf** для настройки очистки по времени для файлов в **/run/volatile**.
- Создайте новый каталог с именем **/webcontent**. И владелец, и группа каталога должны быть **root**. Члены группы **production** должны иметь возможность читать и писать в этот каталог. Пользователь **production1** должен иметь возможность только читать этот каталог. Эти разрешения должны применяться ко всем новым файлам и каталогам, созданным в каталоге **/webcontent**.

Оценка

На рабочей станции **workstation**, запустите сценарий **lab rhcsa-compreview2 grade** чтобы подтвердить успешное выполнение этого упражнения. Исправьте все сообщения об ошибках и перезапускайте сценарий до тех пор, пока он не будет успешно завершён.

```
[student@workstation ~]$ lab rhcsa-compreview2 grade
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab rhcsa-compreview2 finish**, чтобы завершить упражнение. Этот сценарий удаляет файлы и ресурсы, созданные во время упражнения, и обеспечивает чистоту среды.

```
[student@workstation ~]$ lab rhcsa-compreview2 finish
```

На этом всеобъемлющий обзор завершается.

РЕШЕНИЕ

КОНФИГУРИРОВАНИЕ И УПРАВЛЕНИЕ ФАЙЛОВЫМИ СИСТЕМАМИ И ХРАНИЛИЩАМИ

В этом обзоре вы создадите логический том **LVM**, смонтируете сетевую файловую систему, создадите раздел подкачки, который автоматически активируется при загрузке, настроите удаление временных неиспользуемых файлов из системы и используйте **ACL** для защиты каталога.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Создать логический том **LVM**.
- Монтировать сетевую файловую систему.
- Создайте раздел подкачки (**swop**), который автоматически активируется при загрузке.
- Настройте удаление временных неиспользуемых файлов из системы.
- Используйте **ACL** для защиты каталога.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Скопируйте любые файлы или работу, которые вы хотите сохранить, в другие системы перед сбросом. Перезагрузите системы рабочей станции **workstation**, **servera** и **serverb** сейчас, если только вы не завершили их сброс в конце последнего упражнения.

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-compreview2 start**, чтобы начать всестороннюю проверку. Этот сценарий создает необходимые файлы для правильной настройки среды.

```
[student@workstation ~]$ lab rhcsa-compreview2 start
```

ИНСТРУКЦИИ

Выполните следующие задачи на **serverb**, чтобы выполнить всестороннюю проверку.

- Настройте новый логический том размером **1 ГиБ** с именем **vol_home** в новой группе томов размером **2 ГиБ** с именем **extra_storage**. Используйте неразмеченный диск **/dev/vdb** для создания разделов.
- Логический том **vol_home** должен быть отформатирован с использованием файловой системы **XFS** и постоянно монтиrovаться в каталоги **/home**.
- Убедитесь, что сетевая файловая система с именем **/share** постоянно подключена к **/local-share** после перезагрузки. Сервер **NFS servera.lab.example.com** экспортирует сетевую файловую систему **/share**. Путь экспорта **NFS — servera.lab.example.com:/share**.
- Создайте новый раздел размером **512 МБ** на диске **/dev/vdc**, который будет использоваться в качестве пространства подкачки (**swop**). Это пространство подкачки должно автоматически активироваться при загрузке.

- Создайте новую группу под названием **production**. Создайте пользователей **production1**, **production2**, **production3** и **production4**. Убедитесь, что они используют новую группу под названием **production** в качестве дополнительной группы.
- Настройте свою систему так, чтобы она использовала новый каталог с именем **/run/volatile** для хранения временных файлов. Файлы в этом каталоге должны подлежать очистке на основе времени, если к ним не обращались более **30 секунд**. Восьмеричные разрешения для каталога должны быть **0700**. Убедитесь, что вы используете файл **/etc/tmpfiles.d/volatile.conf** для настройки очистки по времени для файлов в **/run/volatile**.
- Создайте новый каталог с именем **/webcontent**. И владелец, и группа каталога должны быть **root**. Члены группы **production** должны иметь возможность читать и писать в этот каталог. Пользователь **production1** должен иметь возможность только читать этот каталог. Эти разрешения должны применяться ко всем новым файлам и каталогам, созданным в каталоге **/webcontent**.

1. С рабочей станции **workstation**, откройте **SSH-сессию** на **serverb** как пользователь **student**.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
```

