Руководство по Ubuntu Server

Руководство по Ubuntu Server

Аннотация

Добро пожаловать в *Руководство Ubuntu Server*! Оно содержит информацию о том, как установить и настроить различные серверные приложения на вашей системе Ubuntu в зависимости от ваших потребностей. Это пошаговое, ориентированное на конкретные задачи руководство по конфигурации и модернизации вашей системы.

Разработчики и лицензия

Этот документ поддерживается командой документации Ubuntu (https://wiki.ubuntu.com/DocumentationTeam). Список авторов приведён ниже.

Авторы этого документа :

- Участники *Проекта документирования Ubuntu*[2]
- Участники *команды Ubuntu Server*[3]
- Contributors to the *Community Help Wiki*[4]
- Информацию о других участниках можно найти в истории версий веток bzr *serverguide*[5] и *ubuntu-docs*[6], доступных на Launchpad.

[1] http://creativecommons.org/licenses/by-sa/3.0/
[2] https://launchpad.net/~ubuntu-core-doc
[3] https://launchpad.net/~ubuntu-server
[4] https://help.ubuntu.com/community/
[5] https://code.launchpad.net/serverguide
[6] https://code.launchpad.net/ubuntu-docs

Содержание

Список таблиц

Глава 1. Введение

Добро пожаловать в *Руководство по Ubuntu Server*!

Здесь вы найдете информацию о том, как устанавливать и настраивать различные серверные приложения. Это пошаговое, ориентированное на конкретные задачи руководство по конфигурации и настройке вашей системы.

This guide assumes you have a basic understanding of your Ubuntu system. Some installation details are covered in *Глава 2, Установка [3]*, but if you need detailed instructions installing Ubuntu please refer to the *Ubuntu Installation Guide*[1].

HTML-версия руководства доступна в Интернете на *сайте документации Ubuntu*[2].

[1] https://help.ubuntu.com/14.04/installation-guide/
[2] https://help.ubuntu.com

1. Поддержка

Существует два варианта поддержки Ubuntu Server Edition: коммерческая поддержка и поддержка сообщества. Основную коммерческую поддержку (и финансирование разработки) осуществляет Canonical Ltd. За умеренную плату она предоставляет поддержку по договорам на каждую настольную систему или на сервер. Подробную информацию смотрите на странице *Canonical Services*[3].

Поддержка со стороны сообщества означает, что поддержку осуществляют специалисты и компании, которые хотят сделать Ubuntu лучшим дистрибутивом. Поддержка предоставляется через почтовые рассылки, IRC-каналы, форумы, блоги, вики, и т.д. Очень много информации можно найти в поисковиках. Для получения дополнительной информации смотрите страницу *Ubuntu Support*[4]

[3] http://www.canonical.com/services/support
[4] http://www.ubuntu.com/support

Глава 2. Установка

This chapter provides a quick overview of installing Ubuntu 14.04 LTS Server Edition. For more detailed instructions, please refer to the *Ubuntu Installation Guide*[1].

[1] https://help.ubuntu.com/14.04/installation-guide/

1. Подготовка к установке

В данном разделе рассмотрены некоторые аспекты, которые необходимо чётко понимать до начала установки.

1.1. Системные требования

Ubuntu 14.04 LTS Server Edition supports three (3) major architectures: Intel x86, AMD64 and ARM. The table below lists recommended hardware specifications. Depending on your needs, you might manage with less than this. However, most users risk being frustrated if they ignore these suggestions.

Таблица 2.1. Рекомендованные минимальные требования

Тип установки	**Процессор**	**ОЗУ**	**Место на жёстком диске**	
			Базовая система	**Установлены все задачи**
Сервер (стандартный)	1 гигагерц	512 мегабайт	1 ГБ	1,75 гигабайт
Сервер (минимальный)	300 МГц	192 megabytes	700 мегабайт	1.4 гигабайт

Server Edition предоставляет основу для всех видов серверных приложений. В платформу уже включены такие сервисы, как: файловый и почтовый серверы, веб-хостинг, сервер электронной почты и т.д.

1.2. Различия между серверной и настольной редакциями

Есть несколько различий между *Ubuntu Server Edition* и *Ubuntu Desktop Edition*. Следует отметить, что обе редакции используют одни и те же репозитории apt, что дает возможность так же легко установить серверные приложения на Desktop Edition, как и на Server Edition.

Различия между этими двумя редакциями заключаются в отсутствии в Server Edition графической оконной среды X и в некоторых особенностях процесса установки.

1.2.1. Различия ядер:

В Ubuntu 10.10 и более ранних версиях использовались два различных ядра для серверной версии и версии для настольных компьютеров. Теперь в Ubuntu нет отдельных вариантов ядра -server и -generic. Они объединены

в один вариант ядра -generic, чтобы снизить расходы на обслуживание в течение жизненного цикла выпуска.

При работе в 64-разрядной версии Ubuntu на 64-битных процессорах вы не ограничены по адресному пространству памяти.

To see all kernel configuration options you can look through `/boot/config-3.13.0-server`. Also, *Linux Kernel in a Nutshell*[2] is a great resource on the options available.

1.3. Создание резервной копии

- Перед установкой Ubuntu Server Edition следует убедиться, что сделана резервная копия всех файлов системы. О способах резервного копирования читайте *Глава 19, Резервное копирование [345]*.

  Если на вашем компьютере уже установлена какая-то другая операционная система, то скорее всего вам нужно будет переразметить ваш диск, чтобы освободить место для Ubuntu.

  Каждый раз при разметке диска, вы должны быть готовы потерять всё на диске, если вы сделаете ошибку или что-то пойдет не так во время разметки. Программы, используемые в установке, достаточно надежны, большинство из них работают не первый год, но они также могут выполнить деструктивные действия.

[2] http://www.kroah.com/lkn/

2. Установка с CD

Основные этапы установки Ubuntu Server Edition с CD такие же, как и при установке любой операционной системы с компакт-диска. В отличие от *Desktop Edition*, *Server Edition* не включает в себя графическую программу установки. Server Edition использует вместо неё консольное меню.

- Прежде всего, скачайте и запишите на диск соответствующий файл ISO с *веб-сайта Ubuntu*[3].
- Загрузите систему с компакт-диска.
- В приглашении загрузчика вам будет предложено выбрать язык.
- В главном меню загрузки есть некоторые дополнительные опции для установки Ubuntu Server Edition. Вы можете установить базовую комплектацию Ubuntu Server, проверить компакт-диск на наличие дефектов, проверить оперативную память системы, загрузиться с первого жёсткого диска, или восстановить повреждённую систему.
- Программа установки спросит, какой язык она должна в дальнейшем использовать. После этого вам будет предложено выбрать ваше местоположение.
- Процесс установки начинается с выбора раскладки клавиатуры. Вы можете разрешить программе установки попытаться автоматически определить раскладку или же можете выбрать необходимую раскладку вручную из списка.
- Затем программа установки определяет вашу аппаратную конфигурацию и настраивает параметры сети с помощью DHCP. Если вы не хотите использовать DHCP, на следующем экране выберите "Назад", и у вас будет возможность "настроить сеть вручную".
- Next, the installer asks for the system's hostname.
- Будет создан новый пользователь; этот пользователь будет иметь *root* доступ через утилиту sudo.
- После того, как настройки пользователя будут завершены, вам будет предложено зашифровать свой каталог (`home`).
- Next, the installer asks for the system's Time Zone.
- Затем вы сможете выбрать один из нескольких вариантов настройки структуры разделов жёсткого диска. После этого вас спросят, какой диск использовать для установки. В зависимости от разбивки диска, прежде чем будет перезаписана таблицы разделов или настроена LVM, вы можете получить запрос на подтверждение данной операции. Если вы выбираете LVM, вам будет предложено ввести размер корневого

[3] http://www.ubuntu.com/download/server/download

логического тома. Для продвинутых вариантов установки смотрите *Раздел 4, «Расширенная установка» [11]*.

- После этого будет установлена базовая система Ubuntu.
- Следующий шаг в процессе установки — это решить, как вы хотите обновить систему. Есть три варианта:
  - *Нет автоматического обновления*: администратору будет необходимо войти в систему и вручную установить обновления.
  - *Установка обновлений безопасности автоматически*: это позволит установить пакет unattended-upgrades, который будет устанавливать обновления безопасности без вмешательства администратора. Для получения дополнительной информации смотрите *Раздел 5, «Автоматические обновления» [32]*.
  - *Управление системой с Landscape*: Landscape — это платный сервис, предоставляемый компанией Canonical, чтобы помочь в управлении вашими компьютерами с Ubuntu. Подробности смотрите на сайте *Landscape*[4].
- Теперь у вас есть возможность установить, или не устанавливать, несколько пакетов задач. Смотрите *Раздел 2.1, «Наборы пакетов (задачи)» [8]* для более подробной информации. Также, есть опция для запуска aptitude, чтобы выбрать отдельные пакеты для установки. Для получения дополнительной информации смотрите *Раздел 4, «Aptitude» [30]*.
- И, наконец, последний шаг перед перезагрузкой — это установить часы относительно UTC.

Если на каком-то этапе установки вас не устроят настройки по умолчанию, используйте функцию «Назад» на любом экране для перехода к меню детальных настроек, которое позволит вам изменить настройки по умолчанию.

В какой-то момент в процессе установки вы можете захотеть получить помощь, предоставляемую системой установки. Вы можете вызвать ее нажатием F1.

Once again, for detailed instructions see the *Ubuntu Installation Guide*[5].

[4] http://www.canonical.com/projects/landscape
[5] https://help.ubuntu.com/14.04/installation-guide/

2.1. Наборы пакетов (задачи)

Во время установки Server Edition у вас будет возможность выбрать установку дополнительных пакетов с установочного CD. Эти пакеты сгруппированы по типам предоставляемых ими сервисов.

- DNS сервер: Выбирает DNS-сервер BIND и документацию по нему.
- LAMP сервер: Выбирает установку готового к работе сервера Linux/Apache/MySQL/PHP.
- Почтовый сервер: Этот набор выбирает множество пакетов, требующихся для типичного почтового сервера.
- OpenSSH сервер: Выбирает пакеты, необходимые для установки сервера OpenSSH.
- База данных PostgreSQL: Этот набор включает клиентские и серверные пакеты для установки системы управления базами данных PostgreSQL.
- Сервер печати: Этот набор делает вашу систему сервером печати.
- Файловый сервер Samba: Этот набор настроит вашу систему, как файловый сервер Samba, который особенно подходит для сетей, где присутствуют как Windows, так и Linux системы.
- Tomcat Java сервер: Установит Apache Tomcat и необходимые зависимости.
- Virtual Machine host: Добавит пакеты, требующиеся для запуска виртуальных машин KVM.
- Выбор пакетов вручную: Запускает aptitude, позволяющий вам выбирать пакеты индивидуально.

Установка групп пакетов (задач) выполняется с помощью утилиты tasksel. Одним из важнейших отличий Ubuntu (или Debian) от других дистрибутивов на основе GNU/Linux является то, что будучи установленным, пакет также получает разумные настройки по умолчанию, запрашивая у вас по необходимости дополнительную информацию. Подобным образом, когда устанавливается группа пакетов, пакеты не только устанавливаются, но и настраиваются для предоставления полностью связанного сервиса.

Когда процесс установки завершится, вы сможете увидеть список доступных задач, введя следующую команду в терминале:

```
tasksel --list-tasks
```

В выводе будут показаны задачи, относящиеся и к другим дистрибутивам на базе Ubuntu, таким как Kubuntu и Edubuntu.

> Заметьте, что если вы выполните команду **tasksel** без ключей, она покажет меню со всеми доступными задачами.

Вы можете увидеть список устанавливаемых пакетов для каждой задачи, используя опцию *--task-packages*. Например, для вывода списка пакетов, устанавливаемых задачей «DNS сервер», введите следующее:

```
tasksel --task-packages dns-server
```

Результатом выполнения команды будет следующее:

```
bind9-doc
bind9utils
bind9
```

Если вы не установили какую-либо задачу в процессе установки, но, например, решили сделать ваш новый сервер LAMP ещё и DNS-сервером, просто вставьте установочный CD и введите в терминале:

```
sudo tasksel install dns-server
```

3. Обновление

Существует несколько путей обновления выпусков Ubuntu с одного на другой. Этот раздел представляет обзор рекомендуемых методов обновления.

3.1. do-release-upgrade

Рекомендуемый вариант обновления установленного Server Edition заключается в использовании утилиты do-release-upgrade. Являясь частью пакета *update-manager-core*, она не имеет графического интерфейса и устанавливается по умолчанию.

Системы, основанные на дистрибутиве Debian, могут также обновляться с использованием команды **apt-get dist-upgrade**. Однако использование do-release-upgrade предпочтительней, поскольку позволяет отслеживать изменения в конфигурациях систем при переходе от выпуска к выпуску.

Для обновления до нового выпуска введите в терминале команду:

```
do-release-upgrade
```

Также существует возможность обновления с помощью do-release-upgrade до разрабатываемой версии Ubuntu. Для этого дополните команду опцией *-d*:

```
do-release-upgrade -d
```

Обновление до выпуска, находящегося в разработке, *не* рекомендуется для работающих «боевых» систем.

4. Расширенная установка

4.1. Программный RAID

Избыточный массив независимых дисков (Redundant Array of Independent Disks, RAID) — это метод использования нескольких дисков для различных сочетаний увеличения надёжности хранения данных и/или увеличения производительности операций чтения/записи в зависимости от используемого уровня RAID. RAID реализуется либо на программном уровне (когда операционная система знает про оба носителя и активно их обслуживает), либо на аппаратном (когда специальный контроллер заставляет ОС думать, что существует только один носитель и обслуживает носители незаметно для системы).

Программное обеспечение для работы с RAID, включенное в текущие версии Linux (и Ubuntu), основано на драйвере 'mdadm' и работает очень хорошо, даже лучше чем многие, так называемые, «аппаратные» RAID-контроллеры. Этот раздел руководства поможет вам установить Ubuntu Server Edition, используя два раздела RAID первого уровня (RAID 1), находящиеся на двух физических жёстких дисках, один для */* (корневого раздела), а другой для раздела подкачки *swap*.

4.1.1. Разметка дисков

Следуйте инструкциям по установке, пока вы не достигнете этапа разметки дисков, а затем:

1. Выберите метод разметки *Вручную*.
2. Выберите первый жёсткий диск и согласитесь с предложением "*Создать новую пустую таблицу разделов на этом устройстве?*".

   Повторите этот шаг для каждого диска, который вы собираетесь включить в RAID массив.
3. Выберите "*СВОБОДНОЕ МЕСТО*"на первом носителе и выберите "*Создать новый раздел*".
4. Далее, выберите *размер* раздела. Этот раздел будет *разделом подкачки*, а общее правило для определения размера раздела подкачки — сделать его равным двойному объёму RAM. Введите размер, далее выберите *Первичный*, затем *Начало*.

   

   Двойной размер раздела подкачки по отношению к оперативной памяти (RAM) не всегда желателен, особенно на системах с большим объемом RAM. Расчёт размера раздела подкачки в

значительной степени зависит от того, как будет использоваться система.

5. Выберите строку "*Использовать как:*" вверху. По умолчанию там установлено "*Журналируемая файловая система Ext4*", измените её на "*физический том для RAID*" затем выберите "*Настройка раздела закончена*".
6. Для раздела */* снова выберите "*СВОБОДНОЕ МЕСТО*" на первом носителе и нажмите "*Создать новый раздел*".
7. Используйте оставшееся свободное на носителе место и выберите *Далее*, а затем *Первичный*.
8. Так же, как и для раздела подкачки, выберите строку "*Использовать как:*" вверху и измените её значение на "*физический том для RAID*". Также поставьте отметку на строке "*Загрузочный флаг:*" "*on*". После этого выберите "*Настройка раздела закончена*".
9. Повторите шаги с третьего по восьмой для всех остальных дисков и разделов.

4.1.2. Настройка RAID

После разметки разделов массив готов к настройке:

1. Вернитесь на основную страницу "Разметка дисков", выберите "*Настройка программного RAID*" сверху.
2. Выберите "*да*" для записи изменений на диск.
3. Выберите "*Создать MD устройство*".
4. Для этого примера выберите "*RAID1*", но если вы используете другую конфигурацию, выберите соответствующий тип (RAID0 RAID1 RAID5).

Для использования *RAID5* нужно по крайней мере *три* диска. Использование RAID0 или RAID1 потребует лишь *двух* дисков.

5. Введите количество активных устройств равное "*2*", или же количество жёстких дисков, которые у вас выделены под массив. После этого нажмите "*Далее*".
6. Далее, введите число резервных устройств "*0*" по умолчанию, после чего нажмите "*Далее*".
7. Выберите используемые разделы. Как правило это будут sda1, sdb1, sdc1, и т.д. Цифры обычно совпадают, а разные буквы соответствуют разным жестким дискам.

   Для *раздела подкачки* выберите *sda1* и *sdb1*. Нажмите "*Далее*" для перехода к следующему шагу.

8. Повторите шаги с *третьего* по *седьмой* для раздела */*, выбрав *sda2* и *sdb2*.
9. По окончании выберите "*Завершить*".

4.1.3. Форматирование

Теперь должен появиться список жёстких дисков и RAID-устройств. Следующим шагом является форматирование и установка точек монтирования для RAID-устройств. Относитесь к RAID-устройствам как к локальным жёстким дискам, отформатируйте и выберите точки монтирования соответственно.

1. Выберите "*#1*" под разделом "*RAID1 устройство #0*".
2. Выберите "*Использовать как:*". Далее выберите "*раздел подкачки*", затем "*Настройка раздела выполнена*".
3. Следующим выберите "*#1*" под разделом "*RAID1 устройство #1*".
4. Выберите "*Использовать как:*". Далее выберите "*Журналируемая файловая система Ext4*".
5. Затем выберите "*Точка подключения*" и выберите "*/ — корневая файловая система*". Измените все необходимые опции и выберите "*Настройка раздела выполнена*".
6. Ну и наконец, выберите "*Завершить разметку и записать изменения на диск*".

Если вы разместили корневой раздел на RAID-массиве, установщик спросит, хотите ли вы загружать систему в состоянии *пониженной работоспособности*. Более подробную информацию читайте в разделе *Раздел 4.1.4, «Повреждённый RAID» [13]* .

Далее процесс установки продолжится как обычно.

4.1.4. Повреждённый RAID

В определенный момент работы компьютера вы можете столкнуться с отказом диска. Когда это случится, при использовании программного RAID, операционная система переведет массив в режим *пониженной работоспособности (degraded state)*.

Если массив повреждён, в связи с возможностью потери данных, по умолчанию Ubuntu Server Edition запустит *начальный загрузчик* через 30 секунд. Как только загрузчик стартует, появится предупреждение на 50 секунд с выбором либо продолжить и загрузить систему, либо сделать попытку восстановления вручную. Запуск загрузчика с предупреждением может быть как желательным, так и нет, особенно если это удалённый

компьютер. Загрузка с повреждённым массивом может быть настроена по-разному:

- Утилита dpkg-reconfigure может быть использована для настройки желательного варианта по умолчанию, и в процессе у вас будет возможность задать дополнительные настройки, связанные с массивом. Такие как слежение, почтовые предупреждения и пр. Для перенастройки mdadm введите следующее:

  ```
  sudo dpkg-reconfigure mdadm
  ```

- Команда **dpkg-reconfigure mdadm** изменит конфигурационный файл `/etc/initramfs-tools/conf.d/mdadm`. У этого файла есть возможность предварительной настройки желаемого поведения системы и он может быть отредактирован вручную:

  ```
  BOOT_DEGRADED=true
  ```

  

  Конфигурационный файл может быть проигнорирован при использовании параметров ядра

- Использование параметра ядра также позволит загрузиться системе с повреждённым массивом:
  - В процессе загрузки сервера нажмите **Shift** для входа в меню Grub.
  - Нажмите **e** для редактирования опций загрузки ядра.
  - Клавишей **стрелка вниз** подсветите строку ядра.
  - Добавьте "*bootdegraded=true*" (без кавычек) в конец строки.
  - Нажмите **Ctrl**+**x** для загрузки системы.

Как только система загружена, вы можете как восстановить массив (смотрите *Раздел 4.1.5, «Обслуживание RAID» [14]*), так и скопировать важные данные на другую машину в случае сильных повреждений устройства.

4.1.5. Обслуживание RAID

Утилита mdadm может быть использована для просмотра статуса массива, добавления дисков в массив, удаления дисков и пр.:

- Для просмотра статуса массива введите в терминале:

  ```
  sudo mdadm -D /dev/md0
  ```

  Опция *-D* указывает mdadm выводить *детальную* информацию об устройстве `/dev/md0`. Замените `/dev/md0` на соответствующее RAID устройство.

- Для просмотра статуса диска в массиве:

  ```
  sudo mdadm -E /dev/sda1
  ```

  Вывод очень похож на команду **mdadm -D**, относительно `/dev/sda1` для каждого диска.
- Если диск вышел из строя и должен быть удален:

  ```
  sudo mdadm --remove /dev/md0 /dev/sda1
  ```

  Замените `/dev/md0` и `/dev/sda1` на необходимые RAID устройство и диск.
- Аналогичным образом можно добавить диск:

  ```
  sudo mdadm --add /dev/md0 /dev/sda1
  ```

Иногда диск может перейти в *неработоспособное* состояние, даже когда нет никаких повреждений устройства. Обычно достаточно удалить устройство из массива и затем повторно его добавить. В этом случае диск повторно синхронизируется с массивом. Если диск не синхронизируется с массивом, это означает действительное повреждение устройства.

Файл `/proc/mdstat` также содержит полезную информацию по RAID устройствам в системе:

```
cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sda1[0] sdb1[1]
      10016384 blocks [2/2] [UU]

unused devices: <none>
```

Следующая команда лучше всего подходит для просмотра статуса синхронизации устройства:

```
watch -n1 cat /proc/mdstat
```

Нажмите *Ctrl+c* для окончания просмотра.

Если вам действительно потребовалось заменить повреждённый диск, после его замены и синхронизации будет необходимо установить grub. Для установки grub на новое устройство введите следующее:

```
sudo grub-install /dev/md0
```

Замените `/dev/md0` на имя соответствующего устройства.

4.1.6. Ресурсы

Тема массивов RAID обширна из-за изобилия вариантов настройки RAID. Пожалуйста посмотрите следующие ссылки для дополнительной информации:

- *Ubuntu Wiki Articles on RAID*[6].
- *Software RAID HOWTO*[7]
- *Managing RAID on Linux*[8]

4.2. Менеджер логических томов (LVM)

Менеджер логических томов, или *LVM*,позволяет администраторам создавать *логические* тома на одном и нескольких жёстких дисках. LVM тома могут быть созданы как на разделах программного RAID, так и на стандартных разделах единичного диска. Тома также могут расширяться, предоставляя большую гибкость системам по изменению предоставляемых ресурсов.

4.2.1. Обзор

Побочным эффектом от мощи и гибкости LVM является *большая* степень сложности. Перед тем, как погружаться в установочный процесс LVM, было бы неплохо ознакомиться с некоторыми терминами.

- *Физический том (Physical Volume — PV):* физический жёсткий диск, раздел диска или раздел программного RAID, отформатированный как LVM PV.
- *Группа томов (Volume Group — VG):* строится из одного или нескольких физических томов. VG могут быть расширены добавлением PV. VG похожи на виртуальные дисковые устройства, которые можно разделять на логические тома.
- *Логический том (Logical Volume — LV):* аналогичен разделу на диске без использования LVM. LV, отформатированный в желаемую файловую систему (EXT3, XFS, JFS и др.) доступен для монтирования и хранения данных.

4.2.2. Установка

Пример в данной секции показывает установку Ubuntu Server Edition с монтированием `/srv` на том LVM. В процессе начальной установки только один физический том (PV) может стать частью группы томов (VG). Другой

[6] https://help.ubuntu.com/community/Installation#raid
[7] http://www.faqs.org/docs/Linux-HOWTO/Software-RAID-HOWTO.html
[8] http://oreilly.com/catalog/9781565927308/

PV будет добавлен после установки для демонстрации того, как VG может быть расширен.

Существует несколько вариантов установки LVM, "*Управляемый — использовать весь диск и настроить LVM*" который также позволит вам выделить часть доступного пространства под LVM, "*Управляемый — использовать целиком и настроить шифрованный LVM*", или установить и настроить LVM *вручную*. На данный момент единственный вариант настроить систему с использованием как LVM, так и стандартных разделов в процессе установки — это использование ручной настройки.

1. Следуйте инструкциям по установке, пока вы не достигнете этапа разметки дисков, а затем:
2. На экране "*Дисковые разделы*" выберите "*Вручную*".
3. Выделите жёсткий диск и на следующем экране выберите "да" в ответ на предложение "*Создать новую таблицу разделов устройства*".
4. Далее создайте стандартные разделы для */boot*, *swap*, и */* с той файловой системой, которую вы предпочитаете.
5. Для размещения */srv* на LVM, создайте новый *логический* раздел. Затем замените "*Использовать как*" на "*физический том LVM*", после чего нажмите "*Настройка разделов завершена*".
6. Теперь выделите "*Настроить менеджер логических томов*" вверху и выберите "*Да*" для сохранения изменений на диск.
7. В качестве "*Действия по настройке LVM*"на следующем экране выберите "*Создать группу томов*". Введите имя для VG типа *vg01*, или что-то более наглядное. После ввода имени выберите раздел, выделенный под LVM, и нажмите "*Continue*".
8. Вернитесь на экран "*Действие по настройке LVM*" выберите "*Создать логический том*". Выделите недавно созданную группу томов и введите имя для LV, например, *srv*, раз уж она предназначена для этой точки монтирования. Затем выберите размер, который может быть на весь раздел, поскольку его всегда можно будет расширить позднее. Нажмите "*Finish*" и вы вернётесь обратно на основную страницу "*Дисковые разделы*".
9. Теперь добавьте файловую систему для нового LVM. Выделите раздел под названием "*LVM VG vg01, LV srv*" или тем именем, которое вы выбрали и нажмите *Использовать как*. Настройте файловую систему как обычно, выбрав */srv*в качестве точки монтирования. По окончании нажмите "*Выполнить настройку разделов*".
10. Наконец нажмите "*Завершить разбиение и сохранить изменения на диск*". Затем подтвердите изменения и продолжите обычную установку.

Есть несколько полезных утилит для просмотра информации по LVM:

- *pvdisplay:* показывает информацию по физическим томам.
- *vgdisplay:* показывает информацию по группам томов.
- *lvdisplay:* показывает информацию по логическим томам.

4.2.3. Расширение групп томов

Продолжая с *srv*, как примером тома LVM, в данной секции рассматривается добавление второго жёсткого диска, создание физического тома (PV), добавление его в группу томов (VG), расширение логического тома `srv` и в конце расширение файловой системы. Этот пример подразумевает, что в систему был добавлен второй жёсткий диск. В данном примере этот жёсткий диск получит имя `/dev/sdb`, и мы будем использовать весь диск под физический том (вы можете выбрать создание разделов и использовать их как другие физические тома).

Убедитесь, что у вас ещё нет `/dev/sdb` перед тем, как выполнять приведенные ниже команды. Вы можете потерять некоторые данные, если выполните эти команды на использующемся диске.

1. Сначала создадим физический том, выполнив в терминале:

   ```
   sudo pvcreate /dev/sdb
   ```

2. Теперь расширим группу томов (VG):

   ```
   sudo vgextend vg01 /dev/sdb
   ```

3. Используйте vgdisplay для поиска свободных физических экстентов (PE) — Free PE / size (размер, который вы можете занять). Предположим, что свободно 511 PE (эквивалентно 2 ГБ при размере PE в 4 МБ) и мы используем всё доступное свободное место. Используйте ваши собственные PE и/или свободное место.

   Логический том (LV) теперь может быть увеличен различными методами, мы будем рассматривать только как использовать PE для расширения LV:

   ```
   sudo lvextend /dev/vg01/srv -l +511
   ```

   Опция *-l* позволяет расширять LV, используя PE. Опция *-L* позволит задавать увеличение LV в МБ, ГБ, ТБ и т.п.

4. Даже если вы считаете, что можете *увеличить* файловую систему ext3 или ext4 без предварительного отключения, будет хорошей практикой

в любом случае отмонтировать её и проверить на целостность, что позволит избежать суматошного дня по уменьшению логического тома (в этом случае ее придется отключить обязательно).

Следующая команда только для файловой системы *EXT3* или *EXT4*. Если вы используете другую файловую систему, возможно, придётся использовать другие утилиты.

```
sudo umount /srv
sudo e2fsck -f /dev/vg01/srv
```

Опция *-f* для e2fsck заставляет принудительно сделать проверку на целостность системы.

5. Наконец, изменяем размер файловой системы:

```
sudo resize2fs /dev/vg01/srv
```

6. Теперь монтируем раздел и проверяем его размер:

```
mount /dev/vg01/srv /srv && df -h /srv
```

4.2.4. Ресурсы

- Смотрите статьи *Ubuntu Wiki LVM*[9].
- Смотрите *LVM HOWTO*[10] для дополнительной информации.
- Ещё одна хорошая статья — *Managing Disk Space with LVM*[11] на сайте O'Reilly's linuxdevcenter.com.
- For more information on fdisk see the *fdisk man page*[12].

[9] https://help.ubuntu.com/community/Installation#lvm
[10] http://tldp.org/HOWTO/LVM-HOWTO/index.html
[11] http://www.linuxdevcenter.com/pub/a/linux/2006/04/27/managing-disk-space-with-lvm.html
[12] http://manpages.ubuntu.com/manpages/trusty/en/man8/fdisk.8.html

5. Отчёт о падении ядра

5.1. Введение

Отчёт о падении ядра (Kernel Crash Dump) является частью содержимого оперативной памяти (RAM), которая копируется на диск всякий раз, когда выполнение ядра прерывается. Следующие события могут стать причиной остановки ядра:

- Критическая ошибка ядра (Kernel Panic)
- Незамаскированное прерывание (NMI)
- Фатальная ошибка проверки машины (MCE)
- Ошибка аппаратного обеспечения
- Ручное вмешательство

Некоторые из этих событий (Kernel Panic, NMI) ядро обрабатывает автоматически и запускает механизм сохранения отчета через *kexec*. В других случаях требуется ручное вмешательство для захвата памяти. Всякий раз, как что-то из перечисленного происходит, важно найти основную причину с целью предотвращения такого события снова. Причина может быть определена с помощью инспектирования содержимого памяти.

5.2. Механизм отчёта о падении ядра

Когда происходит критическая ошибка ядра, ядро полагается на механизм *kexec* для быстрой перезагрузки новой копии ядра в предварительно зарезервированную секцию памяти, которая выделяется при загрузке системы. Это позволяет существующей памяти остаться нетронутой с целью безопасного копирования её содержимого в файл.

5.3. Установка

Утилита сохранения отчёта о падении ядра устанавливается следующей командой:

```
sudo apt-get install linux-crashdump
```

После этого потребуется перезагрузка.

5.4. Конфигурация

Никакой дальнейшей настройки не требуется, чтобы разрешить механизм отчёта о падении ядра.

5.5. Проверка

Для подтверждения, что механизм отчёта о падении ядра доступен, надо проверить несколько вещей. Во-первых, убедитесь, что указан загрузочный параметр *crashkernel* (заметьте, что строка внизу разделена на две, чтобы уместить её в формат документа):

```
cat /proc/cmdline

BOOT_IMAGE=/vmlinuz-3.2.0-17-server root=/dev/mapper/PreciseS-root ro
 crashkernel=384M-2G:64M,2G-:128M
```

Параметр *crashkernel* имеет следующий синтаксис:

```
crashkernel=<range1>:<size1>[,<range2>:<size2>,...][@offset]
    range=start-[end] 'start' is inclusive and 'end' is exclusive.
```

Таким образом с параметром crashkernel, найденным в файле `/proc/cmdline` мы имеем:

```
crashkernel=384M-2G:64M,2G-:128M
```

Значение выше означает:

- если оперативная память меньше 384МБ, то ничего не резервировать (это вариант «спасения»)
- если оперативная память между 384МБ и 2ГБ (привилегированная), то зарезервировать 64МБ
- если оперативная память больше 2ГБ, то зарезервировать 128МБ

Во-вторых, убедитесь, что ядро зарезервировало требуемую память для kdump ядра, выполнив:

```
dmesg | grep -i crash

...
[    0.000000] Reserving 64MB of memory at 800MB for crashkernel (System RAM: 1023MB)
```

5.6. Проверка механизма отчёта о падении ядра

Проверка механизма отчёта о падении ядра вызовет *перезагрузку системы*. В определенных ситуациях это может привести к потере

данных, если система будет сильно загружена. Если вы хотите проверить этот механизм, убедитесь, что система простаивает или загружена очень слабо.

Убедитесь, что механизм *SysRQ* включен, посмотрев значение параметра ядра `/proc/sys/kernel/sysrq`:

```
cat /proc/sys/kernel/sysrq
```

Если возвращается значение *0*, свойство отключено. Включите его следующей командой:

```
sudo sysctl -w kernel.sysrq=1
```

Как только закончите с этим, вам придётся стать суперпользователем (root), поскольку будет недостаточно использовать только **sudo**. От имени пользователя *root* вам нужно выполнить команду **echo c > /proc/sysrq-trigger**. Если вы используете сетевое соединение, вы потеряете связь с системой. Именно поэтому лучше проводить тест с системной консоли. Это позволит сделать процесс отчёта о падении ядра видимым.

Типичный вывод теста будет выглядеть следующим образом:

```
sudo -s
[sudo] password for ubuntu:
# echo c > /proc/sysrq-trigger
[   31.659002] SysRq : Trigger a crash
[   31.659749] BUG: unable to handle kernel NULL pointer dereference at           (null)
[   31.662668] IP: [<ffffffff8139f166>] sysrq_handle_crash+0x16/0x20
[   31.662668] PGD 3bfb9067 PUD 368a7067 PMD 0
[   31.662668] Oops: 0002 [#1] SMP
[   31.662668] CPU 1
....
```

Дальнейший вывод отрезан, но вы можете посмотреть перезагрузку системы и где-нибудь в журнале вы сможете найти следующую строчку:

```
Begin: Saving vmcore from kernel crash ...
```

После завершения система перезагрузится в нормальный рабочий режим. После чего вы сможете найти файл отчёта о падении ядра в каталоге `/var/crash`:

```
ls /var/crash
linux-image-3.0.0-12-server.0.crash
```

5.7. Ресурсы

Отчёт о падении ядра — обширная тема, требующая хорошего знания ядра Linux. Вы сможете найти больше информации по следующим ссылкам:

- *Документация по kdump*[13].
- *Утилита crash*[14]
- *Анализ падений ядра Linux*[15] (Основано на дистрибутиве Fedora, однако предоставляет хороший критический анализ исследований отчетов падения ядра)

[13] http://www.kernel.org/doc/Documentation/kdump/kdump.txt
[14] http://people.redhat.com/~anderson/
[15] http://www.dedoimedo.com/computers/crash-analyze.html

Глава 3. Управление пакетами

Ubuntu содержит всеобъемлющую систему управления пакетами для установки, обновления, настройки и удаления программ. Помимо обеспечения доступа к упорядоченной базе из более 35000 пакетов программного обеспечения для Ubuntu, система управления пакетами также содержит средства разрешения зависимостей и проверки наличия обновлений программ.

Есть несколько программ для работы с системой управления пакетами Ubuntu, начиная с простых консольных утилит, использование которых может быть легко автоматизировано администраторами системы, до простых в использовании программ с графическим интерфейсом, более подходящих для новичков в Ubuntu.

1. Введение

Система управления пакетами создана на основе подобной системы, используемой в дистрибутиве Debian GNU/Linux. Файлы пакетов содержат все необходимые файлы, метаданные и инструкции, чтобы обеспечить использование специфических функций или программного приложения на вашем компьютере под управлением Ubuntu.

Файлы пакетов Debian обычно имеют расширение '.deb' и, как правило, содержатся в *репозиториях* — коллекциях пакетов, которые можно найти на различных носителях, таких как диски CD-ROM, или в Интернете. Пакеты обычно заранее скомпилированы в двоичный формат, поэтому устанавливаются быстро и не требуют выполнения компиляции.

Многие сложные пакеты используют концепцию *зависимостей*. Зависимости — это дополнительные пакеты, необходимые основному пакету для нормальной работы. Например, пакет синтезатора речи festival зависит от пакета libasound2, предоставляющего звуковую библиотеку ALSA, необходимую для воспроизведения звука. Чтобы festival заработал, необходимо установить его и все его зависимости. Средства управления программами в Ubuntu делают это автоматически.

2. dpkg

dpkg — это менеджер пакетов для систем, основанных на *Debian*. Он может устанавливать, удалять и собирать пакеты, но в отличие от других систем управления пакетами, не может автоматически скачивать и устанавливать пакеты или их зависимости. В этом разделе рассматривается применение dpkg для управления локально установленными пакетами:

- Чтобы увидеть список всех установленных в системе пакетов, наберите в приглашении терминала:

  ```
  dpkg -l
  ```

- В зависимости от количества пакетов в вашей системе, вывод данных может быть очень большим. Перенаправьте вывод через grep, чтобы узнать, установлен ли определённый пакет:

  ```
  dpkg -l | grep apache2
  ```

  Замените *apache2* на любое имя пакета, часть имени пакета или любое допустимое выражение.

- Для получения списка файлов, установленных пакетом, в данном случае ufw, введите:

  ```
  dpkg -L ufw
  ```

- Если вы не уверены, какой пакет установил файл, dpkg -S поможет вам узнать это. Например:

  ```
  dpkg -S /etc/host.conf
  base-files: /etc/host.conf
  ```

  Вывод покажет вам, что `/etc/host.conf` принадлежит пакету base-files.

  

  Многие файлы генерируются автоматически в процессе установки пакета и, хотя они находятся в файловой системе, **dpkg -S** может не знать, какому пакету они принадлежат.

- Вы можете установить локальный *.deb*-файл, введя:

  ```
  sudo dpkg -i zip_3.0-4_i386.deb
  ```

  Замените `zip_3.0-4_i386.deb` на реальное имя локального .deb-файла, который вы хотите установить.

- Удаление пакета может быть выполнено так:

```
sudo dpkg -r zip
```

Устанавливать пакеты с помощью dpkg в большинстве случаев *НЕ РЕКОМЕНДУЕТСЯ*. Лучше воспользоваться менеджером пакетов, который обрабатывает зависимости, чтобы исключить возможность возникновения несогласованностей в системе. Например, **dpkg -r zip** удалит пакет zip, но все пакеты, от которых он зависит, останутся установленными и могут в дальнейшем функционировать неправильно.

О других опциях dpkg смотрите руководство: **man dpkg**.

3. Apt-Get

Команда apt-get — это мощный инструмент командной строки, который работает с Ubuntu *Advanced Packaging Tool* (APT), выполняя такие действия, как установка новых пакетов программ, обновление существующих пакетов, обновление индекса списка пакетов и даже полное обновление всей системы Ubuntu.

Являясь простым инструментом командной строки, apt-get имеет множество преимуществ над другими инструментами управления пакетами в Ubuntu для администраторов серверов. Некоторыми из этих преимуществ являются простота работы через SSH и возможность использования в сценариях администрирования системы, которые, в свою очередь, можно автоматизировать с помощью утилиты планировщика cron.

Несколько примеров использования apt-get:

- **Установка пакета**: устанавливать пакеты с помощью apt-get очень просто. Например, чтобы установить сетевой сканер nmap, наберите:

  ```
  sudo apt-get install nmap
  ```

- **Удаление пакета**: удалить пакет (или несколько пакетов) тоже просто. Чтобы удалить пакет, установленный в предыдущем примере, наберите:

  ```
  sudo apt-get remove nmap
  ```

  

  Несколько пакетов: Вы можете указать сразу несколько пакетов для установки или удаления, разделив их названия пробелами.

  При добавлении опции *--purge* к **apt-get remove** будут также удалены конфигурационные файлы пакета. Это может быть, а может и не быть нужным вам эффектом, так что используйте эту опцию с осторожностью.

- **Обновление индекса пакетов**: индекс пакетов APT — это база данных пакетов, доступных из репозиториев, указанных в файле `/etc/apt/sources.list` и каталоге `/etc/apt/sources.list.d`. Чтобы обновить локальный индекс пакетов, согласовав его с последними изменениями в репозиториях, наберите следующее:

  ```
  sudo apt-get update
  ```

- **Обновление пакетов**: через какое-то время в репозиториях могут стать доступными обновлённые версии установленных на вашем компьютере пакетов (например, обновления, связанные с безопасностью). Чтобы

обновить систему, в первую очередь обновите индекс пакетов, как показано выше, и после этого наберите:

```
sudo apt-get upgrade
```

Информацию по переходу на новый релиз Ubuntu смотрите в *Раздел 3, «Обновление» [10]*.

Действия команды apt-get, такие как установка и удаление пакетов, сохраняются в файле журнала /var/log/dpkg.log

Для дополнительной информации об использовании APT прочтите подробное *Руководство пользователя Debian APT*[1] или наберите в терминале:

```
apt-get help
```

[1] http://www.debian.org/doc/user-manuals#apt-howto

4. Aptitude

Запуск Aptitude без параметров предоставит вам текстовый интерфейс с меню для доступа к системе *Advanced Packaging Tool* (APT). Множество общих функций управления пакетами, такие как установка, удаление и обновление, могут быть выполнены в Aptitude однобуквенной командой, набранной обычно в нижнем регистре.

Aptitude лучше всего подходит для терминального окружения без графики, чтобы убедиться в правильном функционировании ключевых команд. Вы можете запустить интерфейс Aptitude с меню под обычным пользователем, введя следующую команду в терминале:

```
sudo aptitude
```

Когда Aptitude запускается, вы можете видеть строку меню вверху экрана и две панели под ней. Верхняя панель содержит категории пакетов, такие как *Новые пакеты* и *Неустановленные пакеты*. Нижняя панель содержит информацию, касающуюся пакетов и категорий пакетов.

Использование Aptitude для управления пакетами является достаточно прямолинейным, а пользовательский интерфейс делает повседневные задачи простыми в выполнении. Вот примеры стандартных функций по управлению пакетами в том виде, в каком они выполняются в Aptitude:

- **Install Packages**: To install a package, locate the package via the *Not Installed Packages* package category, by using the keyboard arrow keys and the **ENTER** key. Highlight the desired package, then press the **+** key. The package entry should turn *green*, indicating that it has been marked for installation. Now press **g** to be presented with a summary of package actions. Press **g** again, and downloading and installation of the package will commence. When finished, press **ENTER**, to return to the menu.
- **Remove Packages**: To remove a package, locate the package via the *Installed Packages* package category, by using the keyboard arrow keys and the **ENTER** key. Highlight the desired package you wish to remove, then press the **-** key. The package entry should turn *pink*, indicating it has been marked for removal. Now press **g** to be presented with a summary of package actions. Press **g** again, and removal of the package will commence. When finished, press **ENTER**, to return to the menu.
- **Update Package Index**: To update the package index, simply press the **u** key. Updating of the package index will commence.
- **Upgrade Packages**: To upgrade packages, perform the update of the package index as detailed above, and then press the **U** key to mark all

packages with updates. Now press **g** whereby you'll be presented with a summary of package actions. Press **g** again, and the download and installation will commence. When finished, press **ENTER**, to return to the menu.

Первая колонка, отображаемая в списке пакетов в верхней панели, при непосредственном просмотре пакетов, отображает текущее состояние пакета и использует следующие символы для индикации этого состояния:

- **i**: Установленный пакет
- **c**: Пакет не установлен, но настройки пакета остались в системе
- **p**: Удалён из системы
- **v**: Виртуальный пакет
- **B**: Испорченный пакет
- **u**: Файлы распакованы, однако настройка пакета не закончена
- **C**: Частично настроенные — настройка завершилась неудачей и требует исправления
- **H**: Частично установленные — удаление завершилось неудачей и требует исправления

Чтобы выйти из Aptitude, просто нажмите клавишу **q** и подтвердите свой выход. Многие другие функции доступны из меню Aptitude, которое доступно по нажатию клавиши **F10**.

4.1. Aptitude в командной строке

Aptitude можно использовать и как инструмент командной строки, подобно apt-get. Чтобы установить пакет nmap со всеми зависимостями, как в примере apt-get, нужно ввести следующую команду:

```
sudo aptitude install nmap
```

Чтобы удалить тот же пакет, нужно использовать команду:

```
sudo aptitude remove nmap
```

Смотрите в man-страницах дополнительные сведения об опциях командной строки Aptitude.

5. Автоматические обновления

Пакет unattended-upgrades может использоваться для автоматической установки обновлённых пакетов и может быть сконфигурирован на обновление всех пакетов или установку только изменений для безопасности системы. Во-первых, установите пакет, введя в командной строке:

```
sudo apt-get install unattended-upgrades
```

Чтобы настроить unattended-upgrades, отредактируйте `/etc/apt/apt.conf.d/50unattended-upgrades` изменив следующие строки так, как вам это необходимо:

```
Unattended-Upgrade::Allowed-Origins {
        "Ubuntu trusty-security";
//      "Ubuntu trusty-updates";
};
```

Некоторые пакеты могут быть указаны *в* "*чёрном*" *списке* что означает, что они не будут обновляться. Чтобы поместить пакет в "чёрный" список:

```
Unattended-Upgrade::Package-Blacklist {
//      "vim";
//      "libc6";
//      "libc6-dev";
//      "libc6-i686";
};
```

Двойная косая черта «//» означает комментарий, то есть всё, что находится после "//", не будет обрабатываться.