2. Переключитесь на пользователя **root**.

```
[student@serverb ~]$ sudo -i
[sudo] password for student: student
[root@serverb ~]#
```

3. Создайте раздел размером **2 ГиБ** в **/dev/vdb**.

```
[root@serverb ~]# parted /dev/vdb mklabel msdos
[root@serverb ~]# parted /dev/vdb mkpart primary 1GiB 3GiB
```

4. Создайте логический том с именем **vol_home**, используя раздел размером **2 ГиБ**, который вы создали в **/dev/vdb**. Назовите группу томов **extra_storage**.

4.1. Объявите блочное устройство **/dev/vdb1** как физический том.

```
[root@serverb ~]# pvcreate /dev/vdb1
...output omitted...
```

4.2. Создайте группу томов **extra_storage**, используя **/dev/vdb1**.

```
[root@serverb ~]# vgcreate extra_storage /dev/vdb1
...output omitted...
```

4.3. Создайте логический том размером **1 ГиБ** с именем **vol_home**.

```
[root@serverb ~]# lvcreate -L 1GiB -n vol_home extra_storage  
...output omitted...
```

5. Отформатируйте **vol_home** с типом файловой системы **XFS** и смонтируйте его в каталоги **/home**.

5.1. Создайте каталог с именем **/home-directories**.

```
[root@serverb ~]# mkdir /home-directories
```

5.2. Отформатируйте **/dev/extra_storage/vol_home** с типом файловой системы **XFS**.

```
[root@serverb ~]# mkfs -t xfs /dev/extra_storage/vol_home  
...output omitted...
```

5.3. Выполните постоянное монтирование **/dev/extra_storage/vol_home** в **/home-directories**. Используйте **UUID** структуры при создании записи в **/etc/fstab**.

```
[root@serverb ~]# lsblk -o UUID /dev/extra_storage/vol_home  
UUID  
988cf149-0667-4733-abca-f80c6ec50ab6  
[root@serverb ~]# echo "UUID=988cf149-0667-4733-abca-f80c6ec50ab6 /home-directories \  
xfs defaults 0 0" >> /etc/fstab  
[root@serverb ~]# mount -a
```

6. Убедитесь, что сетевая файловая система с именем **/share** постоянно подключена к **/localshare** после перезагрузки. Сервер **NFS servera.lab.example.com** экспортирует сетевую файловую систему **/share**. Путь экспорта **NFS — servera.lab.example.com:/share**.

6.1. Создайте каталог **/local-share**.

```
[root@serverb ~]# mkdir /local-share
```

6.2. Добавьте соответствующую запись в **/etc/fstab**, чтобы сетевая файловая система, доступная по адресу **servera.lab.example.com:/share**, постоянно монтировалась в **/local-share** при перезагрузке.

```
[root@serverb ~]# echo "servera.lab.example.com:/share /local-share \  
nfs rw,sync 0 0" >> /etc/fstab
```

6.3. Смонтируйте сетевую файловую систему в **/local-share** на основе записи в **/etc/fstab**.

```
[root@serverb ~]# mount /local-share
```

7. Создайте новый раздел размером **512 МБ** на диске **/dev/vdc**, который будет использоваться в качестве пространства подкачки. Это пространство подкачки должно автоматически активироваться во время загрузки.