Чтобы включить автоматические обновления, отредактируйте `/etc/apt/apt.conf.d/10periodic` и задайте соответствующие конфигурационные параметры apt:

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "1";
APT::Periodic::AutocleanInterval "7";
APT::Periodic::Unattended-Upgrade "1";
```

Приведённая выше конфигурация обновляет список пакетов, скачивает и устанавливает доступные обновления каждый день. Локальный архив загрузок очищается каждую неделю.

Вы можете почитать больше о настройках периодичности apt в заголовке сценария /`/etc/cron.daily/apt`.

Результаты unattended-upgrades будут занесены в `/var/log/unattended-upgrades`.

5.1. Уведомления

Настройка *Unattended-Upgrade::Mail* в `/etc/apt/apt.conf.d/50unattended-upgrades` позволит unattended-upgrades сообщать администратору по электронной почте подробности о любых пакетах, нуждающихся в обновлении или имеющих проблемы.

Ещё один полезный пакет — apticron. apticron настраивает задание cron для отправки электронны писем администратору с информацией о любых пакетах в системе, для которых доступны обновления, а также отчётом об изменениях в каждом пакете.

Для установки пакета apticron в терминале введите:

```
sudo apt-get install apticron
```

После установки пакета отредактируйте `/etc/apticron/apticron.conf`, чтобы установить адрес электронной почты и другие опции:

```
EMAIL="root@example.com"
```

6. Конфигурация

Настройка системных репозиториев *Advanced Packaging Tool* (APT) сохраняется в файле `/etc/apt/sources.list` и каталоге `/etc/apt/sources.list.d`. Пример такого файла приведен здесь вместе с информацией по добавлению или удалению ссылок на репозитории в этом файле.

Вы можете изменять файл для подключения и отключения репозиториев. Например, для отключения требования вставить диск Ubuntu во время выполнения операций с пакетами просто закомментируйте строки с соответствующим диском, которые находятся в начале файла:

```
# no more prompting for CD-ROM please
# deb cdrom:[Ubuntu 14.04 _Trusty Tahr_ - Release i386 (20111013.1)]/ trusty main restricted
```

6.1. Дополнительные репозитории

В дополнение к официально поддерживаемым репозиториям пакетов для Ubuntu существуют дополнительные, поддерживаемые сообществом, репозитории, которые обеспечивают возможность установки ещё нескольких тысяч пакетов. Два наиболее популярных — это репозитории *Universe* и *Multiverse*. Эти репозитории официально не поддерживаются Ubuntu, но, поскольку они поддерживаются сообществом, обычно можно без опасений использовать входящие в них пакеты на компьютере с Ubuntu.

Пакеты в репозитории *Multiverse* часто имеют нюансы с лицензиями, которые не позволяют им распространяться вместе со свободной операционной системой, и их использование может быть незаконным там, где вы находитесь.

Обращаем ваше внимание, что указанные репозитории, *Universe* и *Multiverse*, не содержат официально поддерживаемых пакетов. В частности, может не существовать необходимых обновлений безопасности для этих пакетов.

Доступно большое количество других источников пакетов, иногда предоставляющих доступ лишь к одному пакету (например, в случае пакета с исходными кодами, предоставляемого разработчиком отдельного приложения). Вы должны быть очень осторожны и внимательны при использовании нестандартных источников пакетов. Внимательно изучите как источник, так и пакет перед установкой, так как некоторые источники, и пакеты, предоставляемые ими, могут вызвать нестабильную работу вашей системы или даже полную её неработоспособность.

По умолчанию, репозитории *Universe* и *Multiverse* доступны, но если вы хотите отключить их, измените `/etc/apt/sources.list` и закомментируйте следующие строки:

```
deb http://archive.ubuntu.com/ubuntu trusty universe multiverse
deb-src http://archive.ubuntu.com/ubuntu trusty universe multiverse

deb http://us.archive.ubuntu.com/ubuntu/ trusty universe
deb-src http://us.archive.ubuntu.com/ubuntu/ trusty universe
deb http://us.archive.ubuntu.com/ubuntu/ trusty-updates universe
deb-src http://us.archive.ubuntu.com/ubuntu/ trusty-updates universe

deb http://us.archive.ubuntu.com/ubuntu/ trusty multiverse
deb-src http://us.archive.ubuntu.com/ubuntu/ trusty multiverse
deb http://us.archive.ubuntu.com/ubuntu/ trusty-updates multiverse
deb-src http://us.archive.ubuntu.com/ubuntu/ trusty-updates multiverse

deb http://security.ubuntu.com/ubuntu trusty-security universe
deb-src http://security.ubuntu.com/ubuntu trusty-security universe
deb http://security.ubuntu.com/ubuntu trusty-security multiverse
deb-src http://security.ubuntu.com/ubuntu trusty-security multiverse
```

7. Ссылки

Большинство материалов, имеющихся в этой главе, также доступны в помощи man, многие из которых также доступны в интернете.

- *Wiki-страница установки Ubuntu*[2] содержит дополнительную информацию.
- For more dpkg details see the *dpkg man page*[3].
- The *APT HOWTO*[4] and *apt-get man page*[5] contain useful information regarding apt-get usage.
- See the *aptitude man page*[6] for more aptitude options.
- *HOWTO по добавлению репозиториев (Ubuntu Wiki)*[7] содержат дополнительную информацию по добавлению репозиториев.

[2] https://help.ubuntu.com/community/InstallingSoftware
[3] http://manpages.ubuntu.com/manpages/trusty/en/man1/dpkg.1.html
[4] http://www.debian.org/doc/manuals/apt-howto/
[5] http://manpages.ubuntu.com/manpages/trusty/en/man8/apt-get.8.html
[6] http://manpages.ubuntu.com/manpages/trusty/man8/aptitude.8.html
[7] https://help.ubuntu.com/community/Repositories/Ubuntu

Глава 4. Работа в сети

Сети состоят из двух и более устройств, таких, как компьютерные системы, принтеры и сопутствующее оборудование, которые соединены при помощи физического кабеля или беспроводными каналами с той целью, чтобы делать общей информацию и распространять её между соединёнными устройствами.

Этот раздел содержит общую и специфическую информацию относительно сетевого взаимодействия, включая обзор концепций сети и детальное обсуждение популярных сетевых протоколов.

1. Настройка сети

Ubuntu поставляется с несколькими графическими инструментами для настройки сетевых устройств. Этот документ рассчитан на продвинутых пользователей и фокусируется на управлении сетью с помощью командной строки.

1.1. Интерфейсы Ethernet

Интерфейсы Ethernet обозначаются в системе как *ethX*, где *X* является числом. Первый интерфейс Ethernet обычно обозначается *eth0*, второй *eth1*, и так далее в порядке возрастания чисел.

1.1.1. Определение Ethernet интерфейсов

Для быстрого определения всех доступных сетевых интерфейсов вы можете использовать команду ifconfig, как показано ниже.

```
ifconfig -a | grep eth
eth0      Link encap:Ethernet  HWaddr 00:15:c5:4a:16:5a
```

Другое приложение, которое может помочь идентифицировать все доступные вашей системе сетевые интерфейсы — это команда lshw . В приведённом ниже примере lshw показывает один Ethernet интерфейс с логическим именем *eth0* вместе с информацией по шине, сведениями о драйвере и всеми поддерживаемыми возможностями.

```
sudo lshw -class network
  *-network
       description: Ethernet interface
       product: BCM4401-B0 100Base-TX
       vendor: Broadcom Corporation
       physical id: 0
       bus info: pci@0000:03:00.0
       logical name: eth0
       version: 02
       serial: 00:15:c5:4a:16:5a
       size: 10MB/s
       capacity: 100MB/s
       width: 32 bits
       clock: 33MHz
       capabilities: (snipped for brevity)
       configuration: (snipped for brevity)
       resources: irq:17 memory:ef9fe000-ef9fffff
```

1.1.2. Логические имена интерфейсов Ethernet

Логические имена интерфейсов настраиваются в файле `/etc/udev/rules.d/70-persistent-net.rules`. Если вы захотите определить, какой интерфейс получит определённое логическое имя, найдите строку, соответствующую физическому MAC-адресу интерфейса, и измените значение *NAME=ethX* на желаемое логическое имя. Перегрузите систему для применения изменений.

1.1.3. Настройки интерфейса Ethernet

ethtool — это программа, которая показывает и изменяет настройки сетевых карт, такие как автосогласование (auto-negotiation), скорость порта, режим дуплекса и функция Wake-on-LAN (пробуждение системы через сеть). Эта программа не устанавливается по умолчанию, но доступна к установке из репозиториев.

```
sudo apt-get install ethtool
```

Ниже приведён пример того, как посмотреть возможности карты и настроить параметры интерфейса Ethernet.

```
sudo ethtool eth0
Settings for eth0:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Half 1000baseT/Full
        Supports auto-negotiation: Yes
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Half 1000baseT/Full
        Advertised auto-negotiation: Yes
        Speed: 1000Mb/s
        Duplex: Full
        Port: Twisted Pair
        PHYAD: 1
        Transceiver: internal
        Auto-negotiation: on
        Supports Wake-on: g
        Wake-on: d
        Current message level: 0x000000ff (255)
        Link detected: yes
```

Изменения, сделанные с использованием команды ethtool, временные и будут утеряны после перезагрузки. Если вы хотите сохранить настройки, просто добавьте требуемую команду ethtool в строку *pre-up* в файле `/etc/network/interfaces`.

Ниже приведен пример того, как интерфейс, определённый как *eth0*, может быть постоянно настроен на скорость порта 1000 Мб/с в режиме полного дуплекса.

```
auto eth0
iface eth0 inet static
pre-up /sbin/ethtool -s eth0 speed 1000 duplex full
```

Несмотря на то, что пример выше показывает интерфейс, настроенный *статично*, это работает и с другими методами, такими как DHCP. Этот пример призван просто продемонстрировать правильное размещение строки *pre-up* по отношению к остальной части конфигурационного файла интерфейса.

1.2. Адресация IP

Следующий раздел описывает процесс настройки IP-адреса вашего компьютера и шлюза по умолчанию, необходимых для подключения к локальной сети и интернету.

1.2.1. Временное назначение IP-адреса

Для временной настройки сети вы можете использовать стандартные команды, такие как ip, ifconfig и route, которые присутствуют также и в других системах на базе GNU/Linux. Эти команды позволят изменить настройки, которые будут применены мгновенно, но они не будут постоянными и будут утеряны после перезагрузки.

Для временной настройки IP-адреса вы можете использовать команду ifconfig следующим образом. Только замените IP-адрес и маску подсети на соответствующие требованиям вашей сети.

```
sudo ifconfig eth0 10.0.0.100 netmask 255.255.255.0
```

Для проверки настройки IP-адреса eth0 вы можете использовать команду ifconfig таким образом:

```
ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:15:c5:4a:16:5a
          inet addr:10.0.0.100  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::215:c5ff:fe4a:165a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:466475604 errors:0 dropped:0 overruns:0 frame:0
          TX packets:403172654 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2574778386 (2.5 GB)  TX bytes:1618367329 (1.6 GB)
```

```
          Interrupt:16
```

Для настройки шлюза по умолчанию вы можете использовать команду route следующим образом. Измените адрес шлюза по умолчанию на требуемый для вашей сети.

```
sudo route add default gw 10.0.0.1 eth0
```

Для проверки настройки шлюза по умолчанию используйте команду route таким образом:

```
route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.0.0.0        0.0.0.0         255.255.255.0   U     1      0        0 eth0
0.0.0.0         10.0.0.1        0.0.0.0         UG    0      0        0 eth0
```

If you require DNS for your temporary network configuration, you can add DNS server IP addresses in the file `/etc/resolv.conf`. In general, editing `/etc/resolv.conf` directly is not recommanded, but this is a temporary and non-persistent configuration. The example below shows how to enter two DNS servers to `/etc/resolv.conf`, which should be changed to servers appropriate for your network. A more lengthy description of the proper persistent way to do DNS client configuration is in a following section.

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

Если вам больше не требуется эта конфигурация и вы хотите отменить все IP настройки интерфейса, вы можете использовать команду ip с опцией flush как показано ниже:

```
ip addr flush eth0
```

Flushing the IP configuration using the ip command does not clear the contents of `/etc/resolv.conf`. You must remove or modify those entries manually, or re-boot which should also cause `/etc/resolv.conf`, which is actually now a symlink to `/run/resolvconf/resolv.conf`, to be re-written.

1.2.2. Динамическое назначение IP-адреса (клиент DHCP)

Чтобы настроить ваш сервера на использование DHCP для динамического присвоения адреса, добавьте *dhcp* метод в адресную секцию inet для соответствующего интерфейса в файле `/etc/network/interfaces`. Пример ниже предполагает, что вы настраиваете ваш первый интерфейс Ethernet, обозначенный как *eth0*.

```
auto eth0
iface eth0 inet dhcp
```

Добавив настройку интерфейса как показано выше, вы можете вручную включить интерфейс командой ifup, которая активизирует процесс DHCP через dhclient.

```
sudo ifup eth0
```

Для отключения интерфейса вручную вы можете воспользоваться командой ifdown, которая запустит процесс освобождения DHCP и остановки интерфейса.

```
sudo ifdown eth0
```

1.2.3. Статическое назначение IP-адреса

Для настройки вашей системы под использование статического присвоения IP-адреса добавьте метод *static* в секцию inet для соответствующего интерфейса в файле `/etc/network/interfaces`. Пример ниже предполагает, что вы настраиваете ваш первый интерфейс Ethernet, обозначенный как *eth0*. Измените значения *адреса*, *маски сети*, и *шлюза* для соответствия требованиям вашей сети.

```
auto eth0
iface eth0 inet static
address 10.0.0.100
netmask 255.255.255.0
gateway 10.0.0.1
```

Добавив настройку интерфейса как показано выше, вы можете вручную включить интерфейс командой ifup.

```
sudo ifup eth0
```

Для отключения интерфейса вручную вы можете воспользоваться командой ifdown.

```
sudo ifdown eth0
```

1.2.4. Интерфейс Loopback (обратной петли)

Интерфейс loopback определяется системой как *lo* и по умолчанию задает адрес 127.0.0.1. Он может быть выведен командой ifconfig.

```
ifconfig lo
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2718 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2718 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:183308 (183.3 KB)  TX bytes:183308 (183.3 KB)
```

По умолчанию в `/etc/network/interfaces` должны присутствовать две строки, отвечающих за автоматическую настройку интерфейса loopback. Рекомендуется оставить эти настройки по умолчанию, пока не возникнет специфической причины для их изменения. Пример этих двух строк приведён ниже.

```
auto lo
iface lo inet loopback
```

1.3. Разрешение имён

Под разрешением имени по отношению к IP-сетям подразумевается процесс определения IP-адреса по имени хоста, упрощающий идентификацию ресурса в сети. В следующем разделе показано, как правильно настроить вашу систему для разрешения имён с помощью DNS и статических записей имен хостов.

1.3.1. Настройка клиента DNS

Traditionally, the file `/etc/resolv.conf` was a static configuration file that rarely needed to be changed or automatically changed via DCHP client hooks. Nowadays, a computer can switch from one network to another quite often and the *resolvconf* framework is now being used to track these changes and update the resolver's configuration automatically. It acts as an intermediary between programs that supply nameserver information and applications that need nameserver information. Resolvconf gets populated with information by a set of hook scripts related to network interface configuration. The most notable difference for the user is that any change manually done to `/etc/resolv.conf` will be lost as it gets overwritten each time something triggers resolvconf. Instead, resolvconf uses DHCP client hooks, and `/etc/network/interfaces` to generate a list of nameservers and domains to put in `/etc/resolv.conf`, which is now a symlink:

```
/etc/resolv.conf -> ../run/resolvconf/resolv.conf
```

To configure the resolver, add the IP addresses of the nameservers that are appropriate for your network in the file `/etc/network/interfaces`. You can also add

an optional DNS suffix search-lists to match your network domain names. For each other valid resolv.conf configuration option, you can include, in the stanza, one line beginning with that option name with a **dns-** prefix. The resulting file might look like the following:

```
iface eth0 inet static
    address 192.168.3.3
    netmask 255.255.255.0
    gateway 192.168.3.1
    dns-search example.com
    dns-nameservers 192.168.3.45 192.168.8.10
```

В опции *search* можно также указать несколько доменных имён, в этом случае DNS-запросы будут добавляться в том порядке, в котором они введены. Например, в вашей сети может быть несколько поддоменов для поиска: родительский домен *example.com* и два поддомена, *sales.example.com* и *dev.example.com*.

Если у вас несколько доменов, в которых должен выполняться поиск, ваша конфигурация может выглядеть так:

```
iface eth0 inet static
    address 192.168.3.3
    netmask 255.255.255.0
    gateway 192.168.3.1
    dns-search example.com sales.example.com dev.example.com
    dns-nameservers 192.168.3.45 192.168.8.10
```

Если вы попытаетесь проверить командой ping хост с именем *server1*, ваша система автоматически запросит DNS по их полным доменным именам (FQDN) в следующем порядке:

1. server1**.example.com**
2. server1**.sales.example.com**
3. server1**.dev.example.com**

Если совпадений не будет, DNS сервер предоставит результат *notfound* и запрос DNS потерпит неудачу.

1.3.2. Статические имена хостов

Статические имена хостов — это локально определённые соотношения «имя хоста - IP», находящиеся в файле `/etc/hosts`. Значения, определённые в файле `hosts`, по умолчанию превалируют над DNS. Это означает, что если система пытается разрешить имя и находит его в /etc/hosts, она не будет пытаться смотреть записи в DNS. В некоторых конфигурациях, особенно когда доступ в интернет не требуется, сервера, соединённые

с ограниченным количеством ресурсов, могут просто использовать статический список имён вместо DNS.

Далее приведен пример файла `hosts`, где ряд локальных серверов определены обычными именами хостов, алиасами и их эквивалентами полных имен (FQDN).

```
127.0.0.1 localhost
127.0.1.1 ubuntu-server
10.0.0.11 server1 server1.example.com vpn
10.0.0.12 server2 server2.example.com mail
10.0.0.13 server3 server3.example.com www
10.0.0.14 server4 server4.example.com file
```

В примере выше обратите внимание, что каждый сервер имеет алиас вдобавок к их правильным коротким и полным именам. *Server1* соотносится с именем *vpn*, *server2* определен как *mail*, *server3* как *www*, and *server4* как *file*.

1.3.3. Настройка переключения сервиса имён

Последовательность, в которой ваша система выбирает метод разрешения имен по IP адресам управляется конфигурационным файлом переключателя сервиса имён (NSS) `/etc/nsswitch.conf`. Как отмечено в предыдущей секции, обычно статические имена хостов, определенные в системном файле `/etc/hosts`, имеют приоритет перед разрешением имён через DNS. Далее следует пример строки, отвечающей за этот порядок перебора имён хостов в файле `/etc/hosts`.

```
hosts:          files mdns4_minimal [NOTFOUND=return] dns mdns4
```

- **files** сперва пытается разрешить статическое имя хоста в `/etc/hosts`.
- **mdns4_minimal** пытается разрешить имя с использованием параллельного (multicast) DNS.
- **[NOTFOUND=return]** означает, что любой ответ *notfound*, предшествующий процессу *mdns4_minimal* должен считаться значимым (авторитетным), и что система не будет пытаться продолжать искать ответ.
- **dns** представляет собой наследуемый последовательный (legacy unicast) DNS-запрос.
- **mdns4** представляет параллельный (multicast) DNS-запрос.

Для изменения последовательности вышеупомянутых методов разрешения имен вы можете просто заменить строку *hosts:* на значение по вашему

выбору. Например, если вы предпочитаете использовать последовательный DNS до параллельного DNS, вы можете изменить строку в `/etc/nsswitch.conf` как показано ниже:

```
hosts:          files dns [NOTFOUND=return] mdns4_minimal mdns4
```

1.4. Использование моста

Соединение нескольких интерфейсов — наиболее продвинутая настройка, но очень полезная во множестве сценариев. Один вариант — установка взаимодействия между несколькими сетевыми интерфейсами и затем использование защитного экрана (firewall) для фильтрования трафика между двумя сегментами сети. Другой сценарий — использование связывания на системе с одним интерфейсом для разрешения виртуальным машинам иметь прямой доступ во внешнюю сеть. Следующий пример раскрывает последний сценарий.

Перед настойкой взаимодействия вам потребуется установить пакет bridge-utils. Для установки пакета введите в терминале:

```
sudo apt-get install bridge-utils
```

Далее настройте взаимодействие, отредактировав `/etc/network/interfaces`:

```
auto lo
iface lo inet loopback

auto br0
iface br0 inet static
        address 192.168.0.10
        network 192.168.0.0
        netmask 255.255.255.0
        broadcast 192.168.0.255
        gateway 192.168.0.1
        bridge_ports eth0
        bridge_fd 9
        bridge_hello 2
        bridge_maxage 12
        bridge_stp off
```

Введите соответствующие значения для вашего физического интерфейса и сети.

Now bring up the bridge:

```
sudo ifup br0
```

Теперь новый мост между интерфейсами поднят и работает. Утилита application>brctl

1.5. Ресурсы

- *Страница Ubuntu Wiki Network*[1] содержит ссылки на заметки по более продвинутым настройкам сети.
- *man-страница resolvconf*[2] содержит больше информации по resolvconf.
- *man-страница interfaces*[3] содержит подробности по дополнительным опциям для `/etc/network/interfaces`.
- *man-страница dhclient*[4] содержит подробности по большему количеству опций для настройки клиента DHCP.
- Для дополнительной информации по настройке DNS-клиента смотрите *resolver man page*[5]. Глава 6 руководства O'Reilly *Администрирования сетей Linux*[6] также является хорошим источником по разрешению имён и настройке сервиса имён.
- For more information on *bridging* see the *brctl man page*[7] and the Linux Foundation's *Networking-Bridge*[8] page.

[1] https://help.ubuntu.com/community/Network
[2] http://manpages.ubuntu.com/manpages/man8/resolvconf.8.html
[3] http://manpages.ubuntu.com/manpages/man5/interfaces.5.html
[4] http://manpages.ubuntu.com/manpages/man8/dhclient.8.html
[5] http://manpages.ubuntu.com/manpages/man5/resolver.5.html
[6] http://oreilly.com/catalog/linag2/book/ch06.html
[7] http://manpages.ubuntu.com/manpages/man8/brctl.8.html
[8] http://www.linuxfoundation.org/collaborate/workgroups/networking/bridge

2. TCP/IP

Протокол управления передачей и межсетевой протокол (TCP/IP) — это стандартный набор протоколов, разработанных в конце 70-х годов управлением перспективного планирования оборонных научно-исследовательских работ (DARPA) в качестве средства коммуникации между различными типами компьютеров и компьютерных сетей. Так как сеть Интернет построена на стеке протоколов TCP/IP, они представляют самый популярный набор сетевых протоколов на Земле.

2.1. Введение в TCP/IP

Два компонента протокола TCP/IP представляют разные аспекты компьютерного взаимодействия. *Межсетевой протокол* "IP" в стеке TCP/IP — это протокол без установления соединения, который предоставляет только маршрутизацию пакетов, используя *IP-пакет (датаграмму)* как основной блок сетевой информации. IP-пакет содержит заголовок с последующим сообщением. *Протокол управления передачей* "TCP" в TCP/IP позволяет сетевым хостам устанавливать соединения, которые могут использоваться для передачи потоков данных. TCP также гарантирует, что данные между подключениями будут доставлены, и что они придут на сетевой хост в том же порядке, в каком они были посланы с другого.

2.2. Настройка TCP/IP

Настройка протокола TCP/IP состоит из нескольких элементов, которые должны быть указаны в соответствующих файлах конфигураций, или получены с помощью дополнительных служб таких как сервер протокола динамической настройки хостов (Dynamic Host Configuration Protocol, DHCP), который, в свою очередь, может быть настроен для автоматического предоставления правильных настроек TCP/IP клиентам сети. Следующим параметрам настройки должны быть указаны правильные значения, чтобы обеспечить нормальную работу вашей системы Ubuntu в сети.

Обычные элементы настроек TCP/IP и их назначение таковы:

- **IP адрес**. IP адрес — это уникальная идентификационная строка, представленная в виде четырёх десятичных чисел в диапазоне от нуля (0) до двухсот пятидесяти пяти (255), разделённых точками; каждое из четырёх чисел представляет восемь (8) бит адреса, полная длина которого тридцать два (32) бита. Этот формат называют *dotted quad notation (четырёхкомпонентная система обозначений адресов с точками)*.
- **Маска сети (Netmask)**. Маска подсети (или просто, *netmask*) — это локальная битовая маска, или наборы флагов, отделяющая часть IP-

адреса, значимую для сети, от битов, значимых для *подсети (subnetwork)*. Например, в сети класса C, стандартная маска сети определена как 255.255.255.0, она маскирует первые три байта IP-адреса и позволяет последнему байту IP-адреса оставаться доступным для обозначения хостов в подсети.

- **Сетевой адрес (Номер сети)**. Сетевой адрес (номер сети) представляет собой набор байт, заключающий в себе сетевую часть IP-адреса. Например, хост 12.128.1.2 в сети класса A будет использовать 12.0.0.0 в качестве сетевого адреса, здесь число 12 представляет первый байт IP-адреса (сетевая часть), а нули (0) во всех оставшихся трех байтах представляют потенциальные значения для хоста. Хост в сети, использующий частный IP-адрес 192.168.1.100, будет, в свою очередь, использовать номер сети 192.168.1.0, который определяет первые три байта сети 192.168.1 класса C и ноль (0) для всех возможных хостов в сети.
- **Широковещательный адрес**. Широковещательный адрес — это IP-адрес, который позволяет отправлять данные одновременно всем хостам в данной подсети вместо конкретного хоста. Стандартный основной широковещательный адрес для всех IP-сетей — 255.255.255.255, но этот адрес не может использоваться для отправки широковещательного сообщения каждому хосту в Интернете, потому что его заблокируют маршрутизаторы. Более подходящий широковещательный адрес — это тот, который устанавливается для конкретной подсети. Например, в частной сети 192.168.1.0 класса C, широковещательным адресом является 192.168.1.255. Широковещательные сообщения обычно производятся сетевыми протоколами, такими как ARP (Address Resolution Protocol — протокол разрешения адресов) и RIP (Routing Information Protocol — протокол маршрутизационной информации).
- **Адрес шлюза (Gateway Address)**. Адрес шлюза — это IP-адрес, через который некоторая сеть, или хост в сети, могут быть доступны. Пусть один сетевой хост желает организовать соединение с другим сетевым хостом, но они расположены в разных сетях, в таких случаях должен использоваться *шлюз (gateway)*. Во многих случаях адрес шлюза будет совпадать с адресом маршрутизатора той же сети, который, в свою очередь, будет перенаправлять трафик в другие сети или на другие хосты, такие как хосты Интернет. Адресу шлюза должно быть присвоено правильное значение, в противном случае ваша система не сможет связаться ни с одним хостом, находящимся за пределами вашей сети.
- **Адрес сервера имён**. Адрес сервера имён — это IP-адрес сервера службы доменных имён (DNS), который разрешает сетевые имена хостов в IP-адреса. Существует три уровня адресов серверов имён, которые

могут быть определены в порядке старшинства: *первичный* сервер имён, *вторичный* сервер имён и *третичный* сервер имён. Чтобы у вашей системы была возможность разрешения сетевых имён хостов в соответствующие им IP-адреса, вы должны определить правильные адреса серверов имён, которые вам разрешено использовать в настройках TCP/IP вашей системы. В большинстве случаев эти адреса предоставляются вашим сетевым провайдером, но есть много свободных и публично доступных серверов имён, которые можно использовать — например, сервера Level3 (Verizon) с адресами от 4.2.2.1 до 4.2.2.6.

The IP address, Netmask, Network Address, Broadcast Address, Gateway Address, and Nameserver Addresses are typically specified via the appropriate directives in the file `/etc/network/interfaces`. For more information, view the system manual page for `interfaces`, with the following command typed at a terminal prompt:

Обратитесь к соответствующей странице системного руководства о `interfaces` с помощью комманды:

```
man interfaces
```

2.3. IP-маршрутизация

Маршрутизация IP — это способ задания и определения маршрутов в сети TCP/IP, через которые могут передаваться данные. Маршрутизация использует набор *таблиц маршрутизации* для управления переадресацией сетевых пакетов с данными от их источника к месту назначения, часто через множество промежуточных сетевых узлов, называемых *маршрутизаторами*. Есть две основных разновидности IP-маршрутизации: *статическая маршрутизация* и *динамическая маршрутизация*.

Статическая маршрутизация настраивается путём добавления вручную IP-маршрутов в системную таблицу маршрутизации. Обычно это делается с помощью команды route. Статическая маршрутизация имеет много преимуществ по сравнению с динамической маршрутизацией, таких как простота реализации на небольших сетях, предсказуемость (таблица маршрутизации всегда просчитана заранее и, таким образом, маршруты строго постоянны при каждом использовании), меньшие накладные расходы по сравнению с протоколом динамической маршрутизации. Тем не менее, у статической маршрутизации также есть свои недостатки. Например, статическая маршрутизация ограничена малыми сетями и плохо масштабируема. Статическая маршрутизация также неустойчива к сбоям

в работе сети или сбоям по пути маршрута, что связано с фиксированной природой её маршрутов.

Динамическая маршрутизация меняется в зависимости от состояния больших сетей с множеством возможных IP-маршрутов от источника к приемнику. Она использует специальные протоколы маршрутизации, такие как RIP (Router Information Protocol — протокол маршрутизационной информации), которые осуществляют автоматическую корректировку маршрутизационных таблиц, делающую динамическую маршрутизацию возможной. Динамическая маршрутизация имеет ряд преимуществ перед статической. Например, лучшую масштабируемость и возможность адаптироваться к сбоям в маршрутах и выходам их из строя. Помимо этого, требуется меньше ручной настройки таблиц маршрутизации, так как маршрутизаторы узнают друг у друга про своё существование и доступные маршруты. Эта черта также сводит на нет возможность внесения некорректных записей в таблицы маршрутизации из-за простой человеческой ошибки. Динамическая маршрутизация, тем не менее, не идеальна, и имеет неудобства, такие как повышенная сложность и дополнительная нагрузка на сеть за счёт взаимодействия маршрутизаторов, которая не приносит мгновенной выгоды конечным пользователям, но при этом занимает сетевой трафик.

2.4. TCP и UDP

TCP — протокол с установлением соединения, предоставляющий коррекцию ошибок и гарантированную доставку данных через так называемое *управление передачей (flow control)*. Управление передачей определяет, когда поток данных необходимо остановить и заново отправить предыдущие пакеты данных вследствие таких проблем, как *коллизии (collisions)*. TCP обычно используется при обмене важной информацией, такой как транзакции баз данных.

Протокол пользовательских датаграмм (UDP — User Datagram Protocol), с другой стороны, является протоколом *без установления соединения*, который редко используется для передачи важных данных, поскольку в нём отсутствует управление передачей или другие способы гарантированной доставки данных. UDP обычно используется в приложениях для передачи потокового аудио или видео, в которых он работает быстрее TCP из-за отсутствия коррекции ошибок и управления передачей, и где потеря нескольких пакетов не является катастрофичной.

2.5. ICMP

Протокол управляющих сообщений сети Интернет (Internet Control Messaging Protocol, ICMP) — это расширение Интернет-протокола (IP), определённое в документе RFC#792 (Request For Comments), поддерживающее сетевые пакеты, содержащие управляющие и информационные сообщения, а также сообщения об ошибках. ICMP используется сетевыми приложениями, например, утилитой ping, с помощью которой можно определить доступность сетевого хоста или устройства. Например, сообщения об ошибках, возвращаемых ICMP, которые полезны как хостам в сети, так и устройствам типа маршрутизаторов, включают в себя *«адресат недоступен» (Destination Unreachable)* и *«превышено время ожидания» (Time Exceeded)*.

2.6. Демоны

Демоны — это специальные системные программы, которые, как правило, выполняются постоянно в фоновом режиме и ожидают запросов на функции, которые они предоставляют для других программ. Многие демоны направлены на работу с сетью; то есть, большое число демонов, выполняющихся в фоновом режиме в системе Ubuntu, могут предоставлять сетевую функциональность. В качестве примера таких сетевых демонов можно привести *Hyper Text Transport Protocol Daemon* (httpd), который предоставляет функции веб-сервера, *Secure SHell Daemon* (sshd), который предоставляет безопасный удалённый доступ к консоли и возможность передачи файлов и *Internet Message Access Protocol Daemon* (imapd), который предоставляет службы электронной почты.

2.7. Ресурсы

- There are man pages for *TCP*[9] and *IP*[10] that contain more useful information.
- Также посмотрите *TCP/IP Tutorial and Technical Overview*[11] из «красной книги» IBM.
- Ещё один ресурс — это книга издательства O'Reilly *TCP/IP Network Administration*[12].

[9] http://manpages.ubuntu.com/manpages/trusty/en/man7/tcp.7.html
[10] http://manpages.ubuntu.com/manpages/trusty/man7/ip.7.html
[11] http://www.redbooks.ibm.com/abstracts/gg243376.html
[12] http://oreilly.com/catalog/9780596002978/

3. Протокол динамической настройки хостов (Dynamic Host Configuration Protocol, DHCP)

DHCP (протокол динамической конфигурации узла) — это сетевой сервис, позволяющий компьютерам автоматически получать настройки от сервера в отличие от ручной настройки каждого компьютера в сети. Компьютеры, настроенные в качестве DHCP клиентов, не контролируют параметры, которые они получают от DHCP сервера, а сама настройка прозрачна для пользователя компьютера.

В самом общем случае, настройки, предоставляемые сервером DHCP его клиентам, включают в себя:

- IP-адрес и маску подсети
- IP-адрес шлюза по умолчанию
- IP-адрес серверов DNS

Кроме того, сервер DHCP может дополнительно предоставить такие параметры настроек, как:

- Имя хоста
- Имя домена
- Сервер синхронизации времени
- Сервер печати

Преимущество использования DHCP-сервера в сети состоит в том, что изменения настроек сети, например, изменение адреса DNS-сервера, должны выполняться только на DHCP-сервере. Все остальные компьютеры в сети будут автоматически перенастроены DHCP-клиентами во время следующего опроса ими DHCP-сервера. Дополнительное преимущество состоит в том, что становится проще подключать в сеть новые компьютеры, так как отпадает необходимость проверять доступность IP-адреса. Также сокращается количество конфликтов при назначении IP-адресов.

Сервер DHCP может предоставлять настройки, используя следующие методы:

Выделение вручную (по MAC-адресу)
: Этот метод подразумевает использование DHCP для определения уникального аппаратного адреса каждой сетевой карты, подключенной к сети, и затем продолжительного предоставления неизменной конфигурации каждый раз, когда DHCP-клиент делает запрос на DHCP-сервер, используя это сетевое устройство. Это гарантирует, что определённый адрес будет автоматически присваиваться этой сетевой карте на основе её MAC-адреса.

Динамическое выделение (пул адресов)
: In this method, the DHCP server will assign an IP address from a pool of addresses (sometimes also called a range or scope) for a period of time or lease, that is configured on the server or until the client informs the server that it doesn't need the address anymore. This way, the clients will be receiving their configuration properties dynamically and on a "first come, first served" basis. When a DHCP client is no longer on the network for a specified period, the configuration is expired and released back to the address pool for use by other DHCP Clients. This way, an address can be leased or used for a period of time. After this period, the client has to renegociate the lease with the server to maintain use of the address.

Автоматическое выделение
: Использую этот метод, DHCP автоматически присваивает устройству постоянный IP-адрес, выбранный из пула доступных адресов. Обычно DHCP используется для выдачи временного адреса, но сервер DHCP может использовать бесконечное время аренды.

Последние два метода можно считать «автоматическими», так как в обоих случаях сервер DHCP назначает адрес без необходимости дополнительного вмешательства. Единственное различие между ними в том, на какой срок арендуется IP-адрес, другими словами: будет ли адрес клиента изменяться со временем. В состав Ubuntu входит и сервер DHCP, и клиент. Сервер — это dhcpd (dynamic host configuration protocol daemon). Клиент в Ubuntu — это dhclient, и он должен быть установлен на всех компьютерах, которые должны конфигурироваться автоматически. Обе программы просты в установке и настройке, и они будут автоматически запускаться при загрузке системы.

3.1. Установка

Для установки dhcpd введите следующую команду в терминале:

```
sudo apt-get install isc-dhcp-server
```

Возможно, вам потребуется изменить настройку по умолчанию редактированием /etc/dhcp/dhcpd.conf для удовлетворения вашим потребностям и специфическим настройкам.

Вы также можете исправить /etc/default/isc-dhcp-server для определения интерфейсов, которые должен слушать dhcpd.

ПРИМЕЧАНИЕ: сообщения демона dhcpd пересылаются в syslog. Обращайтесь туда для ознакомления с диагностическими сообщениями.

3.2. Конфигурация

Сообщение об ошибке, с которым заканчивается процесс установки, может быть немного непонятным, но приведённые ниже шаги помогут вам настроить службу:

Наиболее вероятно, вы захотите установить случайную раздачу IP-адресов. Это может быть выполнено сдледующим образом:

```
# minimal sample /etc/dhcp/dhcpd.conf
default-lease-time 600;
max-lease-time 7200;

subnet 192.168.1.0 netmask 255.255.255.0 {
 range 192.168.1.150 192.168.1.200;
 option routers 192.168.1.254;
 option domain-name-servers 192.168.1.1, 192.168.1.2;
 option domain-name "mydomain.example";
}
```

В результате DHCP-сервер будет выдавать клиентам IP-адреса из диапазона 192.168.1.150-192.168.1.200. IP-адрес будет выдаваться в аренду на 600 секунд, если клиент не запросит какой-то другой период времени. В любом случае максимальный (допустимый) срок аренды будет 7200 секунд. Кроме того, сервер будет «советовать» клиенту использовать 192.168.1.254 в качестве шлюза по умолчанию и 192.168.1.1 и 192.168.1.2 в качестве серверов DNS.

После изменения конфигурационного файла необходимо перезапустить dhcpd:

```
sudo service isc-dhcp-server restart
```

3.3. Ссылки

- *dhcp3-server Ubuntu Wiki*[13] содержит дополнительную информацию.
- For more `/etc/dhcp/dhcpd.conf` options see the *dhcpd.conf man page*[14].
- *ISC dhcp-server*[15]

[13] https://help.ubuntu.com/community/dhcp3-server
[14] http://manpages.ubuntu.com/manpages/trusty/en/man5/dhcpd.conf.5.html
[15] http://www.isc.org/software/dhcp

4. Синхронизация времени с NTP

NTP — это протокол TCP/IP для синхронизации времени через сеть. По существу, клиент запрашивает текущее время на сервере и использует результат для установки своего собственного времени.

За этим простым описанием скрывается много сложностей — существуют уровни NTP-серверов, где первый уровень подключен к атомным часам, а второй и третий уровни серверов распределяют на себя нагрузку по актуальным запросам из интернета. Кроме того, клиентское приложение сложнее, чем вы можете подумать — оно компенсирует задержки соединения и регулирует время таким образом, чтобы не навредить другим процессам, запущенным на сервере. Но, к счастью, вся эта сложность скрыта от вас!

Ubuntu использует ntpdate и ntpd.

4.1. ntpdate

Ubuntu в качестве стандарта использует ntpdate, запуская эту программу один раз во время загрузки, чтобы настроить время в соответствии с NTP-сервером Ubuntu.

```
ntpdate -s ntp.ubuntu.com
```

4.2. ntpd

Демон ntpd вычисляет смещение системных часов и постоянно подстраивает их, чтобы не было больших изменений, которые могут привести, например, к противоречивым записям в журналах. Платой за это является некоторое потребление вычислительной мощности и памяти, но на современных серверах это несущественно.

4.3. Установка

Чтобы установить ntpd, введите в терминале:

```
sudo apt-get install ntp
```

4.4. Конфигурация

Отредактируйте `/etc/ntp.conf`, чтобы добавить или удалить серверы. По умолчанию используются следующие серверы:

```
# Use servers from the NTP Pool Project. Approved by Ubuntu Technical Board
# on 2011-02-08 (LP: #104525). See http://www.pool.ntp.org/join.html for
# more information.
server 0.ubuntu.pool.ntp.org
server 1.ubuntu.pool.ntp.org
server 2.ubuntu.pool.ntp.org
server 3.ubuntu.pool.ntp.org
```

После изменения конфигурационного файла необходимо перезагрузить ntpd:

```
sudo service ntp reload
```

4.5. Просмотр статуса

Используйте ntpq, чтобы увидеть больше информации:

```
# sudo ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
+stratum2-2.NTP. 129.70.130.70    2 u    5   64  377   68.461  -44.274 110.334
+ntp2.m-online.n 212.18.1.106     2 u    5   64  377   54.629  -27.318  78.882
*145.253.66.170  .DCFa.           1 u   10   64  377   83.607  -30.159  68.343
+stratum2-3.NTP. 129.70.130.70    2 u    5   64  357   68.795  -68.168 104.612
+europium.canoni 193.79.237.14    2 u   63   64  337   81.534  -67.968  92.792
```

4.6. Ссылки

- Дополнительную информацию смотрите на вики-странице *Ubuntu Time*[16].
- *ntp.org, домашняя страница проекта Network Time Protocol*[17]

[16] https://help.ubuntu.com/community/UbuntuTime
[17] http://www.ntp.org/

Глава 5. Множественное связывание устройств (DM-Multipath)

1. Множественное связывание устройств (Device Mapper Multipathing)

Множественное связывание устройств (DM-Multipath) позволяет вам настроить несколько путей ввода/вывода между серверным узлом и массивом накопителей как одно устройство. Эти пути ввода/вывода являются физическими соединениями сети хранения данных (SAN), которые могут включать различные кабели, переключатели и контроллеры. Множественное связывание объединяет пути ввода/вывода, создавая новое устройство, которое состоит из этих объединяемых путей. Эта глава представляет краткое изложение возможностей DM-Multipath, которые впервые появились в редакции Ubuntu Server 12.04. Затем в этой главе приведен обзор верхнего уровня DM-Multipath и его компонентов, а также описание процесса установки DM-Multipath.

1.1. Новые и изменённые возможности в Ubuntu Server 12.04

Произведён переход с multipath-0.4.8 к multipath-0.4.9.

1.1.1. Переход с 0.4.8

Модули проверки приоритета теперь запускаются не как отдельные программы, а как разделяемые библиотеки. Ключевая часть имён для функций также несколько изменена. Скопируйте атрибут с именем **prio_callout** как **prio**, также измените аргумент имени модуля проверки приоритета, системный путь теперь необязателен. Пример изменений:

```
device {
        vendor "NEC"
        product "DISK ARRAY"
        prio_callout mpath_prio_alua /dev/%n
        prio    alua
}
```

См. таблицу *Преобразование модулей проверки приоритета [59]*, в которой приведён полный список изменений.

Таблица 5.1. Преобразование модулей проверки приоритета

v0.4.8	v0.4.9
prio_callout mpath_prio_emc /dev/%n	**prio emc**

v0.4.8	v0.4.9
prio_callout mpath_prio_alua /dev/%n	**prio alua**
prio_callout mpath_prio_netapp /dev/%n	**prio netapp**
prio_callout mpath_prio_rdac /dev/%n	**prio rdac**
prio_callout mpath_prio_hp_sw /dev/%n	**prio hp_sw**
prio_callout mpath_prio_hds_modular %b	**prio hds**

Поскольку разборщик файла настройки множественного связывания разбирает все пары ключ/значение, которые находит и затем использует, безопасно совместное использование **prio_callout** с **prio**, и рекомендуется вставлять атрибуты **prio** до начала миграции. После этого вы можете безопасно удалить унаследованные атрибуты **prio_calliout** без прерывания работы сервиса.

1.2. Обзор

DM-Multipath может быть использован для обеспечения:

- *Избыточности*. DM-Multipath может предоставлять обход отказа в активной/пассивной настройке. В активной/пассивной настройке только половина путей используется в определённые моменты времени для ввода/вывода. Если какой-то элемент пути ввода/вывода (кабель, переключатель или контроллер) повреждён, DM-Multipath переключается на альтернативный маршрут.
- *Улучшенной производительности.* Работа DM-Multipath может быть настроена в активно/активном режиме, где ввод/вывод распределяется между путями в циклическом режиме. При некоторых конфигурациях DM-Multipath может определять загрузку путей ввода/вывода и динамически её балансировать.

1.3. Обзор массивов носителей

По умолчанию DM-Multipath содержит поддержку большинства массивов носителей, которые поддерживают DM-Multipath. Поддерживаемые устройства могут быть найдены в файле multipath.conf.defaults. Если ваш массив носителей поддерживает DM-Multipath и не настроен по умолчанию в этом файле, вам может понадобиться добавить его в файл настройки DM-Multipath multipath.conf. Для информации по конфигурационному файлу DM-Multipath смотрите раздел *The DM-Multipath Configuration File*. Некоторые массивы носителей требуют специального управления ошибками ввода/вывода и переключением маршрутов. Они требуют отдельных обработчиков оборудования модулей ядра.

1.4. Компоненты DM-Multipath

Таблица *DM-Multipath Components* описывает компоненты пакета DM-Multipath.

Таблица 5.2. Компоненты DM-Multipath

Компонент	Описание
модуль ядра dm_multipath	Перенаправляет ввод/вывод и поддерживает обход **failover** для маршрута и группы маршрутов.
команда multipath	Перечисляет и настраивает устройства **multipath**. Обычно стартует с `/etc/rc.sysinit`, но может также подниматься программой udev всякий раз, когда добавляется блоковое устройство, или она может быть запущена файловой системой initramfs.
сервис multipathd	Отслеживает маршруты; когда маршрут повреждается и восстанавливается, он может инициировать переключатели групповых путей. Обеспечивает интерактивные изменения устройств **multipath**. Этот сервис должен быть перезапущен после любых изменений файла `/etc/multipath.conf` для применения.
команда kpartx	Создаёт устройство переопределения устройств для разделов на устройстве. Необходимо использовать эту команду для DOS-совместимых разделов с DM-Multipath. Команда kpartx поставляется в своем собственном пакете, но пакет **multipath-tools** имеет на него зависимость.

1.5. Настройка DM-Multipath

DM-Multipath содержит встроенные настройки по умолчанию, которые подходят для общих конфигураций множественного связывания. Установка DM-Multipath — обычно достаточно простая процедура. Основная процедура по настройке вашей системы с использованием DM-Multipath следующая:

1. Установите пакеты **multipath-tools** и **multipath-tools-boot**.
2. Создайте пустой настроечный файл `/etc/multipath.conf`, который переопределит *следующее*
3. Если необходимо, отредактируйте конфигурационный файл **multipath.conf** для изменения значений по умолчанию и сохраните его.
4. Запустите сервис multipath
5. Обновите изначальный ramdisk

Для детальных инструкций по настройке multipath смотрите раздел *Setting Up DM-Multipath*.

2. Множественные устройства

Без DM-Multipath каждый канал от серверного узла к контроллеру накопителя рассматривается системой как отдельное устройство, даже когда канал ввода/вывода соединяет тот же сервер с тем же контроллером накопителя. DM-Multipath предоставляет возможность организации каналов ввода/вывода локально через создание единого устройства множественного связывания поверх основных устройств.

2.1. Идентификаторы устройств множественного связывания

Each multipath device has a World Wide Identifier (WWID), which is guaranteed to be globally unique and unchanging. By default, the name of a multipath device is set to its WWID. Alternately, you can set the ***user_friendly_names*** option in the multipath configuration file, which causes DM-Multipath to use a node-unique alias of the form **mpathn** as the name. For example, a node with two HBAs attached to a storage controller with two ports via a single unzoned FC switch sees four devices: **/dev/sda**, **/dev/sdb**, **/dev/sdc**, and **/dev/sdd**. DM-Multipath creates a single device with a unique WWID that reroutes I/O to those four underlying devices according to the multipath configuration. When the ***user_friendly_names*** configuration option is set to **yes**, the name of the multipath device is set to **mpathn**. When new devices are brought under the control of DM-Multipath, the new devices may be seen in two different places under the **/dev** directory: **/dev/mapper/mpathn** and **/dev/dm-n**.

- Устройства в **/dev/mapper** создаются раньше в процессе загрузки. Используйте эти имена для доступа к множественным устройствам, например, при создании логических томов.
- Устройства в виде **/dev/dm-n** только для внутреннего использования и никогда не должны использоваться.

For information on the multipath configuration defaults, including the ***user_friendly_names*** configuration option, see Section , *Configuration File Defaults*. You can also set the name of a multipath device to a name of your choosing by using the ***alias*** option in the **multipaths** section of the multipath configuration file. For information on the **multipaths** section of the multipath configuration file, see Section, *Multipaths Device Configuration Attributes*.

2.2. Согласованные имена множественных устройств в кластере

Когда опция **user_friendly_names** установлена в yes, имя множественного устройства уникально для узла, но не гарантируется то же самое на всех

узлах, использующих это устройство. Также, если вы установили опцию **alias** для устройства в секции **multipaths** конфигурационного файла `multipath.conf`, имя не будет автоматически согласовано со всеми узлами кластера. Это не вызовет сложностей при использовании LVM для создания логических устройств на множественном устройстве, но если вам требуется согласовать имена множественных устройств на всех узлах, рекомендуется оставить **user_friendly_names** значение **no** и не настраивать алиасы для устройств. По умолчанию, если у вас не установлено **user_friendly_names** в yes и не настроен алиас для устройства, имя для этого устройства будет совпадать с WWID, которое всегда одно и то же. Если же вы хотите согласованные дружественные имена, определяемые системой для всех узлов в кластере, следуйте данной процедуре:

1. Установите все множественные устройства на одной машине.
2. Заблокируйте все ваши множественные устройства на других машинах, выполнив команды:

```
# service multipath-tools stop
# multipath -F
```

3. Скопируйте файл `/etc/multipath/bindings` с первой машины на все остальные в кластере.
4. Восстановите работу сервиса multipathd на всех остальных машинах командой:

```
# service multipath-tools start
```

Если вы добавляете новое устройство, вам потребуется повторить этот процесс.

Таким же образом, если вы настроили алиас для устройства, который хотите согласовать на всех узлах в кластере, вам надо убедиться, что файл `/etc/multipath.conf` один и тот же на всех узлах кластера, следуя такой же процедуре:

1. Настройте алиасы для множественных устройств в файле `multipath.conf` на одной машине.
2. Заблокируйте все ваши множественные устройства на других машинах, выполнив команды:

```
# service multipath-tools stop
# multipath -F
```

3. Скопируйте файл `multipath.conf` с первой машины на все остальные в кластере.
4. Восстановите работу сервиса multipathd на всех остальных машинах командой:

```
# service multipath-tools start
```

Если вы добавляете новое устройство, вам потребуется повторить процесс.

2.3. Атрибуты множественных устройств

В дополнение к опциям **user_friendly_names** и **alias**, множественные устройства имеют ряд атрибутов. Вы можете изменить эти атрибуты для определённых **multipaths**, создавая секции этих устройств в секции **multipath** конфигурационного файла. Для информации по секции **multipaths** конфигурационного файла смотрите раздел "*Атрибуты множественности в файле конфигурации*".

2.4. Множественные устройства в логических томах

После создания множественных устройств вы можете использовать их имена так же, как и имя физического устройства при создании физических томов LVM. Например, если имя множественного устройства /dev/mapper/mpatha, следующая команда пометит /dev/mapper/mpatha как физический том:

```
# pvcreate /dev/mapper/mpatha
```

Вы можете использовать полученный физический том LVM при создании группы томов LVM так же, как вы использовали бы другое физическое устройство.

Если вы пытаетесь создать физический том LVM на всё устройство, на котором у вас сконфигурированы разделы, команда pvcreate приведёт к ошибке.

Когда вы создаёте логический том LVM, который использует массив носителей как активно/пассивное множественное устройство в качестве зависимых физических устройств, вы должны включить фильтры в **lvm.conf** для исключения дисков, на которых основано множественное устройство. Это нужно для ситуации, когда массив автоматически меняет активный маршрут на пассивный при получении [ошибки] ввода/вывода и множественное устройство будет обходить ошибку и восстанавливаться после сбоя всякий раз, как LVM сканирует пассивный маршрут, если устройство не отфильтровано. Для активно/пассивных массивов, которые требуют команду для перевода пассивного маршрута в активное состояние, LVM каждый раз выводит предупреждение. Для фильтрования всех SCSI устройств в конфигурационном файле LVM (lvm.conf), включите следующий фильтр в секцию devices файла:

```
filter = [ "r/block/", "r/disk/", "r/sd.*/", "a/.*/" ]
```

После изменений в `/etc/lvm.conf`, необходимо обновить **initrd** так, чтобы этот файл был скопирован туда, где фильтры имеют максимальную важность, во время загрузки. Выполните:

```
update-initramfs -u -k all
```

Каждый раз, когда меняется `/etc/lvm.conf` или `/etc/multipath.conf`, initrd должен быть создан заново для отражения этих изменений. Это обязательно, когда «чёрные списки» и фильтры необходимы для поддержания стабильной настройки хранилища.

3. Обзор установки DM-Multipath

Эта секция предоставляет пример пошаговых процедур для настройки DM-Multipath. Она включает следующие процедуры:

- Общая настройка DM-Multipath
- Игнорирование локальных дисков
- Добавление дополнительных устройств в конфигурационный файл

3.1. Настройка DM-Multipath

До проведения настройки DM-Multipath на вашей системе убедитесь, что система обновлена и содержит пакет **multipath-tools**. Если предусматривается загрузка с внешнего хранилища (SAN), также потребуется пакет **multipath-tools-boot**.

Наличие файла **/etc/multipath.conf** не является обязательным. Когда **multpath** запускается без `/etc/multipath.conf`, он ищет в своей внутренней базе подходящую конфигурацию, а также копирует данные из внутреннего «черного списка». Если после запуска **multipath -ll** без конфигурационного файла не будет обнаружено ни одного множественного устройства (multipaths), то необходимо провести расширенный анализ для определения причин, из-за которых множественные устройства не были созданы. Есть смысл изучить документацию производителей внешних хранилищ (SAN), примеры конфигурационных файлов для multipath, которые находятся в `/usr/share/doc/multipath-tools/examples`, а также проанализировать используемую база multipathd:

```
# echo 'show config' | multipathd -k > multipath.conf-live
```

В случае причудливой работы multipathd, без создания `/etc/multipath.conf`, предыдущая команда ничего не вернёт, поскольку это будет результатом *объединения* `/etc/multipath.conf` с базой в памяти. Для исправления этого либо создайте пустой `/etc/multipath.conf`, используя **touch**, либо создайте его, переопределив значения по умолчанию:

```
defaults {
        user_friendly_names no
}
```

и перезапустив multipathd:

```
# service multipath-tools restart
```

Теперь "show config" будет возвращать актуальную базу.

3.2. Установка с поддержкой множественных устройств

Для включения *поддержки множественных устройств в процессе установки*[1] используйте

```
install disk-detect/multipath/enable=true
```

по запросу установщика. Если множественные устройства найдутся, во время установки они будут показаны как **/dev/mapper/mpath<X>**.

3.3. Игнорирование локальных дисков при создании множественных устройств

Некоторые машины имеют локальные SCSI карты для своих внутренних дисков. DM-Multipath не рекомендуется для таких устройств. Следующая процедура покажет как изменить настройку multipath для игнорирования локальных дисков.

1. Determine which disks are the internal disks and mark them as the ones to blacklist. In this example, **/dev/sda** is the internal disk. Note that as originally configured in the default multipath configuration file, executing the **multipath -v2** shows the local disk, **/dev/sda**, in the multipath map. For further information on the **multipath** command output, see Section *Multipath Command Output*.