7.1. Создайте раздел размером **512 МБ** в **/dev/vdc**.

```
[root@serverb ~]# parted /dev/vdc mklabel msdos  
[root@serverb ~]# parted /dev/vdc mkpart primary 1MiB 513MiB
```

7.2. Сделайте область подкачки на **/dev/vdc1**.

```
[root@serverb ~]# mkswap /dev/vdc1  
...output omitted...
```

7.3. Активируйте пространство подкачки, чтобы оно сохранялось при перезагрузке. Используйте **UUID** структуры при создании записи в **/etc/fstab**.

```
[root@serverb ~]# lsblk -o UUID /dev/vdc1  
UUID  
cc18ccb6-bd29-48a5-8554-546bf3471b69  
[root@serverb ~]# echo "UUID=cc18...1b69 swap \  
swap defaults 0 0" >> /etc/fstab  
[root@serverb ~]# swapon -a
```

8. Создайте пользователей **production1**, **production2**, **production3** и **production4**. Убедитесь, что они используют новую группу под названием **production** в качестве дополнительной группы.

```
[root@serverb ~]# groupadd production  
[root@serverb ~]# for i in 1 2 3 4; do useradd -G production production$i;  
Done
```

9. Настройте свою систему так, чтобы она использовала новый каталог с именем **/run/volatile** для хранения временных файлов. Файлы в этом каталоге должны подлежать очистке на основе времени, если к ним не обращались более **30 секунд**. Восьмеричные разрешения для каталога должны быть **0700**. Убедитесь, что вы используете файл **/etc/tmpfiles.d/volatile.conf** для настройки очистки по времени для файлов в **/run/volatile**.

9.1. Создайте файл с именем **/etc/tmpfiles.d/volatile.conf** со следующим содержимым.

```
d /run/volatile 0700 root root 30s
```

9.2. Используйте команду **systemd-tmpfiles --create**, чтобы создать каталог **/run/volatile**, если он не существует.

```
[root@servera ~]# systemd-tmpfiles --create /etc/tmpfiles.d/  
volatile.conf
```

10. Создайте новый каталог с именем **/webcontent**. И владелец, и владелец группы каталога должны быть **root**. Члены группы **production** должны иметь возможность читать и писать в этот каталог. Пользователь **production1** должен иметь возможность только читать этот каталог. Эти разрешения должны применяться ко всем новым файлам и каталогам, созданным в каталоге **/webcontent**.

10.1. Создайте каталог **/webcontent**.

```
[root@serverb ~]# mkdir /webcontent
```

10.2. Используйте **setfacl** для настройки разрешений на **/webcontent**, чтобы члены группы **production** имели права на чтение и запись, за исключением пользователя **production1**, которому должно быть предоставлено только разрешение на чтение.

```
[root@serverb ~]# setfacl -m u:production1:rx /webcontent  
[root@serverb ~]# setfacl -m g:production:rwx /webcontent
```

10.3. Используйте **setfacl**, чтобы установить разрешения по умолчанию для **/webcontent**, чтобы разрешения, которые вы применили на предыдущем шаге, также применялись ко всем новым файлам и каталогам, созданным в каталоге **/webcontent**.

```
[root@serverb ~]# setfacl -m d:u:production1:rx /webcontent  
[root@serverb ~]# setfacl -m d:g:production:rwx /webcontent
```

10.4. Выйдите из оболочки пользователя **root**.

```
[root@serverb ~]# exit  
logout
```

10.5. Выйти с сервера **serverb**.

```
[student@serverb ~]$ exit
```

```
logout  
Connection to server closed.
```

Оценка

На рабочей станции **workstation**, запустите сценарий **lab rhcsa-comprevew2 grade** чтобы подтвердить успешное выполнение этого упражнения. Исправьте все сообщения об ошибках и перезапускайте сценарий до тех пор, пока он не будет успешно завершён.

```
[student@workstation ~]$ lab rhcsa-comprevew2 grade
```

Завершение

На рабочей станции **workstation**, запустите сценарий **lab rhcsa-comprevew2 finish**, чтобы завершить упражнение. Этот сценарий удаляет файлы и ресурсы, созданные во время упражнения, и обеспечивает чистоту среды.

```
[student@workstation ~]$ lab rhcsa-comprevew2 finish
```

На этом всеобъемлющий обзор завершается.