```
# multipath -v2
create: SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1 undef WINSYS,SF2372
size=33 GB features="0" hwhandler="0" wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 0:0:0:0 sda 8:0  [---------

device-mapper ioctl cmd 9 failed: Invalid argument
device-mapper ioctl cmd 14 failed: No such device or address
create: 3600a0b80001327d80000006d43621677 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:0 sdb 8:16  undef ready  running
    `- 3:0:0:0 sdf 8:80 undef ready  running

create: 3600a0b800013275100000009a436215ec undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:1 sdc 8:32 undef ready  running
    `- 3:0:0:1 sdg 8:96 undef ready  running

create: 3600a0b80001327d800000070436216b3 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
```

[1] http://wiki.debian.org/DebianInstaller/MultipathSupport

```
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:2 sdd 8:48 undef ready  running
    `- 3:0:0:2 sdg 8:112 undef ready  running

create: 3600a0b80001327510000009b4362163e undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:3 sdd 8:64 undef ready  running
    `- 3:0:0:3 sdg 8:128 undef ready  running
```

2. Для исключения из списка устройства **/dev/sda** при использовании multipath, отредактируйте секцию blacklist файла `/etc/multipath.conf` для включения в неё этого устройства. Вы можете заблокировать устройство **sda** используя тип **devnode**, что не является безопасной процедурой, поскольку с этого момента не гарантируется, что **/dev/sda** будет тем же после перезагрузки. Для блокирования индивидуальных устройств, лучше использовать их WWID. Обратите внимание, что в выводе команды **multipath -v2** WWID устройства `/dev/sda` указан как SIBM-ESXSST336732LC___F3ET0EP0Q000072428BX1. Для блокирования этого устройства, включите следующее в файл `/etc/multipath.conf`.

```
blacklist {
      wwid SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1
}
```

3. После изменений файла `/etc/multipath.conf`, вы должны вручную дать команду сервису **multipathd** перечитать конфигурационный файл. Следующая команда применит настройки из изменённого `/etc/multipath.conf`.

```
# service multipath-tools reload
```

4. Запустите следующую команду для удаления множественного устройства:

```
# multipath -f SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1
```

5. To check whether the device removal worked, you can run the **multipath -ll** command to display the current multipath configuration. For information on the **multipath -ll** command, see Section *Multipath Queries with multipath Command*. To check that the blacklisted device was not added back, you can run the multipath command, as in the following example. The multipath command defaults to a verbosity level of **v2** if you do not specify a **-v** option.

```
# multipath

create: 3600a0b80001327d80000006d43621677 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
```

```
    |- 2:0:0:0 sdb 8:16  undef ready  running
      `- 3:0:0:0 sdf 8:80 undef ready  running

  create: 3600a0b80001327510000009a436215ec undef WINSYS,SF2372
  size=12G features='0' hwhandler='0' wp=undef
  `-+- policy='round-robin 0' prio=1 status=undef
    |- 2:0:0:1 sdc 8:32 undef ready  running
      `- 3:0:0:1 sdg 8:96 undef ready  running

  create: 3600a0b80001327d800000070436216b3 undef WINSYS,SF2372
  size=12G features='0' hwhandler='0' wp=undef
  `-+- policy='round-robin 0' prio=1 status=undef
    |- 2:0:0:2 sdd 8:48 undef ready  running
      `- 3:0:0:2 sdg 8:112 undef ready  running

  create: 3600a0b80001327510000009b4362163e undef WINSYS,SF2372
  size=12G features='0' hwhandler='0' wp=undef
  `-+- policy='round-robin 0' prio=1 status=undef
    |- 2:0:0:3 sdd 8:64 undef ready  running
      `- 3:0:0:3 sdg 8:128 undef ready  running
```

3.4. Настройка устройств массивов хранения

По умолчанию DM-Multipath включает поддержку большинства массивов хранения, которые поддерживают работу с DM-Multipath. Значения конфигурационных параметров по умолчанию, включая поддерживаемые устройства, могут быть найдены в файле `multipath.conf.defaults`.

Если вам нужно добавить устройство, не поддерживаемое по умолчанию, отредактируйте файл `/etc/multipath.conf` для добавления информации о требуемом устройстве.

Например, при добавлении информации о HP Open-V series запись будет выглядеть так, где **%n** — имя устройства:

```
devices {
     device {
            vendor "HP"
            product "OPEN-V."
            getuid_callout "/lib/udev/scsi_id --whitelisted --device=/dev/%n"
     }
}
```

Для дополнительной информации смотрите раздел *Устройства в файле конфигурации [82]*.

4. Конфигурационный файл DM-Multipath

По умолчанию DM-Multipath предоставляет конфигурации для большинства множественных устройств. В дополнение к этому DM-Multipath включает поддержку большинства массивов хранения, которые поддерживают DM-Multipath. Значения конфигураций по умолчанию и поддерживаемые устройства можно найти в файле `multipath.conf.defaults`.

Вы можете переопределить настроенные значения по умолчанию для DM-Multipath, изменив конфигурационный файл `/etc/multipath.conf`. Если необходимо, вы можете также добавить массив хранения, который не поддерживается по умолчанию, в конфигурационный файл. Эта глава предоставляет информацию по разбору и изменению файла `multipath.conf`. Она содержит следующие разделы:

В файле конфигурации multipath вам нужно определить только те секции, которые вам потребуются для ваших настроек или те, для которых вы захотите изменить значения по умолчанию, определённые в `multipath.conf.defaults`. Если в файле присутствуют секции, не относящиеся к вашему оборудованию, или для которых вы не хотите менять значения по умолчанию, вы можете оставить их закомментированными, как в изначальном файле.

Файл конфигурации допускает синтаксис регулярных выражений.

Версию конфигурационного файла с комментариями можно найти в архиве `/usr/share/doc/multipath-tools/examples/multipath.conf.annotated.gz`.

4.1. Обзор файла конфигурации

Конфигурационный файл multipath разделяется на следующие секции:

blacklist
Перечисляет специфические устройства, которые не принимаются во внимание multipath.

blacklist_exceptions
Перечисляет кандидатов в множественные устройства, которые иначе будут блокироваться согласно параметрам секции blacklist.

defaults
Общие настройки по умолчанию для DM-Multipath.

multipath
Параметры настроек по характеристикам отдельных множественных устройств. Эти значения переопределяют те, что определены в секциях **defaults** и **devices**.

devices
Параметры настроек для отдельных контроллеров хранилищ. Эти значения переопределяют те, что определены в секции **defaults**. Если вы используете дисковый массив, который не поддерживается по умолчанию, вам может потребоваться создать для него подсекцию в разделе devices.

Когда система определяет атрибут множественного устройства, сначала она ищет совпадения в секции multipath, потом в devices, и только затем использует значения по умолчанию.

4.2. "Чёрный список" в файле конфигурации

Секция blacklist конфигурационного файла multipath определяет устройства, которые не будут использоваться, когда система настраивает множественные устройства. Устройства, внесённые в список блокировки, не будут группироваться в множественные устройства.

- Если вам действительно нужно заблокировать устройства, вы можете сделать это, используя следующие критерии:
  - По WWID, как описано в разделе *Блокировка по WWID [72]*
  - По имени устройства, как описано в разделе *Блокировка по имени устройства [73]*
  - По типу устройства, как описано в разделе *Блокировка по типу устройства [73]*

  По умолчанию множество типов устройств блокируются даже если вы комментируете изначальную секцию blacklist конфигурационного файла. Для информации смотрите *Блокировка по имени устройства [73]*

4.2.1. Блокировка по WWID

Вы можете задать определённые устройства для блокирования по их международному идентификатору (WWID) с использованием метки **wwid** в секции **blacklist** конфигурационного файла.

Следующий пример показывает строки конфигурационного файла, которые будут блокировать устройство с WWID 26353900f02796769.

```
blacklist {
       wwid 26353900f02796769
}
```

4.2.2. Блокировка по имени устройства

Вы можете заблокировать типы устройств по их именам так, что они не будут использоваться для группировки в множественные устройства, задав метку **devnode** в секции **blacklist**.

Следующий пример показывает строки конфигурационного файла, которые заблокируют все SCSI устройства, поскольку они блокируют все устройства sd*.

```
blacklist {
       devnode "^sd[a-z]"
}
```

Вы можете использовать метку **devnode** в секции **blacklist** для определения отдельных блокируемых устройств вместо всех устройств определенного типа. Тем не менее, это не рекомендуется, поскольку, несмотря на то, что они статично определены в правилах udev, нет гарантии что определённое устройство будет иметь то же имя после перезагрузки. Например, после перезагрузки имя устройства может поменяться с `/dev/sda` на `/dev/sdb`.

По умолчанию следующие метки **devnode** включены в список blacklist. Устройства, определяемые в этой секции, как правило не поддерживают DM-Multipath. Чтобы разрешить любое из перечисленного при создании множественных устройств, вы можете определить их в секции **blacklist_exceptions** как показано в разделе *Исключения блокировки [74]*

```
blacklist {
       devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
       devnode "^hd[a-z]"
}
```

4.2.3. Блокировка по типу устройства

Вы можете определить отдельные типы устройств в секцию **blacklist** файла конфигурации с помощью секций device. Следующий пример блокирует все устройства IBM DS4200 и любые производства HP.

```
blacklist {
       device {
```

```
                vendor  "IBM"
                product "3S42"       #DS4200 Product 10
        }
        device {
                vendor  "HP"
                product "*"
        }
}
```

4.2.4. Исключения блокировки

Вы можете использовать секцию **blacklist_exceptions** конфигурационного файла для разрешения множественных устройств, заблокированных по умолчанию.

Например, если у вас множество устройств и вы хотите разрешить только одно из них (с WWID 3600d0230000000000e13955cc3757803), вместо того, чтобы блокировать каждое из них отдельно за исключением требуемого, можно заблокировать все и затем разрешить только одно, добавив следующие строки в файл `/etc/multipath.conf`.

```
blacklist {
        wwid "*"
}

blacklist_exceptions {
        wwid "3600d0230000000000e13955cc3757803"
}
```

Когда добавляете устройства в секцию **blacklist_exceptions** конфигурационного файла, вы должны указывать исключения тем же способом, что и в секции **blacklist**. Например, исключение по WWID не сработает для устройств, определённых в blacklist с помощью **devnode**, даже если заблокированное устройство ассоциируется с данным WWID. Точно так же исключения по devnode применимы только к меткам devnode, а исключения по device — к меткам device.

4.3. Значения по умолчанию в файле конфигурации

Файл конфигурации `/etc/multipath.conf` включает секцию **defaults**, которая устанавливает параметр **user_friendly_names** в **yes**, как показано ниже:

```
defaults {
        user_friendly_names yes
}
```

Это переопределяет значение параметра **user_friendly_names** по умолчанию.

Конфигурационный файл содержит шаблоны настроек по умолчанию. Эта секция комментируется как показано ниже:

```
#defaults {
#       udev_dir                /dev
#       polling_interval        5
#       selector                "round-robin 0"
#       path_grouping_policy    failover
#       getuid_callout          "/lib/dev/scsi_id --whitelisted --device=/dev/%n"
# prio   const
# path_checker  directio
# rr_min_io  1000
# rr_weight  uniform
# failback  manual
# no_path_retry  fail
# user_friendly_names no
#}
```

Для переопределения значения по умолчанию любого настраиваемого параметра вы можете скопировать соответствующую строку из этого шаблона в секцию **defaults** и снять комментарий. Например, чтобы переустановить параметр **path_grouping_policy** в **multibus** вместо определённого по умолчанию **failover**, скопируйте соответствующую строку из шаблона и раскомментируйте ее, как показано ниже:

```
defaults {
        user_friendly_names     yes
        path_grouping_policy    multibus
}
```

Таблица *Настройки Multipath по умолчанию [75]* определяет атрибуты, которые устанавливаются в секции **defaults** файла `multipath.conf`. Эти значения используются DM-Multipath, если они не переопределены атрибутами, заданными в секциях **devices** и **multipaths** в файле `multipath.conf`.

Таблица 5.3. Настройки Multipath по умолчанию

Атрибут	**Описание**
polling_interval	Определяет интервал между двумя проверками маршрутов в секундах. Для правильной работы маршрутов, проверки постепенно увеличиваются до значения (4 * **polling_interval**). Значение по умолчанию **5**.
udev_dir	Каталог, где создаются узлы устройств udev. По умолчанию /dev.

Множественное связывание устройств (DM-Multipath)

Атрибут	Описание
multipath_dir	Каталог, где сохраняются динамические объекты общего доступа. Значение по умолчанию зависит от системы, обычно `/lib/multipath`.
verbosity	Значение уровня комментариев по умолчанию. Большее значение увеличивает количество комментариев. Допустимые значения от 0 до 6. Значение по умолчанию 2.
path_selector	Определяет алгоритм определения следующего маршрута ввода/вывода по умолчанию. Возможные значения: • **round-robin 0**: Цикл по всем маршрутам в группе маршрутов, посылая одинаковый поток в каждый. • **queue-length 0**: Посылать следующую порцию данных по маршруту с наименьшим количеством невыполненных запросов. • **service-time 0**: Посылать следующую порцию данных по маршруту с наименьшими задержками, которые определяются делением общего объема невыполненного ввода/вывода на каждом маршруте на их относительную пропускную способность. Значение по умолчанию **round-robin 0**.
path_grouping_policy	Определяет политику группирования маршрутов по умолчанию для заранее неопределенных множественных устройств. Допустимые значения: • **failover** = 1 маршрут на приоритетную группу • **multibus** = 1 все доступные маршруты на приоритетную группу • **group_by_serial** = 1 приоритетная группа на обнаруженный серийный номер • **group_by_prio** = 1 приоритетная группа на значение приоритета маршрута • **group_by_node_name** = 1 приоритетная группа на целевое имя узла

Атрибут	Описание
	Значение по умолчанию **failover.**
getuid_callout	Определяет программу и аргументы для получения уникального идентификатора маршрута. Требуется абсолютный адрес маршрута. Значение по умолчанию **/lib/udev/scsi_id --whitelisted --device=/dev/%n.**
prio	Определяет функцию вызова для определения значения приоритета маршрута. Например, биты ALUA в спецификации SPC-3 обеспечивают приемлемое значение prio. Возможные значения: • **const**: Устанавливает приоритет 1 для всех маршрутов. • **emc**: Генерирует приоритет маршрута для массивов EMC. • **alua**: Генерирует приоритет маршрута на основе установок ALUA для SCSI-3. • **netapp**: Генерирует приоритет маршрута для массивов NetApp. • **rdac**: Генерирует приоритет маршрута для контроллеров LSI/Engenio RDAC. • **hp_sw**: Генерирует приоритет маршрута для контроллеров Compaq/HP в активно/резервном режиме. • **hds**: Генерирует приоритет маршрута для дисковых массивов Hitachi HDS Modular. Значение по умолчанию **const**.
prio_args	Строка аргументов, передаваемая в функцию prio. Большинство функций prio не требуют аргументов. Установщик приоритетов datacore требует один. Например, **"timeout=1000 preferredsds=foo"**. Значение по умолчанию (null) **""**.
features	Дополнительные особенности множественных устройств. Единственная существующая опция - это **queue_if_no_path**, аналогичная установке **no_path_retry** для **queue**. Для дополнительной

Атрибут	Описание
	информации по проблемам, которые могут возникнуть при использовании этой опции, смотрите секцию "*Issues with queue_if_no_path feature*".
path_checker	Определяет метод по умолчанию для получения статуса маршрута. Возможные значения: • **readsector0**: Читает первый сектор устройства. • **tur**: Передает TEST UNIT READY устройству. • **emc_clariion**: Запрашивает у EMC Clariion специфическую страницу EVPD 0xC0. • **hp_sw**: Определяет статус маршрута массива носителей HP с использованием микропрограммного статуса Активный/ Резервный. • **rdac**: Check the path status for LSI/Engenio RDAC storage controller. • **directio**: Использует прямое чтение первого сектора. Значение по умолчанию **directio**.
failback	Управляет восстановлением после сбоя на группе маршрутов. • Значение **immediate** определяет немедленное восстановление приоритета до высшего у группы маршрутов, которая содержит активные маршруты. • **manual** предписывает, что не нужно немедленно восстанавливаться после сбоя и что восстановление может произойти только при вмешательстве оператора. • Числовое значение больше 0 определяет восстановление после указанного количества секунд. Значение по умолчанию **manual**.
rr_min_io	Определяет количество запросов ввода/вывода для переключения маршрута на другой в текущей группе маршрутов.

Атрибут	Описание
	Значение по умолчанию `1000`.
rr_weight	Если указано **priorities**, то вместо отправки **rr_min_io** запросов до переключения маршрута, количество запросов определяется **path_selector**, умноженное на приоритет маршрута, определяемый функцией **rr_min_io**. Если указано **uniform**, то все маршруты имеют одинаковый вес. Значение по умолчанию **uniform**.
no_path_retry	Числовое значение для этого атрибута определяет количество попыток системы использовать поврежденный маршрут до отключения. Значение **immediate** указывает на немедленное отключение, без запросов. Значение **queue** указывает на безостановочный опрос маршрута до его восстановления. Значение по умолчанию `0`.
user_friendly_names	Если установлено yes, означает, что система использует файл `/etc/multipath/bindings` для назначения постоянного и уникального псевдонима **alias** для множественного устройства **multipath** в виде mpathn. Если указано no, система будет использовать WWID в качестве **alias** для **multipath**. В обоих случаях то, что указано здесь может быть переопределено в секции multipaths конфигурационного файла. Значение по умолчанию **no**.
queue_without_daemon	Если установлено no, сервис **multipathd** отключит опрос всех устройств, когда они выключены. Значение по умолчанию **yes**.
flush_on_last_del	Если установлено yes, то **multipath** отключит опрос устройства, когда последний маршрут к нему будет удален. Значение по умолчанию **no**.

Атрибут	Описание
max_fds	Устанавливает максимальное количество дескрипторов, которое может быть открыто **multipath** и сервисом **multipathd**. Это эквивалент команде ulimit -n. Значение максимума установится в качестве системного ограничения в файле `/proc/sys/fs/nr_open`. Если значение не установлено, максимальное количество дескрипторов открытых файлов берется из вызывающего процесса, обычно значение равно 1024. Для безопасности, стоит установить значение по максимальному количеству путей + 32, если это значение больше 1024.
checker_timer	Ограничение по времени для проверок маршрутов, которое выдают SCSI команды с заданным таймаутом, в секундах. Значение по умолчанию берется из `/sys/block/sdx/device/timeout`, которое установлено в `30` сек. для выпуска Ubuntu 12.04 LTS.
fast_io_fail_tmo	Количество секунд, которое интерфейс SCSI будет ждать после обнаружения проблемы на удаленном порту оптического канала (FC) до установки состояния падения устройства ввода/вывода по этому порту. Это значение должно быть меньше, чем значение dev_loss_tmo. Установка значения off выключает ограничение по времени. Значение по умолчанию зависит от операционной системы.
dev_loss_tmo	Количество секунд, которое интерфейс SCSI будет ждать после обнаружения проблемы на удаленном порту оптического канала (FC) до удаления его из системы. При установке значения infinity будет использован интервал в 2147483647 сек. (68 лет). Значение по умолчанию зависит от операционной системы.

4.4. Атрибуты множественности в файле конфигурации

Таблица *Атрибуты множественности [81]* содержит перечень атрибутов, которые вы можете установить для каждого отдельного множественного устройства в секции **multipaths** конфигурационного файла `multipath.conf`. Эти атрибуты применяются только к одному конкретному множественному устройству. Эти значения по умолчанию используются DM-Multipath и переопределяют атрибуты, установленные в секциях **defaults** и **devices** файла multipath.conf.

Таблица 5.4. Атрибуты множественности

Атрибут	Описание
wwid	Указывает WWID устройства **multipath**, для которого применяется атрибут **multipath**. Этот параметр обязательный для этой секции файла `multipath.conf`.
alias	Определяет символическое имя для устройства **multipath**, для которого применяется атрибут **multipath**. Если вы используете **user_friendly_names**, не устанавливайте это значение в mpathn. Это может привести к конфликту с автоматически присвоенным дружественным именем и предоставить вам некорректные имена устройств.

В дополнение следующие параметры могут быть переопределены в секции **multipath** section

- *path_grouping_policy*
- *path_selector*
- *failback*
- *prio*
- *prio_args*
- *no_path_retry*
- *rr_min_io*
- *rr_weight*
- *flush_on_last_del*

Следующий пример показывает множественные атрибуты, определённые в конфигурационном файле для двух отдельных множественных устройств. Первое устройство имеет WWID 3600508b4000156d70001200000b0000 и символическое имя yellow.

The second multipath device in the example has a WWID of 1DEC_____321816758474 and a symbolic name of red. In this example, the *rr_weight* attributes are set to priorities.

```
multipaths {
       multipath {
              wwid                  3600508b4000156d70001200000b0000
              alias                 yellow
              path_grouping_policy  multibus
              path_selector         "round-robin 0"
              failback              manual
              rr_weight             priorities
              no_path_retry         5
       }
       multipath {
              wwid                  1DEC_____321816758474
              alias                 red
              rr_weight             priorities
        }
}
```

4.5. Устройства в файле конфигурации

Таблица *Атрибуты устройств [83]* показывает атрибуты, которые вы можете поставить для каждого отдельного дискового массива в секции devices конфигурационного файла multipath.conf. Эти атрибуты используются DM-Multipath пока не будут переопределены в секции **multipaths** файла `multipath.conf` для маршрутов, которые включают это устройство. Эти атрибуты переопределяют наборы атрибутов в секции **defaults** файла `multipath.conf`.

Многие устройства, поддерживающие множественные маршруты, включены по умолчанию в настройки multipath. Значения для устройств, поддерживаемых по умолчанию, перечислены в файле `multipath.conf.defaults`. Возможно, вам не потребуется изменять значения для этих устройств, но если потребуется, вы можете переопределить значения по умолчанию, включив метки в файл конфигурации для устройства, которые переопределят эти значения. Вы можете скопировать значения по умолчанию из `multipath.conf.annotated.gz` или, если предпочитаете короткий конфигурационный файл, из файла `multipath.conf.synthetic` для нужного устройства и перезаписать значения, которые вы хотите изменить.

Для добавления в эту секцию конфигурационного файла устройства, которое не настроилось автоматически по умолчанию, вы должны установить параметры **vendor** и **product**. Вы можете найти эти значения, просматривая **/sys/block/device_name/device/vendor** и **/sys/block/**

device_name/device/model, где device_name — это устройство, требующее настройки множественности, как в следующем примере:

```
# cat /sys/block/sda/device/vendor
WINSYS
# cat /sys/block/sda/device/model
SF2372
```

Дополнительные параметры для определения зависят от специфических устройств. Если устройство активно/активное, вам, как правило, не требуется устанавливать дополнительные параметры. Возможно вы захотите установить *path_grouping_policy* в **multibus**. Другие параметры, которые вы, возможно, захотите определить — это *no_path_retry* and *rr_min_io*, как описано в таблице *Атрибуты множественности [81]*.

Если устройство активно/пассивное, но автоматически переключает маршруты ввода/вывода на пассивный маршрут, вам потребуется изменить функцию проверки на ту, которая не посылает поток ввода/вывода для проверки работоспособности (иначе ваше устройство будет постоянно находиться в состоянии сбоя). Это также всегда подразумевает, что вы установили *path_checker* в **tur**; это работает для всех SCSI устройств, которые поддерживают команду Test Unit Ready, а таких большинство.

Если устройство требует специальной команды для переключения маршрутов, то настройка этого устройства требует модуля ядра для управления оборудованием. Доступный в данное время обработчик — это emc. Если это не подходит для вашего устройства, то, возможно, у вас не получится настроить устройство для multipath.

Таблица 5.5. Атрибуты устройств

Атрибут	**Описание**
vendor	Указывает название производителя устройства хранения информации, к которому применяются атрибуты устройства, например, **COMPAQ**.
product	Определяет имя производителя устройства хранения, для которого применяют атрибуты, например, **HSV110 (C)COMPAQ**.
revision	Определяет идентификатор версии устройства хранения.
product_blacklist	Определяет регулярное выражение для блокировки устройства по его названию.

Атрибут	Описание
hardware_handler	Определяет модуль, который будет использован для выполнения специфических действий, когда переключается группа маршрутов или обнаруживается ошибка ввода/вывода. Возможные значения включают: • **1 emc**: обработчик для массивов хранения EMC. • **1 alua**: обработчик для SCSI-3 массивов ALUA. • **1 hp_sw**: обработчик для контроллеров Compaq/HP. • **1 rdac**: обработчик для контроллеров LSI/Engenio RDAC.

В дополнение следующие параметры могут быть переопределены в секции **device**

- *path_grouping_policy*
- *getuid_callout*
- *path_selector*
- *path_checker*
- *features*
- *failback*
- *prio*
- *prio_args*
- *no_path_retry*
- *rr_min_io*
- *rr_weight*
- *fast_io_fail_tmo*
- *dev_loss_tmo*
- *flush_on_last_del*

Каждый раз, как определяется hardware_handler, ваша обязанность проверить, что соответствующий модуль ядра загружен для поддержки указанного интерфейса. Эти модули могут быть найдены в **/lib/modules/`uname -r`/kernel/drivers/scsi/device_handler/** . Необходимый модуль должен быть интегрирован в initrd, чтобы гарантировать обязательное обнаружение и способность обхода-восстановления сбоев, доступные во время загрузки. Например,

```
# echo scsi_dh_alua >> /etc/initramfs-tools/modules  ## append module to file
# update-initramfs -u -k all
```

Следующий пример показывает метку device в конфигурационном файле multipath:

```
#devices {
# device {
#  vendor   "COMPAQ  "
#  product   "MSA1000         "
#  path_grouping_policy multibus
#  path_checker  tur
#  rr_weight  priorities
# }
#}
```

Пробелы, оставленные в полях **vendor**, **product**, и **revision** имеют значение, поскольку multipath выполняет прямое сравнение этих атрибутов, чей формат определён в спецификациях SCSI, особенно команда *Standard INQUIRY*[2]. Когда используются кавычки, поля vendor, product и revision будут интерпретироваться строго по спецификации. Регулярные выражения могут интегрироваться в закавыченные строки. Поля, будучи объявлены без требуемых пробелов, будут скопированы multipath в буфер правильного размера и дополнены требуемым количеством пробелов. Спецификация ожидает, что всё поле будет заполнено печатаемыми символами или пробелами, как видно в примере выше:

- vendor: 8 символов
- product: 16 символов
- revision: 4 имвола

Для создания более надёжного файла конфигурации могут быть также использованы регулярные выражения. Операторы включают **^ $ [ ] . * ? +**. Примеры работающих регулярных выражений могут быть найдены при исследовании примеров живой базы multipath и файла `multipath.conf` , которые находятся в `/usr/share/doc/multipath-tools/examples`:

```
# echo 'show config' | multipathd -k
```

[2] http://en.wikipedia.org/wiki/SCSI_Inquiry_Command

5. Администрирование DM-Multipath и устранение проблем

5.1. Изменение размера работающего множественного устройства

Если вам требуется изменить размер работающего множественного устройства, используйте следующую процедуру:

1. Измените размер вашего физического устройства. Эта операция зависит от платформы хранилища.
2. Используйте следующую команду для поиска маршрутов для логического номера узла (LUN):

   ```
   # multipath -l
   ```

3. Измените размер маршрутов. Для SCSI устройств запись 1 в файл `rescan` этого устройства заставляет SCSI драйвер обновить информацию, как в следующей команде:

   ```
   # echo 1 > /sys/block/device_name/device/rescan
   ```

4. Измените размер множественного устройства запуском команды multipathd:

   ```
   # multipathd -k 'resize map mpatha'
   ```

5. Измените размер файловой системы (предполагается, что не используется LVM и DOS разделы):

   ```
   # resize2fs /dev/mapper/mpatha
   ```

5.2. Перенос корневой файловой системы с одиночного устройства на множественное

Это значительно упрощено за счёт использования UUID для идентификации устройств в качестве естественной метки. Просто установите **multipath-tools-boot** и перегрузитесь. Это перестроит изначальный ramdisk и предоставит multipath возможность построить маршруты до того как корневая система будет смонтирована по UUID.

Каждое обновление `multipath.conf` вынуждает initrd запустить **update-initramfs -u -k all**. Следствием этого является копирование `multipath.conf` на ramdisk и внедрение его для определения доступных устройств для группирования через их секции blacklist и device.

5.3. Перенос файловой системы подкачки с одиночного устройства на множественное

Процедура в точности такая же, как приведена в предыдущем разделе *Перенос корневой файловой системы с одиночного устройства на множественное*.

5.4. Сервис Multipath

Если вы испытываете трудности в настройке multipath, вам надо убедиться, что сервис multipath запущен, как описано в "*Настройка DM-Multipath*". Сервис **multipathd** должен быть запущен для того, чтобы использовать multipathd устройства. Также смотрите раздел *Решение проблем с помощью интерактивной консоли multipathd* касательно взаимодействия с **multipathd**, как со средством отладки.

5.5. Проблемы с queue_if_no_path

Если установлены свойства **features "1 queue_if_no_path"** в файле `/etc/multipath.conf`, то любой процесс, использующий ввод-вывод, будет сбрасываться, пока восстанавливаются один или несколько маршрутов. Для предотвращения этого установите параметр ***no_path_retry*** **N** в `/etc/multipath.conf`.

Когда вы установите параметр **no_path_retry**, удалите также опции **features "1 queue_if_no_path"** из файла `/etc/multipath.conf`. Однако если вы используете множественное устройство, для которого опция `features "1 queue_if_no_path"` скомпилирована по умолчанию, как для множества устройств SAN, вам придётся добавить значение `features "0"` для переопределения этого умолчания. Вы можете это сделать копированием существующей секции **devices** и только этой секции (а не всего файла), из `/usr/share/doc/multipath-tools/examples/multipath.conf.annotated.gz` в `/etc/multipath.conf` и редактированием её по вашим потребностям.

Если вам требуется использовать опцию `features "1 queue_if_no_path"` и вы испытываете отмеченные здесь проблемы, используйте команду для редактирования политики в процессе работы с определенным LUN (т.е. для каждого недоступного маршрута). Например, если вы хотите изменить политику для множественного устройства `mpathc` с `"queue_if_no_path"` на `"fail_if_no_path"` выполните следующую команду:

```
# dmsetup message mpathc 0 "fail_if_no_path"
```

Вы должны использовать псевдоним `mpathN` вместо пути.

5.6. Вывод команды multipath

Когда вы создаёте, изменяете или просматриваете множественные устройства, вы получаете вывод текущих настроек устройства. Формат показан ниже. Для каждого множественного устройства:

```
action_if_any: alias (wwid_if_different_from_alias) dm_device_name_if_known vendor,product
size=size features='features' hwhandler='hardware_handler' wp=write_permission_if_known
```

Для каждой группы маршрутов:

```
-+- policy='scheduling_policy' prio=prio_if_known
status=path_group_status_if_known
```

Для каждого маршрута:

```
 `- host:channel:id:lun devnode major:minor dm_status_if_known path_status
online_status
```

Например, вывод команды multipath может выглядеть следующим образом:

```
3600d0230000000000e13955cc3757800 dm-1 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=active
| `- 6:0:0:0 sdb 8:16  active ready  running
`-+- policy='round-robin 0' prio=1 status=enabled
  `- 7:0:0:0 sdf 8:80  active ready  running
```

Если маршрут поднят и готов к вводу-выводу, статус маршрута **ready (готов)** или *ghost (скрытый)*. Если маршрут погашен, статус **faulty (дефектный)** или **shaky (шаткий)**. Статус маршрута обновляется периодически сервисом multipathd на основе интервала опросов, определённом в файле `/etc/multipath.conf`.

Статус dm аналогичен статусу маршрута, но только с точки зрения ядра. Статус dm имеет два состояния: **failed**, который аналогичен **faulty**, и **active**, который определяет все остальные состояния. Изредка статусы маршрута и dm бывают временно несогласованны.

Возможные значения **online_status** — **running** и **offline**. Статус *offline* означает, что SCSI устройство отключено.

Когда множественное устройство создаётся или изменяется, статус группы маршрутов, имя dm устройства, права на запись и dm статус неизвестны. Также значения бывают не всегда корректны.

5.7. Получение информации через команду multipath

Вы можете использовать опции **-l** и **-ll** команды **multipath** для просмотра текущей конфигурации multipath. Опция **-l** показывает топологию multipath, собранную из информации в sysfs и маршрутизаторе устройств. Опция **-ll**

показывает ту же информацию, что и опция **-l**, а также дополнительную информацию по всем остальным доступным компонентам системы.

При выводе конфигурации multipath существуют три уровня детализации, которые вы можете задавать опцией **-v** команды multipath. Указание **-v0** приводит к отсутствию вывода. Указание **-v1** выводит только имена созданных или обновлённых множественных устройств, которые вы можете затем использовать в других утилитах, таких как kpartx. Указание **-v2** печатает все обнаруженные пути, множественные маршруты и маршрутизаторы устройств.

Уровень **verbosity** multipath по умолчанию равен **2** и может быть изменён глобально установкой *атрибута verbosity* в секции **defaults** файла `multipath.conf`.

Следующий пример показывает пример вывода команды **multipath -l**.

```
# multipath -l
  3600d0230000000000e13955cc3757800 dm-1 WINSYS,SF2372
  size=269G features='0' hwhandler='0' wp=rw
  |-+- policy='round-robin 0' prio=1 status=active
  | `- 6:0:0:0 sdb 8:16  active ready  running
  `-+- policy='round-robin 0' prio=1 status=enabled
    `- 7:0:0:0 sdf 8:80  active ready  running
```

А данный пример показывает вывод команды **multipath -ll**.

```
# multipath -ll
  3600d0230000000000e13955cc3757801 dm-10 WINSYS,SF2372
  size=269G features='0' hwhandler='0' wp=rw
  |-+- policy='round-robin 0' prio=1 status=enabled
  | `- 19:0:0:1 sdc 8:32  active ready  running
  `-+- policy='round-robin 0' prio=1 status=enabled
    `- 18:0:0:1 sdh 8:112 active ready  running
    3600d0230000000000e13955cc3757803 dm-2 WINSYS,SF2372
    size=125G features='0' hwhandler='0' wp=rw
    `-+- policy='round-robin 0' prio=1 status=active
      |- 19:0:0:3 sde 8:64  active ready  running
        `- 18:0:0:3 sdj 8:144 active ready  running
```

5.8. Опции команды multipath

В таблице *Полезные опции команды multipath [89]* описаны несколько параметров команды **multipath**, которые могут быть вам полезны.

Таблица 5.6. Полезные опции команды multipath

Опция	Описание
-l	Показывает текущую настройку multipath, собранную из **sysfs** и маршрутизатора устройств.

Опция	Описание
-ll	Показывает текущую конфигурацию multipath, собранную из **sysfs**, маршрутизатора устройств и всех иных доступных компонентов в системе.
-f device	Удалить именованное множественное устройство.
-F	Удалить все неиспользуемые множественные устройства.

5.9. Определение меток маршрутизации устройств командой dmsetup

Вы можете использовать команду **dmsetup** для поиска того, какие метки маршрутизаторов устройств соответствуют каким **множественным** устройствам.

Следующая команда показывает все маршрутизаторы устройств и их старшие и младшие номера. Младшие номера определяют имя dm устройства. Например, младший номер **3** соответствует множественному устройству **/dev/dm-3**.

```
# dmsetup ls
mpathd  (253, 4)
mpathep1        (253, 12)
mpathfp1        (253, 11)
mpathb  (253, 3)
mpathgp1        (253, 14)
mpathhp1        (253, 13)
mpatha  (253, 2)
mpathh  (253, 9)
mpathg  (253, 8)
VolGroup00-LogVol01     (253, 1)
mpathf  (253, 7)
VolGroup00-LogVol00     (253, 0)
mpathe  (253, 6)
mpathbp1        (253, 10)
mpathd  (253, 5)
```

5.10. Решение проблем с помощью интерактивной консоли multipathd

Команда **multipathd -k** — это интерактивный интерфейс к сервису **multipathd**. Ввод этой команды запускает интерактивную консоль multipath. После ввода этой команды вы можете ввести help для получения

списка доступных команд, интерактивную команду или нажать **CTRL-D** для выхода.

Интерактивная консоль multipathd может быть использована для решения проблем, которые могут возникнуть на вашей системе. Например, следующая последовательность команд показывает конфигурацию multipath, включая умолчания, до выхода из консоли. Смотрите статью IBM "*Tricks with Multipathd*"[3] для дополнительных примеров.

```
# multipathd -k
  > > show config
  > > CTRL-D
```

Следующая последовательность команд подтверждает что multipath подхватила все изменения в multipath.conf.

```
# multipathd -k
> > reconfigure
> > CTRL-D
```

Используйте следующую последовательность команд, чтобы убедиться, что контроль маршрутов работает правильно.

```
# multipathd -k
> > show paths
> > CTRL-D
```

Команды могут также передаваться через поток stdin в multipathd, как показано ниже:

```
# echo 'show config' | multipathd -k
```

[3] http://www-01.ibm.com/support/docview.wss?uid=isg3T1011985

Глава 6. Удалённое администрирование

Существует много способов удалённого администрирования Linux-сервера. В этой главе рассмотрены три наиболее популярных приложения: OpenSSH, Puppet и Zentyal.

1. Сервер OpenSSH

1.1. Введение

Этот раздел руководства по Ubuntu Server представляет мощную коллекцию инструментов для удалённого управления и обмена данными с сетевыми компьютерами, которая называется *OpenSSH*. Вы также изучите некоторые конфигурационные настройки, доступные для серверного приложения OpenSSH, и то, как изменять их в вашей системе Ubuntu.

OpenSSH — это свободно распространяемая версия семейства инструментов для удалённого управления компьютерами и передачи файлов с использованием протокола Secure Shell (SSH). Традиционные инструменты, используемые для этих целей, такие как telnet и rcp, небезопасны и передают пользовательский пароль открытым текстом. OpenSSH предоставляет серверный демон и клиентские приложения для облегчения операций защиты, зашифрованного удалённого управления и передачи файлов, эффективно заменяя устаревшие инструменты.

Серверная компонента OpenSSH, sshd, постоянно ожидает клиентских соединений от любых клиентских программ. Когда приходит запрос на соединение, sshd устанавливает правильный тип соединения, в зависимости от типа подключаемого клиента. Например, если удалённый компьютер пытается подключиться с помощью клиента ssh, то сервер OpenSSH после аутентификации запустит сеанс удалённого управления. Если же удалённый пользователь подключается с помощью scp, серверный демон OpenSSH после аутентификации организует безопасное копирование файлов между сервером и клиентом. OpenSSH может использовать множество методов аутентификации, включая обычный пароль, использование открытого ключа и сертификаты Kerberos.

1.2. Установка

Установка клиента и сервера OpenSSH проста. Для установки клиента OpenSSH на вашу систему Ubuntu используйте следующую команду в строке терминала:

```
sudo apt-get install openssh-client
```

Для установки сервера OpenSSH и всех необходимых файлов выполните эту команду в строке терминала:

```
sudo apt-get install openssh-server
```

Пакет openssh-server также может быть выбран для установки во время инсталляции Server Edition.

1.3. Конфигурация

Вы можете настроить режим работы по умолчанию серверного приложения OpenSSH, sshd, редактируя файл `/etc/ssh/sshd_config`. Для получения информации о конфигурационных директивах, используемых в этом файле, вы можете просмотреть соответствующее руководство с помощью следующей команды, выполненной в командной строке терминала:

```
man sshd_config
```

Существует множество директив в конфигурационном файле sshd, управляющих такими вещами, как настройки соединений и способы аутентификации. Далее следуют примеры конфигурационных директив, которые могут быть изменены редактированием файла `/etc/ssh/sshd_config`.

Перед внесением изменений в файл настроек вы должны сделать копию оригинального файла и защитить её от записи. Благодаря этому, вы всегда сможете посмотреть оригинальные настройки, а в случае необходимости вы сможете вернуться к этим настройкам.

Создайте копию файла `/etc/ssh/sshd_config` и защитите её от записи, введя в терминале следующие команды:

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.original
sudo chmod a-w /etc/ssh/sshd_config.original
```

Ниже даны примеры конфигурационных директив, которые вы можете изменить:

- Чтобы настроить демона OpenSSH в режим прослушивания TCP порта 2222, вместо стандартного TCP порта 22, измените директиву Port таким образом:

  Port 2222

- Чтобы sshd разрешал использовать процедуру идентификации пользователя с помощью данных, основанных на открытом ключе, просто добавьте или измените следующую строку:

  PubkeyAuthentication yes

  Если строка уже присутствует, убедитесь, что она не закомментирована.

- Чтобы ваш сервер OpenSSH отображал содержимое файла `/etc/issue.net` в качестве сообщения перед логином, просто добавьте или измените следующую строку:

  Banner /etc/issue.net

  В файле `/etc/ssh/sshd_config`.

После внесения изменений в файл `/etc/ssh/sshd_config`, сохраните его и, чтобы изменения вступили в силу, перезапустите серверное приложение sshd, выполнив следующую команду в терминале:

```
sudo service ssh restart
```

Существует множество других директив конфигурации sshd для изменения поведения серверного приложения под ваши нужды. Однако учтите, что если единственный способ доступа к серверу — это ssh, и вы допустили ошибку конфигурации sshd в файле `/etc/ssh/sshd_config`, то ваш сервер может оказаться заблокированным, пока его не перезагрузите. В дополнение, если указана неправильная конфигурационная директива, сервер sshd может отказаться загружаться, поэтому будьте очень осторожны, когда редактируете этот файл на удалённом сервере.

1.4. Ключи SSH

Ключи SSH разрешают аутентификацию пользователей между двумя узлами без необходимости ввода пароля. Аутентификация по ключам SSH использует два ключа: *секретный* и *открытый*.

Чтобы сгенерировать ключи, наберите в приглашении терминала:

```
ssh-keygen -t dsa
```

Эта команда сгенерирует ключи с помощью метода *Digital Signature Algorithm (DSA)*. Во время процесса генерации вам будет предложено ввести пароль. Просто нажмите *Enter* на запрос о создании ключа.

По умолчанию *открытый* ключ сохраняется в файл `~/.ssh/id_dsa.pub`, в то время как *секретный* — в `~/.ssh/id_dsa`. Теперь скопируйте файл `id_dsa.pub` на удалённый компьютер и добавьте его к `~/.ssh/authorized_keys` командой:

```
ssh-copy-id username@remotehost
```

В конце дважды проверьте права доступа к файлу `authorized_keys`. Только аутентифицированный пользователь должен имть права на чтение и запись этого файла. Если права установлены некорректно, измените их:

```
chmod 600 .ssh/authorized_keys
```

Теперь у вас есть возможность соединиться по SSH с этим узлом без ввода пароля.

1.5. Ссылки

- Страница *SSH на Ubuntu Wiki*[1].
- *Сайт OpenSSH*[2]
- *Страница Wiki о расширенных настройках OpenSSH*[3]

[1] https://help.ubuntu.com/community/SSH
[2] http://www.openssh.org/
[3] https://wiki.ubuntu.com/AdvancedOpenSSH

2. Puppet

Puppet — это кроссплатформенный фреймворк, позволяющий системным администраторам выполнять различные задачи путём написания программного кода. Этот код может делать различные вещи: от установки новых программ до проверки прав доступа к файлам или обновления учётных записей пользователей. Puppet полезен не только в процессе первоначальной установки системы, но и в течение всего её жизненного цикла. В большинстве случаев puppet используется в клиент-серверной конфигурации.

Этот раздел посвящён установке и настройке Puppet в конфигурации клиент/сервер. Этот простой пример демонстрирует, как установить Apache с использованием Puppet.

2.1. Preconfiguration

Прежде чем настраивать puppet, вам, возможно, захочется добавить запись DNS *CNAME* для *puppet.example.com*, где *example.com* — это ваш домен. По умолчанию клиенты Puppet проверяют DNS на наличие puppet.example.com в качестве имени сервера puppet (*Puppet Master*). Посетите раздел *Глава 8, Служба доменных имён (DNS) [157]* для дополнительных деталей использования DNS.

Если вы не предполагаете использовать DNS, вы можете добавить записи в файл `/etc/hosts` на сервере и клиенте. Например, в файл `/etc/hosts` сервера Puppet добавьте:

```
127.0.0.1 localhost.localdomain localhost puppet
192.168.1.17 puppetclient.example.com puppetclient
```

На каждом клиенте Puppet добавьте запись для сервера:

```
192.168.1.16 puppetmaster.example.com puppetmaster puppet
```

Замените указанные в приведённом выше примере IP-адреса и доменные имена на реальные адреса и доменные имена вашего сервера и клиента.

2.2. Установка

Для установки Puppet наберите в терминале *сервера*:

```
sudo apt-get install puppetmaster
```

На *клиентском* компьютере (или нескольких компьютерах), введите:

```
sudo apt-get install puppet
```

2.3. Конфигурация

Create a folder path for the apache2 class:

```
  sudo mkdir -p /etc/puppet/modules/apache2/manifests
```

Now setup some resources for apache2. Create a file `/etc/puppet/modules/apache2/manifests/init.pp` containing the following:

```
class apache2 {
  package { 'apache2':
    ensure => installed,
  }

  service { 'apache2':
    ensure  => true,
    enable  => true,
    require => Package['apache2'],
  }
}
```

Next, create a node file `/etc/puppet/manifests/site.pp` with:

```
node 'puppetclient.example.com' {
   include apache2
}
```

Замените *puppetclient.example.com* на ваше реальное имя хоста клиента Puppet.

Финальным шагом для этого простого сервера Puppet является перезапуск демона:

```
sudo service puppetmaster restart
```

Теперь на сервере Puppet всё настроено, и пришло время настроить клиента.

First, configure the Puppet agent daemon to start. Edit `/etc/default/puppet`, changing *START* to yes:

```
START=yes
```

Далее запустите сервис:

```
sudo service puppet start
```

Просмотрите отпечаток сертфиката клиента

```
sudo puppet agent --fingerprint
```

Back on the Puppet server, view pending certificate signing requests:

```
sudo puppet cert list
```

On the Puppet server, verify the fingerprint of the client and sign puppetclient's cert:

```
sudo puppet cert sign puppetclient.example.com
```

On the Puppet client, run the puppet agent manually in the foreground. This step isn't strictly speaking necessary, but it is the best way to test and debug the puppet service.

```
sudo puppet agent --test
```

Check `/var/log/syslog` on both hosts for any errors with the configuration. If all goes well the apache2 package and it's dependencies will be installed on the Puppet client.

Этот пример *очень* простой и не показывает многие возможности и преимущества Puppet. Для дополнительной информации смотрите *Раздел 2.4, «Ресурсы» [99]*.

2.4. Ресурсы

- Посетите сайт *официальной документации Puppet*[4].
- Смотрите *Puppet forge*[5], онлайн-репозиторий модулей puppet.
- Также смотрите *Pro Puppet*[6].

[4] http://docs.puppetlabs.com/
[5] http://forge.puppetlabs.com/
[6] http://www.apress.com/9781430230571

3. Zentyal

Zentyal — это Linux-сервер для малого бизнеса, который может быть сконфигурирован как шлюз (Gateway), инфраструктурный менеджер (Infrastructure Manager), защитный сервер (Unified Threat Manager), офисный сервер (Office Server), коммуникационный сервер (Unified Communication Server) или любое их сочетание. Все сетевые сервисы, управляемые Zentyal, тесно интегрированы, автоматизируя большинство задач. Это помогает избежать ошибок в сетевых настройках и администрировании, и, как следствие, сэкономить время. Zentyal имеет открытый исходный код, опубликованный под лицензией GNU General Public License (GPL) и запускается поверх Ubuntu GNU/Linux.

Zentyal consists of a series of packages (usually one for each module) that provide a web interface to configure the different servers or services. The configuration is stored on a key-value Redis database but users, groups and domains related configuration is on OpenLDAP . When you configure any of the available parameters through the web interface, final configuration files are overwritten using the configuration templates provided by the modules. The main advantages of using Zentyal are: unified, graphical user interface to configure all network services and high, out-of-the-box integration between them.

3.1. Установка

Zentyal 2.3 доступен в Ubuntu 12.04 в репозитории Universe. Доступны следующие модули:

- zentyal-core и zentyal-common: ядро интерфейса Zentyal и общие библиотеки окружения. Также включают модули журналирования и событий, которые обеспечивают администратору интерфейс для просмотра журналов и генерации событий из него.
- zentyal-network: управляет настройкой сети. От интерфейсов (поддерживая статичные IP, DHCP, VLAN, мосты или PPPoE) до множественных шлюзов, когда существует более одного соединения с интернетом, балансировки нагрузки и расширенной маршрутизации, статической маршрутизации или динамического DNS.
- zentyal-objects & zentyal-services: provide an abstraction level for network addresses (e.g. LAN instead of 192.168.1.0/24) and ports named as services (e.g. HTTP instead of 80/TCP).
- zentyal-firewall: настройка правил iptables для блокирования запрещённых соединений, сетевой трансляции адресов (NAT) и перенаправления портов.

- zentyal-ntp: устанавливает сервис NTP, чтобы контролировать время на сервере и позволять клиентам в сети синхронизовать свои часы с серверными.
- zentyal-dhcp: настраивает сервер ISC DHCP, поддерживающий диапазоны, статические выделения и другие расширенные опции, такие как NTP, WINS, обновления динамического DNS и загрузка через сеть с помощью PXE.
- zentyal-dns: добавляет DNS-сервер ISC Bind9 на ваш компьютер для кэширования локальных запросов, работая в качестве перенаправляющего DNS-сервера (DNS forwarder) или доверенного сервера (authoritative server) для настроенных доменов. Позволяет настраивать записи A, CNAME, MX, NS, TXT и SRV.
- zentyal-ca: интегрирует управление центром сертификации в Zentyal таким образом, что пользователи могут использовать сертификаты для аутентификации сервисов, подобно OpenVPN.
- zentyal-openvpn: позволяет настроить несколько VPN серверов и клиентов, использующих OpenVPN с настройкой динамической маршрутизации с помощью Quagga.
- zentyal-users: предоставляет интерфейс настройки и управления пользователями и группами в OpenLDAP. Другие сервисы Zentyal авторизуются по LDAP, имея централизованное управления пользователями и группами. Это также позволяет синхронизировать пользователей, пароли и группы из домена Microsoft Active Directory.
- zentyal-squid: настраивает Squid и Dansguardian для ускорения просмотра благодаря возможностям кэширования и фильтрования контента.
- zentyal-samba: позволяет настраивать Samba и интеграцию с существующим LDAP. Из этого же интерфейса вы можете задавать политику паролей, создавать ресурсы общего доступа и устанавливать права доступа.
- zentyal-printers: интегрирует CUPS с Samba и позволяет не только настраивать принтеры, но и предоставлять им права доступа на основе пользователей и групп LDAP.

Для установки Zentyal в терминале на *сервере* введите следующее (здесь <zentyal-module

```
sudo apt-get install <zentyal-module>
```

Zentyal выпускает по одной стабильной версии в год (в сентябре), в качестве базового дистрибутива для которой используется

последний выпуск Ubuntu LTS. Стабильные выпуски всегда имеют чётное значение младшей части версии (например, 2.2, 3.0), а бета-версии — нечётное (2.1, 2.3). Ubuntu 12.04 поставляется с версией пакетов Zentyal 2.3. Если вы хотите обновиться до последней стабильной версии, опубликованной после выпуска Ubuntu 12.04, используйте *Zentyal Team PPA*[7]. Обновление до новейшей стабильной версии может предоставить вам исправления незначительных ошибок, которые не будут бэкпортироваться в версию 2.3 для Precise, а также добавить новые возможности.

If you need more information on how to add packages from a PPA see *Add a Personal Package Archive (PPA)*[8].

В *Zentyal Team PPA*[9] вы можете найти следующие модули, отсутствующие в репозитории Ubuntu Universe:

- zentyal-antivirus: интегрирует антивирус ClamAV с другими модулями, такими как прокси, общего доступа к файлам и почтового фильтра.
- zentyal-asterisk: настраивает Asterisk для обеспечения работы PBX (Private branch exchange, офисная АТС) на основе LDAP-аутентификации.
- zentyal-bwmonitor: позволяет отслеживать использование пропускной способности вашей локальной сети.
- zentyal-captiveportal: интегрирует captive portal (механизм регистрации доступа в интернет) с защитным сервером (firewall), а также пользователями и группами LDAP.
- zentyal-ebackup: позволяет выполнять резервное копирование по расписанию, используя популярное средство резервного копирования duplicity.
- zentyal-ftp: настраивает FTP-сервер на использование аутентификации по LDAP.
- zentyal-ids: добавляет систему обнаружения сетевых вторжений.
- zentyal-ipsec: позволяет настраивать IPsec туннели с использованием OpenSwan.
- zentyal-jabber: интегрирует XMPP-сервер ejabberd с пользователями и группами LDAP.
- zentyal-thinclients: терминальный сервер (LTSP) для "тонких" клиентов.

[7] https://launchpad.net/~zentyal/
[8] https://help.ubuntu.com/14.04/ubuntu-help/addremove-ppa.html
[9] https://launchpad.net/~zentyal/

- zentyal-mail: полный почтовый стек, включая Postfix и Dovecot с LDAP.
- zentyal-mailfilter: настраивает amavisd на работу с почтовым стеком для фильтрации спама и прикрепленных вирусов.
- zentyal-monitor: добавляет collectd для отслеживания производительности сервера и запущенных сервисов.
- zentyal-pptp: настраивает PPTP VPN сервер.
- zentyal-radius: интегрирует FreeRADIUS с пользователями и группами LDAP.
- zentyal-software: простой интерфейс для управления установленными модулями Zentyal и системными обновлениями.
- zentyal-trafficshaping: настраивает правила ограничения трафика для уменьшения полосы пропускания и уменьшения задержек.
- zentyal-usercorner: разрешает пользователям редактировать их собственные атрибуты LDAP, используя веб-браузер.
- zentyal-virt: простой интерфейс для создания и управления виртуальными машинами на базе libvirt.
- zentyal-webmail: позволяет осуществлять доступ к вашей почте, используя популярный веб-интерфейс Roundcube.
- zentyal-webserver: настраивает интернет сервер Apache для обслуживания различных сайтов на вашей машине.
- zentyal-zarafa: интегрирует средство групповой работы Zarafa с почтовым стеком Zentyal и LDAP.

3.2. Первые шаги

Любой системный пользователь, принадлежащий к группе sudo, имеет возможность войти в веб-интерфейс Zentyal. Если вы используете пользователя, созданного при установке системы, то он входит в группу sudo по умолчанию.

Если вам надо добавить другого пользователя к группе sudo, просто выполните:

```
sudo adduser username sudo
```

Для доступа к веб-интерфейсу Zentyal подключитесь к адресу https://localhost/ (или IP-адресу вашего удалённого сервера). Поскольку Zentyal создаёт собственный самозаверенный сертификат SSL, вы получите предупреждение системы безопасности в вашем браузере.

Подключившись, вы увидите панель управления (dashboard) с обзором всего вашего сервера. Для настройки любого свойства установленного вами модуля, перейдите к нужной секции в меню слева. Когда вы делаете изменения, в правом верхнем углу появляется красная кнопка *Save changes*, которую надо нажать для сохранения всех изменений настроек. Для применения этих изменений на сервере, вначале модуль нужно подключить, что вы можете сделать при выборе *Module Status* в меню слева. Каждый раз как вы включаете модуль, будет появляться всплывающее окно подтверждения о выполнении необходимых действий и изменений на вашем сервере и в файлах настроек.

Если вам требуется настроить под себя какой-либо файл конфигурации или выполнить определенные действия (сценарий или команду) по настройке, не доступной из Zentyal, поместите шаблон конфигурационного файла в Zentyal и указатели (hooks) в /etc/zentyal/hooks/<module>.<action>.

3.3. Ссылки

Официальная страница документации Zentyal[10]

Смотрите также страницу *документации сообщества Zentyal* [11]

И не забудьте посетить *форум* [12]сообщества для поддержки, обратной связи, запросов на доработку и пр.

[10] http://doc.zentyal.org/
[11] http://trac.zentyal.org/wiki/Documentation
[12] http://forum.zentyal.org/

Глава 7. Сетевая аутентификация

В этом разделе рассматривается применение LDAP для аутентификации и авторизации.

1. Сервер OpenLDAP

Lightweight Directory Access Protocol (LDAP) — это протокол запросов и изменений к сервису каталогов на базе X.500, работающий поверх TCP/IP. Текущая версия LDAP — LDAPv3, как определено в *RFC4510*[1], а реализация LDAP в Ubuntu — это OpenLDAP, текущей версии 2.4.25 (Oneiric) (2.4.28 для Precise — прим. переводчика).

Итак, этот протокол обеспечивает доступ к каталогам LDAP. Здесь приведены некоторые ключевые понятия и термины:

- Каталог LDAP — это дерево данных в виде *записей*, иерархичных по своей природе, которое называется деревом каталогов информации (Directory Information Tree, или DIT).
- Запись состоит из набора *атрибутов*.
- Атрибут имеет *тип* (имя/описание) и одно или несколько *значений*.
- Каждый атрибут должен быть определён как минимум в одном *объектном классе (objectClass)*.
- Атрибуты и объектные классы определяются в *схемах* (объектный класс фактически рассматривается как специальный вид атрибута).
- Каждая запись имеет уникальный идентификатор — *отличительное имя* (Distinguished Name, или DN). Оно состоит из *относительного отличительного имени* (RDN), за которым следует запись родительского DN.
- DN записи — это не атрибут. Оно не является частью собственно записи.

Термины *объект*, *контейнер*, and *узел* (node) имеют определенный подтекст, но они все по существу обозначают такую вещь, как запись, технически корректный термин.

Например, далее мы имеем одну запись, содержащую 11 атрибутов. Её DN — это "cn=John Doe,dc=example,dc=com"; её RDN — это "cn=John Doe"; а родительский DN — "dc=example,dc=com".

```
dn: cn=John Doe,dc=example,dc=com
cn: John Doe
givenName: John
sn: Doe
telephoneNumber: +1 888 555 6789
telephoneNumber: +1 888 555 1232
mail: john@example.com
manager: cn=Larry Smith,dc=example,dc=com
```

[1] http://tools.ietf.org/html/rfc4510

```
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: person
objectClass: top
```

Вышеприведённая запись — это формат *LDIF* (LDAP Data Interchange Format, то есть формат обмена данными LDAP). Любая информация, которую вы помещаете в ваш DIT, должна быть в таком формате. Это определено в *RFC2849*[2].

Хотя данное руководство описывает, как использовать его для централизованной идентификации, LDAP хорош для всего, что затрагивает большое количество запросов к системе, основанной на атрибутах (имя:значение) и ориентированной преимущественно на чтение. В качестве примеров можно привести адресную книгу, список адресов электронной почты и конфигурацию почтового сервера.

1.1. Установка

Установите демон сервера OpenLDAP и традиционные утилиты управления LDAP. Они находятся в пакетах slapd и ldap-utils, соответственно.

Установка slapd создаст работающую конфигурацию. В частности, она создаст экземпляр базы данных, которую вы можете использовать для хранения своих данных. Однако суффикс (или базовый DN) этого экземпляра будет определён из доменного имени localhost. Если вы хотите использовать что-то другое, отредактируйте `/etc/hosts` и замените доменное имя на подходящее. Например, если вам нужен суффикс *dc=example,dc=com*, то ваш файл должен иметь подобную строку:

```
127.0.1.1       hostname.example.com hostname
```

Вы можете отменить изменения после установки пакета.

Это руководство будет использовать суффикс базы данных *dc=example,dc=com*.

Приступаем к установке:

```
sudo apt-get install slapd ldap-utils
```

Начиная с Ubuntu 8.10 slapd проектируется так, чтобы настраиваться самостоятельно, выделяя отдельный DIT для этой цели. Это позволяет

[2] http://tools.ietf.org/html/rfc2849

динамически настраивать slapd без необходимости перезапускать сервис. Эта конфигурационная база данных состоит из набора текстовых LDIF-файлов, расположенных в `/etc/ldap/slapd.d`. Этот вариант работы известен под разными названиями: метод slapd-config, RTC-метод (от Real Time Configuration — настройка в реальном времени) или метод cn=config. Вы всё ещё можете использовать традиционный метод плоского файла (slapd.conf), но это не рекомендуется; данная функциональность в конечном счете будет убрана.

В настоящее время Ubuntu использует метод *slapd-config* для настройки slapd, и данное руководство это отражает.

Во время установки вам будет предложено указать учётные данные администратора. Это LDAP-данные для *rootDN* вашего экземпляра базы данных. По умолчанию DN этого пользователя: *cn=admin,dc=example,dc=com*. Также по умолчанию не создается административного пользователя для базы данных slapd-config и вы, следовательно, будете вынуждены использовать внешнюю аутентификацию LDAP для доступа к ней. Мы рассмотрим, как это делается, позднее.

Некоторые классические схемы (cosine, nis, inetorgperson) выпускаются теперь для slapd. Это также включает базовую (core) схему, которая предполагается для любой рабочей схемы.

1.2. Проверка после установки

Процесс установки создаст два DIT. Один для slapd-config и один для ваших данных (dc=example,dc=com). Давайте взглянем:

- Здесь показано, как выглядит дерево (DIT) базы данных slapd-config. Напомним, что эта база основана на LDIF и находится в `/etc/ldap/slapd.d`:

```
/etc/ldap/slapd.d/
/etc/ldap/slapd.d/cn=config
/etc/ldap/slapd.d/cn=config/cn=module{0}.ldif
/etc/ldap/slapd.d/cn=config/cn=schema
/etc/ldap/slapd.d/cn=config/cn=schema/cn={0}core.ldif
/etc/ldap/slapd.d/cn=config/cn=schema/cn={1}cosine.ldif
/etc/ldap/slapd.d/cn=config/cn=schema/cn={2}nis.ldif
/etc/ldap/slapd.d/cn=config/cn=schema/cn={3}inetorgperson.ldif
/etc/ldap/slapd.d/cn=config/cn=schema.ldif
/etc/ldap/slapd.d/cn=config/olcBackend={0}hdb.ldif
/etc/ldap/slapd.d/cn=config/olcDatabase={0}config.ldif
/etc/ldap/slapd.d/cn=config/olcDatabase={-1}frontend.ldif
```

```
/etc/ldap/slapd.d/cn=config/olcDatabase={1}hdb.ldif
/etc/ldap/slapd.d/cn=config.ldif
```

Не редактируйте базу slapd-config напрямую. Вносите изменения через протокол LDAP (утилитами).

- Здесь показано, как выглядит дерево slapd-config через LDAP протокол:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=config dn

dn: cn=config

dn: cn=module{0},cn=config

dn: cn=schema,cn=config

dn: cn={0}core,cn=schema,cn=config

dn: cn={1}cosine,cn=schema,cn=config

dn: cn={2}nis,cn=schema,cn=config

dn: cn={3}inetorgperson,cn=schema,cn=config

dn: olcBackend={0}hdb,cn=config

dn: olcDatabase={-1}frontend,cn=config

dn: olcDatabase={0}config,cn=config

dn: olcDatabase={1}hdb,cn=config
```

Пояснения к записям:

- *cn=config*: глобальные настройки
- *cn=module{0},cn=config*: динамически загружаемый модуль
- *cn=schema,cn=config*: содержит жёстко запрограммированную схему системного уровня
- *cn={0}core,cn=schema,cn=config*: жёстко запрограммированная базовая (core) схема
- *cn={1}cosine,cn=schema,cn=config*: схема cosine
- *cn={2}nis,cn=schema,cn=config*: схема nis
- *cn={3}inetorgperson,cn=schema,cn=config*: схема inetorgperson
- *olcBackend={0}hdb,cn=config*: тип хранилища 'hdb' заднего плана
- *olcDatabase={-1}frontend,cn=config*: база переднего плана, настройка по умолчанию для других баз данных

  - *olcDatabase={0}config,cn=config*: конфигурационная база slapd (cn=config)
  - *olcDatabase={1}hdb,cn=config*: экземпляр вашей базы данных (dc=examle,dc=com)
- А здесь показано как выглядит дерево dc=example,dc=com:

```
ldapsearch -x -LLL -H ldap:/// -b dc=example,dc=com dn

dn: dc=example,dc=com

dn: cn=admin,dc=example,dc=com
```

  Пояснения к записям:

  - *dc=example,dc=com*: базовый уровень вашего дерева (DIT)
  - *cn=admin,dc=example,dc=com*: администратор (rootDN) данного дерева (заполняется в процессе установки пакета)

1.3. Изменение/заполнение вашей базы данных

Давайте введём некоторые данные в нашу базу. Мы добавим следующее:

- узел (node) с названием *People* (для хранения пользователей)
- узел с названием *Groups* (для хранения групп)
- группу с названием *miners*
- пользователя с именем *john*

Создайте следующий LDIF файл и назовите его `add_content.ldif`:

```
dn: ou=People,dc=example,dc=com
objectClass: organizationalUnit
ou: People

dn: ou=Groups,dc=example,dc=com
objectClass: organizationalUnit
ou: Groups

dn: cn=miners,ou=Groups,dc=example,dc=com
objectClass: posixGroup
cn: miners
gidNumber: 5000

dn: uid=john,ou=People,dc=example,dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
```

```
uid: john
sn: Doe
givenName: John
cn: John Doe
displayName: John Doe
uidNumber: 10000
gidNumber: 5000
userPassword: johnldap
gecos: John Doe
loginShell: /bin/bash
homeDirectory: /home/john
```

Важно, чтобы значения uid и gid в вашем каталоге не совпадали с локальными значениями. Используйте диапазон больших чисел, начинающийся, например, с 5000. Установка больших значений uid и gid для ldap также позволяет упростить контроль за тем что могут делать локальные пользователи, а что ldap. Подробнее об этом смотрите далее.

Добавляем данные:

```
ldapadd -x -D cn=admin,dc=example,dc=com -W -f add_content.ldif

Enter LDAP Password: ********
adding new entry "ou=People,dc=example,dc=com"

adding new entry "ou=Groups,dc=example,dc=com"

adding new entry "cn=miners,ou=Groups,dc=example,dc=com"

adding new entry "uid=john,ou=People,dc=example,dc=com"
```

Мы можем проверить что информация добавлена правильно с помощью утилиты ldapsearch:

```
ldapsearch -x -LLL -b dc=example,dc=com 'uid=john' cn gidNumber

dn: uid=john,ou=People,dc=example,dc=com
cn: John Doe
gidNumber: 5000
```

Объяснения ключей команды:

- *-x:* "простое" связывание; не будет использоваться метод SASL по умолчанию
- *-LLL:* отключить вывод посторонней информации
- *uid=john:* — «фильтр» для поиска пользователя john

- *cn gidNumber:* запрос на вывод определенных атрибутов (по умолчанию выводятся все атрибуты)

1.4. Изменение базы данных настройки slapd

Дерево (DIT) slapd-config также может запрашиваться и изменяться. Здесь приведено несколько примеров.

- Используйте ldapmodifyудля добавления индекса (атрибут DbIndex) для вашей {1}hdb,cn=config базы (dc=example,dc=com). Создайте файл с названием `uid_index.ldif` следующего содержания:

```
dn: olcDatabase={1}hdb,cn=config
add: olcDbIndex
olcDbIndex: uid eq,pres,sub
```

  Затем выполните команду:

```
sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f uid_index.ldif

modifying entry "olcDatabase={1}hdb,cn=config"
```

  Вы можете подтвердить изменения следующим способом:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b \ cn=config '(olcDatabase={1}hdb)' olcDbIndex

dn: olcDatabase={1}hdb,cn=config
olcDbIndex: objectClass eq
olcDbIndex: uid eq,pres,sub
```

- Давайте добавим схему. Сначала её нужно преобразовать в формат LDIF. Вы можете найти не преобразованные схемы в добавление к преобразованным в каталоге `/etc/ldap/schema`.

  - Удаление схемы из базы slapd-config — нетривиальная задача. Потренируйтесь добавлять схемы на тестовой системе.
  - Перед добавлением любой схемы вам следует проверить, какие схемы уже установлены (показан вывод по умолчанию, для состояния "из коробки"):

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b \ cn=schema,cn=config dn

dn: cn=schema,cn=config

dn: cn={0}core,cn=schema,cn=config
```

```
dn: cn={1}cosine,cn=schema,cn=config

dn: cn={2}nis,cn=schema,cn=config

dn: cn={3}inetorgperson,cn=schema,cn=config
```

В следующем примере мы добавим схему CORBA.

1. Создайте конфигурационный файл преобразования `schema_convert.conf`, содержащий следующие строки:

```
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/collective.schema
include /etc/ldap/schema/corba.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/duaconf.schema
include /etc/ldap/schema/dyngroup.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/java.schema
include /etc/ldap/schema/misc.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/openldap.schema
include /etc/ldap/schema/ppolicy.schema
include /etc/ldap/schema/ldapns.schema
include /etc/ldap/schema/pmi.schema
```

2. Создайте выходной каталог `ldif_output`.
3. Определите индекс схемы:

```
slapcat -f schema_convert.conf -F ldif_output -n 0 | grep corba,cn=schema

cn={1}corba,cn=schema,cn=config
```

When slapd ingests objects with the same parent DN it will create an *index* for that object. An index is contained within braces: {X}.

4. Используйте slapcat для выполнения преобразования:

```
slapcat -f schema_convert.conf -F ldif_output -n0 -H \ ldap:///cn={1}corba,cn=schema,cn=confi
```

Сконвертированная (преобразованная) схема теперь в `cn=corba.ldif`

5. Редактируйте `cn=corba.ldif` по достижении следующих атрибутов:

```
dn: cn=corba,cn=schema,cn=config
...
cn: corba
```

Также удалите следующие строки в конце:

```
structuralObjectClass: olcSchemaConfig
entryUUID: 52109a02-66ab-1030-8be2-bbf166230478
creatorsName: cn=config
createTimestamp: 20110829165435Z
entryCSN: 20110829165435.935248Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20110829165435Z
```

Значения ваших атрибутов могут быть другими.

6. Наконец, используйте ldapadd для добавления новой схемы к дереву slapd-config:

```
sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f cn\=corba.ldif

adding new entry "cn=corba,cn=schema,cn=config"
```

7. Проверьте текущую загруженную схему:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=schema,cn=config dn

dn: cn=schema,cn=config

dn: cn={0}core,cn=schema,cn=config

dn: cn={1}cosine,cn=schema,cn=config

dn: cn={2}nis,cn=schema,cn=config

dn: cn={3}inetorgperson,cn=schema,cn=config

dn: cn={4}corba,cn=schema,cn=config
```

Для аутентификации с помощью LDAP внешних приложений и клиентов, они должны быть специфически настроены. Обратитесь к соответствующей документации по поводу деталей.

1.5. Ведение журнала

Ведение журнала активности для slapd обязательно, когда осуществляется решение на базе OpenLDAP, поэтому его требуется включить вручную после установки приложения. Иначе только элементарные сообщения будут доступны в журналах. Ведение журналов, как и другие настройки slapd, подключаются через базу данных slapd-config.

OpenLDAP поставляется с несколькими подсистемами (уровнями) журналирования, каждая из которых включает подчиненную (дополнительную). Хороший вариант, который стоит попробовать — это *stats*. Страница *slapd-config*[3] содержит больше информации по иным подсистемам.

Создайте файл `logging.ldif` со следующим содержимым:

```
dn: cn=config
changetype: modify
add: olcLogLevel
olcLogLevel: stats
```

Производим изменения:

```
sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f logging.ldif
```

Это породит значительный объем записи в журнал и вы захотите уменьшить уровень детализации когда ваша система станет боевой. С таким уровнем детализации система журналирования вашего хоста (rsyslog) может отнимать значительное время процессора, а также пропускать сообщения:

```
rsyslogd-2177: imuxsock lost 228 messages from pid 2547 due to rate-limiting
```

Вы можете решить изменить настройки rsyslog. В файл `/etc/rsyslog.conf` поместите следующее:

```
# Disable rate limiting
# (default is 200 messages in 5 seconds; below we make the 5 become 0)
$SystemLogRateLimitInterval 0
```

А затем перезапустите демон rsyslog:

```
sudo service rsyslog restart
```

1.6. Репликация

Сервис LDAP становится всё более и более важным, поскольку большинство сетевых систем начинают зависеть от него. В этом контексте стандартной практикой является встраивание избыточности (высокой доступности) в LDAP для защиты от опустошения, которое сделает сервер неработающим. Это достигается с помощью *репликации LDAP*.

[3] http://manpages.ubuntu.com/manpages/en/man5/slapd-config.5.html

Репликация доступна через механизм *Syncrepl*. Он позволят синхронизировать изменения используя модель *Потребитель - Поставщик*. Специфический вид репликации, который мы будем реализовывать в этом руководстве, является комбинацией следующих режимов: *refreshAndPersist* и *delta-syncrepl*. Это подразумевает что Потребитель передает измененные записи Поставщику, как только они появляются, но при этом посылаются только актуальные изменения, а не все записи.

1.6.1. Настройка Поставщика

Начнем с настройки *Поставщика*.

1. Создайте файл LDIF со следующим содержимым и назовите его `provider_sync.ldif`:

```
# Add indexes to the frontend db.
dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: entryCSN eq
-
add: olcDbIndex
olcDbIndex: entryUUID eq

#Load the syncprov and accesslog modules.
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov
-
add: olcModuleLoad
olcModuleLoad: accesslog

# Accesslog database definitions
dn: olcDatabase={2}hdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {2}hdb
olcDbDirectory: /var/lib/ldap/accesslog
olcSuffix: cn=accesslog
olcRootDN: cn=admin,dc=example,dc=com
olcDbIndex: default eq
olcDbIndex: entryCSN,objectClass,reqEnd,reqResult,reqStart

# Accesslog db syncprov.
dn: olcOverlay=syncprov,olcDatabase={2}hdb,cn=config
changetype: add
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
```

```
olcOverlay: syncprov
olcSpNoPresent: TRUE
olcSpReloadHint: TRUE

# syncrepl Provider for primary db
dn: olcOverlay=syncprov,olcDatabase={1}hdb,cn=config
changetype: add
objectClass: olcOverlayConfig
objectClass: olcSyncProvConfig
olcOverlay: syncprov
olcSpNoPresent: TRUE

# accesslog overlay definitions for primary db
dn: olcOverlay=accesslog,olcDatabase={1}hdb,cn=config
objectClass: olcOverlayConfig
objectClass: olcAccessLogConfig
olcOverlay: accesslog
olcAccessLogDB: cn=accesslog
olcAccessLogOps: writes
olcAccessLogSuccess: TRUE
# scan the accesslog DB every day, and purge entries older than 7 days
olcAccessLogPurge: 07+00:00 01+00:00
```

Замените rootDN в LDIF файле на соответствующий вашему каталогу.

2. Профиль apparmor для slapd нужно будет отрегулировать для расположения базы accesslog. Отредактируйте `/etc/apparmor.d/local/usr.sbin.slapd`, добавив следующее:

```
/var/lib/ldap/accesslog/ r,
/var/lib/ldap/accesslog/** rwk,
```

Создаём каталог, устанавливаем файл настроек базы данных и перезагружаем профиль apparmor:

```
sudo -u openldap mkdir /var/lib/ldap/accesslog
sudo -u openldap cp /var/lib/ldap/DB_CONFIG /var/lib/ldap/accesslog
sudo service apparmor reload
```

3. Добавляем новый контент и, поскольку изменили apparmor, перезапускаем сервис:

```
sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f provider_sync.ldif
sudo service slapd restart
```

Теперь поставщик настроен.

1.6.2. Настройка Потребителя

А теперь настроим *Потребителя*.

1. Установим программное обеспечение как указано в *Раздел 1.1, «Установка» [107]*. Убедитесь, что база slapd-config аналогична базе Поставщика. Особенно проверьте, что одинаковы схемы и суффикс базы.
2. Создайте файл LDIF со следующим содержимым и назовите его `consumer_sync.ldif`:

```
dn: cn=module{0},cn=config
changetype: modify
add: olcModuleLoad
olcModuleLoad: syncprov

dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: entryUUID eq
-
add: olcSyncRepl
olcSyncRepl: rid=0 provider=ldap://ldap01.example.com bindmethod=simple binddn="cn=admin,dc=exa
 credentials=secret searchbase="dc=example,dc=com" logbase="cn=accesslog"
 logfilter="(&(objectClass=auditWriteObject)(reqResult=0))" schemachecking=on
 type=refreshAndPersist retry="60 +" syncdata=accesslog
-
add: olcUpdateRef
olcUpdateRef: ldap://ldap01.example.com
```

   Убедитесь, что следующие атрибуты имеют правильные значения:

   - *provider* (hostname сервера Поставщика — в этом примере — или IP-адрес)
   - *binddn* (DN администратора, которым вы пользуетесь)
   - *credentials* (пароль для DN администратора, который вы используете)
   - *searchbase* (суффикс базы, которую вы используете)
   - *olcUpdateRef* (hostname сервера Поставщика или его IP адрес)
   - *rid* (Replica ID, уникальное трёхзначное число, идентифицирующее данную копию. Каждый Потребитель должен иметь минимум один rid)
3. Добавьте новое содержимое:

```
sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f consumer_sync.ldif
```

Вы сделали это! Две базы (суффикс: dc=example,dc=com) будут теперь синхронизированы.

1.6.3. Тестирование

Как только репликация стартует, вы можете отслеживать ее запустив:

```
ldapsearch -z1 -LLLQY EXTERNAL -H ldapi:/// -s base -b dc=example,dc=com contextCSN

dn: dc=example,dc=com
contextCSN: 20120201193408.178454Z#000000#000#000000
```

как на Поставщике, так и на Потребителе. Как только вывод (`20120201193408.178454Z#000000#000#000000`в примере выше) на обеих машинах совпадет, вы провели репликацию. Каждый раз, как происходят изменения на Поставщике, это значение будет изменяться и должно стать таким же на Поставщике.

Если ваше соединение медленное и/или ваша база LDAP велика, процесс приведения в соответствие *contextCSN* Потребителя и Поставщика может быть протяженным. Но, вы должны знать, что процесс запускается как только *contextCSN* Потребителя неизбежно увеличивается.

Если *contextCSN* Потребителя отсутствует или не совпадает со значением Поставщика, вы должны остановиться и понять причину проблемы перед тем как продолжить. Попробуйте проверить slapd (syslog — системный журнал) и файлы журналов аутентификации Поставщика, чтобы увидеть удачны ли были запросы аутентификации Потребителя и не возвращались ли ошибки в ответ на запросы данных (они будут видны как множество записей ldapsearch).

Чтобы проверить, что всё работает, просто запросите на Потребителе DN из базы:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b dc=example,dc=com dn
```

Вы должны увидеть пользователя 'john' и группу 'miners', также как ноды 'People' и 'Groups'.

1.7. Управление доступом

Управление тем, какой тип доступа пользователей (чтение, запись и пр.) должен быть предоставлен к ресурсам, известно как *контроль доступа*. Используемые для этого директивы называются *списками контроля доступа* (access control lists, или ACL).

Когда мы устанавливали пакет slapd, различные ACL были установлены автоматически. Мы рассмотрим некоторые важные следствия этих умолчаний и, занимаясь этим, мы поймём идею того, как работают ACL и как их настраивать.

Для получения эффективных ACL для запроса LDAP нам надо посмотреть на ACL записи запрашиваемой базы данных также как и на записи специального экземпляра базы данных переднего плана. По умолчанию используются ACL, полученные последним действием, в случае, если они не совпадают с правилами из предыдущего варианта. База данных переднего плана опрашивается во вторую очередь и применяется ACL по первому совпадению среди этих двух источников ACL. Следующие команды покажут соответственно ACL базы hdb ("dc=example,dc=com") и они же из базы переднего плана:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b \ cn=config '(olcDatabase={1}hdb)' olcAccess

dn: olcDatabase={1}hdb,cn=config
olcAccess: {0}to attrs=userPassword,shadowLastChange by self write by anonymous
              auth by dn="cn=admin,dc=example,dc=com" write by * none
olcAccess: {1}to dn.base="" by * read
olcAccess: {2}to * by self write by dn="cn=admin,dc=example,dc=com" write by *
  read
```

rootDN всегда имеет полный доступ к своей базе данных. Добавление их к ACL обеспечивает полную конфигурацию, но при этом становится причиной снижения быстродействия slapd.

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b \ cn=config '(olcDatabase={-1}frontend)' olcAcc

dn: olcDatabase={-1}frontend,cn=config
olcAccess: {0}to * by dn.exact=gidNumber=0+uidNumber=0,cn=peercred,
              cn=external,cn=auth manage by * break
olcAccess: {1}to dn.exact="" by * read
olcAccess: {2}to dn.base="cn=Subschema" by * read
```

Самый первый ACL очень важен:

```
olcAccess: {0}to attrs=userPassword,shadowLastChange by self write by anonymous
              auth by dn="cn=admin,dc=example,dc=com" write by * none
```

Это может быть представлено по-другому для лучшего понимания:

```
to attrs=userPassword
 by self write
 by anonymous auth
 by dn="cn=admin,dc=example,dc=com" write
 by * none

to attrs=shadowLastChange
 by self write
```

```
 by anonymous auth
 by dn="cn=admin,dc=example,dc=com" write
 by * none
```

Этот составной ACL (их два) предписывает следующее:

- Анонимный 'auth' доступ обеспечивается к атрибуту *userPassword* для осуществления изначального соединения. Возможно потребуется counter-intuitively для 'by anonymous auth' даже когда анонимный доступ к DIT не требуется. Как только удаленное соединение установлено, требуется аутентификация (см. следующий пункт).
- Должна пройти аутентификация, поскольку все пользователи имеют доступ на чтение (вследствие 'by self write') к атрибуту *userPassword*.
- Атрибут *userPassword* не доступен для всех других пользователей за исключением rootDN, который имеет полный доступ.
- Для того чтобы пользователи могли менять собственные пароли, используя **passwd** или иные утилиты, атрибут *shadowLastChange* должен быть доступен как только пользователь авторизовался.

Поиск по этому DIT может быть проведен анонимно из-за 'by * read' в данном ACL:

```
to *
 by self write
 by dn="cn=admin,dc=example,dc=com" write
 by * read
```

Если это нежелательно, то вам потребуется изменить набор ACL. Для принуждения к аутентификации в процессе связывающего (bind) запроса в качестве альтернативы (или в комбинации с измененным ACL) вам надо использовать директиву 'olcRequire: authc'.

Как указывалось ранее, для базы slapd-config не создаётся никаких административных пользователей. Однако существует идентификация SASL, которая обеспечивает полный доступ к ней. Она подобна суперпользователю для localhost (root/sudo). Вот она:

```
dn.exact=gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth
```

Следующая команда покажет ACL базы slapd-config:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b \ cn=config '(olcDatabase={0}config)' olcAccess

dn: olcDatabase={0}config,cn=config
olcAccess: {0}to * by dn.exact=gidNumber=0+uidNumber=0,cn=peercred,
              cn=external,cn=auth manage by * break
```

Since this is a SASL identity we need to use a SASL *mechanism* when invoking the LDAP utility in question and we have seen it plenty of times in this guide. It is the EXTERNAL mechanism. See the previous command for an example. Note that:

1. Вы должны использовать *sudo* для идентификации как root, чтобы ACL сработали.
2. Механизм EXTERNAL работает через *IPC* (доменные сокеты UNIX). Это означает, что вы должны использовать *ldapi* формат адресации (URI).