ЛАБОРАТОРНАЯ РАБОТА

НАСТРОЙКА И УПРАВЛЕНИЕ БЕЗОПАСНОСТЬЮ СЕРВЕРА

В этом обзоре вы настроите аутентификацию на основе ключа **SSH**, измените настройки брандмауэра, настройте режим **SELinux** и логическое значение **SELinux**, а также устранитте проблемы с **SELinux**.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Настройте ключи **SSH** для аутентификации на основе ключей.
- Настройте параметры брандмауэра.
- Настройте режим **SELinux** и логические значения **SELinux**.
- Устранить неполадки **SELinux**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-comprevew3 start**, чтобы начать всестороннюю проверку. Этот сценарий создает необходимые файлы для правильной настройки среды.

```
[student@workstation ~]$ lab rhcsa-comprevew3 start
```

ИНСТРУКЦИИ

Выполните следующие задачи, чтобы завершить всесторонний обзор:

- Сгенерируйте **SSH-ключи** для учащегося на **serverb**. Не защищайте закрытый ключ парольной фразой.
- На сервере **servera** настройте пользователя **student** для принятия аутентификации при входе с использованием **пары ключей SSH**, созданной для **student** на **serverb**. Пользователь **student** на **serverb** должен иметь возможность войти в **servera** с помощью **SSH без ввода пароля**. При необходимости используйте **student** в качестве пароля пользователя **student**.
- На сервере **servera**, измените режим **SELinux** по умолчанию на **permissive**.
- Настройте **serverb** для автоматического монтирования домашнего каталога **production5**, когда пользователь входит в систему, используя сетевую файловую систему **/home-directories/production5**. Сетевая файловая система экспортируется с сервера **servera.lab.example.com**. Настройте соответствующее логическое значение **SELinux**, чтобы **production5** мог использовать смонтированный **NFS** домашний каталог на **serverb** после аутентификации с помощью аутентификации на **основе ключа SSH**. Пароль пользователя **production5** — **redhat**.

- На **serverb** настройте параметры брандмауэра таким образом, чтобы соединения **SSH**, исходящие от **servera**, отвергались.
- На **serverb** изучите и устраните проблему с демоном **Apache HTTPD**, который настроен на прослушивание порта **30080/TCP**, но не запускается. Настройте параметры брандмауэра соответствующим образом, чтобы **порт 30080/TCP** был открыт для входящих подключений.

Оценка

На рабочей станции **workstation**, запустите сценарий **lab rhcsa-comprevew3 grade**, чтобы подтвердить успешное выполнение упражнения. Исправьте все сообщения об ошибках и перезапустите сценарий до тех пор, пока он не будет успешным.

```
[student@workstation ~]$ lab rhcsa-comprevew3 grade
```

Завершение

На рабочей станции запустите **lab rhcsa-comprevew3 finish**, чтобы завершить это упражнение. Этот сценарий удаляет файлы и ресурсы, созданные во время упражнения, и обеспечивает чистоту среды.

```
[student@workstation ~]$ lab rhcsa-comprevew3 finish
```

Сохраните любые файлы или работу, которые вы хотите сохранить, в других системах, а затем перезагрузите рабочую станцию **workstation**, серверы **servera** и **serverb**.

На этом исчерзывающий обзор заканчивается.

РЕШЕНИЕ

НАСТРОЙКА И УПРАВЛЕНИЕ БЕЗОПАСНОСТЬЮ СЕРВЕРА

В этом обзоре вы настроите аутентификацию на основе ключа **SSH**, измените настройки брандмауэра, настройте режим **SELinux** и логическое значение **SELinux**, а также устранитте проблемы с **SELinux**.

В РЕЗУЛЬТАТЕ

Вы должны быть способны:

- Настройте ключи **SSH** для аутентификации на основе ключей.
- Настройте параметры брандмауэра.
- Настройте режим **SELinux** и логические значения **SELinux**.
- Устранить неполадки **SELinux**.

ПРЕЖДЕ ЧЕМ ВЫ НАЧНЕТЕ

Войдите на рабочую станцию **workstation** как пользователь **student**, используя в качестве пароля слово **student**.