Короткий путь для получения всех ACL выглядит следующим образом:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b \ cn=config '(olcAccess=*)' olcAccess olcSuffix
```

Есть ещё много что сказать по контролю доступа. Смотрите страницу руководства по *slapd.access*[4].

1.8. TLS

Когда происходит аутентификация на OpenLDAP сервере, лучше всего это делать, используя зашифрованную сессию. Это может быть достигнуто использованием транспортного уровня шифрования (TLS).

Здесь мы организуем свой собственный *Центр сертификации* (Certificate Authority — CA) и затем создадим и подпишем сертификат нашего LDAP сервера от имени этого CA. Поскольку slapd скомпилирован с использованием библиотеки gnutls, мы будем использовать для выполнения этих задач утилиту certtool.

1. Установите пакеты gnutls-bin и ssl-cert:

   ```
   sudo apt-get install gnutls-bin ssl-cert
   ```
2. Создайте секретный ключ для центра сертификации

   ```
   sudo sh -c "certtool --generate-privkey > /etc/ssl/private/cakey.pem"
   ```
3. Создаём временный файл `/etc/ssl/ca.info` для определения CA:

   ```
   cn = Example Company
   ca
   cert_signing_key
   ```
4. Создаём самоподписанный сертификат центра:

[4] http://manpages.ubuntu.com/manpages/en/man5/slapd.access.5.html

```
sudo certtool --generate-self-signed \ --load-privkey /etc/ssl/private/cakey.pem \ --template /
```

5. Создайте секретный ключ для сервера:

```
sudo certtool --generate-privkey \ --bits 1024 \ --outfile /etc/ssl/private/ldap01_slapd_key.pe
```

Замените *ldap01* в имени файла на имя вашего сервера (hostname). Имена сертификата и ключа для узла и сервиса, которые будут их использовать, помогут сохранять ясность понимания.

6. Создайте файл `/etc/ssl/ldap01.info`, содержащий:

```
organization = Example Company
cn = ldap01.example.com
tls_www_server
encryption_key
signing_key
expiration_days = 3650
```

Данный сертификат будет действителен 10 лет. Вы можете выбрать другое значение.

7. Создайте сертификат для сервера:

```
sudo certtool --generate-certificate \ --load-privkey /etc/ssl/private/ldap01_slapd_key.pem \ -
```

Создайте файл `certinfo.ldif` со следующим содержимым (подставляйте свои значения, наш пример предполагает использование https://www.cacert.org):

```
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/ldap01_slapd_cert.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/ldap01_slapd_key.pem
```

Используйте команду ldapmodify, чтобы сообщить slapd о работе нашего TLS через базу данных slapd-config:

```
sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f /etc/ssl/certinfo.ldif
```

Вопреки распространённому мнению, вам не обязательно указывать *ldaps:///* в `/etc/default/slapd` чтобы использовать шифрование. Вам достаточно указать:

```
SLAPD_SERVICES="ldap:/// ldapi:///"
```

LDAP поверх TLS/SSL (ldaps:///) осуждается в пользу *StartTLS*. Последний опирается на существующую LDAP сессию (прослушивание TCP порта 389), защищённую TLS/SSL в то время как LDAPS, подобно HTTPS, является другим защищённым-с-самого-начала протоколом, который работает через TCP порт 636.

Сужаем права на владение и доступ:

```
sudo adduser openldap ssl-cert
sudo chgrp ssl-cert /etc/ssl/private/ldap01_slapd_key.pem
sudo chmod g+r /etc/ssl/private/ldap01_slapd_key.pem
sudo chmod o-r /etc/ssl/private/ldap01_slapd_key.pem
```

Перезапустите OpenLDAP:

```
sudo service slapd restart
```

Проверьте журналы вашего хоста (/var/log/syslog), чтобы убедиться, что сервер запущен правильно.

1.9. Репликация и TLS

Если вы настроили репликацию между серверами, существует общая практика шифровать (StartTLS) трафик репликации для исключения прослушивания. Лучше всего использовать шифрование с аутентификацией, как мы делали выше. В этом разделе мы будем основываться на проделанной работе по TLS-аутентификации.

Здесь предполагается, что вы настроили репликацию между Поставщиком и Провайдером в соответствии с *Раздел 1.6, «Репликация» [115]* и настроили TLS для аутентификации на Поставщике, следуя инструкциям *Раздел 1.8, «TLS» [122]*.

Как утверждалось ранее, цель (для нас) репликации — это высокая доступность сервиса LDAP. Поскольку мы имеем TLS для аутентификации на Поставщике, мы нуждаемся в этом и на Потребителе. Однако в дополнение к этому мы хотим зашифровать трафик репликации. Что остается сделать, так это создать ключ и сертификат для Потребителя и затем провести соответствующую настройку. Мы создадим ключ и сертификат на Поставщике для предотвращения создания другого Центра сертификатов, а затем перенесем необходимые данные на Потребителя.

1. На Поставщике:

Создаём промежуточный каталог (который будет использоваться для переноса) и затем секретный ключ Потребителя:

```
mkdir ldap02-ssl
cd ldap02-ssl
sudo certtool --generate-privkey \ --bits 1024 \ --outfile ldap02_slapd_key.pem
```

Создаём информационный файл `ldap02.info` для сервера Потребителя; подставляйте свои соответствующие значения:

```
organization = Example Company
cn = ldap02.example.com
tls_www_server
encryption_key
signing_key
expiration_days = 3650
```

Создаём сертификат Потребителя:

```
sudo certtool --generate-certificate \ --load-privkey ldap02_slapd_key.pem \ --load-ca-certific
```

Получаем копию сертификата CA:

```
cp /etc/ssl/certs/cacert.pem .
```

Всё готово. Теперь переносим каталог `ldap02-ssl` на сервер Потребителя. Здесь мы использовали scp (данные изменяем соответственно):

```
cd ..
scp -r ldap02-ssl user@consumer:
```

2. На Потребителе:

   Настраиваем TLS-аутентификацию:

   ```
   sudo apt-get install ssl-cert
   sudo adduser openldap ssl-cert
   sudo cp ldap02_slapd_cert.pem cacert.pem /etc/ssl/certs
   sudo cp ldap02_slapd_key.pem /etc/ssl/private
   sudo chgrp ssl-cert /etc/ssl/private/ldap02_slapd_key.pem
   sudo chmod g+r /etc/ssl/private/ldap02_slapd_key.pem
   sudo chmod o-r /etc/ssl/private/ldap02_slapd_key.pem
   ```

   Создаём файл `/etc/ssl/certinfo.ldif` со следующим содержимым (исправляйте соответственно):

```
dn: cn=config
add: olcTLSCACertificateFile
olcTLSCACertificateFile: /etc/ssl/certs/cacert.pem
-
add: olcTLSCertificateFile
olcTLSCertificateFile: /etc/ssl/certs/ldap02_slapd_cert.pem
-
add: olcTLSCertificateKeyFile
olcTLSCertificateKeyFile: /etc/ssl/private/ldap02_slapd_key.pem
```

Настраиваем базу slapd-config:

```
sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f certinfo.ldif
```

Настраиваем `/etc/default/slapd` как на Поставщике (SLAPD_SERVICES).

3. На Потребителе:

   Настраиваем TLS для репликации на стороне Потребителя. Изменяем существующий атрибут *olcSyncrepl* присоединяя некоторые TLS опции. Делая это, мы увидим в первый раз как изменять значения атрибутов.

   Создайте файл `consumer_sync_tls.ldif` со следующим содержимым:

```
dn: olcDatabase={1}hdb,cn=config
replace: olcSyncRepl
olcSyncRepl: rid=0 provider=ldap://ldap01.example.com bindmethod=simple
 binddn="cn=admin,dc=example,dc=com" credentials=secret searchbase="dc=example,dc=com"
 logbase="cn=accesslog" logfilter="(&(objectClass=auditWriteObject)(reqResult=0))"
 schemachecking=on type=refreshAndPersist retry="60 +" syncdata=accesslog
 starttls=critical tls_reqcert=demand
```

   Дополнительные опции определяют, соответственно, что Потребитель должен использовать StartTLS и что CA сертификат требуется для идентификации Поставщика. Также обратите внимание на LDIF синтаксис для изменения значений атрибута ('replace').

   Применяем эти изменения:

```
sudo ldapmodify -Y EXTERNAL -H ldapi:/// -f consumer_sync_tls.ldif
```

   И перезапустите slapd:

```
sudo service slapd restart
```

4. На Поставщике:

Проверяем, что TLS сессия устанавливается. В `/var/log/syslog`, предполагая что вы настроили уровень журналирования 'conns', вы сможете увидеть подобные записи:

```
slapd[3620]: conn=1047 fd=20 ACCEPT from IP=10.153.107.229:57922 (IP=0.0.0.0:389)
slapd[3620]: conn=1047 op=0 EXT oid=1.3.6.1.4.1.1466.20037
slapd[3620]: conn=1047 op=0 STARTTLS
slapd[3620]: conn=1047 op=0 RESULT oid= err=0 text=
slapd[3620]: conn=1047 fd=20 TLS established tls_ssf=128 ssf=128
slapd[3620]: conn=1047 op=1 BIND dn="cn=admin,dc=example,dc=com" method=128
slapd[3620]: conn=1047 op=1 BIND dn="cn=admin,dc=example,dc=com" mech=SIMPLE ssf=0
slapd[3620]: conn=1047 op=1 RESULT tag=97 err=0 text
```

1.10. Установление подлинности через LDAP

Теперь, когда у вас есть работающий LDAP-сервер, нужно будет установить на клиент библиотеки, которые знают как и когда связываться с ним. В Ubuntu это традиционно достигается установкой пакета libnss-ldap. Этот пакет установит и другие инструменты, которые помогут вам на этапе конфигурации. Установим этот пакет сейчас:

```
sudo apt-get install libnss-ldap
```

У вас будут запрошены подробности по вашему LDAP серверу. Если вы сделаете ошибку, вы можете попробовать снова, используя:

```
sudo dpkg-reconfigure ldap-auth-config
```

Результат диалога можно увидеть в `/etc/ldap.conf`. Если ваш сервер требует опции, недоступные в меню, редактируйте этот файл самостоятельно.

Теперь настраиваем LDAP профиль для NSS:

```
sudo auth-client-config -t nss -p lac_ldap
```

Настраиваем систему на использование LDAP для аутентификации:

```
sudo pam-auth-update
```

Из меню, выберите LDAP и любые другие механизмы аутентификации, которые вам требуются.

Теперь вы имеете возможность входить в систему, используя учётные записи на основе LDAP.

Клиентам LDAP потребуются ссылки на несколько серверов, если используется репликация. В `/etc/ldap.conf` вам надо иметь что-то похожее:

```
uri ldap://ldap01.example.com ldap://ldap02.example.com
```

Запросы имеют таймаут и будет попытка обратиться к Потребителю (ldap02), если Поставщик (ldap01) станет недоступным.

Если вы собираетесь использовать LDAP для хранения пользователей SAMBA, вам потребуется настроить SAMBA сервер на использование LDAP. Смотрите *Раздел 2, «Samba и LDAP» [134]* для подробностей.

Альтернативой пакету libnss-ldap является пакет libnss-ldapd. Однако он добавит в систему пакет nscd, который, возможно, нежелателен. Просто впоследствии удалите его.

1.11. Управление пользователями и группами

Пакет ldap-utils поставляется с достаточным количеством утилит для управления каталогами, но необходимость использовать длинные строки с опциями делает их применение обременительным. Пакет ldapscripts содержит обёрточные сценарии (wrapper scripts) для этих утилит, которые некоторые находят более удобными в использовании.

Установите пакет:

```
sudo apt-get install ldapscripts
```

Затем отредактируйте файл `/etc/ldapscripts/ldapscripts.conf`, чтобы получить что-то наподобие следующего:

```
SERVER=localhost
BINDDN='cn=admin,dc=example,dc=com'
BINDPWDFILE="/etc/ldapscripts/ldapscripts.passwd"
SUFFIX='dc=example,dc=com'
GSUFFIX='ou=Groups'
USUFFIX='ou=People'
MSUFFIX='ou=Computers'
GIDSTART=10000
UIDSTART=10000
MIDSTART=10000
```

Затем редактируем файл `ldapscripts.passwd` для получения нечто похожего на следующее:

```
sudo sh -c "echo -n 'secret' > /etc/ldapscripts/ldapscripts.passwd"
```

```
sudo chmod 400 /etc/ldapscripts/ldapscripts.passwd
```

Замените «secret» на действующий пароль для пользователя rootDN вашей базы.

Теперь этот сценарий готов помочь вам в управлении вашим каталогом. Вот несколько примеров его использования:

- Создание нового пользователя:

  ```
  sudo ldapadduser george example
  ```

  Это создаст пользователя с uid *george* и установит gid *example* в качестве первичной пользовательской группы.

- Изменение пароля пользователя:

  ```
  sudo ldapsetpasswd george
  Changing password for user uid=george,ou=People,dc=example,dc=com
  New Password:
  New Password (verify):
  ```

- Удаление пользователя:

  ```
  sudo ldapdeleteuser george
  ```

- Добавление группы:

  ```
  sudo ldapaddgroup qa
  ```

- Удаление группы:

  ```
  sudo ldapdeletegroup qa
  ```

- Добавление пользователя в группу:

  ```
  sudo ldapadduser george qa
  ```

  Вы теперь можете увидеть атрибут *memberUid* для группы *qa* со значением для *george*.

- Удаление пользователя из группы:

  ```
  sudo ldapdeleteuserfromgroup george qa
  ```

  Атрибут *memberUid* теперь будет удален из группы *qa*.

- Сценарий ldapmodifyuser позволяет добавлять, удалять или заменять пользовательские атрибуты. Сценарий использует тот же синтаксис, что и утилита ldapmodify. Например:

```
sudo ldapmodifyuser george
# About to modify the following entry :
dn: uid=george,ou=People,dc=example,dc=com
objectClass: account
objectClass: posixAccount
cn: george
uid: george
uidNumber: 1001
gidNumber: 1001
homeDirectory: /home/george
loginShell: /bin/bash
gecos: george
description: User account
userPassword:: e1NTSEF9eXFsTFcyWlhwWkF1eGUybVdFWHZKRzJVMjFTSG9vcHk=

# Enter your modifications here, end with CTRL-D.
dn: uid=george,ou=People,dc=example,dc=com
replace: gecos
gecos: George Carlin
```

Поле имени пользователя *gecos* теперь «George Carlin».

- A nice feature of ldapscripts is the template system. Templates allow you to customize the attributes of user, group, and machine objects. For example, to enable the *user* template edit `/etc/ldapscripts/ldapscripts.conf` changing:

```
UTEMPLATE="/etc/ldapscripts/ldapadduser.template"
```

В каталоге `/etc/ldapscripts` имеются *образцы* шаблонов. Скопируйте или переименуйте файл `ldapadduser.template.sample` в `/etc/ldapscripts/ldapadduser.template`:

```
sudo cp /usr/share/doc/ldapscripts/examples/ldapadduser.template.sample \ /etc/ldapscripts/ldapad
```

Отредактируйте новый шаблон для добавления желаемых атрибутов. Следующее создаст новых пользователей с объектным классом inetOrgPerson:

```
dn: uid=<user>,<usuffix>,<suffix>
objectClass: inetOrgPerson
objectClass: posixAccount
cn: <user>
sn: <ask>
uid: <user>
uidNumber: <uid>
gidNumber: <gid>
homeDirectory: <home>
loginShell: <shell>
```

```
gecos: <user>
description: User account
title: Employee
```

Обратите внимание на опцию *<ask>*, использованную для атрибута *sn*. Это заставит ldapadduser запросить у вас его значение.

В пакете имеются утилиты, которые не были рассмотрены здесь. Вот полный список:

ldaprenamemachine[5]
ldapadduser[6]
ldapdeleteuserfromgroup[7]
ldapfinger[8]
ldapid[9]
ldapgid[10]
ldapmodifyuser[11]
ldaprenameuser[12]
lsldap[13]
ldapadddusertogroup[14]
ldapsetpasswd[15]
ldapinit[16]
ldapaddgroup[17]
ldapdeletegroup[18]
ldapmodifygroup[19]
ldapdeletemachine[20]
ldaprenamegroup[21]
ldapaddmachine[22]
ldapmodifymachine[23]
ldapsetprimarygroup[24]
ldapdeleteuser[25]

[5] http://manpages.ubuntu.com/manpages/en/man1/ldaprenamemachine.1.html
[6] http://manpages.ubuntu.com/manpages/en/man1/ldapadduser.1.html
[7] http://manpages.ubuntu.com/manpages/en/man1/ldapdeleteuserfromgroup.1.html
[8] http://manpages.ubuntu.com/manpages/en/man1/ldapfinger.1.html
[9] http://manpages.ubuntu.com/manpages/en/man1/ldapid.1.html
[10] http://manpages.ubuntu.com/manpages/en/man1/ldapgid.1.html
[11] http://manpages.ubuntu.com/manpages/en/man1/ldapmodifyuser.1.html
[12] http://manpages.ubuntu.com/manpages/en/man1/ldaprenameuser.1.html
[13] http://manpages.ubuntu.com/manpages/en/man1/lsldap.1.html
[14] http://manpages.ubuntu.com/manpages/en/man1/ldapadddusertogroup.1.html
[15] http://manpages.ubuntu.com/manpages/en/man1/ldapsetpasswd.1.html
[16] http://manpages.ubuntu.com/manpages/en/man1/ldapinit.1.html
[17] http://manpages.ubuntu.com/manpages/en/man1/ldapaddgroup.1.html
[18] http://manpages.ubuntu.com/manpages/en/man1/ldapdeletegroup.1.html
[19] http://manpages.ubuntu.com/manpages/en/man1/ldapmodifygroup.1.html
[20] http://manpages.ubuntu.com/manpages/en/man1/ldapdeletemachine.1.html
[21] http://manpages.ubuntu.com/manpages/en/man1/ldaprenamegroup.1.html
[22] http://manpages.ubuntu.com/manpages/en/man1/ldapaddmachine.1.html
[23] http://manpages.ubuntu.com/manpages/en/man1/ldapmodifymachine.1.html
[24] http://manpages.ubuntu.com/manpages/en/man1/ldapsetprimarygroup.1.html
[25] http://manpages.ubuntu.com/manpages/en/man1/ldapdeleteuser.1.html

1.12. Резервное копирование и восстановление

Есть утилиты из пакета, которые здесь не рассматривались. Вот их полный список:

Что нам требуется, это способ сделать резервные копии для базы данных ldap, специфичные для данных баз заднего (cn=config) и переднего плана (dc=example,dc=com). Если мы собираемся сохранить эти базы, скажем, в `/export/backup`, мы можем использовать slapcat как показано в следующем сценарии с именем `/usr/local/bin/ldapbackup`:

```
#!/bin/bash

BACKUP_PATH=/export/backup
SLAPCAT=/usr/sbin/slapcat

nice ${SLAPCAT} -n 0 > ${BACKUP_PATH}/config.ldif
nice ${SLAPCAT} -n 1 > ${BACKUP_PATH}/example.com.ldif
nice ${SLAPCAT} -n 2 > ${BACKUP_PATH}/access.ldif
chmod 640 ${BACKUP_PATH}/*.ldif
```

Это несжатые текстовые файлы, содержащие все данные из вашей ldap базы, включая расположение дерева, имена пользователей и каждый пароль. Поэтому вы можете решить сделать `/export/backup` шифрованным разделом и даже иметь сценарии шифрования этих файлов сразу после создания. В идеале вы можете сделать и то и другое, но это зависит от ваших требований безопасности.

Затем имеет смысл создать сценарии cron для запуска этой программы настолько часто, насколько вам будет комфортно. Для большинства достаточно одного раза в день. Для некоторых требуется чаще. Здесь пример сценария cron, названного `/etc/cron.d/ldapbackup`, который срабатывает каждую ночь в 22:45:

```
MAILTO=backup-emails@domain.com
45 22 * * *  root    /usr/local/bin/ldapbackup
```

Теперь файлы созданы и они могут быть скопированы на резервный сервер.

Предположим мы сделали переустановку ldap; процесс восстановления будет подобен следующему: sudo service slapd stop

```
sudo service slapd stop
sudo mkdir /var/lib/ldap/accesslog
sudo slapadd -F /etc/ldap/slapd.d -n 0 -l /export/backup/config.ldif
sudo slapadd -F /etc/ldap/slapd.d -n 1 -l /export/backup/domain.com.ldif
```

```
sudo slapadd -F /etc/ldap/slapd.d -n 2 -l /export/backup/access.ldif
sudo chown -R openldap:openldap /etc/ldap/slapd.d/
sudo chown -R openldap:openldap /var/lib/ldap/
sudo service slapd start
```

1.13. Ресурсы

- Основной ресурс — это документация из апстрима: *www.openldap.org*[26]
- Существует много страниц руководств пакета slapd. Здесь наиболее важные, особенно в плане рассматриваемых в этом руководстве материалов:

  slapd[27]
  slapd-config[28]
  slapd.access[29]
  slapo-syncprov[30]
- Другие man-страницы:

  auth-client-config[31]
  pam-auth-update[32]
- *LDAP for Rocket Scientists*[33] от Zytrax; руководство менее педантичное, но содержащее всесторонне рассмотренный LDAP.
- *OpenLDAP wiki*[34] страница сообщества Ubuntu имеет коллекцию заметок.
- *LDAP System Administration*[35] от O'Reilly (текст, 2003)
- *Mastering OpenLDAP*[36] от Packt (текст, 2007)

[26] http://www.openldap.org/
[27] http://manpages.ubuntu.com/manpages/en/man8/slapd.8.html
[28] http://manpages.ubuntu.com/manpages/en/man5/slapd-config.5.html
[29] http://manpages.ubuntu.com/manpages/en/man5/slapd.access.5.html
[30] http://manpages.ubuntu.com/manpages/en/man5/slapo-syncprov.5.html
[31] http://manpages.ubuntu.com/manpages/en/man8/auth-client-config.8.html
[32] http://manpages.ubuntu.com/manpages/en/man8/pam-auth-update.8.html
[33] http://www.zytrax.com/books/ldap/
[34] https://help.ubuntu.com/community/OpenLDAPServer
[35] http://www.oreilly.com/catalog/ldapsa/
[36] http://www.packtpub.com/OpenLDAP-Developers-Server-Open-Source-Linux/book

2. Samba и LDAP

В этом разделе описана интеграция Samba и LDAP. В этом случае сервер Samba выполняет роль «отдельного» сервера, а LDAP обеспечивает уровень аутентификации с описанием пользователя, группы и информации о пользователе компьютера для нормального функционирования и выполнения своих ролей (из 3 возможных). Отправной точкой в этом может служить сервер OpenLDAP с определённой директорией, которая принимает запросы аутентификации. Подробнее эта процедура описана в предыдущей главе *Раздел 1, «Сервер OpenLDAP» [106]*. После прочтения этой главы, вы должны решить для себя что вы хотите от вашего сервера Samba, а затем настроить его относительно ваших потребностей.

2.1. Установка программного обеспечения

Для интеграции Samba с LDAP необходимы три пакета: samba, samba-doc и smbldap-tools.

Строго говоря, пакет smbldap-tools не обязателен, но если у вас нет других способов управления различными сущностями Samba (пользователями, группами, компьютерами) в контексте LDAP, то вам следует установить его.

Установите эти пакеты сейчас:

```
sudo apt-get install samba samba-doc smbldap-tools
```

2.2. Конфигурация LDAP

Теперь настроим LDAP-сервер, чтобы он мог хранить данные Samba. Для этого нам необходимо выполнить три пункта:

1. Импортировать схему
2. Индексировать записи
3. Добавить объекты

2.2.1. Схема Samba

Для того чтобы OpenLDAP использовался как дополнение к Samba, теоретически в дереве (DIT) должны присутствовать атрибуты, которые корректно описывают данные Samba. Такие атрибуты могут быть получены путём введения схемы Samba в LDAP. Сейчас мы это сделаем.

Для более детальной информации о схемах и их установке смотрите *Раздел 1.4, «Изменение базы данных настройки slapd» [112]*.

1. Такая схема находится в свежеустановленном вами пакете samba-doc. Её требуется скопировать и разархивировать в каталог `/etc/ldap/schema`:

```
sudo cp /usr/share/doc/samba-doc/examples/LDAP/samba.schema.gz /etc/ldap/schema
sudo gzip -d /etc/ldap/schema/samba.schema.gz
```

2. Получаем файл конфигурации `schema_convert.conf`, который должен содержать следующие строки:

```
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/collective.schema
include /etc/ldap/schema/corba.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/duaconf.schema
include /etc/ldap/schema/dyngroup.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/java.schema
include /etc/ldap/schema/misc.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/openldap.schema
include /etc/ldap/schema/ppolicy.schema
include /etc/ldap/schema/ldapns.schema
include /etc/ldap/schema/pmi.schema
include /etc/ldap/schema/samba.schema
```

3. Оставляем каталог `ldif_output` для вывода.
4. Определите индекс схемы:

```
slapcat -f schema_convert.conf -F ldif_output -n 0 | grep samba,cn=schema

dn: cn={14}samba,cn=schema,cn=config
```

5. Конвертируем схему в формат LDIF:

```
slapcat -f schema_convert.conf -F ldif_output -n0 -H \ ldap:///cn={14}samba,cn=schema,cn=config
```

6. Редактируем созданный файл `cn=samba.ldif`, удаляя индексную информацию, по достижению:

```
dn: cn=samba,cn=schema,cn=config
...
cn: samba
```

Удалите нижние строки:

```
structuralObjectClass: olcSchemaConfig
entryUUID: b53b75ca-083f-102d-9fff-2f64fd123c95
creatorsName: cn=config
```

```
createTimestamp: 20080827045234Z
entryCSN: 20080827045234.341425Z#000000#000#000000
modifiersName: cn=config
modifyTimestamp: 20080827045234Z
```

Значения ваших атрибутов могут быть другими.

7. Добавляем новую схему:

```
sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f cn\=samba.ldif
```

Для запроса и просмотра новой схемы введите:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b cn=schema,cn=config 'cn=*samba*'
```

2.2.2. Индексы Samba

Теперь, когда slapd знает о атрибутах Samba, мы можем создать несколько индексов на их основе. Индексация записей является способом повышения производительности, когда клиент осуществляет выборочный поиск в дереве (DIT).

Создайте файл `samba_indices.ldif` со следующим содержимым:

```
dn: olcDatabase={1}hdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: uidNumber eq
olcDbIndex: gidNumber eq
olcDbIndex: loginShell eq
olcDbIndex: uid eq,pres,sub
olcDbIndex: memberUid eq,pres,sub
olcDbIndex: uniqueMember eq,pres
olcDbIndex: sambaSID eq
olcDbIndex: sambaPrimaryGroupSID eq
olcDbIndex: sambaGroupType eq
olcDbIndex: sambaSIDList eq
olcDbIndex: sambaDomainName eq
olcDbIndex: default sub
```

Используйте утилиту ldapmodify для загрузки новых индексов:

```
sudo ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f samba_indices.ldif
```

Если всё настроено правильно, вы увидите новые индексы, используя утилиту ldapsearch:

```
sudo ldapsearch -Q -LLL -Y EXTERNAL -H \ ldapi:/// -b cn=config olcDatabase={1}hdb olcDbIndex
```

2.2.3. Добавление объектов Samba к LDAP

Далее, настройте пакет smbldap-tools, который соответствовал вашей архитектуре. Пакет должен вступить с конфигурационным скриптом помощником (smbldap-config.pl, раньше был configure.pl) который будет задавать вопросы и предлагать необходимые варианты, но есть *bug*[37] ,в котором описано что smbldap-config.pl не устанавливается (но можно найти его в исходном коде; 'apt-get source smbldap-tools').

Чтобы вручную сконфигурировать пакет, нужно создать и отредактировать файлы `/etc/smbldap-tools/smbldap.conf` и `/etc/smbldap-tools/smbldap_bind.conf`.

Сценарий smbldap-populate затем добавит объекты LDAP, необходимые для Samba. Не помешает сначала сделать резервную копию DIT с помощью slapcat:

```
sudo slapcat -l backup.ldif
```

После создания резервной копии, приступите к наполнению каталога:

```
sudo smbldap-populate
```

Вы можете создать файл LDIF, содержащий новые объекты Samba, выполнив **sudo smbldap-populate -e samba.ldif**. Это позволит просматривать изменения, убедившись, что всё работает правильно. Если это так, перезапустите сценарий без опции '-e'. Либо вы можете взять файл LDIF и импортировать его данные в обычном режиме.

Теперь ваш каталог LDAP содержит всю необходимую информацию для аутентификации пользователей Samba.

2.3. Настройка Samba

There are multiple ways to configure Samba. For details on some common configurations see *Глава 18, Samba [324]*. To configure Samba to use LDAP, edit it's configuration file `/etc/samba/smb.conf` commenting out the default *passdb backend* parameter and adding some ldap-related ones:

```
#   passdb backend = tdbsam

# LDAP Settings
   passdb backend = ldapsam:ldap://hostname
   ldap suffix = dc=example,dc=com
```

[37] https://bugs.launchpad.net/serverguide/+bug/997172

```
   ldap user suffix = ou=People
   ldap group suffix = ou=Groups
   ldap machine suffix = ou=Computers
   ldap idmap suffix = ou=Idmap
   ldap admin dn = cn=admin,dc=example,dc=com
   ldap ssl = start tls
   ldap passwd sync = yes
...
   add machine script = sudo /usr/sbin/smbldap-useradd -t 0 -w "%u"
```

Измените значения для вашей конфигурации.

Перезапустите samba, чтобы задействовать новые настройки:

```
sudo restart smbd
sudo restart nmbd
```

Теперь укажите Samba пароль пользователя rootDN (который создан при установке пакета slapd):

```
sudo smbpasswd -w password
```

Если у вас уже есть существующие пользователи LDAP, которых вы хотите включить в вашу конфигурацию Samba, они должны иметь необходимые атрибуты. Утилита smbpasswd подойдёт для этого наилучшим образом (ваш компьютер должен иметь возможность видеть (нумеровать) этих пользователей через NSS; или же должны быть установлены и настроены пакеты libnss-ldapd или libnss-ldap):

```
sudo smbpasswd -a username
```

Вам будет предложено ввести пароль. Он будет считаться новым паролем для этого пользователя. Разумным решением будет сделать его таким же, как прежде.

Для настройки пользователей, групп и учётных записей на компьютерах используйте стандартные утилиты, предоставляемые пакетом smbldap-tools. Вот несколько примеров:

- Чтобы добавить нового пользователя:

  ```
  sudo smbldap-useradd -a -P username
  ```

  Опция *-a* добавляет атрибут Samba, а опция *-P* вызывает утилиту smbldap-passwd, после того как пользователь создан, позволяя создать новый пароль для этого пользователя.
- Чтобы удалить пользователя:

```
sudo smbldap-userdel username
```

В этой команде также можно использовать опцию *-r* для удаления домашней директории пользователя.

- Чтобы добавить группу:

```
sudo smbldap-groupadd -a groupname
```

Как и для smbldap-useradd, опция *-a* добавляет атрибуты Samba.

- Чтобы сделать существующего пользователя членом группы:

```
sudo smbldap-groupmod -m username groupname
```

Опция *-m* позволяет добавить сразу несколько пользователей, перечислив их через запятую.

- Чтобы удалить пользователя из группы:

```
sudo smbldap-groupmod -x username groupname
```

- Добавить в Samba учетную запись компьютера:

```
sudo smbldap-useradd -t 0 -w username
```

Замените *username* на имя рабочей станции. Опция *-t 0* создает учетную запись без задержки, в то время как опция *-w* определяет пользователя как учетную запись компьютера. Также обратите внимание, что параметр *add machine script* в `/etc/samba/smb.conf` изменён, чтобы использовался smbldap-useradd.

В пакете smbldap-tools есть пакеты, которые здесь не были рассмотрены. Вот полный список:

smbldap-groupadd[38]
smbldap-groupdel[39]
smbldap-groupmod[40]
smbldap-groupshow[41]
smbldap-passwd[42]
smbldap-populate[43]

[38] http://manpages.ubuntu.com/manpages/en/man8/smbldap-groupadd.8.html
[39] http://manpages.ubuntu.com/manpages/en/man8/smbldap-groupdel.8.html
[40] http://manpages.ubuntu.com/manpages/en/man8/smbldap-groupmod.8.html
[41] http://manpages.ubuntu.com/manpages/en/man8/smbldap-groupshow.8.html
[42] http://manpages.ubuntu.com/manpages/en/man8/smbldap-passwd.8.html
[43] http://manpages.ubuntu.com/manpages/en/man8/smbldap-populate.8.html

smbldap-useradd[44]
smbldap-userdel[45]
smbldap-userinfo[46]
smbldap-userlist[47]
smbldap-usermod[48]
smbldap-usershow[49]

2.4. Ресурсы

- Для более подробной информации об установке и настройке Samba смотрите раздел *Глава 18, Samba [324]* этого руководства по Ubuntu Server.
- Существует несколько мест, где документированы LDAP и Samba в апстриме *Samba HOWTO Collection*[50].
- Относительно предыдущей ссылки, смотрите отдельно *passdb section*[51].
- Хотя он и устарел (2007 год), ресурс *Linux Samba-OpenLDAP HOWTO*[52] содержит ценную информацию.
- Основная страница *Samba Ubuntu community documentation*[53] содержит множество ссылок на статьи, которые могут оказаться полезными.

[44] http://manpages.ubuntu.com/manpages/en/man8/smbldap-useradd.8.html
[45] http://manpages.ubuntu.com/manpages/en/man8/smbldap-userdel.8.html
[46] http://manpages.ubuntu.com/manpages/en/man8/smbldap-userinfo.8.html
[47] http://manpages.ubuntu.com/manpages/en/man8/smbldap-userlist.8.html
[48] http://manpages.ubuntu.com/manpages/en/man8/smbldap-usermod.8.html
[49] http://manpages.ubuntu.com/manpages/en/man8/smbldap-usershow.8.html
[50] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/
[51] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/passdb.html
[52] http://download.gna.org/smbldap-tools/docs/samba-ldap-howto/
[53] https://help.ubuntu.com/community/Samba#samba-ldap

3. Kerberos

Kerberos — это система сетевой аутентифиации, основанная на принципах доверия третьей стороне. Другие две стороны — это пользователь и сервис, на котором он хочет авторизоваться. Не все сервисы и приложения могут использовать Kerberos, но те, которые могут, приближают сетевое окружение на один шаг к технологии единого входа (Single Sign On — SSO).

В этом разделе рассматривается установка и настройка сервера Kerberos, а также некоторые примеры настройки клиентов.

3.1. Обзор

Этот раздел раскрывает установку и настройку сервера Kerberos, а также некоторые примеры клиентских настроек.

- *Учётная запись (Principal):* любые пользователи, компьютеры или сервисы, предоставляемые серверами, должны быть определены, как учётные записи Kerberos.
- *Требования (Instances):* используются для сервисных и специальных административных учетных записей.
- *Области (Realms):* уникальная область управления, обеспечиваемая установкой Kerberos. Представляйте её себе как домен или группу ваших компьютеров и пользователей, ей принадлежащих. По умолчанию Ubuntu использует имя DNS домена в верхнем регистре (EXAMPLE.COM) в качестве имени области.
- *Центр распространения ключей (KDC):* состоит из трёх частей: базы данных всех учетных записей, сервера аутентификации и сервера предоставления билетов. Для каждой области должен быть хотя бы один KDC.
- *Билет для получения билета (TGT):* изданный сервером аутентификации, TGT зашифровывается на пароле пользователя, который известен только пользователю и KDC.
- *Сервер распространения билетов (TGS):* выпускает сервисные билеты для клиентов по запросу.
- *Билеты (Tickets):* подтверждение идентичности двух учётных записей. Одна учётная запись — пользователь, а другая — сервис, запрашиваемый этим пользователем. Билеты устанавливают секретный ключ, используемый для защищённого соединения во время авторизованной сессии.
- *Файлы ключей (Keytab Files):* файлы, извлечённые из базы учетных записей KDC и содержащие ключ шифрования для сервиса или компьютера.

Чтобы сложить все вместе: область содержит как минимум один KDC, лучше больше для обеспечения безотказности, которые содержат базу данных учётных записей. Когда пользователь под учётной записью заходит на рабочую станцию, которая настроена на Kerberos аутентификацию, KDC выпускает билет для получения билетов (TGT). Если пользователь предоставляет совпадающие параметры, он считается аутентифицированным и может запрашивать билеты для сервисов, поддерживающих Kerberos, на сервере распространения билетов (TGS). Сервисные билеты позволяют пользователю аутентифицироваться на сервисах без ввода имени и пароля.

3.2. Сервер Kerberos

3.2.1. Установка

Протокол Kerberos разработан в Масачусетском технологическом университете (MIT), поэтому полное название протокола MIT Kerberos. (прим. переводчика). Мы создадим домен MIT Kerberos со следующими характеристиками (измените их под свои нужды):

- *Realm:* EXAMPLE.COM
- *Primary KDC:* kdc01.example.com (192.168.0.1)
- *Secondary KDC:* kdc02.example.com (192.168.0.2)
- *Учетная запись пользователя:* steve
- *Учетная запись администратора:* steve/admin

Настоятельно рекомендуется, чтобы ваши авторизованные в сети пользователи имели uid в отдельном диапазоне от ваших локальных пользователей (скажем, начиная с 5000).

Перед установкой сервера Kerberos требуется правильно настроить DNS-сервер для вашего домена. Поскольку область Kerberos по соглашению совпадает с именем домена, этот раздел использует домен *EXAMPLE.COM*, настроенный как Primary Master по документации *Раздел 2.3, «Первичный мастер» [160]*.

Кроме того, Kerberos — протокол, зависимый от времени. Поэтому если локальное время системы на клиентской машине и на сервере отличается более чем на 5 минут (по умолчанию), рабочая станция не будет аутентифицирована. Для решения проблемы все узлы сети должны синхронизировать своё время по одному серверу *Network Time Protocol (NTP)*. Детали настройки NTP смотрите в разделе *Раздел 4, «Синхронизация времени с NTP» [56]*.

Первый шаг по созданию области Kerberos — это установка пакетов krb5-kdc и krb5-admin-server. Введите в терминале:

```
sudo apt-get install krb5-kdc krb5-admin-server
```

В конце установки у вас запросят сетевые имена серверов Kerberos и административного, которые могут быть одним и тем же или разными серверами для определённой области.

По умолчанию область создаётся из доменного имени KDC.

Далее создаём новую область с помощью утилиты kdb5_newrealm:

```
sudo krb5_newrealm
```

3.2.2. Конфигурация

Вопросы, задаваемые в процессе установки, используются для настройки файла `/etc/krb5.conf`. Если вам требуется скорректировать настройки KDC, просто измените файл и перезапустите службу krb5-kdc. Если вам требуется перенастроить Kerberos с самого начала, возможно, для изменения имени области, вы можете это сделать, набрав следующее:

```
sudo dpkg-reconfigure krb5-kdc
```

1. Как только KDC запущен правильно, требуется административный пользователь *учётная запись администратора*. Рекомендуется использовать имя пользователя, отличное от вашего повседневного пользователя. Для использования утилиты kadmin.local наберите в терминале:

   ```
   sudo kadmin.local
   Authenticating as principal root/admin@EXAMPLE.COM with password.
   kadmin.local: addprinc steve/admin
   WARNING: no policy specified for steve/admin@EXAMPLE.COM; defaulting to no policy
   Enter password for principal "steve/admin@EXAMPLE.COM":
   Re-enter password for principal "steve/admin@EXAMPLE.COM":
   Principal "steve/admin@EXAMPLE.COM" created.
   kadmin.local: quit
   ```

   В примере выше *steve — учётная запись*, */admin — требование*, а *@EXAMPLE.COM* — определяет область. "*Ежедневная*" учётная запись, она же *пользовательская — steve@EXAMPLE.COM*; она будет иметь только обычные права пользователя.

Замените *EXAMPLE.COM* и *steve* на ваши имена области и администратора.

2. Далее, новому пользователю-администратору требуется предоставить соответствующие права доступа ACL. Права настраиваются в файле `/etc/krb5kdc/kadm5.acl`:

```
steve/admin@EXAMPLE.COM        *
```

Эта запись предоставляет для *steve/admin* возможность выполнять любые операции над любыми учётными записями в этой области. Вы можете настроить учётные записи более ограниченными правами, которые удобны, если вам требуется учётная запись младшего администратора, которую можно использовать на клиентах Kerberos. Пожалуйста, посмотрите страницу руководства (man) по *kadm5.acl*.

3. Теперь перезапустите krb5-admin-server, чтобы применились новые ACL:

```
sudo service krb5-admin-server restart
```

4. Новая пользовательская учётная запись может быть протестирована утилитой kinit:

```
kinit steve/admin
steve/admin@EXAMPLE.COM's Password:
```

После ввода пароля используйте утилиту klist, чтобы увидеть информацию о билете для получения билетов (TGT):

```
klist
Credentials cache: FILE:/tmp/krb5cc_1000
        Principal: steve/admin@EXAMPLE.COM

  Issued           Expires          Principal
Jul 13 17:53:34  Jul 14 03:53:34  krbtgt/EXAMPLE.COM@EXAMPLE.COM
```

где имя файла кэша `krb5cc_1000` составлено из префикса `krb5cc_` и идентификатора пользователя (UID), который в нашем случае `1000`. У вас может возникнуть необходимость добавить запись в `/etc/hosts` для KDC, чтобы клиент мог его найти. Например:

```
192.168.0.1   kdc01.example.com       kdc01
```

Замените *192.168.0.1* на IP-адрес вашего KDC. Обычно такое требуется, когда ваша область Kerberos охватывает различные сети, разделенные маршрутизаторами.

5. Лучший способ позволить клиентам автоматически определить KDC для области — это использование SRV-записей DNS. Добавьте следующие записи в `/etc/named/db.example.com`:

```
_kerberos._udp.EXAMPLE.COM.     IN SRV 1  0 88  kdc01.example.com.
_kerberos._tcp.EXAMPLE.COM.     IN SRV 1  0 88  kdc01.example.com.
_kerberos._udp.EXAMPLE.COM.     IN SRV 10 0 88  kdc02.example.com.
_kerberos._tcp.EXAMPLE.COM.     IN SRV 10 0 88  kdc02.example.com.
_kerberos-adm._tcp.EXAMPLE.COM. IN SRV 1  0 749 kdc01.example.com.
_kpasswd._udp.EXAMPLE.COM.      IN SRV 1  0 464 kdc01.example.com.
```

Замените *EXAMPLE.COM*, *kdc01*, и*kdc02* на ваши имя домена, первичный и вторичный KDC

Смотрите *Глава 8, Служба доменных имён (DNS) [157]* для детальных инструкций по настройке DNS.

Ваша новая область Kerberos теперь готова аутентифицировать клиентов.

3.3. Вторичный KDC

Когда у вас есть один центр распространения ключей (KDC) в сети, хорошей практикой является создание вторичного KDC на случай, если первичный будет недоступен. Также, если у вас клиенты Kerberos расположены в различных сетях (возможно разделённых маршрутизаторами, использующими NAT), разумно будет поместить вторичные KDC в каждую такую сеть.

1. Сначала установим пакеты и на вопросы о Kerberos и административном серверах введем имя первичного KDC:

```
sudo apt-get install krb5-kdc krb5-admin-server
```

2. Как только пакеты установлены, создайте учетную запись вторичного KDC. Из терминала набираем:

```
kadmin -q "addprinc -randkey host/kdc02.example.com"
```

Впоследствии, при выполнении любых команд kadmin, у вас будет запрашиваться пароль вашей учётной записи *username/admin@EXAMPLE.COM*.

3. Извлекаем файл *keytab*:

```
kadmin -q "ktadd -norandkey -k keytab.kdc02 host/kdc02.example.com"
```

4. Теперь в текущем каталоге появился `keytab.kdc02`, переместите его в `/etc/krb5.keytab`:

```
sudo mv keytab.kdc02 /etc/krb5.keytab
```

Если путь до файла `keytab.kdc02` иной, замените соответственно.

Также вы можете вывести список учётных записей в файл keytab, который может быть полезен при решении проблем, используя утилиту klist:

```
sudo klist -k /etc/krb5.keytab
```

Опция -k показывает, что это keytab файл.

5. Затем на каждом KDC должен быть файл `kpropd.acl`, который содержит список всех KDC в области. В нашем примере на первичном и вторичном KDC создайте `/etc/krb5kdc/kpropd.acl`:

```
host/kdc01.example.com@EXAMPLE.COM
host/kdc02.example.com@EXAMPLE.COM
```

6. Создаём пустую базу данных на *вторичном KDC*:

```
sudo kdb5_util -s create
```

7. Теперь запускаем службу kpropd, которая слушает соединения от утилиты kprop. kprop используется для передачи файлов выгрузки данных:

```
sudo kpropd -S
```

8. Из терминала на *первичном KDC* создаём файл выгрузки для базы данных учетных записей:

```
sudo kdb5_util dump /var/lib/krb5kdc/dump
```

9. Извлекаем *keytab* файл первичного KDC и копируем его в `/etc/krb5.keytab`:

```
kadmin -q "ktadd -k keytab.kdc01 host/kdc01.example.com"
sudo mv keytab.kdc01 /etc/krb5.keytab
```

Убедитесь, что это *host* для *kdc01.example.com*, перед извлечением Keytab.

10. Используя утилиту kprop, загрузите базу данных на вторичный KDC:

```
sudo kprop -r EXAMPLE.COM -f /var/lib/krb5kdc/dump kdc02.example.com
```

Должно вернуться сообщение *SUCCEEDED*, если распространение сработало. Если вернулось сообщение об ошибке, проверьте `/var/log/syslog` на вторичном KDC для дополнительной информации.

Вы можете также создать задачу cron для периодического обновления базы данных на вторичных KDC. Например, следующий код будет выгружать базу данных каждый час (обратите внимание, что длинная строка разделена чтобы соответствовать формату документа):

```
# m h  dom mon dow   command
0 * * * * /usr/sbin/kdb5_util dump /var/lib/krb5kdc/dump &&
/usr/sbin/kprop -r EXAMPLE.COM -f /var/lib/krb5kdc/dump kdc02.example.com
```

11. Вернёмся на *Secondary KDC*, создадим *stash* (stash) файл для хранения Kerberos master key (главного ключа Kerberos):

```
sudo kdb5_util stash
```

12. Под конец запустим сервис krb5-kdc на вторичном KDC:

```
sudo service krb5-kdc start
```

Вторичный KDC теперь должен иметь возможность выдавать билеты для своей области. Вы можете это проверить, остановив службу krb5-kdcна первичном KDC и затем запросив билет с помощью kinitЕсли всё пойдет хорошо, вы получите билет со вторичного KDC. В противном случае проверяйте `/var/log/syslog` и `/var/log/auth.log` на вторичном KDC.

3.4. Клиент Kerberos для Linux

Эта часть освещает настройку клиента Kerberos на системе Linux. Это позволит получить доступ к любому керберезированному сервису, как только пользователь удачно авторизуется в системе.

3.4.1. Установка

Чтобы аутентифицироваться в области Kerberos, требуются пакеты krb5-user и libpam-krb5, а также некоторые другие, которые не являются необходимыми, но делают жизнь проще. Для установки пакетов наберите следующую команду в терминале:

```
sudo apt-get install krb5-user libpam-krb5 libpam-ccreds auth-client-config
```

Пакет auth-client-config позволяет просто настроить PAM для аутентификации множества сервисов, а libpam-ccreds будет кэшировать

параметры аутентификации, позволяя вам подключаться, когда центр распространения ключей (KDC) недоступен. Этот пакет также полезен для переносных компьютеров, которые могут авторизовываться с использованием Kerberos в корпоративной сети, но также должны быть доступны и вне сети.

3.4.2. Конфигурация

Для настройки клиента наберите в терминале:

```
sudo dpkg-reconfigure krb5-config
```

Вас попросят ввести имя области Kerberos. Также, если у вас нет DNS-сервера с настроенными записями Kerberos *SRV*, меню запросит у вас сетевое имя центра распространения ключей (KDC) и административного сервера области.

dpkg-reconfigure добавит записи в файл `/etc/krb5.conf` для вашей области. У вас будут записи, похожие на следующие:

```
[libdefaults]
        default_realm = EXAMPLE.COM
...
[realms]
        EXAMPLE.COM = {
                kdc = 192.168.0.1
                admin_server = 192.168.0.1
        }
```

Если вы установите uid каждого вашего авторизованного в сети пользователя начиная с 5000, как предложено в *Раздел 3.2.1, «Установка» [142]*, вы затем сможете указать pam аутентифицировать с помощью Kerberos только пользователей с uid > 5000:

```
# Kerberos should only be applied to ldap/kerberos users, not local ones. for i in common-a
```

Это поможет избежать запросов (несуществующих) паролей Kerberos для локально аутентифицированных пользователей при смене у них пароля с помощью **passwd**.

Вы можете проверить настройки запросив билет с помощью утилиты kinit. Например:

```
kinit steve@EXAMPLE.COM
Password for steve@EXAMPLE.COM:
```

Когда билет будет предоставлен, детали можно увидеть с помощью klist:

```
klist
Ticket cache: FILE:/tmp/krb5cc_1000
Default principal: steve@EXAMPLE.COM

Valid starting     Expires            Service principal
07/24/08 05:18:56  07/24/08 15:18:56  krbtgt/EXAMPLE.COM@EXAMPLE.COM
        renew until 07/25/08 05:18:57


Kerberos 4 ticket cache: /tmp/tkt1000
klist: You have no tickets cached
```

Затем используйте auth-client-config для настройки модуля libpam-krb5 для запроса билета в процессе входа:

```
sudo auth-client-config -a -p kerberos_example
```

Теперь вы будете получать билет в случае удачной аутентификации на входе.

3.5. Ресурсы

- Для дополнительной информации по версии MIT Kerberos смотрите сайт *MIT Kerberos*[54].
- Страница *Ubuntu Wiki Kerberos*[55] содержит дополнительные подробности.
- *Kerberos: The Definitive Guide*[56] от O'Reilly — великолепное руководство по установке Kerberos.
- Посетите также IRC-каналы *#ubuntu-server* и *#kerberos* на *Freenode*[57], если у вас остались вопросы по Kerberos.

[54] http://web.mit.edu/Kerberos/
[55] https://help.ubuntu.com/community/Kerberos
[56] http://oreilly.com/catalog/9780596004033/
[57] http://freenode.net/

4. Kerberos и LDAP

Большинство людей не используют Kerberos сам по себе; как только пользователь аутентифицировался (Kerberos), нам нужно вычислить что пользователь может делать (авторизация). И это становится задачей таких программ, как LDAP.

Репликация базы данных учётных записей (принципалов) Kerberos между двумя серверами может быть сложной и добавляет в вашу сеть дополнительную базу данных пользователей. К счастью, MIT Kerberos можно сконфигурировать для использования каталога LDAP в качестве базы данных принципалов. В этом разделе рассматривается конфигурирование первичного и вторичного серверов Kerberos для использования OpenLDAP для базы данных принципалов.

Приведенные здесь примеры предполагают использование MIT Kerberos и OpenLDAP.

4.1. Настройка OpenLDAP

В первую очередь, *schema* должен быть загружен на OpenLDAP сервер, который имеет подключения к сети на Первичном и Вторичном KDC. Далее в этом разделе предполагается, что у вас также настроена репликация LDAP, как минимум, между двумя серверами. Для получения информации о настройке OpenLDAP смотрите *Раздел 1, «Сервер OpenLDAP» [106]*.

Необходимо также настроить OpenLDAP для TLS и SSL-соединений, чтобы трафик между KDC и LDAP сервером был в зашифрованном виде. Подробнее смотрите в *Раздел 1.8, «TLS» [122]* .

`cn=admin,cn=config` — пользователь, которого мы создали с правами редактирования базы ldap. Много раз это был RootDN. Измените его значение для соответствия вашим настройкам.

- Для загрузки схемы в LDAP, на сервере LDAP установите пакет krb5-kdc-ldap. В терминале введите:

```
sudo apt-get install krb5-kdc-ldap
```

- Далее распакуйте файл `kerberos.schema.gz`:

```
sudo gzip -d /usr/share/doc/krb5-kdc-ldap/kerberos.schema.gz
sudo cp /usr/share/doc/krb5-kdc-ldap/kerberos.schema /etc/ldap/schema/
```

- Схема *kerberos* должна быть добавлена к дереву *cn=config*. Процедура добавления новой схемы к slapd детально описана в секции *Раздел 1.4, «Изменение базы данных настройки slapd» [112]*.

  1. Сначала создадим файл настроек с именем `schema_convert.conf` или другим значащим именем, содержащим следующие строки:

     ```
     include /etc/ldap/schema/core.schema
     include /etc/ldap/schema/collective.schema
     include /etc/ldap/schema/corba.schema
     include /etc/ldap/schema/cosine.schema
     include /etc/ldap/schema/duaconf.schema
     include /etc/ldap/schema/dyngroup.schema
     include /etc/ldap/schema/inetorgperson.schema
     include /etc/ldap/schema/java.schema
     include /etc/ldap/schema/misc.schema
     include /etc/ldap/schema/nis.schema
     include /etc/ldap/schema/openldap.schema
     include /etc/ldap/schema/ppolicy.schema
     include /etc/ldap/schema/kerberos.schema
     ```

  2. Создадим временный каталог для хранения LDIF файлов:

     ```
     mkdir /tmp/ldif_output
     ```

  3. Теперь используем slapcat для конвертирования файлов схемы:

     ```
     slapcat -f schema_convert.conf -F /tmp/ldif_output -n0 -s \ "cn={12}kerberos,cn=schema,cn=cor
     ```

     Измените имена файла и каталога выше для соответствия вашим именам, если они отличаются.

  4. Отредактируйте созданный файл `/tmp/cn\=kerberos.ldif`, изменив следующие атрибуты:

     ```
     dn: cn=kerberos,cn=schema,cn=config
     ...
     cn: kerberos
     ```

     И удалите следующие строки в конце файла:

     ```
     structuralObjectClass: olcSchemaConfig
     entryUUID: 18ccd010-746b-102d-9fbe-3760cca765dc
     creatorsName: cn=config
     createTimestamp: 20090111203515Z
     entryCSN: 20090111203515.326445Z#000000#000#000000
     modifiersName: cn=config
     modifyTimestamp: 20090111203515Z
     ```

Значения атрибутов могут отличаться, просто убедитесь, что атрибуты удалены.

5. Загрузите новую схему с помощью ldapadd:

```
ldapadd -x -D cn=admin,cn=config -W -f /tmp/cn\=kerberos.ldif
```

6. Добавьте индекс для атрибута *krb5principalname*:

```
ldapmodify -x -D cn=admin,cn=config -W
Enter LDAP Password:
dn: olcDatabase={1}hdb,cn=config
add: olcDbIndex
olcDbIndex: krbPrincipalName eq,pres,sub

modifying entry "olcDatabase={1}hdb,cn=config"
```

7. В конце обновите списки контроля доступа (ACL):

```
ldapmodify -x -D cn=admin,cn=config -W
Enter LDAP Password:
dn: olcDatabase={1}hdb,cn=config
replace: olcAccess
olcAccess: to attrs=userPassword,shadowLastChange,krbPrincipalKey by
 dn="cn=admin,dc=example,dc=com" write by anonymous auth by self write by * none
-
add: olcAccess
olcAccess: to dn.base="" by * read
-
add: olcAccess
olcAccess: to * by dn="cn=admin,dc=example,dc=com" write by * read

modifying entry "olcDatabase={1}hdb,cn=config"
```

Ну вот, теперь ваш каталог LDAP готов обслуживать базу данных учётных записей Kerberos.

4.2. Настройка первичного KDC

С настроенным OpenLDAP самое время настроить KDC.

- Сначала установите необходимые пакеты, набрав в теминале:

```
sudo apt-get install krb5-kdc krb5-admin-server krb5-kdc-ldap
```

- Теперь редактируем `/etc/krb5.conf`, добавив следующие опции в соответствующие секции:

```
[libdefaults]
```

```
        default_realm = EXAMPLE.COM

...

[realms]
        EXAMPLE.COM = {
                kdc = kdc01.example.com
                kdc = kdc02.example.com
                admin_server = kdc01.example.com
                admin_server = kdc02.example.com
                default_domain = example.com
                database_module = openldap_ldapconf
        }

...

[domain_realm]
        .example.com = EXAMPLE.COM


...

[dbdefaults]
        ldap_kerberos_container_dn = dc=example,dc=com

[dbmodules]
        openldap_ldapconf = {
                db_library = kldap
                ldap_kdc_dn = "cn=admin,dc=example,dc=com"

                # this object needs to have read rights on
                # the realm container, principal container and realm sub-trees
                ldap_kadmind_dn = "cn=admin,dc=example,dc=com"

                # this object needs to have read and write rights on
                # the realm container, principal container and realm sub-trees
                ldap_service_password_file = /etc/krb5kdc/service.keyfile
                ldap_servers = ldaps://ldap01.example.com ldaps://ldap02.example.com
                ldap_conns_per_server = 5
        }
```

Замените *example.com*, *dc=example,dc=com*, *cn=admin,dc=example,dc=com*, и *ldap01.example.com* на соответствующие домен, LDAP объект и LDAP сервер вашей сети.

- Далее используем утилиту kdb5_ldap_util для создания области:

```
sudo kdb5_ldap_util -D cn=admin,dc=example,dc=com create -subtrees \ dc=example,dc=com -r EXAMPLE
```

- Создаём тайник для пароля, используемого для подключения к LDAP-серверу. Этот пароль используется опциями *ldap_kdc_dn* и *ldap_kadmin_dn* в `/etc/krb5.conf`:

```
sudo kdb5_ldap_util -D cn=admin,dc=example,dc=com stashsrvpw -f \ /etc/krb5kdc/service.keyfile cn
```

- Копируем сертификат CA из сервера LDAP:

```
scp ldap01:/etc/ssl/certs/cacert.pem .
sudo cp cacert.pem /etc/ssl/certs
```

и редактируем `/etc/ldap/ldap.conf` для использования этого сертификата:

```
TLS_CACERT /etc/ssl/certs/cacert.pem
```

Сертификат также нужно скопировать на вторичный KDC, чтобы позволить соединение с LDAP-серверами с использованием LDAPS.

Вы можете добавить учётные записи Kerberos в базу LDAP, и они будут скопированы на все LDAP-серверы, настроенные на репликацию. Для добавления учётной записи с использованием утилиты kadmin.local введите:

```
sudo kadmin.local
Authenticating as principal root/admin@EXAMPLE.COM with password.
kadmin.local:  addprinc -x dn="uid=steve,ou=people,dc=example,dc=com" steve
WARNING: no policy specified for steve@EXAMPLE.COM; defaulting to no policy
Enter password for principal "steve@EXAMPLE.COM":
Re-enter password for principal "steve@EXAMPLE.COM":
Principal "steve@EXAMPLE.COM" created.
```

Теперь будут добавлены атрибуты krbPrincipalName, krbPrincipalKey, krbLastPwdChange и krbExtraData к объекту пользователя *uid=steve,ou=people,dc=example,dc=com*. Используйте утилитыkinit и klist для проверки, что пользователю действительно выдали билет.

Если объект пользователя уже создан, потребуется опция *-x dn="...*" для добавления атрибутов Kerberos. Иначе будет создан *новый* объект учетной записи в поддереве области.

4.3. Настройка вторичного KDC

Настройка вторичного KDC с использованием LDAP похожа на настройку обычной базы Kerberos.

1. Во-первых, установите необходимые пакеты. В терминале введите:

```
sudo apt-get install krb5-kdc krb5-admin-server krb5-kdc-ldap
```

2. Далее редактируем `/etc/krb5.conf` для использования LDAP:

```
[libdefaults]
        default_realm = EXAMPLE.COM

...

[realms]
        EXAMPLE.COM = {
                kdc = kdc01.example.com
                kdc = kdc02.example.com
                admin_server = kdc01.example.com
                admin_server = kdc02.example.com
                default_domain = example.com
                database_module = openldap_ldapconf
        }

...

[domain_realm]
        .example.com = EXAMPLE.COM

...

[dbdefaults]
        ldap_kerberos_container_dn = dc=example,dc=com

[dbmodules]
        openldap_ldapconf = {
                db_library = kldap
                ldap_kdc_dn = "cn=admin,dc=example,dc=com"

                # this object needs to have read rights on
                # the realm container, principal container and realm sub-trees
                ldap_kadmind_dn = "cn=admin,dc=example,dc=com"

                # this object needs to have read and write rights on
                # the realm container, principal container and realm sub-trees
                ldap_service_password_file = /etc/krb5kdc/service.keyfile
                ldap_servers = ldaps://ldap01.example.com ldaps://ldap02.example.com
                ldap_conns_per_server = 5
        }
```

3. Создаём тайник для пароля соединения с LDAP:

```
sudo kdb5_ldap_util -D cn=admin,dc=example,dc=com stashsrvpw -f \ /etc/krb5kdc/service.keyfile
```

4. Теперь на *первичном KDC* копируем `/etc/krb5kdc/.k5.EXAMPLE.COM` *тайник с главным ключом* на вторичный KDC. Убедитесь, что копирование файла происходит через зашифрованное соединение, такое как scp или через физический носитель.

```
sudo scp /etc/krb5kdc/.k5.EXAMPLE.COM steve@kdc02.example.com:~
sudo mv .k5.EXAMPLE.COM /etc/krb5kdc/
```

Снова замените *EXAMPLE.COM* на вашу актуальную область.

5. Возвращаемся на *Secondary KDC*, чтобы только (пере)запустить ldap сервер:

```
sudo service slapd restart
```

6. И в конце запускаем сервис krb5-kdc:

```
sudo service krb5-kdc start
```

7. Убедитесь, что два LDAP-сервера (и kerberos вдобавок) синхронизированы.

Теперь у вас в сети избыточные KDC, и с избыточными LDAP серверами вы можете продолжать аутентифицировать пользователей, если один LDAP сервер, один Kerberos сервер или один LDAP с одним Kerberos сервером станут недоступны.

4.4. Ресурсы

- *Kerberos Admin Guide*[58] содержит некоторые дополнительные детали.
- For more information on kdb5_ldap_util see *Section 5.6*[59] and the *kdb5_ldap_util man page*[60].
- Another useful link is the *krb5.conf man page*[61].
- Также смотрите *Kerberos and LDAP*[62] на Ubuntu wiki.

[58] http://web.mit.edu/Kerberos/krb5-1.6/krb5-1.6.3/doc/krb5-admin.html#Configuring-Kerberos-with-OpenLDAP-back_002dend
[59] http://web.mit.edu/Kerberos/krb5-1.6/krb5-1.6.3/doc/krb5-admin.html#Global-Operations-on-the-Kerberos-LDAP-Database
[60] http://manpages.ubuntu.com/manpages/trusty/en/man8/kdb5_ldap_util.8.html
[61] http://manpages.ubuntu.com/manpages/trusty/en/man5/krb5.conf.5.html
[62] https://help.ubuntu.com/community/Kerberos#kerberos-ldap

Глава 8. Служба доменных имён (DNS)

Служба доменных имён (Domain Name Service, DNS) — это служба интернета, которая ставит в соответствие друг с другом IP-адреса и полные доменные имена (Fully Qualified Domain Names, FQDN). Таким образом, DNS избавляет от необходимости запоминать IP-адреса. Компьютеры, на которых запущен сервер DNS, называются *серверами имён*. Ubuntu включает в себя BIND (Berkley Internet Naming Daemon), наиболее распространенную программу для обслуживания серверов имён в Linux.

1. Установка

Для установки bind наберите в терминале следующую команду:

```
sudo apt-get install bind9
```

Очень полезный пакет для тестирования и решения проблем с DNS — это пакет dnsutils. Очень часто эти инструменты уже установлены, но для проверки и/или установки dnsutils введите следующее:

```
sudo apt-get install dnsutils
```

2. Конфигурация

Существует много способов настроить BIND9. Наиболее распространенные конфигурации — это кэширующий сервер имён, первичный мастер и вторичный мастер.

- Когда BIND9 настроен как кэширующий сервер, он ищет ответы на запросы имени и запоминает ответ на случай, если запрос придёт повторно.
- В качестве первичного мастера BIND9 читает данные зоны из локального файла и является ответственным за эту зону.
- В качестве вторичного мастера BIND9 получает данные по зоне (целиком) с другого сервера имён, отвечающего за эту зону.

2.1. Обзор

Файлы настройки DNS сохраняются в каталоге `/etc/bind`. Основной файл конфигурации — это `/etc/bind/named.conf`.

Строки *include* определяют имена файлов, которые содержат опции DNS. Строка *directory* в файле `/etc/bind/named.conf.options` сообщает DNS, где искать файлы. Пути ко всем файлам, используемым BIND, будут относительными к этому каталогу.

Файл с именем `/etc/bind/db.root` описывает корневые сервера имён в мире. Сервера со временем меняются, поэтому файл `/etc/bind/db.root` должен время от времени обслуживаться. Обычно это делается через обновлений к пакету bind9. Секция *zone* определяет мастер сервер, и она хранится в файле, определяемом опцией *file*.

Существует возможность настроить один сервер как кэширующий сервер имён, первичный мастер и вторичный мастер одновременно. Сервер может быть началом Authority (Start of Authority, SOA) для одной зоны, при этом предоставляя вторичный сервис для другой, и при всём этом предоставлять кэширующий сервис в локальной сети (LAN).

2.2. Кэширующий сервер имён

По умолчанию конфигурация настраивается на работу в качестве кэширующего сервера. Всё, что для этого требуется — это добавить IP-адреса DNS-серверов вашего интернет-провайдера. Просто раскомментируйте и исправьте следующее в `/etc/bind/named.conf.options`:

```
forwarders {
```

```
        1.2.3.4;
        5.6.7.8;
    };
```

Замените *1.2.3.4* и *5.6.7.8* на актуальные IP-адреса серверов имён.

Теперь перегружаем DNS-сервер для применения новой конфигурации. Наберите в терминале:

```
sudo service bind9 restart
```

Смотрите *Раздел 3.1.2, «dig» [165]* для информации по тестированию кэширующего DNS-сервера.

2.3. Первичный мастер

В этом разделе BIND9 будет настроен как первичный мастер для домена *example.com*. Просто замените *example.com* на ваше FQDN (Fully Qualified Domain Name).

2.3.1. Файл прямой зоны

Чтобы добавить зону DNS в BIND9, превратив его в сервер первичного мастера, первым шагом отредактируем `/etc/bind/named.conf.local`:

```
zone "example.com" {
 type master;
        file "/etc/bind/db.example.com";
};
```

(Обратите внимание, если bind будет получать автоматические обновления в файл `/var/lib/bind/db.example.com`, а не `/etc/bind/db.example.com` как здесь, так и в команде копирования ниже.)

Теперь используем существующий файл зоны в качестве шаблона для создания файла `/etc/bind/db.example.com`:

```
sudo cp /etc/bind/db.local /etc/bind/db.example.com
```

Редактируем новый файл зоны `/etc/bind/db.example.com`, заменив *localhost.* на FQDN нашего сервера, оставляя дополнительную "." в конце. Заменим *127.0.0.1* на IP-адрес сервера имён и *root.localhost* на правильный адрес электронной почты, но с "." вместо символа "@", опять же оставляя "." в конце. Измените комментарий для указания домена, для которого этот файл сделан.

Создайте *запись A* для базового домена *example.com*. Также создайте *запись A* для *ns.example.com* — сервера имён в данном примере:

```
;
; BIND data file for example.com
;
$TTL    604800
@       IN      SOA     example.com. root.example.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
        IN      A       192.168.1.10
;
@       IN      NS      ns.example.com.
@       IN      A       192.168.1.10
@       IN      AAAA    ::1
ns      IN      A       192.168.1.10
```

Вы должны увеличивать *порядковый номер (Serial)* каждый раз, когда делаете изменения в файле зоны. Если вы делаете множественные изменения перед перезапуском BIND9, просто увеличьте Serial на единицу один раз.

Теперь вы можете добавлять DNS записи в конец файла зоны. Смотрите подробности в разделе *Раздел 4.1, «Общие типы записей» [169]*.

Многие администраторы предпочитают использовать дату последнего редактирования в качестве порядкового номера (Serial) зоны в виде *2012010100*, что соответствует формату yyyymmddss (где *ss* — порядковый номер)

Как только вы произвели изменения в файле зоны, требуется перегрузить BIND9 для применения изменений:

```
sudo service bind9 restart
```

2.3.2. Файл обратной зоны

Теперь, когда зона создана и разрешает имена в IP-адреса, требуется создать также *обратную зону*. Обратная зона позволяет DNS определять имя по IP-адресу.

Редактируем /etc/bind/named.conf.local и добавляем следующее:

```
zone "1.168.192.in-addr.arpa" {
```

```
        type master;
        file "/etc/bind/db.192";
};
```

Замените *1.168.192* на первые три октета адресов сети, которую вы используете. Также дайте имя файлу зоны `/etc/bind/db.192` в соответствии с первым октетом вашей сети.

Теперь создаём файл `/etc/bind/db.192`:

```
sudo cp /etc/bind/db.127 /etc/bind/db.192
```

Далее редактируем `/etc/bind/db.192`, изменяя в основном те же опции, что и в `/etc/bind/db.example.com`:

```
;
; BIND reverse data file for local 192.168.1.XXX net
;
$TTL    604800
@       IN      SOA     ns.example.com. root.example.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns.
10      IN      PTR     ns.example.com.
```

*Порядковый номер (Serial)*в обратной зоне также требуется увеличивать при каждом изменении. Для каждой *Записи A*, которую вы настроите в `/etc/bind/db.example.com`, то есть для каждого адреса, вы должны создать *запись PTR* в `/etc/bind/db.192`.

После создания файла обратной зоны перегрузите BIND9:

```
sudo service bind9 restart
```

2.4. Вторичный мастер

Поскольку *первичный мастер (Primary Master)* настроен, требуется *Secondary Master* для того, чтобы поддерживать домен при недоступности первичного мастера.

Для начала на первичном мастере надо разрешить передачу зоны. Добавьте опцию *allow-transfer* к определениям прямой и обратной зон в `/etc/bind/named.conf.local`:

```
zone "example.com" {
        type master;
 file "/etc/bind/db.example.com";
        allow-transfer { 192.168.1.11; };
};

zone "1.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.192";
 allow-transfer { 192.168.1.11; };
};
```

Замените *192.168.1.11* на IP-адрес вашего вторичного сервера имён.

Перезапустим BIND9 на первичном мастере:

```
sudo service bind9 restart
```

Далее, на вторичном мастере установите пакет bind9 так же, как делали на первичном. Затем отредактируем `/etc/bind/named.conf.local` и добавим следующие определения к прямой и обратной зонам:

```
zone "example.com" {
 type slave;
        file "db.example.com";
        masters { 192.168.1.10; };
};

zone "1.168.192.in-addr.arpa" {
 type slave;
        file "db.192";
        masters { 192.168.1.10; };
};
```

Замените *192.168.1.10* на IP-адрес вашего первичного сервера имён.

Перегружаем BIND9 на вторичном мастере:

```
sudo service bind9 restart
```

В `/var/log/syslog` вы сможете увидеть нечто похожее на (некоторые строки разделены для соответствия формату документа):

```
client 192.168.1.10#39448: received notify for zone '1.168.192.in-addr.arpa'
zone 1.168.192.in-addr.arpa/IN: Transfer started.
transfer of '100.18.172.in-addr.arpa/IN' from 192.168.1.10#53:
```

```
 connected using 192.168.1.11#37531
zone 1.168.192.in-addr.arpa/IN: transferred serial 5
transfer of '100.18.172.in-addr.arpa/IN' from 192.168.1.10#53:
 Transfer completed: 1 messages,
6 records, 212 bytes, 0.002 secs (106000 bytes/sec)
zone 1.168.192.in-addr.arpa/IN: sending notifies (serial 5)

client 192.168.1.10#20329: received notify for zone 'example.com'
zone example.com/IN: Transfer started.
transfer of 'example.com/IN' from 192.168.1.10#53: connected using 192.168.1.11#38577
zone example.com/IN: transferred serial 5
transfer of 'example.com/IN' from 192.168.1.10#53: Transfer completed: 1 messages,
8 records, 225 bytes, 0.002 secs (112500 bytes/sec)
```

Обратите внимание, что передача зоны произойдет только если *порядковый номер (Serial)* на первичном сервере больше значения на вторичном. Если вы хотите, чтобы первичный мастер DNS сообщал вторичному DNS серверу об изменении зоны, вы можете добавить *also-notify { ipaddress; };* в `/etc/bind/named.conf.local`, как показано в примере ниже:

```
zone "example.com" {
 type master;
 file "/etc/bind/db.example.com";
 allow-transfer { 192.168.1.11; };
 also-notify { 192.168.1.11; };
 };

zone "1.168.192.in-addr.arpa" {
 type master;
 file "/etc/bind/db.192";
 allow-transfer { 192.168.1.11; };
 also-notify { 192.168.1.11; };
 };
```

Каталог по умолчанию для файлов неавторизованных зон — `/var/cache/bind/`. Этот каталог также настроен в AppArmor для разрешения доступа сервису named на запись в него. Для дополнительной информации по AppArmor смотрите *Раздел 4, «AppArmor» [189]*.

3. Устранение проблем

Этот раздел посвящён способам определения причины проблем, возникающих с DNS и BIND9.

3.1. Тестирование

3.1.1. resolv.conf

The first step in testing BIND9 is to add the nameserver's IP Address to a hosts resolver. The Primary nameserver should be configured as well as another host to double check things. Refer to *Раздел 1.3.1, «Настройка клиента DNS» [43]* for details on adding nameserver addresses to your network clients, and afterwards check that the file `/etc/resolv.conf` contains (for this example):

```
nameserver 192.168.1.10
nameserver 192.168.1.11
```

Nameservers that listen at 127.* are responsible for adding their own IP addresses to resolv.conf (using resolvconf). This is done via the file `/etc/default/bind9` by changing the line RESOLVCONF=no to RESOLVCONF=yes.

Вам надо добавить также IP-адрес вторичного сервера имен на случай недоступности первичного.

3.1.2. dig

Если вы установили пакет dnsutils, вы можете проверить свою установку, используя обзорную утилиту DNS dig:

- После установки BIND9 примените dig к интерфейсу обратной петли (loopback), чтобы убедиться, что порт 53 прослушивается. Из терминала наберите:

  ```
  dig -x 127.0.0.1
  ```

  Вы должны увидеть строки вывода, похожие на следующее:

  ```
  ;; Query time: 1 msec
  ;; SERVER: 192.168.1.10#53(192.168.1.10)
  ```

- Если BIND9 настроен у вас как *кэширующий* сервер, используйте "dig" для замера времени при разрешении имени внешнего домена:

  ```
  dig ubuntu.com
  ```

  Обратите внимание на время в конце вывода результата команды:

```
;; Query time: 49 msec
```

После повторного вызова dig должно произойти улучшение:

```
;; Query time: 1 msec
```

3.1.3. ping

Теперь для демонстрации, как приложения могут использовать DNS для разрешения сетевых имён, используйте утилиту ping для отправки эхо-запроса ICMP. Из терминала наберите следующее:

```
ping example.com
```

Это проверит, может ли сервер имён разрешить имя *ns.example.com* в IP-адрес. Вывод команды будет напоминать следующее:

```
PING ns.example.com (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.800 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=0.813 ms
```

3.1.4. named-checkzone

Хороший способ проверить ваши файлы зон — это использовать утилиту named-checkzone, установленную вместе с пакетом bind9.Эта утилита позволяет вам убедиться в корректности настроек до перезапуска BIND9 и применения изменений.

- Для тестирования нашего файла прямой зоны из примера введите следующее в командной строке:

  ```
  named-checkzone example.com /etc/bind/db.example.com
  ```

  Если всё настроено верно, вы сможете увидеть вывод, похожий на:

  ```
  zone example.com/IN: loaded serial 6
  OK
  ```

- Аналогично, для тестирования файла обратной зоны введите следующее:

  ```
  named-checkzone 1.168.192.in-addr.arpa /etc/bind/db.192
  ```

  Вывод должен напоминать следующее:

  ```
  zone 1.168.192.in-addr.arpa/IN: loaded serial 3
  OK
  ```

Порядковый номер (Serial) вашего файла зоны может отличаться.

3.2. Ведение журнала

BIND9 имеет широкий набор доступных опций настроек журналов. Существуют две основные опции. С помощью опции *channel* указывается, где вести журналы, а опция *category* определяет, какую информацию писать в журнал.

Если опции журналов отсутствуют, по умолчанию применяется следующее:

```
logging {
     category default { default_syslog; default_debug; };
     category unmatched { null; };
};
```

Этот раздел раскрывает, как настроить BIND9 для отправки *отладочных* сообщений, связанных с DNS запросами, в отдельный файл.

- Сначала нам надо настроить канал (channel) для определения того, в какой файл посылать сообщения. Редактируем `/etc/bind/named.conf.local` и добавляем следующее:

```
logging {
    channel query.log {
        file "/var/log/query.log";
        severity debug 3;
    };
};
```

- Затем настраиваем категорию (category) для отправки всех DNS запросов в файл:

```
logging {
    channel query.log {
        file "/var/log/query.log";
        severity debug 3;
    };
    category queries { query.log; };
};
```

Обратите внимание на опцию *debug*, которая может принимать значения от 1 до 3. Если уровень отладки не указан, по умолчанию используется 1.

- Поскольку сервис *named daemon* запускается от имени *bind*, надо создать файл `/var/log/query.log` и сменить его владельца:

```
sudo touch /var/log/query.log
sudo chown bind /var/log/query.log
```

- Перед тем, как сервис named сможет писать в новый файл журнала, нужно изменить профиль AppArmor. Сначала редактируем файл `/etc/apparmor.d/usr.sbin.named`, добавив:

  ```
  /var/log/query.log w,
  ```

  Затем перезагружаем профиль:

  ```
  cat /etc/apparmor.d/usr.sbin.named | sudo apparmor_parser -r
  ```

  Дополнительную информацию по AppArmor смотрите в разделе *Раздел 4, «AppArmor» [189]*
- Теперь перезагружаем BIND9 для применения изменений:

  ```
  sudo service bind9 restart
  ```

Теперь вы можете увидеть файл `/var/log/query.log`, заполненный информацией о запросах. Это простейший пример использования опций журналирования BIND9. По использованию расширенных опций смотрите раздел *Раздел 4.2, «Дополнительная информация» [169]*.

4. Ссылки

4.1. Общие типы записей

Этот раздел покажет некоторые наиболее общие типы записей DNS.

- Запись *A*: Эта запись указывает IP-адрес для сетевого имени (hostname).

```
www     IN    A      192.168.1.12
```

- Запись *CNAME*: Используется для создания псевдонима (alias) записи A. Нельзя создавать запись CNAME, указывающую на другую запись CNAME.

```
web     IN    CNAME  www
```

- Запись *MX*: Используется для определения, куда должна отправляться электронная почта. Должна указывать на запись A, не на CNAME.

```
        IN    MX  1   mail.example.com.
mail    IN    A       192.168.1.13
```

- Запись *NS*: Используется для определения, какие сервера поддерживают копии зоны. Должна указывать на запись A, не на CNAME. Ею определяются первичные и вторичные сервера зоны.

```
        IN    NS     ns.example.com.
        IN    NS     ns2.example.com.
ns      IN    A      192.168.1.10
ns2     IN    A      192.168.1.11
```

4.2. Дополнительная информация

- The *BIND9 Server HOWTO*[1] в Ubuntu Wiki содержит большое количество полезной информации.
- *DNS HOWTO*[2] на The Linux Documentation Project также содержит много информации по настройке BIND9.
- *Bind9.net*[3] содержит ссылки на большую коллекцию ресурсов по DNS и BIND9.
- *DNS and BIND*[4] — популярная книга, вышедшая уже в пятой редакции. Есть также книга *DNS and BIND on IPv6*[5].

[1] https://help.ubuntu.com/community/BIND9ServerHowto
[2] http://www.tldp.org/HOWTO/DNS-HOWTO.html
[3] http://www.bind9.net/
[4] http://shop.oreilly.com/product/9780596100575.do
[5] http://shop.oreilly.com/product/0636920020158.do

- Хорошее место для вопросов поддержки BIND9 и вовлечения в сообщество Ubuntu Server — это канал IRC *#ubuntu-server* на *freenode*[6].

[6] http://freenode.net

Глава 9. Защита

Безопасность всегда следует учитывать во время установки, развёртывания и использования любого вида компьютерных систем. Несмотря на то, что чистая установка Ubuntu весьма безопасна для немедленного использования в Интернете, важно иметь хорошее понимание состояния безопасности ваших систем, на основании того, как они будут использоваться после развёртывания.

This chapter provides an overview of security related topics as they pertain to Ubuntu 14.04 LTS Server Edition, and outlines simple measures you may use to protect your server and network from any number of potential security threats.

1. Управление пользователями

Управление пользователями — это важная часть контроля безопасности системы. Неэффективное управление пользователями и привилегиями часто приводит многие системы к компрометации. Однако, важно, чтобы вы понимали, как можно защитить ваш сервер простыми и эффективными техниками управления пользовательскими учётными записями.

1.1. Где root?

Разработчики Ubuntu сделали сознательное решение по умолчанию отключить административную учётную запись root во всех инсталляциях Ubuntu. Это не означает, что аккаунт root был удалён или что к нему нельзя получить доступ. Он просто получил пароль, соответствующий невозможному значению, так что войти в систему напрямую под root нельзя.

Вместо этого, пользователям для выполнения административных задач предлагается использовать инструмент sudo. Sudo позволяет авторизованному пользователю временно повышать свои привилегии, пользуясь их собственным паролем вместо пароля учётной записи root. Эта простая и эффективная методика предоставляет возможность учёта всех действий пользователей и даёт администратору точечный контроль на тем, какие действия пользователь может выполнять с указанными привилегиями.

- Если по какой-то причине вам хочется включить учётную запись root, просто дайте ей пароль:

  

  Для конфигураций с паролями root поддержка не предоставляется.

  ```
  sudo passwd
  ```

  Sudo запросит ваш пароль, а затем предложит ввести новый пароль для root, как показано ниже:

  ```
  [sudo] password for username: (введите ваш пароль)
  Enter new UNIX password: (введите новый пароль для root)
  Retype new UNIX password: (повторите новый пароль для root)
  passwd: password updated successfully
  ```

- Чтобы отключить пароль учётной записи root, используйте следующий синтаксис passwd:

```
sudo passwd -l root
```

However, to disable the root account itself, use the following command:

```
usermod --expiredate 1
```

- Вам стоит прочитать больше о Sudo, обратившись к его man-странице.

```
man sudo
```

По умолчанию первый пользователь, созданный программой установки Ubuntu, является членом группы "*sudo*", которая добавлена в файл `/etc/sudoers`, как авторизованный пользователь sudo. Если вы хотите предоставить любому другому пользователю полный root-доступ через sudo, просто добавьте его в группу *sudo* group.

1.2. Добавление и удаление пользователей

Процесс управления локальными пользователями и группами совершенно понятный и почти не отличается от большинства других операционных систем семейства GNU/Linux. В Ubuntu и других дистрибутивах, основанных на Debian, рекомендуется использовать для управления учётными записями пакет "adduser".

- Чтобы добавить учётную запись пользователя, используйте следующий синтаксис и следуйте указаниям системы, чтобы назначить учётной записи пароль и указать другие данные, такие как полное имя, номер телефона и прочее.

```
sudo adduser username
```

- Чтобы удалить учётную запись пользователя и его основную группу, используется следующий синтаксис:

```
sudo deluser username
```

Удаление учётной записи не уничтожает соответствующий ей домашний каталог. На вас остаётся решение, удалить ли папку вручную или оставить в соответствии с желаемыми политиками хранения.

Помните, любой пользователь, добавленный позже с такими же UID/GID, как предыдущий владелец, получит доступ к этому каталогу, если вы не примете требуемых мер предосторожности.

Вы можете захотеть заменить значения UID/GID на что-то более подходящее, такое как учётная запись root, и, возможно, даже переместить каталог во избежание будущих конфликтов.

```
sudo chown -R root:root /home/username/
sudo mkdir /home/archived_users/
sudo mv /home/username /home/archived_users/
```

- Чтобы временно заблокировать или разблокировать учётную запись пользователя, используется соответственно следующий синтаксис:

```
sudo passwd -l username
sudo passwd -u username
```

- Чтобы добавить или удалить конкретную группу, используется соответственно следующий синтаксис:

```
sudo addgroup groupname
sudo delgroup groupname
```

- Чтобы добавить пользователя в группу, используется следующее:

```
sudo adduser username groupname
```

1.3. Безопасность пользовательских профилей

Когда создаётся новый пользователь, утилита adduser создаёт новый соответствующий домашний каталог, называющийся `/home/username`. Профиль по умолчанию создаётся на основе содержимого, найденного в папке `/etc/skel`, включающей всё базовое содержимое профиля.

Если ваш сервер будет использоваться многими пользователями, нужно уделить большое внимание правам доступа к домашним каталогам пользователей, чтобы гарантировать конфиденциальность. По умолчанию, пользовательские домашние каталоги в Ubuntu создаются с правами чтения/выполнения для всех. Это означает, что все пользователи могут просматривать и читать содержимое домашних каталогов других пользователей. Это может не подходить для вашего рабочего окружения.

- Чтобы проверить текущие права доступа к домашним каталогам ваших пользователей, используйте следующий синтаксис:

```
ls -ld /home/username
```

Следующий вывод показывает, что у всех есть право доступа к папке `/home/username` на чтение.

```
drwxr-xr-x  2 username username    4096 2007-10-02 20:03 username
```

- Вы можете удалить полномочия чтения для всех, используя следующую команду:

```
sudo chmod 0750 /home/username
```

Некоторые люди необдуманно склоняются к использованию рекурсивной опции (-R), которая модифицирует все дочерние папки и файлы, однако это не требуется и может привести к нежелательным результатам. Изменения прав для родительской папки достаточно для предотвращения неавторизованного доступа ко всему её содержимому.

Гораздо более эффективным подходом к вопросу будет изменение глобальных полномочий по умолчанию на создание пользовательских домашних каталогов для adduser. Просто откройте файл `/etc/adduser.conf` и измените переменную `DIR_MODE` на что-нибудь подходящее, и на все новые домашние каталоги будут устанавливаться корректные права доступа.

```
DIR_MODE=0750
```

- После изменения прав доступа к папке с использованием любого из ранее показанных способов, проверьте результат, используя следующий синтаксис:

```
ls -ld /home/username
```

Результаты ниже показывают, что полномочия на чтение для всех были удалены:

```
drwxr-x---   2 username username    4096 2007-10-02 20:03 username
```

1.4. Политики паролей

Хорошая политика паролей — это один из наиболее важных аспектов состояния безопасности. Для многих успешных взломов против слабых паролей использовался простой брутфорс и перебор по словарю. Если вы планируете предоставить любой вид удалённого доступа с использованием локальной системы паролей, убедитесь, что вы в достаточной мере обдумали требования к минимально сложности пароля, максимальному сроку его жизни и частоте аудита ваших систем аутентификации.

1.4.1. Минимальная длина пароля

По умолчанию Ubuntu требует минимальную длину пароля в 6 символов, также выполняет некоторые базовые проверки энтропии. Эти параметры управляются файлом `/etc/pam.d/common-password` и приведены ниже:

```
password        [success=1 default=ignore]      pam_unix.so obscure sha512
```

Если вы хотите установить минимальную длину в 8 символов, измените соответствующую переменную на min=8. Изменения приведены ниже:

```
password        [success=1 default=ignore]      pam_unix.so obscure sha512 minlen=8
```

Базовые проверки на качество (энтропию) и минимальную длину пароля не применяются к администратору, использующему команды уровня sudo для настройки нового пользователя.

1.4.2. Истечение срока действия пароля

При создании пользовательских учётных записей вам следует сделать политику минимального и максимального срока действия пароля, принуждая пользователей менять их пароли при истечении срока.

- Чтобы просмотреть текущее состояние пользовательской учётной записи, используйте следующий синтаксис:

  sudo chage -l username

  Листинг ниже демонстрирует интересные факты о пользовательской учётной записи, а именно, что не применены никакие политики:

  ```
  Последнее изменение пароля: 20 января 2008 года
  Действие пароля заканчивается: никогда
  Пароль становится неактивным: никогда
  Действие аккаунта заканчивается: никогда
  Изменение пароля возможно, если прошло не менее (указывается количество дней): 0
  Изменение пароля возможно, если прошло не более (указывается количество дней): 99999
  Количество дней, за которое высвечиватся предупреждение об истечении действия пароля: 7
  ```

- Чтобы установить любой из этих параметров, просто воспользуйтесь следующей командой и следуйте указаниям:

  sudo chage username

  Ниже показан пример того, как вы можете вручную изменить дату завершения действия пароля (-E) на 01/31/2008, минимальное время действия пароля (-m) 5 дней, максимальное время действия 90 дней,

период бездействия (-I) 5 дней по окончанию срока действия пароля и период предупреждения (-W) на 14 дней до завершения действия пароля.

```
sudo chage -E 01/31/2011 -m 5 -M 90 -I 30 -W 14 username
```

- Чтобы проверить изменения, воспользуйтесь способом, применявшимся ранее:

```
sudo chage -l username
```

Листинг ниже показывает новые политики, установленные для учётной записи:

```
Последнее изменение пароля: 20 января 2008 года
Действие пароля заканчивается: 19 апреля 2008 года
Пароль становится неактивным: 19 мая 2008 года
Действие аккаунта заканчивается: 31 января 2008 года
Изменение пароля возможно, если прошло не менее (указывается количество дней): 5
Изменение пароля возможно, если прошло не более (указывается количество дней): 90
Количество дней, за которое высвечивается предупреждение об истечении действия пароля: 14
```

1.5. Иные предложения по безопасности

Многие приложения используют собственные механизмы аутентификации, которые могут быть не замечены даже опытными системными администраторами. Поэтому важно понимать и контролировать, как пользователи проходят аутентификацию и получают доступ к сервисам и приложениям на вашем сервере.

1.5.1. Доступ отключенных пользователей по SSH

Простое отключение или блокирование пользовательской учётной записи не помешает пользователю войти на ваш сервер удалённо, если предварительно они установили аутентификацию по открытому ключу RSA. У них всё ещё будет возможность получить скрытый доступ к серверу без потребности в каком бы то ни было пароле. Не забудьте проверить домашний каталог пользователя на файлы, которые разрешают данный вид аутентичного доступа по SSH, как например `/home/username/.ssh/authorized_keys`.

Удалите или переименуйте каталог `.ssh/` в пользовательском домашнем каталоге, чтобы предотвратить дальнейшее использование возможностей аутентификации по SSH.

Обязательно проверьте наличие любых SSH соединений, установленных отключенными пользователями, так как возможно, что они могут иметь

существующее входящее или исходящее подключение. Закройте их, если таковые имеются.

```
who |grep username  (to get the pts/# terminal)
sudo pkill -f pts/#
```

Разрешайте доступ по SSH только тем учётным записям пользователей, которым он нужен. Например, вы можете создать группу "sshlogin" и добавить её в переменную, связанную с переменной `AllowGroups`, расположенной в файле `/etc/ssh/sshd_config`.

```
AllowGroups sshlogin
```

Затем добавьте пользователей, которым разрешён доступ по SSH, в группу "sshlogin" и перезапустите сервис SSH.

```
sudo adduser username sshlogin
sudo service ssh restart
```

1.5.2. Аутентификация базы данных сторонних пользователей

Большинство корпоративных сетей требуют централизованную аутентификацию и контроль доступа ко всем системным ресурсам. Если вы настроили ваш сервер на аутентификацию пользователей с помощью внешних баз данных, убедитесь, что вы отключили пользовательские учётные записи для удалённого и локального использования. Так вы удостоверитесь, что невозможен локальных обход аутентификации.

2. Безопасность консоли

Как и в случае с любым другим барьером безопасности, который вы выстраиваете для защиты вашего сервера, требуется довольно жёстко защититься от невообразимого ущерба, который может возникнуть от физического доступа кого-то лица к вашему оборудованию, например, воровства жёстких дисков, сбоя по питанию, отказа в обслуживании и т.п. Поэтому безопасность консоли стоит рассматривать просто как ещё один компонент вашей общей стратегии физической безопасности. Блокируемая "ширма" (screen door) может защитить от случайного криминала и очень сильно замедлить активное воздействие, поэтому очень желательно соблюдать простейшие предосторожности по отношению к безопасности консоли.

Нижеприведенные инструкции помогут вам обезопасить сервер от таких событий, которые в ином случае могли бы вызвать серьёзные последствия.

2.1. Отключение Ctrl+Alt+Delete

Прежде всего, любой, кто имеет доступ к клавиатуре, просто может нажать комбинацию **Ctrl**+**Alt**+**Delete** для перезагрузки сервера и без необходимости входить в систему. Конечно, можно просто отключить шнур питания, но отключение этой команды на рабочем сервере по-прежнему необходимо. Это заставит злоумышленника предпринимать более радикальные меры по перезагрузке сервера, и в тоже время предотвратит случайные перезагрузки.

- Для отключения перезагрузки по нажатию комбинации **Ctrl**+**Alt**+**Delete** закомментируйте следующую строку в файле `/etc/init/control-alt-delete.conf`.

```
#exec shutdown -r now "Control-Alt-Delete pressed"
```

3. Брандмауэр

3.1. Введение

Ядро Linux включает подсистему *Netfilter*, которая используется для регулирования сетевого траффика, входящего на или проходящего через вашу систему. Все современные средства межсетевой защиты Linux используют эту систему для фильтрации пакетов.

Система фильтрации пакетов ядра была бы малопригодной для администраторов без пользовательского интерфейса управления ею. Для этого предназначено приложение iptables. Когда пакет достигает вашего сервера, он передаётся подсистеме Netfilter для приёма, обработки или отклонения, в зависимости от правил, передаваемых ей из рабочего пространства пользователя с помощью iptables. Таким образом, если вы хорошо знакомы с iptables — это всё, что вам необходимо для управления межсетевым экраном. Однако существует множество программ, предоставляющих интерфейс для упрощения этой задачи.

3.2. ufw — Uncomplicated Firewall

По умолчанию, программой для настройки брандмауэра в Ubuntu является приложение ufw. Разработанное для облегчения настройки iptables, приложение ufw предоставляет дружелюбный пользователю способ по созданию брандмауэра для адресов в формате IPv4 или IPv6, устанавливаемый на машину пользователя.

Приложение ufw по умолчанию отключено. Выдержка из man-страницы ufw:

«Приложение ufw не может предоставить полной функциональности брандмауэра через свой интерфейс командной строки, однако, такое приложение предлагает лёгкий способ добавления или удаления несложных правил. В основном, такое приложение используется для создания брандмауэра, устанавливаемого на компьютере пользователя.»

Ниже идут примеры по использованию приложения ufw:

- Сначала необходимо активировать ufw. Введите в терминале:

```
sudo ufw enable
```

- Для того, чтобы открыть порт (в нашем примере — ssh), введите:

```
sudo ufw allow 22
```

- Также правила могут быть добавлены в *числовом* формате:

```
sudo ufw insert 1 allow 80
```

- Похожим образом, чтобы закрыть порт, введите:

```
sudo ufw deny 22
```

- Чтобы удалить правило, введите delete и, далее, удаляемое правило:

```
sudo ufw delete deny 22
```

- Также возможно разрешить доступ с определённых узлов или сетей на конкретный порт. Нижеследующий пример позволяет доступ к порту ssh с узла 192.168.0.2 на любой IP-адрес на данном компьютере:

```
sudo ufw allow proto tcp from 192.168.0.2 to any port 22
```

  Замените 192.168.0.2 на 192.168.0.0/24, чтобы разрешить доступ к порту ssh из целой подсети.

- Указание опции *--dry-run* у команды *ufw*, приведёт к выводу результирующих правил без их применения. Например, следующее будет применено, если открыт порт HTTP:

```
sudo ufw --dry-run allow http
```

```
*filter
:ufw-user-input - [0:0]
:ufw-user-output - [0:0]
:ufw-user-forward - [0:0]
:ufw-user-limit - [0:0]
:ufw-user-limit-accept - [0:0]
### ПРАВИЛА ###

### tuple ### allow tcp 80 0.0.0.0/0 any 0.0.0.0/0
-A ufw-user-input -p tcp --dport 80 -j ACCEPT

### ОКОНЧАНИЕ ПРАВИЛ ###
-A ufw-user-input -j RETURN
-A ufw-user-output -j RETURN
-A ufw-user-forward -j RETURN
-A ufw-user-limit -m limit --limit 3/minute -j LOG --log-prefix "[UFW LIMIT]: "
-A ufw-user-limit -j REJECT
-A ufw-user-limit-accept -j ACCEPT
COMMIT
Правила изменены
```

- Можно отключить приложение ufw, для этого введите:

```
sudo ufw disable
```

- Введите для просмотра статуса брандмауэра:

```
sudo ufw status
```

- Для более детального отображения статуса введите следующую команду:

```
sudo ufw status verbose
```

- Для просмотра *числового* формата:

```
sudo ufw status numbered
```

Если порт, который вы хотите открыть или закрыть, определён в файле `/etc/services`, вы можете использовать имя порта вместо его номера. Для этого в приведённых выше примерах можно заменить число *22* на *ssh*.

Это краткое введение в использование ufw. Пожалуйста, обратитесь к справочной странице программы ufw для более подробной информации.

3.2.1. Интеграция приложений с ufw

Приложения, открывающие порты, могут включать ufw профиль, который определяет порты, необходимые для корректной работы приложения. Эти профили содержатся в `/etc/ufw/applications.d` и могут быть изменены, если были изменены порты «по умолчанию».

- Для просмотра списка приложений с установленными профилями введите в терминале следующую команду:

```
sudo ufw app list
```

- Аналогично, чтобы разрешить трафик через порт с помощью профиля приложения, введите:

```
sudo ufw allow Samba
```

- Также доступен расширенный синтаксис:

```
ufw allow from 192.168.0.0/24 to any app Samba
```

Замените *Samba* и *192.168.0.0/24* профилем используемого вами приложения и адресом вашей сети соответственно.

Нет необходимости указывать *протокол* для приложения, т.к. эта информация уже содержится в профиле. Также, обратите внимание, что имя приложения — *app* — заменяет номер *порта*.

- Для просмотра деталей по портам, протоколам и т.д. для приложения, введите:

```
sudo ufw app info Samba
```

Не для всех приложений, которые требуют открытие сетевого порта, поставляется профиль ufw, но если у вас есть профиль для приложения, и вы хотите чтобы этот файл был включен в пакет приложения, зарегистрируйте ошибку о пакете на сайте Launchpad.

```
ubuntu-bug имя_пакета
```

3.3. IP маскировка

Назначение IP маскировки в том, чтобы позволить машинам в вашей сети с частными, не маршрутизируемыми IP-адресами, иметь доступ в Интернет через машину, осуществляющую маскировку. Трафик из вашей сети, предназначенный для Интернета, должен быть обработан так, чтобы ответы могли вернуться обратно на машину, которая организовала запрос. Чтобы это сделать, ядро должно изменить IP-адрес *источника* в каждом пакете так, чтобы ответы возвращались на сервер, а не на частный IP-адрес (что невозможно в Интернете), с которого сделан запрос. Linux использует *Connection Tracking* (conntrack) для хранения записи о том, каким машинам принадлежат соединения, и перенаправляет каждый возвращенный пакет соответствующим образом. Таким образом, трафик, покидающий вашу сеть, "замаскирован", как будто исходит от машины, которая выполняет роль шлюза. В документации Microsoft этот процесс упоминается как технология Internet Connection Sharing.

3.3.1. Маскарадинг ufw

Маскарадинг IP может быть реализован использованием пользовательских правил ufw. Это возможно благодаря тому, что текущий бэк-энд для ufw — это iptables-restore с файлами правил, хранящихся в `/etc/ufw/*.rules`. Эти файлы — замечательный способ добавить родные правила iptables без участия ufw, и эти правила больше ориентированы на шлюз или мост сети.

Правила разделены на два файла: те, которые должны быть выполнены до правил командной строки ufw, и те, которые выполняются после правил командной строки ufw .

- В начале, перенаправление пакетов должно быть включено в ufw. Для этого необходимо настроить два конфигурационных файла: в `/etc/default/ufw` измените *DEFAULT_FORWARD_POLICY* на «ACCEPT»:

```
DEFAULT_FORWARD_POLICY="ACCEPT"
```

  После этого отредактируйте `/etc/ufw/sysctl.conf` и раскомментируйте:

```
net/ipv4/ip_forward=1
```

  Аналогично, для форвардинга IPv6, раскомментируйте

```
net/ipv6/conf/default/forwarding=1
```

- Теперь мы добавим правила в файл `/etc/ufw/before.rules`. Правила по умолчанию задают только таблицу *фильтрации*, а для включения маскарадинга должна быть отредактирована таблица *nat*. Добавьте нижеследующее в начало файла, сразу после комментариев в заголовке:

```
# nat Table rules
*nat
:POSTROUTING ACCEPT [0:0]

# Forward traffic from eth1 through eth0.
-A POSTROUTING -s 192.168.0.0/24 -o eth0 -j MASQUERADE

# don't delete the 'COMMIT' line or these nat table rules won't be processed
COMMIT
```

  Комментарии не обязательны, но считается хорошей практикой документировать свою конфигурацию. При этом, редактируя любой файл *правил* в `/etc/ufw`, убедитесь, что эти строки являются последними во всех изменённых таблицах:

```
# не удаляйте строку "COMMIT", иначе эти правила не будут обрабатываться
COMMIT
```

  Для каждой *таблицы* требуется соответствующий оператор правила *COMMIT*. В этих примерах показаны только таблицы *nat* и *filter*, но вы можете точно так же добавить правила для таблиц *raw* и *mangle*.

  В приведённом примере замените *eth0*, *eth1* и *192.168.0.0/24* на подходящие для вашей сети интерфейсы и диапазон IP.

- Наконец, отключите и заново включите ufw для того, чтобы изменения вступили в силу:

```
sudo ufw disable && sudo ufw enable
```

Маскарадинг IP должен быть включён. Также вы можете добавить дополнительные правила FORWARD в `/etc/ufw/before.rules`. Рекомендуется добавлять эти правила в цепочку *ufw-before-forward*.

3.3.2. Маскарадинг iptables

iptables также может быть использован для маскировки соединений.