На рабочей станции **workstation** выполните скрипт **lab rhcsa-comprevew3 start**, чтобы начать всестороннюю проверку. Этот сценарий создает необходимые файлы для правильной настройки среды.

```
[student@workstation ~]$ lab rhcsa-comprevew3 start
```

ИНСТРУКЦИИ

Выполните следующие задачи, чтобы завершить всесторонний обзор:

- Сгенерируйте **SSH-ключи** для учащегося на **serverb**. Не защищайте закрытый ключ парольной фразой.
- На сервере **servera** настройте пользователя **student** для принятия аутентификации при входе с использованием **пары ключей SSH**, созданной для **student** на **serverb**. Пользователь **student** на **serverb** должен иметь возможность войти в **servera** с помощью **SSH без ввода пароля**. При необходимости используйте **student** в качестве пароля пользователя **student**.
- На сервере **servera**, измените режим **SELinux** по умолчанию на **permissive**.
- Настройте **serverb** для автоматического монтирования домашнего каталога **production5**, когда пользователь входит в систему, используя сетевую файловую систему **/home-directories/production5**. Сетевая файловая система экспортируется с сервера **servera.lab.example.com**. Настройте соответствующее логическое значение **SELinux**, чтобы **production5** мог использовать смонтированный **NFS** домашний каталог на **serverb** после аутентификации с помощью аутентификации на **основе ключа SSH**. Пароль пользователя **production5** — **redhat**.

- На **serverb** настройте параметры брандмауэра таким образом, чтобы соединения **SSH**, исходящие от **servera**, отвергались.
- На **serverb** изучите и устраните проблему с демоном **Apache HTTPD**, который настроен на прослушивание порта **30080/TCP**, но не запускается. Настройте параметры брандмауэра соответствующим образом, чтобы **порт 30080/TCP** был открыт для входящих подключений.

1. С рабочей станции **workstation**, откройте **SSH-сессию** на **serverb** в качестве пользователя **student**.

```
[student@workstation ~]$ ssh student@serverb
...output omitted...
```

2. Сгенерируйте **SSH-ключи** для студента на **serverb** с помощью команды **ssh-keygen**. Не защищайте закрытый ключ парольной фразой.

```
[student@serverb ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/student/.ssh/id_rsa): Enter
Created directory '/home/student/.ssh'.
Enter passphrase (empty for no passphrase): Enter
Enter same passphrase again: Enter
Your identification has been saved in /home/student/.ssh/id_rsa.
Your public key has been saved in /home/student/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:1TPZ4TXYwiGWfExUGtRTHgfKQbF9hVuLa+VmH4vgkFY
student@serverb.lab.example.com
The key's randomart image is:
+---[RSA 2048]----+
| .+@B0** |
| .=.#+B* |
| . X.*o= |
| . E +.+ |
| S o + |
| + . o = |
| . o o + +|
| . . . . |
|           |
+---[SHA256]----+
```

3. На сервере **servera** настройте пользователя **student** для принятия аутентификации при входе с помощью **пары ключей SSH**, которую вы создали для учащегося на **serverb**. Пользователь **student** на **serverb** должен иметь возможность войти в **servera** с помощью **SSH** без ввода пароля. При необходимости используйте **student** в качестве пароля пользователя **student**.

- 3.1. Используйте команду **ssh-copy-id** для передачи открытого ключа пары ключей **SSH** **student** на **serverb** **student** на **servera**. Используйте **student** в качестве пароля пользователя **student**, если будет предложено.

```
[student@serverb ~]$ ssh-copy-id student@servera
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/
student/.ssh/id_rsa.pub"
The authenticity of host 'servera (172.25.250.10)' can't be
established.
ECDSA key fingerprint is SHA256:g/
fIMtVzDWTbTi1l0OwC30sL6cHmro9Tf563NxmeyyE.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s),
to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you
are prompted now it is to install the new keys
student@servera's password: student
```

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'student@servera'"
and check to make sure that only the key(s) you wanted were added.