- Аналогично случаю с ufw, первым шагом будет включение перенаправления пакетов IPv4. Для этого отредактируйте `/etc/sysctl.conf` и раскомментируйте следующую строчку

  ```
  net.ipv4.ip_forward=1
  ```

  Если вы хотите включить и перенаправление IPv6, раскомментируйте:

  ```
  net.ipv6.conf.default.forwarding=1
  ```

- Затем выполните команду sysctl для включения новых настроек в конфигурационном файле:

  ```
  sudo sysctl -p
  ```

- Маскарадинг IP теперь может быть завершён одним правилом iptables, которое может слегка отличаться, в зависимости от конфигурации вашей сети:

  ```
  sudo iptables -t nat -A POSTROUTING -s 192.168.0.0/16 -o ppp0 -j MASQUERADE
  ```

  Команда вверху предполагает, что ваш внутрисетевой диапазон адресов — 192.168.0.0/16, а интерфейс, смотрящий в Интернет — ppp0. Синтаксис выглядит следующим образом:

  - -t nat — правило для обращения к таблице NAT
  - -A POSTROUTING — правило, добавляемое (-A) к цепочке POSTROUTING
  - -s 192.168.0.0/16 — правило применяется для трафика, происходящего из обозначенного адресного пространства
  - -o ppp0 — правило применяется к трафику, который планируется направить через определенное сетевое устройство
  - -j MASQUERADE — трафик, попадающий под данное правило, должен быть перенаправлен "jump" (-j) с маскировкой (MASQUERADE) для обработки, как описано выше

- Также, каждая цепочка в таблице фильтров (таблица по умолчанию, в которой происходит большая часть фильтрации) имеет *политику* по умолчанию для правила ACCEPT, но если вы создаёте брандмауэр в дополнение к устройству шлюза, вы можете установить политики в DROP или REJECT. В этом случае ваш замаскированный трафик должен быть разрешён в цепочке FORWARD, для того, чтобы правило вверху работало:

```
sudo iptables -A FORWARD -s 192.168.0.0/16 -o ppp0 -j ACCEPT
sudo iptables -A FORWARD -d 192.168.0.0/16 -m state \
--state ESTABLISHED,RELATED -i ppp0 -j ACCEPT
```

  Верхняя команда разрешит все соединения из вашей локальной сети с Интернетом, и весь трафик, относящийся к этим соединениям, будет возвращаться машинам, их установившим.

- Если вы хотите включить маскарадинг после перезагрузки, что вы уже, вероятно сделали, отредактируйте `/etc/rc.local` и добавьте любую из перечисленных выше команд. Например, добавьте первую команду без фильтрации:

```
iptables -t nat -A POSTROUTING -s 192.168.0.0/16 -o ppp0 -j MASQUERADE
```

3.4. Журналирование

Журналы брандмауэра — это ценные данные при определении атак, нахождения проблем в правилах и причины необычной активности вашей сети. Вы также должны включить правила регистрации событий брандмауэра, и эти правила должны предшествовать любому применяемому завершаемому правилу (правило, целью которого является определение судьбы пакета: ACCEPT, DROP, or REJECT)

Если вы используете ufw, вы можете включить регистрацию событий, введя следующую команду в терминале:

```
sudo ufw logging on
```

Для отключения регистрации событий в ufw просто замените *on* на *off* в приведённой выше команде.

Если используется iptables вместо ufw, введите:

```
sudo iptables -A INPUT -m state --state NEW -p tcp --dport 80 \
-j LOG --log-prefix "NEW_HTTP_CONN: "
```

Запрос, поступивший на порт 80 от компьютера в локальной сети, затем сгенерирует текст журнала в dmesg, который выглядит примерно так (одна строка разделена на три, чтобы уместить её в этом документе):

```
[4304885.870000] NEW_HTTP_CONN: IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00
SRC=127.0.0.1 DST=127.0.0.1 LEN=60 TOS=0x00 PREC=0x00 TTL=64 ID=58288 DF PROTO=TCP
SPT=53981 DPT=80 WINDOW=32767 RES=0x00 SYN URGP=0
```

Вышеупомянутый текст журнала также появится в файлах `/var/log/messages`, `/var/log/syslog` и `/var/log/kern.log`. Это поведение можно изменить, отредактировав соответствующим образом файл `/etc/syslog.conf` или с помощью установки и настройки ulogd и использования ULOG вместо LOG. Демон ulogd — это сервер, работающий в пространстве пользователя, который слушает инструкции журналирования от ядра специально для межсетевых экранов и может записывать журнал в любой выбранный вами файл, и даже в базы данных PostgreSQL или MySQL. Для того, чтобы легко разобраться в файлах журнала, можно использовать их анализаторы, такие как logwatch, fwanalog, fwlogwatch или lire.

3.5. Другие инструменты

Есть много инструментов, предназначенных помочь вам создать полноценный брандмауэр без каких-либо знаний iptables. Для GUI-ориентированных:

- *fwbuilder*[1] очень мощный инструмент, будет удобен администраторам, уже имевшим дело с коммерческими брандмауэрами, например, Checkpoint FireWall-1.

Если вы предпочитаете инструменты командной строки с текстовыми конфигурационными файлами:

- *Shorewall*[2] — очень мощное решение, призванное помочь вам настроить улучшенный брандмауэр для любой сети.

3.6. Ссылки

- Вики-страница *Ubuntu Firewall*[3] содержит необходимую информацию о работе с ufw..
- Также руководство пользователя по ufw содержит много полезной информации: **man ufw**.
- Больше информации по использованию iptables ищите на страничке *packet-filtering-HOWTO*[4]

[1] http://www.fwbuilder.org/
[2] http://www.shorewall.net/
[3] https://wiki.ubuntu.com/UncomplicatedFirewall
[4] http://www.netfilter.org/documentation/HOWTO/packet-filtering-HOWTO.html

- Страничка *nat-HOWTO*[5] содержит дополнительную информацию о маскарадинге.
- *IPTables HowTo*[6] Ubuntu Вики также отличный источник.

[5] http://www.netfilter.org/documentation/HOWTO/NAT-HOWTO.html
[6] https://help.ubuntu.com/community/IptablesHowTo

4. AppArmor

AppArmor — это реализация в виде модулей безопасности Linux (LSM) основанного на именах мандатного управления доступом. AppArmor ограничивает доступ отдельных программ к перечисленному набору файлов и возможностей, указанных в стандартах 1003.1e posix.

Приложение AppArmor по умолчанию установлено и запущено. Оно использует *профили* приложений, чтобы определить, какие файлы и права требуются данному приложению. Некоторые пакеты будут устанавливать собственные профили, дополнительные профили могут быть найдены в пакете apparmor-profiles.

Для установки пакета apparmor-profiles наберите в терминале:

```
sudo apt-get install apparmor-profiles
```

Профили AppArmor имеют два режима выполнения:

- Жалоб/Обучения (Complaining/Learning): нарушения профиля разрешаются и фиксируются. Удобен для тестирования и разработки новых профилей.
- Принудительный/Ограниченный (Enforced/Confined): принудительно применяет политику профиля и регистрирует нарушения.

4.1. Использование AppArmor

Пакет apparmor-utils содержит утилиты для командной стоки, которые вы можете использовать для изменения режима выполнения AppArmor, находить статус профиля, создавать новые профили и т.д.

- apparmor_status используется для просмотра текущего статуса профилей AppArmor.

  ```
  sudo apparmor_status
  ```

- aa-complain переводит профиль в режим *жалоб*.

  ```
  sudo aa-complain /path/to/bin
  ```

- aa-enforce переводит профиль в *принудительный* режим.

  ```
  sudo aa-enforce /path/to/bin
  ```

- Каталог `/etc/apparmor.d` хранит профили AppArmor. Он может использоваться для управления *режимом* всех профилей.

Введите следующее для перевода всех профилей в режим жалоб:

```
sudo aa-complain /etc/apparmor.d/*
```

Для перевода всех профилей в принудительный режим:

```
sudo aa-enforce /etc/apparmor.d/*
```

- apparmor_parser используется для загрузки профиля в ядро. Он также может использоваться для перезагрузки текущего загруженного профиля с помощью опции *-r*. Для загрузки профиля:

```
cat /etc/apparmor.d/profile.name | sudo apparmor_parser -a
```

Для перезагрузки профиля:

```
cat /etc/apparmor.d/profile.name | sudo apparmor_parser -r
```

- `service apparmor` можно использовать для *перезагрузки* всех профилей:

```
sudo service apparmor reload
```

- Каталог `/etc/apparmor.d/disable` может использоваться совместно с опцией apparmor_parser -R для *отключения* профиля.

```
sudo ln -s /etc/apparmor.d/profile.name /etc/apparmor.d/disable/
sudo apparmor_parser -R /etc/apparmor.d/profile.name
```

Для *повторного включения* отключённого профиля удалите символическую ссылку на профиль в `/etc/apparmor.d/disable/`. После этого загрузите профиль с помощью опции *-a*.

```
sudo rm /etc/apparmor.d/disable/profile.name
cat /etc/apparmor.d/profile.name | sudo apparmor_parser -a
```

- Приложение AppArmor может быть отключено, а модули ядра — выгружены, если вы введёте нижеследующее:

```
sudo service apparmor stop
sudo update-rc.d -f apparmor remove
```

- Для повторного включения AppArmor введите:

```
sudo service apparmor start
sudo update-rc.d apparmor defaults
```

Замените *profile.name* на имя того профиля, которым вы хотите управлять. Также замените `/path/to/bin/` на действительный путь к файлу. Например, для команды ping используйте `/bin/ping`

4.2. Профили

Профили AppArmor являются простыми текстовыми файлами, расположенными в `/etc/apparmor.d/`. Имена этих файлов состоят из полного пути к исполняемому файлу с заменой "/" на ".". Например, `/etc/apparmor.d/bin.ping` является профилем AppArmor для команды `/bin/ping`.

Есть два главных типа правил в профилях:

- *Path entries:* описывает, к каким файлам приложение может иметь доступ в файловой системе.
- *Capability entries:* определяет, какие привилегии ограниченному процессу разрешено использовать.

Например, взгляните на файл `/etc/apparmor.d/bin.ping`:

```
#include <tunables/global>
/bin/ping flags=(complain) {
  #include <abstractions/base>
  #include <abstractions/consoles>
  #include <abstractions/nameservice>

  capability net_raw,
  capability setuid,
  network inet raw,

  /bin/ping mixr,
  /etc/modules.conf r,
}
```

- *#include <tunables/global>:* включает операторы из других файлов. Это позволяет помещать в общий файл операторы, относящиеся к нескольким приложениям.
- */bin/ping flags=(complain):* путь к профилированной программе, также настройки режима *complain*
- *capability net_raw,:* даёт доступ к возможности CAP_NET_RAW Posix.1e.
- */bin/ping mixr,:* даёт приложению права доступа на чтение и исполнение файла.

После редактирования файла профиля он должен быть перезагружен. Более подробно смотрите *Раздел 4.1, «Использование AppArmor» [189]* .

4.2.1. Создание профиля

- *Разработка тест-плана:* Попробуйте продумать, как приложение должно быть использовано. Тест-план должен быть разделён на небольшие тестовые случаи. Каждый тестовый случай должен иметь краткое описание и список шагов, по которым нежно следовать.

  Некоторые стандартные тестовые случаи:

  - Запуск программы.
  - Завершение программы.
  - Перезагрузка программы.
  - Проверка всех команд, поддерживаемых скриптом init.
- *Генерация нового профиля:* используйте aa-genprof, чтобы сгенерировать новый профиль. Введите в консоли:

  ```
  sudo aa-genprof executable
  ```

  Например:

  ```
  sudo aa-genprof slapd
  ```
- Чтобы ваш новый профиль был включён в пакет apparmor-profiles, отправьте на *Launchpad* сообщение об ошибке в пакете *AppArmor*[7]:
  - Включите ваш план тестирования и контрольные примеры.
  - Укажите в сведениях об ошибке ваш профиль.

4.2.2. Обновление профилей

Когда программа неправильно работает, сообщения аудита посылаются в файлы журналов. Программа aa-logprof может использоваться для сканирования журналов на предмет сообщений аудита AppArmor, их проверки и обновления профилей. Введите в терминале:

```
sudo aa-logprof
```

4.3. Ссылки

- Расширенные опции конфигурирования можно найти в *Руководстве администратора по AppArmor* [8]
- Чтобы подробнее узнать о том, как использовать AppArmor с другими выпусками Ubuntu, зайдите на страницу *AppArmor Community Wiki*[9].

[7] https://bugs.launchpad.net/ubuntu/+source/apparmor/+filebug
[8] http://www.novell.com/documentation/apparmor/apparmor201_sp10_admin/index.html?page=/documentation/apparmor/apparmor201_sp10_admin/data/book_apparmor_admin.html
[9] https://help.ubuntu.com/community/AppArmor

- Ещё одним введением в AppArmor является страница *OpenSUSE AppArmor*[10].
- Отличное место для получения помощи по AppArmor, а также для участия в сообществе Ubuntu Server — это IRC канал *#ubuntu-server* на *freenode*[11].

[10] http://en.opensuse.org/SDB:AppArmor_geeks
[11] http://freenode.net

5. Сертификаты

Одной из наиболее распространённых видов криптографии на сегодняшний день является криптосистема с *открытым ключом*. Криптографическая система с открытым ключом использует *открытый ключ* и *секретный ключ*. Система *шифрует* информацию с помощью открытого ключа. Такая информация может быть *дешифрована* только с помощью секретного ключа.

Обычное применение криптосистемы с открытым ключом — шифрование трафика приложений с помощью соединений через Secure Socket Layer (SSL) или Transport Layer Security (TLS). Например, конфигурирование Apache для поддержки HTTPS (протокола HTTP через SSL). Это обеспечивает возможность шифровать трафик, используя протокол, который сам по себе не обеспечивает шифрования.

Сертификат является методом, используемым для распространения *публичного ключа* и другой информации о сервере и организации, которая за него отвечает. Сертификаты могут иметь цифровую подпись от центра сертификации (*CA*). Центр сертификации является доверенным третьим лицом, которое подтверждает, что информация, содержащаяся в сертификате, является точной.

5.1. Типы сертификатов

Чтобы создать защищённый сервер с использованием криптографии открытого ключа, в большинстве случаев, вы посылаете запрос на сертификат (с открытым ключом), подтверждаете подлинность данных о своей компании и оплачиваете услуги удостоверяющего центра (CA). Удостоверяющий центр проверяет ваш запрос и присылает вам сертификат для вашего сервера. В качестве альтернативы вы можете создать свой собственный *самоподписанный* сертификат.

Учтите, что самоподписанные сертификаты не должны использоваться в большинстве серьёзных производственных систем.

Продолжая пример с HTTPS, подписанный CA сертификат имеет две важных особенности, которых сампоподписанный сертификат не имеет:

- Браузеры (обычно) автоматически определяют сертификаты и разрешают безопасные соединения без подтверждения пользователя.
- Выданный CA подписанный сертификат является гарантией подлинности организации, предоставляющей веб-страницы браузеру.

Most Web browsers, and computers, that support SSL have a list of CAs whose certificates they automatically accept. If a browser encounters a certificate whose authorizing CA is not in the list, the browser asks the user to either accept or decline the connection. Also, other applications may generate an error message when using a self-signed certificate.

Процесс получения сертификата от CA довольно прост. Краткие сведения об этом:

1. Создайте пару ключей шифрования, открытый и закрытый.
2. Создайте запрос сертификата, основанный на открытом ключе. Данный запрос содержит в себе информацию о вашем сервере и компании, где он размещается.
3. Отправьте запрос сертификата вместе с документами, подтверждающими вашу личность, в CA. Мы не можем рекомендовать вам, какой удостоверяющий центр выбрать. Ваше решение может основываться на вашем прошлом опыте, на опыте ваших друзей и коллег, или просто на финансовых факторах.

   Если вы определились с CA, вам необходимо следовать инструкциям, которые он предоставит для получения его сертификата.
4. Когда CA установит, что вы являетесь тем, за кого себя выдаёте, он пришлёт вам цифровой сертификат.
5. Установите этот сертификат на ваш защищённый сервер и настройте соответствующие приложения на использование сертификата.

5.2. Генерация запроса на подпись сертификата (Certificate Signing Request, или CSR)

Получаете ли вы сертификат от CA или генерируете его собственноручно, первым шагом должно быть создание ключа.

Если сертификат будет использоваться системными сервисами, такими как Apache, Postfix, Dovecot и т.п., уместно создать ключ без пароля. Отсутствие пароля позволяет сервису запускаться без вмешательства пользователя, обычно это предпочтительный вариант запуска сервиса.

В этом разделе показано, как создать ключ с паролем и без него. Ключ без пароля затем будет использован для создания сертификата, который можно использовать для различных системных сервисов.

Запуск вашего защищённого сервиса без пароля удобен потому, что вам не потребуется вводить пароль при каждом запуске данного

сервиса. Однако это небезопасно и компрометация ключа будет означать и компрометацию сервера.

Для генерации *ключей* запроса подписи сертификата (CSR) запустите следующую команду из строки терминала:

```
openssl genrsa -des3 -out server.key 2048
```

```
Generating RSA private key, 2048 bit long modulus
.........................++++++
.......++++++
e is 65537 (0x10001)
Enter pass phrase for server.key:
```

Теперь вы можете ввести свою парольную фразу. Для наилучшей безопасности она должна содержать не менее восьми символов. Минимальная длина — четыре символа. Пароль должен содержать цифры и/или специальные символы и не являться словом из словаря. Запомните то, что вы введёте.

Для подтверждения наберите парольную фразу ещё раз. Как только вы наберете её правильно, ключ к серверу будет создан и сохранён в файле `server.key`.

Теперь создадим небезопасный ключ, без кодовой фразы, и перетасуем имена ключей:

```
openssl rsa -in server.key -out server.key.insecure
mv server.key server.key.secure
mv server.key.insecure server.key
```

Небезопасный ключ теперь называется `server.key`, и вы можете использовать его для создания CSR без кодовой фразы.

Для создания CSR выполните следующую команду в терминале:

```
openssl req -new -key server.key -out server.csr
```

У вас будет запрошена парольная фраза (при использовании ключа с паролем - прим. пер.). Если пароль введён правильно, у вас запросят название компании, имя сайта, адрес электронной почты и пр. Как только вы введёте все эти подробности, будет создан запрос CSR и сохранен в файл `server.csr`.

Теперь вы можете отправить этот CSR-файл в CA для обработки. CA, используя этот CSR-файл, выпустит сертификат. С другой стороны, вы

можете создать самоподписанный сертификат сами, используя тот же CSR-файл.

5.3. Создание сертификата со своей подписью

Для того, чтобы создать самоподписанный сертификат, исполните следующую команду в терминале:

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

Вышеприведённая команда предложит ввести парольную фразу. При вводе правильной парольной фразы ваш сертификат будет создан и сохранён в файле `server.crt`.

Если ваш защищённый сервер будет использоваться в производственной среде, вам, скорее всего, необходим сертификат, подписанный CA. В данном случае не рекомендуется использовать самоподписанный сертификат.

5.4. Установка сертификата

Вы можете установить ключевой файл `server.key` и файл сертификата `server.crt`, или файл сертификата, выданный вам CA, запустив следующую команду в строке терминала:

```
sudo cp server.crt /etc/ssl/certs
sudo cp server.key /etc/ssl/private
```

Теперь просто сконфигурируйте любые приложения, имеющие поддержку криптографии с открытым ключом, для использования файлов *сертификата* и ключа. Например, Apache может использовать HTTPS, Dovecot — IMAPS и POP3S и т.д.

5.5. Центр Сертификации

Если для ваших сетевых сервисов требуется много самоподписанных сертификатов, стоит затратить дополнительные усилия и установить свой собственный *центр сертификации (CA)*. Использование сертификатов, подписанных вашим собственным CA, позволяет различным использующим сертификаты сервисам проще доверять другим сервисам, использующим сертификаты от этого же CA.

1. Сначала создайте каталоги для хранения сертификата CA и связанных с ним файлов:

```
sudo mkdir /etc/ssl/CA
sudo mkdir /etc/ssl/newcerts
```

2. Для работы CA требует несколько дополнительных файлов: один содержит запись о последнем серийном номере, выданном CA (каждый сертификат должен иметь уникальный серийный номер), другой файл предназначен для записи, какие сертификаты были выданы:

```
sudo sh -c "echo '01' > /etc/ssl/CA/serial"
sudo touch /etc/ssl/CA/index.txt
```

3. Третьим файлом является файл конфигурации CA. Хотя он не обязателен, тем не менее, он обеспечивает удобство при выдаче нескольких сертификатов. Отредактируйте `/etc/ssl/openssl.cnf` и в *[CA_default]* измените:

```
dir = /etc/ssl/ # Где все сохраняется
database = $dir/CA/index.txt # база данных файла index.
certificate = $dir/certs/cacert.pem # CA сертификат
serial = $dir/CA/serial # Верный серийный номер
private_key = $dir/private/cakey.pem# Частный ключ
```

4. Next, create the self-signed root certificate:

```
openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out cacert.pem -days 3650
```

Затем вас попросят ввести описание сертификата.

5. Теперь установите корневой сертификат и ключ:

```
sudo mv cakey.pem /etc/ssl/private/
sudo mv cacert.pem /etc/ssl/certs/
```

6. Теперь вы готовы к подписыванию сертификатов. Первое, что вам необходимо, это запрос на подпись сертификата (CSR), подробнее смотрите *Раздел 5.2, «Генерация запроса на подпись сертификата (Certificate Signing Request, или CSR)» [195]*. Как только у вас будет CSR, можно перейти к получению сертификата, подписанного центром сертификации:

```
sudo openssl ca -in server.csr -config /etc/ssl/openssl.cnf
```

После ввода пароля для ключа центра сертификации, вас попросят дважды подписать сертификат, для его подтверждения. Затем вы увидите большое количество генерируемых данных процесса создания сертификата.

7. Теперь у вас должен появиться новый файл `/etc/ssl/newcerts/01.pem`, с таким же содержанием, что и в предыдущем выводе. Выделите и

скопируйте всё, начиная со строки *-----BEGIN CERTIFICATE-----* и до строки *----END CERTIFICATE-----* в файл с названием, соответствующим сетевому имени сервера, где он будет установлен. Например, `mail.example.com.crt` — вполне хорошее описательное имя.

Последующие сертификаты будут называться `02.pem`, `03.pem`, и т.д.

Замените *mail.example.com.crt* своим описательным именем.

8. Наконец, скопируйте новый сертификат на компьютер, для которого он выпущен, и настройте соответствующие приложения на его использование. Место по умолчанию для установки сертификатов — каталог `/etc/ssl/certs`. Это позволяет нескольким сервисам использовать один и тот же сертификат без чрезмерного усложнения прав доступа к файлу.

   Для приложений, требующих использования сертификата CA, вы должны скопировать файл `/etc/ssl/certs/cacert.pem` в каталог `/etc/ssl/certs/` на каждом сервере.

5.6. Ссылки

- Для получения более подробных инструкций по использованию криптографии смотрите *SSL Certificates HOWTO*[12] на tlpd.org
- На странице Википедии *HTTPS*[13] вы найдете больше информации о протоколе HTTPS
- Для дополнительной информации об *OpenSSL* смотрите *домашнюю страницу OpenSSL*[14].
- Также хорошим подробным руководством является *Network Security with OpenSSL*[15] издательства O'Reilly.

[12] http://tldp.org/HOWTO/SSL-Certificates-HOWTO/index.html
[13] http://ru.wikipedia.org/wiki/Https
[14] http://www.openssl.org/
[15] http://oreilly.com/catalog/9780596002701/

6. eCryptfs

eCryptfs is a POSIX-compliant enterprise-class stacked cryptographic filesystem for Linux. Layering on top of the filesystem layer *eCryptfs* protects files no matter the underlying filesystem, partition type, etc.

Во время установки предлагается возможность шифрования раздела `/home`. Это позволит автоматически настроить всё, что необходимо для шифрования, и смонтировать раздел.

В качестве примера в этом разделе приводится шифрование `/srv` с помощью *eCryptfs*.

6.1. Использование eCryptfs

Сначала установите необходимые пакеты. Введите в командной строке:

```
sudo apt-get install ecryptfs-utils
```

Теперь смонтируйте раздел для шифрования:

```
sudo mount -t ecryptfs /srv /srv
```

Вам будет задано несколько вопросов о том, как ecryptfs должен зашифровать данные.

Чтобы убедиться, что файлы, находящиеся в `/srv`, действительно являются зашифрованной копией каталога `/etc/default` в `/srv`:

```
sudo cp -r /etc/default /srv
```

Теперь отмонтируйте `/srv` и попробуйте просмотреть файл:

```
sudo umount /srv
cat /srv/default/cron
```

Перемонтирование `/srv` с помощью ecryptfs сделает данные снова доступными.

6.2. Автоматическое монтирование зашифрованных разделов

Существует пара способов автоматически монтировать файловую систему, зашифрованную ecryptfs, на этапе загрузки. В этом примере используется

файл `/root/.ecryptfsrc`, содержащий опции монтирования, совместно с файлом парольной фразы, расположенным на USB ключе.

Сначала создайте файл `/root/.ecryptfsrc`, содержащий:

```
key=passphrase:passphrase_passwd_file=/mnt/usb/passwd_file.txt
ecryptfs_sig=5826dd62cf81c615
ecryptfs_cipher=aes
ecryptfs_key_bytes=16
ecryptfs_passthrough=n
ecryptfs_enable_filename_crypto=n
```

Измените параметр *ecryptfs_sig* на сигнатуру из файла `/root/.ecryptfs/sig-cache.txt`.

Далее создадим файл парольной фразы `/mnt/usb/passwd_file.txt`:

```
passphrase_passwd=[secrets]
```

Теперь добавьте необходимые строки в `/etc/fstab`:

```
/dev/sdb1       /mnt/usb        ext3    ro      0 0
/srv /srv ecryptfs defaults 0 0
```

Удостоверьтесь, что USB-носитель смонтирован перед шифруемым разделом.

Наконец, перегрузитесь и `/srv` будет смонтирован с использованием *eCryptfs*.

6.3. Другие утилиты

Пакет ecryptfs-utils содержит несколько других полезных утилит:

- *ecryptfs-setup-private:* создаёт каталог `~/Private`, информация в котором хранится в зашифрованном виде. Эту утилиту могут запустить пользователи без административных полномочий, чтобы защитить личные данные от других пользователей системы.
- *ecryptfs-mount-private и ecryptfs-umount-private:* соответственно смонтирует и отмонтирует пользовательский каталог `~/Private`.
- *ecryptfs-add-passphrase:* ecryptfs-add-passphrase: добавляет новую парольную фразу в хранилище ключей ядра.
- *ecryptfs-manager:* управляет объектами eCryptfs, например ключами.
- *ecryptfs-stat:* позволит вам увидеть метаинформацию ecryptfs для файла.

6.4. Ссылки

- Дополнительная информация о *eCryptfs* доступна на *странице проекта на Launchpad*[16].
- Существует также статья в *Linux Journal*[17], посвящённая *eCryptfs*.
- Also, for more ecryptfs options see the *ecryptfs man page*[18].
- *eCryptfs Ubuntu Wiki*[19] также содержит дополнительную информацию.

16 https://launchpad.net/ecryptfs
17 http://www.linuxjournal.com/article/9400
18 http://manpages.ubuntu.com/manpages/trusty/en/man7/ecryptfs.7.html
19 https://help.ubuntu.com/community/eCryptfs

Глава 10. Мониторинг

1. Обзор

Мониторинг основных серверов и служб является важной составляющей системы администрирования. Большинство сетевых служб контролируются для наблюдения за их производительностью, доступностью или и тем и другим. В этом разделе описаны установка и настройка систем Nagios для мониторинга и Munin для слежения за производительностью.

Для примера, в этом разделе использованы два сервера с именами *server01* и *server02*. *Server01* настроен на работу с системой Nagios для мониторинга сервисов и служб самого себя и *server02*. На server01 также будет установлен пакет munin для сбора информации по сети. Используя пакет munin-node, *server02* будет отправлять информацию на *server01*.

Надеемся, что эти простые примеры позволят вам контролировать дополнительные сервера и службы в вашей сети.

2. Nagios

2.1. Установка

Для начала на *server01* необходимо установить пакет nagios. Для этого введите в терминале:

```
sudo apt-get install nagios3 nagios-nrpe-plugin
```

Вам будет предложено ввести пароль для пользователя *nagiosadmin*. Учётные записи пользователя находятся в `/etc/nagios3/htpasswd.users`.
Для смены пароля пользователя *nagiosadmin* или добавления других пользователей для выполнения CGI скриптов Nagios используйте утилиту htpasswd, которая является частью пакета apache2-utils. apache2-utils.

Например, для смены пароля пользователя *nagiosadmin* введите в терминале:

```
sudo htpasswd /etc/nagios3/htpasswd.users nagiosadmin
```

Для добавления пользователя:

```
sudo htpasswd /etc/nagios3/htpasswd.users steve
```

Далее, на *server02* установите пакет nagios-nrpe-server. В терминале на server02 введите:

```
sudo apt-get install nagios-nrpe-server
```

NRPE позволяет выполнять локальные проверки на удалённом компьютере. Но существуют и другие способы достижения этой цели, используя другие плагины Nagios, также как и другие способы проверок.

2.2. Обзор конфигурации

Существует несколько каталогов, содержащих конфигурационные файлы Nagios а также файлы проверок.

- `/etc/nagios3`: содержит конфигурационные файлы для работы демона nagios, CGI-файлов, хостов и др.
- `/etc/nagios-plugins`: файлы конфигурации для служебных проверок.
- `/etc/nagios`: содержит конфигурационные файлы на удаленном компьютере nagios-nrpe-server.

- `/usr/lib/nagios/plugins/`: тут находятся бинарные проверки. Для просмотра опций проверки используйте ключ-*h*.

  Например: **/usr/lib/nagios/plugins/check_dhcp -h**

Существует множество проверок Nagios, которые могут быть настроены для выполнения на любом компьютере. В этом примере NagiosNagios будет настроен на проверку дискового пространства, службы DNS, а также группы пользователей MySQL. Проверка DNS будет осуществятся на *server02*, а группа компьютеров MySQL будет включать в себя как *server01*, так и *server02*.

Смотрите раздел *Раздел 1, «HTTPD - веб сервер Apache2» [213]* для более детальных настроек Apache, *Глава 8, Служба доменных имён (DNS) [157]* для настройки DNS, а также *Раздел 1, «MySQL» [237]* для настройки MySQL.

В дополнение к этому будут приведены несколько терминов, которые помогут вам облегчить настройку Nagios:

- *Host*: сервер, рабочая станция, сетевое устройство и т.д., которое отслеживается.
- *Host Group*: группа подобных компьютеров. Например вы можете сгруппировать все веб-серверы, файловые серверы и т.д.
- *Service*: служба, которая отслеживается на компьютере. Например HTTP, DNS, NFS и т.д.
- *Группа служб*: позволяет объединить несколько служб вместе. Например это будет полезным для объединения нескольких веб-серверов.
- *Контакт*: человек, который будет уведомлён при каком-либо событии. Nagios может быть настроен на отправку email, SMS-сообщений и т.д.

По умолчанию Nagios настроен на проверку HTTP, дискового пространства, SSH, текущих пользователей, процессов и слежение за уровнем загрузки на локальном компьютере (*localhost*). Nagios также выполняет проверку шлюза посредством команды *ping*.

Настроить Nagios на множестве компьютеров может быть довольно сложно. Начать лучше с нескольких компьютеров, одного или двух, настроить всё оптимальным образом, а затем расширить настройку для большего количества компьютеров.

2.3. Конфигурация

- 1. Для начала необходимо создать *конфигурационный файл* для *server02*. Если не указанно иное, выполните все эти команды на *server01*. Введите в терминале:

```
sudo cp /etc/nagios3/conf.d/localhost_nagios2.cfg \ /etc/nagios3/conf.d/server02.cfg
```

В вышеуказанном, а также следующем примере замените "*server01*", "*server02*", *172.18.100.100* и *172.18.100.101* на имя и IP-адреса ваших серверов.

2. Далее отредактируйте файл `/etc/nagios3/conf.d/server02.cfg`:

```
define host{
        use                     generic-host  ; Name of host template to use
        host_name               server02
        alias                   Server 02
        address                 172.18.100.101
}

# check DNS service.
define service {
        use                             generic-service
        host_name                       server02
        service_description             DNS
        check_command                   check_dns!172.18.100.101
}
```

3. Перезагрузите демон nagios для активации новых настроек:

```
sudo service nagios3 restart
```

- 1. Теперь добавим служебное описание для проверки MySQL путём добавления следующих строк в `/etc/nagios3/conf.d/services_nagios2.cfg`:

```
# check MySQL servers.
define service {
        hostgroup_name        mysql-servers
        service_description   MySQL
        check_command         check_mysql_cmdlinecred!nagios!secret!$HOSTADDRESS
        use                   generic-service
        notification_interval 0 ; set > 0 if you want to be renotified
}
```

2. Сейчас должны быть определена группа *mysql-servers*. Отредактируйте `/etc/nagios3/conf.d/hostgroups_nagios2.cfg`, добавив следующее:

```
# MySQL hostgroup.
define hostgroup {
        hostgroup_name  mysql-servers
                alias           MySQL servers
                members         localhost, server02
        }
```

3. Проверка Nagios должна пройти аутентификацию в MySQL. Для добавления пользователя *nagios* в MySQL введите:

```
mysql -u root -p -e "create user nagios identified by 'secret';"
```

Пользователь *nagios* должен присутствовать на всех компьютерах рабочей группы серверов *mysql-servers*.

4. Перезагрузите nagios для проверки сервера MySQL.

```
sudo service nagios3 restart
```

- 1. Наконец, необходимо настроить NRPE для проверки дискового пространства на *server02*.

   На *server01* добавим служебную проверку в `/etc/nagios3/conf.d/server02.cfg`:

```
# NRPE disk check.
define service {
        use                     generic-service
        host_name               server02
        service_description     nrpe-disk
        check_command           check_nrpe_1arg!check_all_disks!172.18.100.101
}
```

2. Теперь на *server02* отредактируем `/etc/nagios/nrpe.cfg`:

```
allowed_hosts=172.18.100.100
```

   А в строку объявления команды добавим:

```
command[check_all_disks]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -e
```

3. В конце перезагрузим nagios-nrpe-server:

```
sudo service nagios-nrpe-server restart
```

4. На *server01* также необходимо перезагрузить nagios:

```
sudo service nagios3 restart
```

Теперь вы должны видеть ваши серверы и служебные проверки в файлах Nagios CGI. Для доступа к ним наберите в строке браузера http://server01/nagios3. Вам будет предложено ввести имя пользователя и пароль для *nagiosadmin*.

2.4. Ссылки

В этом разделе были описаны лишь незначительные возможности Nagios. nagios-plugins-extra и nagios-snmp-plugins содержат намного больше файлов проверки служб.

- Для более детальной информации обратитесь к документации на официальном сайте *Nagios*[1].
- А особенно на сайте *онлайн-документации*[2].
- Существует несколько *книг*[3] посвященных Nagios и мониторингу сети.
- The *Nagios Ubuntu Wiki*[4] page also has more details.

[1] http://www.nagios.org/
[2] http://nagios.sourceforge.net/docs/3_0/
[3] http://www.nagios.org/propaganda/books/
[4] https://help.ubuntu.com/community/Nagios3

3. Munin

3.1. Установка

Перед установкой Munin на *server01* необходимо установить веб-сервер apache2. Стандартной конфигурации будет достаточно для запуска сервера munin. Для более детальной информации по настройке apache2, обратитесь к разделу *Раздел 1, «HTTPD - веб сервер Apache2» [213]*.

Для начала на *server01* установим munin. Введите в терминале:

```
sudo apt-get install munin
```

Теперь на *server02* установим пакет munin-node:

```
sudo apt-get install munin-node
```

3.2. Конфигурация

На *server01* отредактируйте файл `/etc/munin/munin.conf` добавив IP-адрес *server02*:

```
## First our "normal" host.
[server02]
       address 172.18.100.101
```

Замените *server02* и *172.18.100.101* на имя компьютера и IP-адрес вашего сервера.

Далее настроим пакет munin-node на *server02*. Отредактируйте файл `/etc/munin/munin-node.conf` для доступа *server01*:

```
allow ^172\.18\.100\.100$
```

Замените *^172\.18\.100\.100$* на IP-адрес вашего сервера munin.

Теперь перезагрузите munin-node на *server02* для принятия изменений:

```
sudo service munin-node restart
```

Наконец, в строке браузера введите *http://server01/munin*, и вы увидите ссылки, которые отображают информацию из стандартных плагинов *munin-plugins* для дисков, сети, процессов и системы.

Поскольку это новая установка, может пройти некоторое время перед отображением какой-либо полезной информации.

3.3. Дополнительные плагины

Пакет munin-plugins-extra содержит дополнительные проверки производительности служб, таких как DNS, DHCP, Samba и т.д. Для установки пакета введите в терминале:

```
sudo apt-get install munin-plugins-extra
```

Убедитесь в том, что вы установили пакет как на сервер, так и на связывающие компьютеры.

3.4. Ссылки

- Посетите официальный сайт *Munin*[5] для более детальной информации.
- Узконаправленная *документация Munin*[6] содержит информацию о дополнительных плагинах, написании плагинов и т.д.
- Также можно прочесть книгу на немецком языке издательства «Open Source Press»: *Munin Graphisches Netzwerk- und System-Monitoring*[7].

[5] http://munin.projects.linpro.no/
[6] http://munin.projects.linpro.no/wiki/Documentation
[7] https://www.opensourcepress.de/index.php?26&backPID=178&tt_products=152

Глава 11. Веб-серверы

Веб-сервер — это программное обеспечение, ответственное за приём HTTP-запросов от клиентов, известных как веб-браузеры, и отправки им HTTP-ответов вместе с опциональными данными, которые обычно являются веб-страницами, такими как документы HTML и присоединённые объекты (изображения и т.д.).

1. HTTPD - веб сервер Apache2

Apache является наиболее часто используемым веб-сервером на системах Linux. Веб-серверы используются для обслуживания веб-страниц, запрашиваемых клиентскими компьютерами. Клиенты обычно запрашивают веб-страницы с помощью приложений для просмотра Интернета, таких как Firefox, Opera, Chromium или Mozilla.

Пользователи вводят Uniform Resource Locator (URL), чтобы указать на веб-сервер посредством Fully Qualified Domain Name (FQDN) и пути к необходимому ресурсу. Например, чтобы просмотреть домашнюю страницу *веб-сайта Ubuntu*[1], пользователь будет вводить только FQDN:

```
www.ubuntu.com
```

Чтобы посмотреть подстраницу *community*[2], пользователь вводит FQDN, сопровождаемый путём:

```
www.ubuntu.com/community
```

Самый часто используемый протокол для передачи веб-страниц — это HTTP (Hyper Text Transfer Protocol). Также поддерживаются такие протоколы, как HTTP over Secure Sockets Layer (HTTPS) и Transfer Protocol (FTP), протокол для передачи файлов.

Веб-сервер Apache часто используется в связке с движком баз данных MySQL, скриптовым языком PHP и другими популярными скриптовыми языками — Python и Perl. Данная конфигурация обозначена аббревиатурой LAMP (Linux, Apache, MySQL, Perl/Python/PHP) и формирует собой мощный набор инструментов для разработки и использования веб-приложений.

1.1. Установка

Веб-сервер Apache2 доступен в Ubuntu Linux. Чтобы установить Apache2:

- В терминале введите следующую команду:

```
sudo apt-get install apache2
```

[1] http://www.ubuntu.com
[2] http://www.ubuntu.com/community

1.2. Конфигурация

Apache2 настраивается путём редактирования или добавления *директив* в обычных текстовых конфигурационных файлах. Эти *директивы* размещаются в следующих файлах и каталогах:

- *apache2.conf:* главный конфигурационный файл Apache2. Содержит параметры, которые являются *глобальными* для Apache2.
- *httpd.conf:* исторически был главным конфигурационным файлом Apache2, названным по имени демона httpd. Теперь этого файла не существует. В более старых версиях Ubuntu файл может присутствовать, но быть пустым, поскольку все конфигурационные опции были перенесены в перечисленные ниже каталоги.
- *conf-available:* this directory contains available configuration files. All files that were previously in `/etc/apache2/conf.d` should be moved to `/etc/apache2/conf-available`.
- *conf-enabled:* holds *symlinks* to the files in `/etc/apache2/conf-available`. When a configuration file is symlinked, it will be enabled the next time apache2 is restarted.
- *envvars:* файл, где устанавливается *переменные окружения* Apache2.
- *mods-available:* этот каталог содержит файлы конфигурации для загрузки *модулей* и их настройки. Однако, не все модули будут иметь конкретные конфигурационные файлы.
- *mods-enabled:* содержит *символические ссылки* на файлы в `/etc/apache2/mods-available`. Если на конфигурационный файл модуля поставить символическую ссылку, модуль будет загружен при следующем перезапуске apache2.
- *ports.conf:* содержит директивы, определяющие, на каких TCP портах Apache2 принимает соединения.
- *sites-available:* эта папка содержит конфигурационные файлы виртуальных хостов (Virtual Hosts) Apache2. Виртуальные хосты позволяют сконфигурировать Apache2 так, чтобы каждый сайт имел отдельную конфигурацию.
- *sites-enabled:* подобна mods-enabled, `sites-enabled` содержит символьные ссылки на каталог `/etc/apache2/sites-available`. Соответственно, когда на конфигурационный файл в sites-available будет сделана ссылка, то она вступит в действие, как только Apache2 будет перезапущен.
- *magic:* instructions for determining MIME type based on the first few bytes of a file.

В дополнение, другие конфигурационные файлы могут быть добавлены, используя директиву *Include*. Для включения нескольких

конфигурационных файлов могут использоваться метасимволы. Любая директива может быть помещена в любой из этих конфигурационных файлов. Изменения в основных конфигурационных файлах распознаются Apache2 в процессе запуска или перезапуска.

Сервер также читает файл, содержащий mime-типы документов; имя файла задается директивой *TypesConfig*, как правило, через `/etc/apache2/mods-available/mime.conf`, которая может также включать дополнения и коррекции, и `/etc/mime.types` по умолчанию.

1.2.1. Основные настройки

В этом разделе описаны основные параметры конфигурации сервера Apache2. Обратитесь к *Документации Apache2*[3] для более подробной информации.

- По умолчанию Apache 2 имеет конфигурацию, совместимую с виртуальными хостами. В его настройках указан единственный виртуальный хост (через директиву *VirtualHost*), который может быть оставлен как есть, если у вас всего один сайт, либо использован как шаблон для других виртуальных хостов, если сайтов у вас несколько. Если оставить его настройку, как есть, виртуальный хост по умолчанию будет обслуживать ваш основной сайт, или сайт, который увидят пользователи, если URL, по которому они попали на ваш сервер, не обрабатывается ни одним из остальных виртуальных хостов (т.е. если имя хоста не найдено ни в одной директиве *ServerName*). Чтобы изменить виртуальный хост по умолчанию, отредактируйте файл `/etc/apache2/sites-available/default`.

Директивы, установленные для виртуального хоста, применяются только для того виртуального хоста, для которого они установлены. Если директива установлена в основной конфигурации сервера и не установлена для конкретного виртуального хоста, то будет использовано значение по умолчанию. Например, вы можете указать адрес электронной почты вебмастера в основном конфигурационном файле сервера и не указывать его для каждого виртуального хоста.

Если вы хотите настроить новый виртуальный хост или сайт, скопируйте этот файл в ту же папку, дав ему выбранное вами имя. Например:

```
sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/mynewsite
```

[3] http://httpd.apache.org/docs/2.2/

Отредактируйте новый файл, чтобы настроить новый сайт, используя некоторые директивы, описанные ниже.

- Директива *ServerAdmin* определяет почтовый адрес администратора сервера, который будет отображаться пользователям. Значение по умолчанию — webmaster@localhost. Данная переменная должна быть изменена на доступный для вас почтовый адрес (если вы являетесь администратором сервера). Если на вашем сайте возникнут проблемы, Apache2 отобразит ошибку, в которой также будет отображен указанный почтовый адрес с целью сообщения проблемы. Вы можете найти эту директиву в вашем файле конфигурации сайтов, в каталоге /etc/apache2/sites-available.
- Директива *Listen* определяет порт и, при указании, IP-адрес, на котором должен работать Apache2. Если IP-адрес не указан, Apache2 работает на всех IP-адресах, которые доступны компьютеру, на котором он запущен. Значение директивы по умолчанию — порт 80. Вы можете изменить значение на 127.0.0.1:80, чтобы Apache2 работал только на локальном интерфейсе и не был доступен извне. Также можно указать, например, значение 81 для изменения порта сервера или оставить всё как есть для работы по умолчанию. Данная директива может быть найдена и изменена в её собственном файле `/etc/apache2/ports.conf`
- Директива *ServerName* является необязательной и определяет, на какие FQDN должен отвечать ваш сайт. По умолчанию виртуальный хост не имеет установленной директивы ServerName, поэтому он будет отвечать на все запросы, которые не совпадают с директивой ServerName на другом виртуальном хосте. Если вы только что приобрели доменное имя ubunturocks.com и хотите разместить его на своём Ubuntu-сервере, то значение директивы ServerName в конфигурационном файле вашего виртуального хоста должно быть ubunturocks.com. Добавьте эту директиву к новому файлу виртуального хоста, который вы создали ранее (`/etc/apache2/sites-available/mynewsite`).

  Возможно вы захотите, чтобы ваш сайт отвечал на www.ubunturocks.com, поскольку многие пользователи сочтут подходящим использовать префикс www. Для этого используйте директиву *ServerAlias*. В директиве ServerAlias вы также можете использовать метасимволы.

  Например, следующая конфигурация заставит ваш сайт отвечать на любой запрос домена, оканчивающийся на *.ubunturocks.com*.

```
ServerAlias *.ubunturocks.com
```

- Директива *DocumentRoot* указывает, где Apache2 должен искать файлы, составляющие сайт. Значением по умолчанию является /var/www, как указано в `/etc/apache2/sites-available/default`. Если необходимо, измените это значение в файле виртуального хоста вашего сайта, и не забудьте при необходимости создать соответствующий каталог!

Активируйте новый *VirtualHost*, используя утилиту a2ensite, и перезапустите Apache2:

```
sudo a2ensite mynewsite
sudo service apache2 restart
```

Не забудьте заменить *mynewsite* более подходящим именем для VirtualHost. Один из способов — это назвать файл так же, как в директиве *ServerName* для VirtualHost.

Аналогично, используйте утилиту a2dissite для выключения сайтов. Это может быть полезным для устранения неполадок в конфигурации для нескольких виртуальных хостов.

```
sudo a2dissite mynewsite
sudo service apache2 restart
```

1.2.2. Настройки по умолчанию

Данный раздел описывает настройку параметров Apache2 по умолчанию. Они необходимы, например, если вы добавляете виртуальный хост, настраиваете нужные директивы, а некоторые не указываете. В этом случае используются значения по умолчанию.

- *DirectoryIndex* указывает на страницу (файл) по умолчанию, которую выдаёт пользователю сервер при запросе индекса каталога, указывая слеш (/) в конце имени каталога.

  Например, когда пользователь запрашивает страницу http://www.example.com/this_directory/, он получит либо страницу DirectoryIndex, если она существует, либо сгенерированный сервером список каталогов, если она не существует и задана опция indexes, либо страницу отказа в доступе (Permission Denied). Сервер попытается найти один из файлов, перечисленных в директиве DirectoryIndex и вернёт первый найденный. Если он не найдёт ни один из этих файлов и, если задана опция *Options Indexes* для этого каталога, сервер сгенерирует и вернёт в формате HTML список подкаталогов и файлов в каталоге. Значения по умолчанию из `/etc/apache2/mods-available/dir.conf` — это "index.html index.cgi index.pl index.php index.xhtml index.htm". Таким образом, если Apache2 находит в

запрашиваемом каталоге файл, соответствующий любому из этих имён, он возвращает первый найденный файл.