- 3.2.** Используйте команду **ssh**, чтобы убедиться, что пользователь **student** может войти на **servera** с **serverb** без ввода пароля.

```
[student@serverb ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

- 4.** На сервере измените режим **SELinux** по умолчанию на разрешающий (**permissive**).

- 4.1.** Отредактируйте файл **/etc/sysconfig/selinux**, чтобы установить значение **permissive** параметра **SELINUX**. Вы можете использовать команду **sudo vi /etc/sysconfig/selinux** для редактирования файла конфигурации от имени суперпользователя. Используйте пароль **student**, если будет предложено.

```
...output omitted...
#SELINUX=enforcing
SELINUX=permissive
...output omitted...
```

- 4.2.** Используйте команду **sudo systemctl reboot**, чтобы перезагрузить систему в качестве суперпользователя.

```
[student@servera ~]$ sudo systemctl reboot
Connection to servera closed by remote host.
Connection to servera closed.
[student@serverb ~]$
```

5. Настройте **serverb** для автоматического монтирования домашнего каталога пользователя **production5**, когда пользователь входит в систему, используя сетевую файловую систему **/home-directories/production5**. Эта сетевая файловая система экспортируется с сервера **servera.lab.example.com**. Настройте соответствующее логическое значение **SELinux**, чтобы **production5** мог использовать смонтированный **NFS** домашний каталог на **serverb** после аутентификации с помощью аутентификации на основе ключа **SSH**. Пароль пользователя **production5** — **redhat**.

- 5.1. На **serverb** используйте команду **sudo -i**, чтобы переключиться на учетную запись пользователя **root**.

```
[student@serverb ~]$ sudo -i  
[sudo] password for student:  
[root@serverb ~]#
```

- 5.2. Установите пакет **autofs**.

```
[root@serverb ~]# yum install autofs  
...output omitted...  
Is this ok [y/N]: y  
...output omitted...  
Installed:  
    autofs-1:5.1.4-29.el8.x86_64  
  
Complete!
```

- 5.3. Создайте файл мастер сопоставления **autofs** с именем **/etc/auto.master.d/production5.autofs** со следующим содержимым.

```
/- /etc/auto.production5
```

- 5.4. Создайте файл **/etc/auto.production5** со следующим содержимым.

```
/localhome/production5 -rw servera.lab.example.com:/home-directories/  
production5
```

- 5.5. Перезапустите службу **autofs**.

```
[root@serverb ~]# systemctl restart autofs
```

6. На сервере **servera** убедитесь, что пользователь **production5** не может войти на **serverb**, используя аутентификацию с **открытым ключом SSH**. Логическое значение **SELinux** вызывает эту проблему, которую вы исправите в следующих шагах.

6.1. С рабочей станции **workstation**, откройте сеанс SSH на **servera** как пользователь **student**.

```
[student@workstation ~]$ ssh student@servera
...output omitted...
[student@servera ~]$
```

6.2. Переключитесь на пользователя **production5**, используя пароль **redhat**.

```
[student@servera ~]$ su - production5
Password: redhat
[production5@servera ~]$
```

6.3. Используйте команду **ssh-keygen** для создания ключей SSH **production5**.

```
[production5@servera ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/production5/.ssh/
id_rsa): Enter
Created directory '/home/production5/.ssh'.
Enter passphrase (empty for no passphrase): Enter
Enter same passphrase again: Enter
Your identification has been saved in /home/production5/.ssh/id_rsa.
Your public key has been saved in /home/production5/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:zmin1nmCt4H8LA+4FPimtdg81n17ATbInUFW3HSPxk4
    production5@servera.lab.example.com
The key's randomart image is:
+---[RSA 2048]---+
|       .00.0. . |
|       ... .o o |
|       . o o     E .|
|       . o *     + |
|       . . .So   . |
|       . + =   . |
|       *.*+=.. . |
|       0o+***.o   |
|       o.=o.=**   |
+---[SHA256]-----+
```

6.4. Используйте команду **ssh-copy-id** для передачи открытого ключа **пары ключей SSH** от **production5** на **servera** до **production5** на **serverb**. Используйте **redhat** в качестве пароля пользователя **production5**, если будет предложено.

```
[production5@servera ~]$ ssh-copy-id production5@serverb
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/
production5/.ssh/id_rsa.pub"
The authenticity of host 'serverb (172.25.250.11)' can't be
established.
ECDSA key fingerprint is
SHA256:ciCkaRWF4g6eR9nSdPxQ7KL8czpViXal6BousK544TY.
Are you sure you want to continue connecting (yes/no)? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s),
to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you
are prompted now it is to install the new keys
production5@serverb's password: redhat
```

Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'production5@serverb'"
and check to make sure that only the key(s) you wanted were added.

- 6.5.** Используйте аутентификацию на основе открытого ключа **SSH** вместо аутентификации на основе пароля, чтобы войти на **serverb** как пользователь **production5**. Эта команда должна завершиться ошибкой.

```
[production5@servera ~]$ ssh -o pubkeyauthentication=yes \
-o passwordauthentication=no production5@serverb
production5@serverb: Permission denied (publickey,gssapi-keyex,gssapiwith-
mic,password).
```

- 7.** Установите соответствующую логическую настройку **SELinux** на **serverb**, чтобы пользователь **production5** мог войти на **serverb**, используя аутентификацию на основе открытого ключа **SSH**, и используйте домашний каталог.

- 7.1.** На **serverb** с правами **root** установите для логического значения **use_nfs_home_dirs** **SELinux** значение **true**.

```
[root@serverb ~]# setsebool -P use_nfs_home_dirs true
```

- 7.2.** Используйте аутентификацию на основе открытого ключа **SSH** вместо аутентификации на основе пароля, чтобы войти на **serverb** как пользователь **production5**. Команда должна выполниться успешно.

```
[production5@servera ~]$ ssh -o pubkeyauthentication=yes \
-o passwordauthentication=no production5@serverb
...output omitted...
[production5@serverb ~]$
```

8. На **serverb** настройте параметры брандмауэра так, чтобы SSH-соединения, исходящие с сервера **servera**, отвергались. Система **servera** использует адрес IPv4 **172.25.250.10**.

- 8.1. Используйте команду **firewall-cmd**, чтобы добавить IPv4-адрес **servera** в зону **firewalld** с именем **block**.

```
[root@serverb ~]# firewall-cmd --add-source=172.25.250.10/32 \
--zone=block --permanent
success
```

- 8.2. Используйте команду **firewall-cmd --reload**, чтобы перезагрузить изменения в настройках брандмауэра.

```
[root@serverb ~]# firewall-cmd --reload
success
```

9. На **serverb**, изучите и устраните проблему с демоном **Apache HTTPD**, который настроен на прослушивание **порта 30080/TCP**, но не запускается. Настройте параметры брандмауэра соответствующим образом, чтобы порт **30080/TCP** был открыт для входящих подключений.

- 9.1. Используйте команду **systemctl** для перезапуска службы **httpd**. Эта команда не может перезапустить службу.

```
[root@serverb ~]# systemctl restart httpd.service
Job for httpd.service failed because the control process exited with
error code.
See "systemctl status httpd.service" and "journalctl -xe" for details.
```

- 9.2. Используйте команду **systemctl status**, чтобы выяснить причину сбоя службы **httpd**.

```
[root@serverb ~]# systemctl status httpd.service
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled;
   vendor preset: disabled)
     Active: failed (Result: exit-code) since Mon 2019-04-15 06:42:41
   EDT; 5min ago
       Docs: man:httpd.service(8)
     Process: 27313 ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
               (code=exited, status=1/FAILURE)
   Main PID: 27313 (code=exited, status=1/FAILURE)
      Status: "Reading configuration..."

Apr 15 06:42:41 serverb.lab.example.com systemd[1]: Starting The Apache
HTTP Server...
Apr 15 06:42:41 serverb.lab.example.com httpd[27313]: (13)Permission
denied: AH00072: make_sock: could not bind to address [::]:30080
Apr 15 06:42:41 serverb.lab.example.com httpd[27313]: (13)Permission
denied: AH00072: make_sock: could not bind to address 0.0.0.0:30080
Apr 15 06:42:41 serverb.lab.example.com httpd[27313]: no listening
sockets available, shutting down
Apr 15 06:42:41 serverb.lab.example.com httpd[27313]: AH00015: Unable
to open logs
Apr 15 06:42:41 serverb.lab.example.com systemd[1]: httpd.service: Main
process exited, code=exited, status=1/FAILURE
Apr 15 06:42:41 serverb.lab.example.com systemd[1]: httpd.service:
Failed with result 'exit-code'.
Apr 15 06:42:41 serverb.lab.example.com systemd[1]: Failed to start The
Apache HTTP Server.
```

Обратите внимание на ошибку разрешения в предыдущем выводе, которая означает, что демону **httpd** не удалось выполнить привязку к **порту 30080/TCP**. Политика **SELinux** может быть потенциальным ограничением привязки приложения к порту. Нажмите **q**, чтобы выйти из предыдущей команды **systemctl**.

9.3. Используйте команду **sealert**, чтобы определить, препятствует ли политика **SELinux** привязке **httpd** к порту **30080/TCP**.

```
[root@serverb ~]# sealert -a /var/log/audit/audit.log
100% done
found 1 alerts in /var/log/audit/audit.log
-----
SELinux is preventing /usr/sbin/httpd from name_bind access on the
tcp_socket port 30080.

***** Plugin bind_ports (92.2 confidence) suggests
***** If you want to allow /usr/sbin/httpd to bind to network port 30080
Then you need to modify the port type.
Do
# semanage port -a -t PORT_TYPE -p tcp 30080
  where PORT_TYPE is one of the following: http_cache_port_t,
http_port_t, jboss_management_port_t, jboss.messaging_port_t,
ntop_port_t, puppet_port_t.
```

Предыдущее сообщение журнала показывает, что **порт 30080/TCP** не имеет соответствующего контекста **SELinux http_port_t**, из-за чего **SELinux** предотвращает привязку **httpd** к этому порту. В сообщении журнала также отображается синтаксис команды **semanage port**, чтобы вы могли легко устранить проблему.

- 9.4.** Используйте команду **semanage port**, чтобы установить соответствующий контекст **SELinux** на порту **30080/TCP**, чтобы **httpd** привязывался к нему.

```
[root@serverb ~]# semanage port -a -t http_port_t -p tcp 30080
```

- 9.5.** Используйте команду **systemctl** для перезапуска **httpd**. Эта команда должна успешно перезапустить службу.

```
[root@serverb ~]# systemctl restart httpd
```

- 9.6.** Добавьте порт **30080/TCP** в зону **firewalld** по умолчанию, которая называется **public**.

```
[root@serverb ~]# firewall-cmd --add-port=30080/tcp --permanent
success
[root@serverb ~]# firewall-cmd --reload
success
```

- 9.7.** Выходите из оболочки пользователя **root**.

```
[root@serverb ~]# exit
```

```
logout
```

9.8. Выйдите из **serverb**.

```
[student@serverb ~]$ exit  
logout  
Connection to serverb closed.
```

Оценка

На рабочей станции **workstation**, запустите сценарий **lab rhcsa-compreview3 grade**, чтобы подтвердить успешное выполнение упражнения. Исправьте все сообщения об ошибках и перезапустите сценарий до тех пор, пока он не будет успешным.

```
[student@workstation ~]$ lab rhcsa-compreview3 grade
```

Завершение

На рабочей станции запустите **lab rhcsa-compreview3 finish**, чтобы завершить это упражнение. Этот сценарий удаляет файлы и ресурсы, созданные во время упражнения, и обеспечивает чистоту среды.

```
[student@workstation ~]$ lab rhcsa-compreview3 finish
```

Сохраните любые файлы или работу, которые вы хотите сохранить, в других системах, а затем перезагрузите рабочую станцию **workstation**, серверы **servera** и **serverb**.

На этом исчерпывающий обзор завершается.