- Директива *ErrorDocument* позволяет указать Apache2 файл для вывода определённых ошибок. Например, если пользователь запрашивает несуществующий ресурс, возникнет ошибка 404. По умолчанию Apache2 просто вернёт код HTTP 404. Обратитесь к `/etc/apache2/conf.d/localized-error-pages` за подробными инструкциями по использованию ErrorDocument, включая файлы примеров.
- По умолчанию сервер записывает журнал передачи данных в файл `/var/log/apache2/access.log`. Вы можете изменить это в вашем файле конфигурации виртуальных хостов с помощью директивы *CustomLog* для каждого хоста, или пропустить его, чтобы принять значение по умолчанию, указанное в `/etc/apache2/conf.d/other-vhosts-access-log`. Вы также можете указать файл, в котором регистрируются ошибки, через директиву *ErrorLog*, которая по умолчанию указывает на `/var/log/apache2/error.log`. Они хранятся отдельно от журналов передачи данных, чтобы помочь в решении проблем с вашим сервером Apache2. Вы также можете задать уровень журналирования в *LogLevel* (по умолчанию "warn") и формат журнала в *LogFormat* (Смотрите `/etc/apache2/apache2.conf` для значений по умолчанию).
- Некоторые опции указываются для каталога, а не для сервера. Одна из таких директив — *Options*. Группа строк Directory заключена в теги XML следующим образом:

```
<Directory /var/www/mynewsite>
...
</Directory>
```

  Директива *Options* внутри группы строк Directory может принимать одно или несколько из следующих значений (помимо прочих), разделяемых пробелом:

  - **ExecCGI** — разрешает запуск CGI-скриптов. CGI-скрипты не будут запускаться, если эта опция не установлена.
  - **Includes** — разрешает расширения на стороне сервера. Расширения на стороне сервера позволяют HTML-файлу *включать в себя* другие файлы. Смотрите *документацию Apache SSI (сообщества Ubuntu)*[4] для получения дополнительной информации.
  - **IncludesNOEXEC** — разрешает использовать расширения на стороне сервера, но отключает команды *#exec* и *#include* в CGI скриптах.

[4] https://help.ubuntu.com/community/ServerSideIncludes

- **Indexes** — отображать форматированный список содержимого каталога, если в опрашиваемом каталоге нет файла из *Индекса каталога* (такого как index.html).

Из соображений безопасности обычно эта опция не устанавливается и, естественно, не должна устанавливаться для каталога DocumentRoot. Включайте эту опцию только для отдельных каталогов и только в том случае, если уверены, что хотите, чтобы пользователи могли просматривать всё содержимое каталога.

- **Multiview** — поддержка множества видов страницы в зависимости от содержимого; по соображениям безопасности этот параметр по умолчанию выключен. Смотрите *Документацию Apache2 по этому параметру*[5].
- **SymLinksIfOwnerMatch** — переходить по символическим ссылкам только в случае, если у файла/каталога и ссылки один и тот же владелец.

1.2.3. Настройки httpd

Этот раздел объясняет некоторые основные настройки демона httpd

LockFile — директива LockFile устанавливает путь к lock-файлу сервера, который используется, если сервер собран с параметрами USE_FCNTL_SERIALIZED_ACCEPT или USE_FLOCK_SERIALIZED_ACCEPT. Он должен располагаться на локальном диске. Значение директивы должно быть оставлено по умолчанию за исключением случая, когда каталог журналов находится в разделе NFS. Доступ к файлу должен быть только у суперпользователя (root).

PidFile — директива PidFile устанавливает имя файла, в который сервер записывает свой номер поцесса (process ID — pid). Файл должен читаться только суперпользователем (root). В большинстве случаев следует оставить значение по умолчанию.

User — директива User устанавливает идентификатор пользователя (userid), используемый сервером для ответа на запросы. Эта настройка определяет доступ к серверу. Любые файлы, недоступные для этого пользователя, будут также недоступны посетителям вашего сайта. Значение по умолчанию для User: "www-data".

Без полного понимания того, что вы делаете, не устанавливайте директиву User в значение root. Использование суперпользователя

[5] http://httpd.apache.org/docs/2.2/mod/mod_negotiation.html#multiviews

(root) как пользователя веб-сервера создаст очень серьёзные дыры в безопасности вашего сервера.

Group — директива Group подобна директиве User. Group устанавливает группу, под которой сервер будет отвечать на запросы. По умолчанию Group: "www-data".

1.2.4. Модули Apache2

Apache2 — модульный сервер. Это значит, что в ядро сервера включены только базовые функции. Расширенные возможности доступны в виде модулей, которые могут быть загружены в Apache2. По умолчанию, базовый набор модулей включается в сервер во время компиляции. Если сервер скомпилирован с возможностью использования динамически загруженных модулей, модули могут быть скомпилированы отдельно и добавлены в любое время с помощью директивы LoadModule. Иначе, Apache2 должен быть перекомпилирован для добавления и/или удаления модулей.

Ubuntu компилирует Apache2 с возможностью динамической загрузки модулей. Конфигурационные директивы могут быть включены для присутствия конкретного модуля при условии заключения их в блок *<IfModule>*.

Вы можете установить дополнительные модули для Apache2 и использовать их на вашем веб-сервере. Например, запустите следующую команду из командной строки терминала, чтобы установить модуль *MySQL Authentication*:

```
sudo apt-get install libapache2-mod-auth-mysql
```

Для дополнительных модулей смотрите каталог `/etc/apache2/mods-available`.

Чтобы включить модуль, используйте утилиту a2enmod:

```
sudo a2enmod auth_mysql
sudo service apache2 restart
```

Аналогично, a2dismod отключит модуль:

```
sudo a2dismod auth_mysql
sudo service apache2 restart
```

1.3. Настройка HTTPS

Модуль mod_ssl добавляет серверу Apache2 важную особенность — возможность защищённых коммуникаций. Соответственно, когда ваш

браузер соединяется с использованием SSL, в адресной строке браузера перед URL используется префикс https://.

Модуль mod_ssl доступен в составе пакета apache2-common. Выполните нижеследующую команду в терминале для включения модуля mod_ssl:

```
sudo a2enmod ssl
```

Пример настройки HTTPS содержится в файле `/etc/apache2/sites-available/default-ssl`. Для работы Apache2 с HTTPS также необходимы файлы *сертификата* и *ключа*. Базовая конфигурация HTTPS использует ключ и сертификат, генерируемые пакетом ssl-cert. Они подходят для тестирования, но позже должны быть заменены на уникальные для данного сайта или сервера. Прочесть о генерировании ключа и получении сертификата можно здесь *Раздел 5, «Сертификаты» [194]*.

Чтобы настроить Apache2 для HTTPS, введите следующее:

```
sudo a2ensite default-ssl
```

Каталоги `/etc/ssl/certs` и `/etc/ssl/private` — это места по умолчанию. Если вы установили сертификат и ключ в другие каталоги, убедитесь, что *SSLCertificateFile* и *SSLCertificateKeyFile* тоже изменены.

Теперь Apache2 сконфигурирован для HTTPS, перезапустите службу, чтобы активировать новые настройки:

```
sudo service apache2 restart
```

В зависимости от того, как вы получили сертификат, может потребоваться ввести пароль при запуске Apache2.

Вы можете получить доступ к страницам через безопасное соединение, набрав в адресной строке браузера https://your_hostname/url/.

1.4. Права разделения записи

Чтобы несколько пользователей имели возможность выполнять запись в один и тот же каталог, необходимо предоставить права записи группе, к которой они оба относятся. В следующем примере предоставляются права записи в `/var/www` группе "webmasters".

```
sudo chgrp -R webmasters /var/www
sudo find /var/www -type d -exec chmod g=rwxs "{}" \;
```

```
sudo find /var/www -type f -exec chmod g=rws "{}" \;
```

Если доступ к каталогу должен быть предоставлен более чем одной группе, необходимо включить списки контроля доступа (ACL).

1.5. Ссылки

- *Apache2 Documentation*[6] содержит подробную информацию по конфигурационным директивам Apache2. Смотрите также пакет apache2-doc, содержащий официальную документацию Apache2.
- Смотрите сайт *Mod SSL Documentation*[7] для дополнительной информации по SSL.
- Книга *Apache Cookbook*[8] издательства O'Reilly — хороший источник для освоения специфических настроек Apache2.
- По поводу специфических для Ubuntu вопросов по Apache2 обращайтесь на IRC канал *#ubuntu-server* в сети *freenode.net*[9].
- Хороший ресурс по интеграции PHP и MySQL *Apache MySQL PHP Ubuntu Wiki* [10].

[6] http://httpd.apache.org/docs/2.2/
[7] http://www.modssl.org/docs/
[8] http://oreilly.com/catalog/9780596001919/
[9] http://freenode.net/
[10] https://help.ubuntu.com/community/ApacheMySQLPHP

2. PHP5 — язык сценариев

PHP — язык сценариев общего назначения, применяемый веб-программистами. Сценари PHP могут встраиваться в HTML. В этом разделе описывается, как установить и настроить PHP5 в системе Ubuntu с Apache2 и MySQL.

В этом разделе предполагается, что вы установили и настроили веб-сервер Apache2 и сервер баз данных MySQL. Вы можете обратиться к разделам, посвящённым Apache2 и MySQL в данном документе, чтобы установить и настроить Apache2 и MySQL, соответственно.

2.1. Установка

PHP5 доступен в Ubuntu Linux. В отличие от python и perl, которые уже установлены в системе, PHP должен быть добавлен.

- Чтобы установить PHP5, вам нужно ввести следущую команду в терминале:

  ```
  sudo apt-get install php5 libapache2-mod-php5
  ```

  Вы можете запускать сценарии PHP5 из командной строки. Чтобы сделать это, вам следует установить пакет php5-cli. Для установки этого пакета введите в терминале:

  ```
  sudo apt-get install php5-cli
  ```

  Вы также можете запускать сценарии PHP5 без установленного модуля PHP5 Apache. Чтобы добиться этого, вам следует установить пакет php5-cgi. Для этого наберите в терминале:

  ```
  sudo apt-get install php5-cgi
  ```

  Для того, чтобы иметь возможность использовать MySQL с PHP5, вам необходимо установить пакет php5-mysql. Для установки php5-mysql вы можете воспользоваться следующей командой в окне терминала:

  ```
  sudo apt-get install php5-mysql
  ```

  Аналогично, для использования PostgerSQL с PHP5, вам понадобится установить пакет php5-pgsql. Для установки php5-pgsql введите в строке терминала:

```
sudo apt-get install php5-pgsql
```

2.2. Конфигурация

Установив PHP5, вы можете выполнять сценарии PHP5 на сервере по запросу вашего браузера. Если вы установили пакет php5-cli, то можете выполнять сценарии PHP5 из командной строки.

По умолчанию, веб-сервер Apache 2 сконфигурирован для выполнения сценариев PHP5. Другими словами, модуль PHP5 автоматически включается, когда вы устанавливаете модуль. Проверьте, существуют ли файлы `/etc/apache2/mods-enabled/php5.conf` и `/etc/apache2/mods-enabled/php5.load`. Если эти файлы отсутствуют, вы можете включить модуль с помощью команды **a2ebmod**.

После того, как вы установите соответствующие пакеты PHP5 и активируете модули PHP5 в Apache2, необходимо будет перезапустить Web-сервер Apache2, чтобы скрипты PHP5 исполнялись. Вы можете запустить следующую команду в терминале для перезапуска веб-сервера:

```
sudo service apache2 restart
```

2.3. Тестирование

Для проверки успешности установки, вы можете выполнить следующий PHP5 phpinfo скрипт:

```
<?php
  phpinfo();
?>
```

Вы можете сохранить содержимое в файле `phpinfo.php` и поместить его в каталог **DocumentRoot** веб-сервера Apache2. Указав в браузере адрес `http://hostname/phpinfo.php`, вы увидите значения различных конфигурационных параметров PHP5.

2.4. Ссылки

- Более полную информацию можно получить из документации на *php.net*[11].
- Существует множество книг по языку PHP. Можно порекомендовать две хорошие книги O'Reilly *Learning PHP 5*[12] and the *PHP Cook Book*[13].

[11] http://www.php.net/docs.php
[12] http://oreilly.com/catalog/9780596005603/
[13] http://oreilly.com/catalog/9781565926813/

- Кроме того, смотрите страницу *Apache MySQL PHP Ubuntu Wiki*[14] для дополнительной информации.

[14] https://help.ubuntu.com/community/ApacheMySQLPHP

3. Прокси-сервер Squid

Squid является полнофункциональным прокси сервером, который позволяет кэшировать HTTP, FTP, и другие популярные сетевые протоколы. Squid может реализовать кэширование и проксирование Secure Sockets Layer (SSL) запросов и кэширования DNS запросов, а также выполнять прозрачное кэширование. Squid также поддерживает широкий спектр протоколов кэширования, таких как интернет Cache Protocol (ICP), гипертекстовый протокол кэширования (HTCP), Cache Array Routing Protocol (CARP) и Web Cache Coordination Protocol (WCCP).

Сервер кэша прокси Squid - отличное решение для различных кэш и прокси-потребностей серверов - как филиалов, так и серверов корпоративного уровня, так как обеспечивает обширный, гранулярный механизм контроля доступа и наблюдение за критически важными параметрами через протокол простого управления сетью (SNMP). При выборе компьютерной системы для использования в качестве сервера кэша прокси Squid для большого количества пользователей убедитесь, что система сконфигурирована для работы с большими объёмами физической памяти, так как Squid сохраняет кэш в памяти для увеличения быстродействия.

3.1. Установка

В строке терминала введите следующую команду для установки сервера Squid:

```
sudo apt-get install squid3
```

3.2. Конфигурация

Squid настраивается путём редактирования директив, содержащихся в конфигурационном файле `/etc/squid3/squid.conf`. Следующие примеры иллюстрируют некоторые директивы, которые можно изменить, чтобы повлиять на поведение сервера Squid. За более подробной информацией о настройке Squid обратитесь к разделу «Ссылки».

Перед редактированием конфигурационного файла следует сделать копию исходного файла и защитить её от записи, чтобы у вас был образец исходных настроек и возможность восстановить их при необходимости. Сделайте копию и защитите её от записи с помощью следующих команд:

```
sudo cp /etc/squid3/squid.conf /etc/squid3/squid.conf.original
sudo chmod a-w /etc/squid3/squid.conf.original
```

- Для того, чтобы настроить порт, на котором будет работать сервер Squid, на 8888 (по умолчанию 3128), вам нужно изменить значение директивы http_port следующим образом:

```
http_port 8888
```

- Измените директиву visible_hostname для указания имени серверу Squid. Не обязательно, чтобы имя сервера Squid совпадало с именем компьютера. В данном примере оно установлено как *weezie*

```
visible_hostname weezie
```

- Используя контроль доступа Squid, можно разрешить использовать интернет-сервисы через Squid только пользователям с определённых IP-адресов. Для примера, мы покажем как предоставить доступ только пользователям из подсети 192.168.42.0/24:

  Добавьте следующее в **низ** секции ACL вашего файла `/etc/squid3/squid.conf`:

```
acl fortytwo_network src 192.168.42.0/24
```

  Затем добавьте следующее в **верх** секции http_access файла `/etc/squid3/squid.conf`:

```
http_access allow fortytwo_network
```

- Используя превосходные возможности управления доступом Squid, можно настроить доступность интернет-сервисов через Squid только в обычные рабочие часы. Например, мы продемонстрируем настройку доступа для сотрудников, которые работают с 9:00 до 17:00 с понедельника по пятницу и используют подсеть 10.1.42.0/24:

  Добавьте следующее в **низ** секции ACL вашего файла `/etc/squid3/squid.conf`:

```
acl biz_network src 10.1.42.0/24
acl biz_hours time M T W T F 9:00-17:00
```

  Затем добавьте следующее в **верх** секции http_access файла `/etc/squid3/squid.conf`:

```
http_access allow biz_network biz_hours
```

После внесения изменений в файл `/etc/squid3/squid.conf` сохраните файл и перезапустите сервер squid, чтобы задействовать изменения, введя в терминале следующую команду:

```
sudo service squid3 restart
```

3.3. Ссылки

Веб-сайт Squid[15]

Страница *Ubuntu Wiki Squid*[16].

[15] http://www.squid-cache.org/
[16] https://help.ubuntu.com/community/Squid

4. Ruby on Rails

Ruby on Rails — это веб-инфраструктура с открытым исходным кодом для разработки веб-приложений с базами данных. Она оптимизирована для обеспечения стабильной продуктивности работы программиста, поскольку она позволяет программисту писать код, предпочитая конвенцию конфигурации.

4.1. Установка

Перед установкой Rails необходимо установить Apache и MySQL. Для установки Apache, пожалуйста, обратитесь к *Раздел 1, «HTTPD - веб сервер Apache2» [213]*. Для инструкций по установке MySQL, обратитесь к *Раздел 1, «MySQL» [237]*.

Установив Apache и MySQL, можно приступать к установке Ruby on Rails.

Для установки базовых пакетов Ruby и Ruby on Rails, вы можете выполнить следующие команды в терминале:

```
sudo apt-get install rails
```

4.2. Конфигурация

Измените конфигурационный файл `/etc/apache2/sites-available/default` для настройки ваших доменов.

Первое, что подлежит изменению — это директива *DocumentRoot*:

```
DocumentRoot /path/to/rails/application/public
```

Далее, измените директиву <Directory "/path/to/rails/application/public">:

```
<Directory "/path/to/rails/application/public">
        Options Indexes FollowSymLinks MultiViews ExecCGI
        AllowOverride All
        Order allow,deny
        allow from all
        AddHandler cgi-script .cgi
</Directory>
```

Также следует разрешить Apache использовать модуль mod_rewrite. Для этого выполните следующее в строке терминала:

```
sudo a2enmod rewrite
```

Наконец, вам понадобится установить права владения каталогами `/path/to/rails/application/public` и `/path/to/rails/application/tmp` пользователю, используемому для запуска процесса Apache:

```
sudo chown -R www-data:www-data /path/to/rails/application/public
sudo chown -R www-data:www-data /path/to/rails/application/tmp
```

Вот и всё! Теперь ваш сервер может работать с приложениями Ruby on Rails.

4.3. Ссылки

- Для более детальной информации смотрите веб-сайт *Ruby on Rails*[17].
- Также существует великолепный сайт *Agile Development with Rails*[18].
- Дополнительная информация на странице *Ruby on Rails Ubuntu Wiki*[19].

[17] http://rubyonrails.org/
[18] http://pragprog.com/titles/rails3/agile-web-development-with-rails-third-edition
[19] https://help.ubuntu.com/community/RubyOnRails

5. Apache Tomcat

Apache Tomcat — это веб-контейнер, позволяющий вам обслуживать веб-приложения Java Servlets и JSP (Java Server Pages).

Ubuntu поддерживает пакеты для обеих версий — Tomcat 6 и 7. Tomcat 6 — это старая версия, а Tomcat 7 — текущая версия с реализованными новыми возможностями. Обе считаются стабильными. Данное руководство будет сосредоточено на Tomcat 7, но большинство сведений о конфигурации действительно для обеих версий.

Пакеты Tomcat в Ubuntu поддерживают два различных способа работы. Вы можете установить их как классическое приложение tomcat6 или tomcat7, которое будет работать сразу после включения компьютера с правами обычного непривилегированного пользователя. А также можете установить свои собственные инструкции, которые будут работать с вашими собственными правами пользователя и которые вы должны будете запускать и останавливать самостоятельно. Второй способ будет особенно полезен при разработке сервера, когда сразу несколько различных пользователей проводят тесты на своих личных экземплярах Tomcat.

5.1. Общесистемная установка

Для установки сервера Tomcat можно ввести следующую команду в приглашении терминала:

```
sudo apt-get install tomcat7
```

Это установит сервер Tomcat только со встроенным веб-приложением ROOT, которое выводит простейшую страницу "It works".

5.2. Конфигурация

Tomcat configuration files can be found in `/etc/tomcat7`. Only a few common configuration tweaks will be described here, please see *Tomcat 7.0 documentation*[20] for more.

5.2.1. Изменение портов по умолчанию

По умолчанию Tomcat открывает HTTP соединение на порту 8080 и AJP соединение на порту 8009. Возможно, вы захотите поменять порты по умолчанию, чтобы избежать конфликта с другими приложениями

[20] http://tomcat.apache.org/tomcat-7.0-doc/index.html

в системе. Это делается изменением следующих строк в `/etc/tomcat7/server.xml`:

```
<Connector port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
...
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
```

5.2.2. Изменение используемой JVM

По умолчанию Tomcat при запуске отдаёт предпочтение OpenJDK JVM, затем пытается найти Sun JVM, затем какие-либо другие виртуальные машины Java. Можно заставить Tomcat использовать указанную виртуальную машину, задав значение переменной среды JAVA_HOME в `/etc/default/tomcat7`:

```
JAVA_HOME=/usr/lib/jvm/java-6-sun
```

5.2.3. Объявление пользователей и ролей

Имена пользователей, пароли и роли (группы) могут быть определены централизованно в контейнере сервлетов. Это делается в файле `/etc/tomcat7/tomcat-users.xml`:

```
<role rolename="admin"/>
<user username="tomcat" password="s3cret" roles="admin"/>
```

5.3. Использование стандартных веб-приложений Tomcat

Tomcat поставляется с веб-приложениями, которые вы можете установить для документирования, администрирования или с демонстрационными целями.

5.3.1. Документация по Tomcat

Пакет tomcat7-docs содержит документацию Tomcat, упакованную как веб-приложение, доступ к которому можно получить по адресу http://ваш_сервер:8080/docs. Вы можете установить этот пакет, введя следующую команду в терминале:

```
sudo apt-get install tomcat7-docs
```

5.3.2. Веб-приложения для администрирования Tomcat

Пакет tomcat7-admin содержит два веб-приложения, которые можно использовать для администрирования сервера Tomcat через веб-интерфейс. Их можно установить с помощью следующей команды:

```
sudo apt-get install tomcat7-admin
```

Первое — это веб-приложение *manager*, которое по умолчанию доступно на http://yourserver:8080/manager/html. Оно в основном используется для получения статуса сервера и перезапуска web-приложений.

Доступ к приложению *manager* закрыт по умолчанию: для того, чтобы получить к нему доступ, нужно определить пользователя с ролью "manager-gui" в `/etc/tomcat7/tomcat-users.xml`.

Второе — это веб-приложение *host-manager*, которое по умолчанию доступно на http://yourserver:8080/host-manager/html. Оно может использоваться для динамического создания виртуальных хостов.

Доступ к приложению *host-manager* также закрыт по умолчанию: для доступа к нему необходимо определить пользователя с ролью "admin-gui" в `/etc/tomcat7/tomcat-users.xml`.

По соображениям безопасности пользователь tomcat7 по умолчанию не имеет права записи в каталог `/etc/tomcat7`. Некоторые функции этих административных веб-приложений (развёртывание приложений, создание виртуального хоста) требуют разрешения на запись в этот каталог. Если вы хотите использовать эти функции, выполните следующие действия, чтобы дать пользователям из группы tomcat7 необходимые права:

```
sudo chgrp -R tomcat7 /etc/tomcat7
sudo chmod -R g+w /etc/tomcat7
```

5.3.3. Примеры веб-приложений Tomcat

Пакет tomcat7-examples включает два сетевых приложения, которые могут быть использованы для тестирования или демонстрации сервлетов и возможностей JSP, по умолчанию вы можете получить к ним доступ по адресу http://yourserver:8080/examples. Вы можете установить их с помощью следующей команды:

```
sudo apt-get install tomcat7-examples
```

5.4. Использование частных сущностей

Tomcat широко используется в разработке и тестировании сценариев, используя одну систему не отвечающую требованиям нескольких пользователей.Tomcat пакеты в Ubuntu распространяются с инструментами для реализации собственных приложений, ориентированных на

пользователя , что позволяет каждому пользователю системы запускать (без прав администратора) отдельные экземпляры программы в то же время используя систему установленных библиотек.

Возможен запуск общесистемной и частных сущностей параллельно, так как они не используют одни и те же порты TCP.

5.4.1. Установка поддержки частных сущностей

Вы можете установить все необходимое для запуска частных сущностей, выполнив в терминале следующую команду:

```
sudo apt-get install tomcat7-user
```

5.4.2. Создание частной сущности

Вы можете создать каталог частной сущности, выполнив в терминале следующую команду:

```
tomcat7-instance-create my-instance
```

Это создаст новый каталог `my-instance` со всеми необходимыми подкаталогами и скриптами. Вы можете, например, установить общие библиотеки в подкаталог `lib/` и развернуть веб-приложения в подкаталоге `webapps/`. По умолчанию никакие веб-приложения не устанавливаются.

5.4.3. Настраиваем вашу частную сущность

Исходные конфигурационные файлы Tomcat для вашей частной установки, находятся в подкаталоге `conf/`. Вы можете, например, отредактировать файл `conf/server.xml` для изменения портов, используемых по умолчанию вашей персональной установкой Tomcat, во избежание конфликтов с другими запущенными установками.

5.4.4. Запуск/остановка вашей частной сущности

Вы можете запустить вашу частную сущность, введя следующую команду в строке терминала (предполагается, что сущность располагается в каталоге `my-instance`)

```
my-instance/bin/startup.sh
```

Вам следует проверить подкаталог `logs/` на наличие ошибок. Если вы получили ошибку *java.net.BindException: Address already in use<null>:8080*, это значит, что используемый вами порт уже занят, и вам следует изменить его.

Вы можете остановить вашу сущность, введя следующую команду в строке терминала (предполагается, что сущность располагается в каталоге `my-instance`)

```
my-instance/bin/shutdown.sh
```

5.5. Ссылки

- Для более детальной информации посетите сайт *Apache Tomcat*[21].
- *Tomcat: Полное руководство*[22] - хороший ресурс для сборки веб-приложений на основе Tomcat.
- Здесь можно увидеть список дополнительной литературы *Tomcat Books*[23].

[21] http://tomcat.apache.org/
[22] http://shop.oreilly.com/product/9780596003180.do
[23] http://wiki.apache.org/tomcat/Tomcat/Books

Глава 12. Базы данных

Ubuntu предоставляет два популярных сервера баз данных. Это:

- MySQL™
- PostgreSQL

Они доступны в основном хранилище (репозитории). В этом разделе рассматривается, как установить и настроить эти сервера баз данных:

1. MySQL

MySQL — это быстрый, многопоточный, многопользовательский и устойчивый SQL-сервер базы данных. Он предназначен как для ответственных сильнозагруженных производственных систем, так и для встраивания в массовое программное обеспечение.

1.1. Установка

Для установки MySQL выполните следующую команду в терминале:

```
sudo apt-get install mysql-server
```

В процессе установки у вас запросят пароль для пользователя root под MySQL.

Как только установка будет окончена, сервер MySQL должен будет автоматически запущен. Для того, чтобы проверить, запущен ли сервер MySQL или нет, можно воспользоваться командой:

```
sudo netstat -tap | grep mysql
```

После того, как вы запустите эту команду, вы должны увидеть нечто похожее:

```
tcp        0      0 localhost:mysql         *:*                 LISTEN      2556/mysqld
```

Если сервер не был запущен, то для запуска можно попробовать эту команду:

```
sudo service mysql restart
```

1.2. Конфигурация

Вы можете отредактировать файл `/etc/mysql/my.cnf` для настройки основных параметров — файл журнала, номер порта и пр. Например, чтобы настроить MySQL на ожидание подключений от компьютеров в сети, измените параметр *bind-address* на IP-адрес сервера:

```
bind-address            = 192.168.0.5
```

Замените 192.168.0.5 на реальное значение адреса вашего сервера.

После изменений в `/etc/mysql/my.cnf` сервис MySQL нужно перезагрузить:

```
sudo service mysql restart
```

Если вам потребовалось сменить пароль пользователя *root* в MySQL, введите в терминале:

```
sudo dpkg-reconfigure mysql-server-5.5
```

Сервис MySQL будет остановлен и вас попросят ввести новый пароль.

1.3. Драйверы базы данных

Хотя конфигурация по умолчанию для MySQL, предоставляемая пакетами Ubuntu, имеет великолепную функциональность и работает достаточно хорошо, есть некоторые вещи, которые вы можете решить до того как продолжить.

MySQL разработан так, что позволяет хранить данные по-разному. Эти варианты относятся к драйверам (управляющим модулям — engines) как баз данных, так и хранилищ. Существует два основных драйвера, которые вам могут быть интересны: InnoDB и MyISAM. Драйверы хранилищ прозрачны (незаметны) конечным пользователям. MySQL управляет событиями по-разному на нижнем уровне, но независимо от того, какая система хранения данных используется, вы будете взаимодействовать с базой одним и тем же способом.

Каждый драйвер имеет свои преимущества и недостатки.

Хотя смешивание и связывание драйверов баз данных на уровне таблиц разрешается и может быть привлекательным, это снижает эффективность настройки производительности, которую вы смогли бы провести при разделении ресурсов между двумя системами вместо замешивания их в одно целое.

- MyISAM — более старая из двух. Она может быть быстрее InnoDB при определенных обстоятельствах и предпочтительна при рабочей нагрузке, ориентированной на чтение данных. Некоторые интернет-приложения настроены на использование именно MyISAM (однако это не означает, что они будут медленнее под InnoDB). MyISAM также поддерживает тип данных FULLTEXT, который позволяет осуществлять очень быстрый поиск по большому количеству текстовых данных. Однако MyISAM поддерживает блокировку записи только на уровне таблиц. Это означает, что только один процесс может изменять данные в таблице в один момент времени. Поскольку некоторые приложения, использующие таблицу, могут масштабироваться (работать несколькими экземплярами

— scales), это может стать серьёзной помехой. Здесь также отсутствует журналирование, что может усложнить восстановление данных после сбоя. Следующая ссылка предоставляет некоторые соображения по использованию *MyISAM on a production database*[1].

- InnoDB — более современный драйвер, созданный по принципам *ACID compliant*[2], что гарантирует надежную обработку транзакций базы данных. Блокировка записи производится на уровне одной записи в таблице. Это означает возможность нескольких изменений в одной таблице одновременно. Кэширование данных происходит также и в оперативной памяти внутри драйвера базы данных, позволяя кэшировать более эффективно чем на уровне блоков файлов. В соответствии с ACID все транзакции журналируются независимо от основных таблиц. Это позволяет намного более надёжно восстанавливать данные при проверке целостности данных.

Начиная MySQL 5.5, InnoDB является драйвером по умолчанию и настоятельно рекомендуется вместо MyISAM, если только у вас нет специфических потребностей, уникальных для этого драйвера.

1.4. Расширенные настройки

1.4.1. Создание настроенного файла my.cnf

Существует ряд параметров, которые могут быть указаны в файле настроек MySQL, что со временем позволит вам повысить производительность вашего сервера. Для начальной настройки вам может пригодиться *Percona's my.cnf generating tool*[3]. Этот инструмент позволит вам создать файл my.cnf, более оптимизированный под специфические возможности вашего сервера и ваши требования.

Do not replace your existing my.cnf file with Percona's one if you have already loaded data into the database. Some of the changes that will be in the file will be incompatible as they alter how data is stored on the hard disk and you'll be unable to start MySQL. If you do wish to use it and you have existing data, you will need to carry out a mysqldump and reload:

```
mysqldump --all-databases --routines -u root -p > ~/fulldump.sql
```

This will then prompt you for the root password before creating a copy of the data. It is advisable to make sure there are no other users or processes using the database whilst this takes place. Depending on how much data you've got

[1] http://www.mysqlperformanceblog.com/2006/06/17/using-myisam-in-production/
[2] http://en.wikipedia.org/wiki/ACID
[3] http://tools.percona.com/members/wizard

in your database, this may take a while. You won't see anything on the screen during this process.

Once the dump has been completed, shut down MySQL:

```
sudo service mysql stop
```

Now backup the original my.cnf file and replace with the new one:

```
sudo cp /etc/mysql/my.cnf /etc/mysql/my.cnf.backup
sudo cp /path/to/new/my.cnf /etc/mysql/my.cnf
```

Then delete and re-initialise the database space and make sure ownership is correct before restarting MySQL:

```
sudo rm -rf /var/lib/mysql/*
sudo mysql_install_db
sudo chown -R mysql: /var/lib/mysql
sudo service mysql start
```

Finally all that's left is to re-import your data. To give us an idea of how far the import process has got you may find the 'Pipe Viewer' utility, pv, useful. The following shows how to install and use pv for this case, but if you'd rather not use it just replace pv with cat in the following command. Ignore any ETA times produced by pv, they're based on the average time taken to handle each row of the file, but the speed of inserting can vary wildly from row to row with mysqldumps:

```
sudo apt-get install pv
pv ~/fulldump.sql | mysql
```

Once that is complete all is good to go!

Эта операция не обязательна для всех изменений my.cnf. Многие значения, которые вы захотите поменять для улучшения производительности, сработают даже на работающем сервере. Но как всегда не забудьте сделать надёжную копию файлов настроек и данных перед внесением изменений.

1.4.2. MySQL Tuner

MySQL Tuner — это полезный инструмент, который подсоединяется к работающему MySQL и предлагает варианты, как можно улучшить настройки для вашей рабочей нагрузки. Чем дольше работает сервер, тем лучше рекомендации предоставит mysqltuner. Для рабочего окружения подождите как минимум 24 часа, прежде чем запускать утилиту. Вы можете установить mysqltuner из хранилища Ubuntu:

```
sudo apt-get install mysqltuner
```

После установки запустите её:

```
mysqltuner
```

и ждите её финального отчета. Верхняя секция предоставляет общую информацию а сервере баз данных, а нижняя часть содержит рекомендации по настройке, необходимые для изменения вашего my.cnf. Многие из них могут быть поправлены вживую на сервере без перезагрузки. Смотрите официальную документацию MySQL (указанную в разделе Ссылки) для перечня параметров, изменяемых "на лету". Далее часть примерного отчета по работающей базе, который показывает, что можно извлечь некоторую пользу от увеличения размера кэша запросов:

```
-------- Recommendations -------------------------------------------------------
General recommendations:
    Run OPTIMIZE TABLE to defragment tables for better performance
    Increase table_cache gradually to avoid file descriptor limits
Variables to adjust:
    key_buffer_size (> 1.4G)
    query_cache_size (> 32M)
    table_cache (> 64)
    innodb_buffer_pool_size (>= 22G)
```

Один финальный комментарий по настройке базы данных: Хотя мы можем утверждать что определённые настройки самые лучшие, производительность может изменяться от приложения к приложению. Например, что работает великолепно для Wordpress, может оказаться не лучшим для Drupal, Joomla или проприетарных приложений. Производительность зависит от типов запросов, использования индексов, насколько эффективно спроектирована база данных и т.д. Вы можете посчитать полезным потратить некоторое время на поиск настроек базы данных под используемые вами приложения. Как только вы пройдёте определённую точку, любые ваши изменения будут приводить к минимальным улучшениям и вам будет лучше либо заняться улучшением ваших приложений, либо масштабировать вашу базу данных, используя более производительное оборудование или добавляя зависимые сервера.

1.5. Ресурсы

- Смотрите *MySQL Home Page*[4] для дополнительной информации.

[4] http://www.mysql.com/

- Полная документация доступна в форматах как онлайн, так и оффлайн по ссылке *MySQL Developers portal*[5]
- Для общей информации по SQL смотрите *Using SQL Special Edition*[6] от Rafe Colburn.
- Страница *Apache MySQL PHP Ubuntu Wiki*[7] также содержит полезную информацию.

[5] http://dev.mysql.com/doc/
[6] http://www.informit.com/store/product.aspx?isbn=0768664128
[7] https://help.ubuntu.com/community/ApacheMySQLPHP

2. PostgreSQL

PostgreSQL — это объектно-реляционная система управления базами данных, объединяющая возможности традиционных коммерческих систем управления базами данных с улучшениями, имеющимися в СУБД нового поколения.

2.1. Установка

Для установки PostgreSQL выполните в терминале следующую команду:

```
sudo apt-get install postgresql
```

После того, как установка будет завершена, вам следует настроить сервер PostgreSQL в соответствии с вашими потребностями, хотя конфигурация по умолчанию также вполне жизнеспособна.

2.2. Конфигурация

PostgreSQL supports multiple client authentication methods. IDENT authentication method is used for postgres and local users, unless otherwise configured. Please refer to the *PostgreSQL Administrator's Guide*[8] if you would like to configure alternatives like Kerberos.

The following discussion assumes that you wish to enable TCP/IP connections and use the MD5 method for client authentication. PostgreSQLconfiguration files are stored in the `/etc/postgresql/<version>/main` directory. For example, if you install PostgreSQL 9.1, the configuration files are stored in the `/etc/postgresql/9.1/main` directory.

Для настройки *ident* аутентификации, необходимо добавить записи в файл `/etc/postgresql/9.1/main/pg_ident.conf`. В файле есть подробные комментарии, чтобы помочь вам.

Чтобы позволить другим компьютерам подключаться к вашему серверу PostgreSQL, отредактируйте файл `/etc/postgresql/9.1/main/postgresql.conf`

Найдите строку *#listen_addresses = 'localhost'* и замените её на:

```
listen_addresses = '*'
```

Чтобы разрешить как соединения IPv4, так и IPv6, замените 'localhost' на '::'

[8] http://www.postgresql.org/docs/9.1/static/admin.html

Вы можете также отредактировать все остальные параметры, если знаете, что делаете! За подробностями обратитесь к конфигурационному файлу или документации PostgreSQL.

Now that we can connect to our PostgreSQL server, the next step is to set a password for the *postgres* user. Run the following command at a terminal prompt to connect to the default PostgreSQL template database:

```
sudo -u postgres psql template1
```

The above command connects to PostgreSQL database *template1* as user *postgres*. Once you connect to the PostgreSQL server, you will be at a SQL prompt. You can run the following SQL command at the psql prompt to configure the password for the user *postgres*.

```
ALTER USER postgres with encrypted password 'your_password';
```

После настройки пароля, отредактируйте файл `/etc/postgresql/9.1/main/pg_hba.conf` для использования *MD5* аутентификации с пользователем *postgres*:

```
local   all         postgres                          md5
```

Под конец вам потребуется перезапустить сервис PostgreSQL для применения новых настроек. Из терминала выполните следующее для перезапуска PostgreSQL:

```
sudo service postgresql restart
```

The above configuration is not complete by any means. Please refer to the *PostgreSQL Administrator's Guide*[9] to configure more parameters.

You can test server connections from other machines by using the PostgreSQL client.

```
sudo apt-get install postgresql-client
psql -h postgres.example.com -U postgres -W
```

Замените указанное доменное имя на доменное имя вашего реального сервера.

[9] http://www.postgresql.org/docs/9.1/static/admin.html

2.3. Резервное копирование

PostgreSQL databases should be backed up regularly. Refer to the *PostgreSQL Administrator's Guide*[10] for different approaches.

2.4. Ресурсы

- As mentioned above the *PostgreSQL Administrator's Guide*[11] is an excellent resource. The guide is also available in the postgresql-doc-9.1 package. Execute the following in a terminal to install the package:

  ```
  sudo apt-get install postgresql-doc-9.1
  ```

  Чтобы просмотреть руководство, введите **file:///usr/share/doc/postgresql-doc-9.1/html/index.html** в адресную строку вашего браузера.
- Для общей информации по SQL смотрите *Using SQL Special Edition*[12] от Rafe Colburn.
- Также смотрите страницу *PostgreSQL Ubuntu Wiki*[13] для дополнительной информации.

[10] http://www.postgresql.org/docs/9.1/static/backup.html
[11] http://www.postgresql.org/docs/9.1/static/admin.html
[12] http://www.informit.com/store/product.aspx?isbn=0768664128
[13] https://help.ubuntu.com/community/PostgreSQL

Глава 13. Приложения LAMP

1. Обзор

Установка LAMP (Linux + Apache + MySQL + PHP/Perl/Python) является популярным вариантом настройки серверов Ubuntu. Существует множество приложений с открытым кодом, написанных с использованием стека приложений LAMP. Популярными приложениями LAMP являются wiki-энциклопедии, системы управления содержимым (CMS) и управляющие приложения, такие как phpMyAdmin.

Одним из преимуществ LAMP является значительная гибкость в выборе различных баз данных, веб-серверов и языков сценариев. Популярной заменой для MySQL служат PostgreSQL и SQLite. Python, Perl и Ruby также часто заменяют PHP. A Nginx, Cherokee и Lighttpd могут заменять Apache.

Самым быстрым способом установить LAMP является использование tasksel. Tasksel — это инструмент Debian/Ubuntu, который устанавливает несколько зависимых пакетов в вашу систему в качестве единой "задачи". Для установки LAMP сервера:

- В терминале введите следующую команду:

```
sudo tasksel install lamp-server
```

После установки вы можете поставить большинство *LAMP* приложений следующим образом:

- Загрузите архив, содержащий файлы с исходным кодом приложения.
- Распакуйте архив в каталог, доступный веб-серверу.
- В зависимости от того, куда распакованы файлы, настройте веб-сервер на их обработку.
- Настройте приложение на доступ к базе данных.
- Выполните сценарий (script) или загрузите страницу приложения для установки базы данных, необходимой приложению.
- Когда шаги, указанные выше или подобные им, выполнены, вы готовы начать использовать приложение.

Неудобство использования такого подхода заключается в нестандартном способе установки файлов приложения на файловую систему, что может привести к беспорядку в выборе мест установки приложений. Другим большим неудобством является обновление приложений. При выпуске новой версии, этот же процесс используется для установки обновляемого приложения.

К счастью, ряд приложений *LAMP* уже упакованы для Ubuntu и доступны для установки так же, как и обычные (не-LAMP) приложения. Однако для некоторых таких приложений могут потребоваться дополнительные шаги по установке и настройке.

В этом разделе описано, как установить некоторые приложения *LAMP*.

2. Moin Moin

MoinMoin — это Wiki-движок реализованный на языке Python, основанный на движке Wiki PikiPiki и распространяемый под лицензией GNU GPL.

2.1. Установка

Для установки MoinMoin выполните следующую команду в командной строке:

```
udo apt-get install python-moinmoin
```

Вам также понадобится установить веб-сервер apache2. Для этого, просмотрите подсекцию *Раздел 1.1, «Установка» [213]* в секции *Раздел 1, «HTTPD - веб сервер Apache2» [213]*.

2.2. Конфигурация

Для настройки своего первого приложения Wiki выполните следующий набор команд. Предположим, что вы создаете Wiki с именем *mywiki*:

```
cd /usr/share/moin
sudo mkdir mywiki
sudo cp -R data mywiki
sudo cp -R underlay mywiki
sudo cp server/moin.cgi mywiki
sudo chown -R www-data.www-data mywiki
sudo chmod -R ug+rwX mywiki
sudo chmod -R o-rwx mywiki
```

Сейчас вам рекомендуется настроить MoinMoin чтобы найти вашу новую Wiki *mywiki*. Для настройки MoinMoin откройте файл `/etc/moin/mywiki.py` и измените следующую строку:

```
data_dir = '/org/mywiki/data'
```

на

```
data_dir = '/usr/share/moin/mywiki/data'
```

Также ниже *data_dir* добавьте опцию *data_underlay_dir*:

```
data_underlay_dir='/usr/share/moin/mywiki/underlay'
```

Если файла `/etc/moin/mywiki.py` не существует, вы можете скопировать файл `/usr/share/moin/config/wikifarm/mywiki.py` в `/etc/moin/mywiki.py` и провести соответствующие изменения, описанные выше.

Если вы назвали Wiki как *my_wiki_name*, то введите строку «("my_wiki_name", r".*")» в файл `/etc/moin/farmconfig.py` после строки «("mywiki", r".*")».

После того, как вы настроили MoinMoin для поиска *mywiki*, нужно настроить apache2 и подготовить его для вашего Wiki-приложения.

Добавьте следующие строки в файл `/etc/apache2/sites-available/default` внутри тега «<VirtualHost *>»:

```
### moin
  ScriptAlias /mywiki "/usr/share/moin/mywiki/moin.cgi"
  alias /moin_static193 "/usr/share/moin/htdocs"
  <Directory /usr/share/moin/htdocs>
  Order allow,deny
  allow from all
  </Directory>
### end moin
```

После того, как вы настроили веб-сервер apache2 и подготовили его для вашего приложения Wiki, перезапустите его. Вы можете выполнить следующею команду, чтобы перезапустить веб-сервер apache2:

```
sudo service apache2 restart
```

2.3. Проверка

Вы можете проверить приложение Wiki и убедиться, что оно работает, введя следующий URL:

```
http://localhost/mywiki
```

За дополнительными подробностями обратитесь к веб-сайту *MoinMoin*[1].

2.4. Ссылки

- Для дополнительной информации смотрите *moinmoin Wiki*[2].
- Также обратитесь к странице *Ubuntu Wiki MoinMoin*[3].

[1] http://moinmo.in/
[2] http://moinmo.in/
[3] https://help.ubuntu.com/community/MoinMoin

3. MediaWiki

MediaWiki является веб-ориентированным Wiki-приложением, написанном на языке PHP. Оно может использовать систему управления базами данных MySQL или PostgreSQL.

3.1. Установка

Before installing MediaWiki you should also install Apache2, the PHP5 scripting language and a Database Management System. MySQL or PostgreSQL are the most common, choose one depending on your need. Please refer to those sections in this manual for installation instructions.

Для установки MediaWiki выполните следующую команду в командной строке:

```
sudo apt-get install mediawiki php5-gd
```

Для расширения функциональности MediaWiki смотрите пакет mediawiki-extensions.

3.2. Конфигурация

Конфигурационный файл Apache `mediawiki.conf` для MediaWiki установлен в каталог `/etc/apache2/conf.d/`. Вам нужно раскомментировать следующую строку в этом файле для доступа к приложениям MediaWiki.

```
# Alias /mediawiki /var/lib/mediawiki
```

После того, как вы раскомментируете указанную выше строку, перезапустите сервер Apache и осуществите доступ к MediaWiki по следующему URL:

```
http://localhost/mediawiki/config/index.php
```

Пожалуйста, прочтите раздел «Проверка окружения...» на этой странице. Вы сможете решить многие вопросы, внимательно прочтя его.

После завершения настройки вам нужно скопировать файл `LocalSettings.php` в каталог `/etc/mediawiki`:

```
sudo mv /var/lib/mediawiki/config/LocalSettings.php /etc/mediawiki/
```

Вы можете также отредактировать `/etc/mediawiki/LocalSettings.php`, чтобы установить лимит используемой памяти (отключено по умолчанию):

```
ini_set( 'memory_limit', '64M' );
```

3.3. Расширения

Расширения добавляют новые возможности и расширяют функциональность приложения MediaWiki. Расширения дают wiki-администраторам и конечным пользователям возможность подстраивать MediaWiki под их требования.

Вы можете загрузить расширения MediaWiki в виде архива или получить их из репозитория Subversion. Вам нужно будет скопировать их в каталог `/var/lib/mediawiki/extensions`. Также потребуется добавить следующую строку в конец файла: `/etc/mediawiki/LocalSettings.php`.

```
require_once "$IP/extensions/ExtentionName/ExtentionName.php";
```

3.4. Ссылки

- Для получения более подробной информации перейдите на сайт *MediaWiki*[4].
- *MediaWiki Administrators*' *Tutorial Guide*[5] содержит множество информации для новых администраторов MediaWiki.
- Также хорошим ресурсом является страница *Ubuntu Wiki MediaWiki*[6] .

[4] http://www.mediawiki.org
[5] http://www.packtpub.com/Mediawiki/book
[6] https://help.ubuntu.com/community/MediaWiki

4. phpMyAdmin

phpMyAdmin — это приложение LAMP, специально созданное для администрирования серверов MySQL. Написанное на PHP и доступное через веб-браузер, приложение phpMyAdmin предоставляет графический интерфейс для задач администрирования базы данных.

4.1. Установка

Перед установкой phpMyAdmin вам понадобится доступ к базе данных MySQL либо на том же хосте, на который установлен phpMyAdmin, либо на хосте, доступном по сети. Для дополнительной информации смотрите *Раздел 1, «MySQL» [237]*. Наберите в терминале:

```
sudo apt-get install phpmyadmin
```

По запросу выберите, какой веб-сервер будет настроен для phpMyAdmin. В этом разделе предполагается использование в качестве веб-сервера Apache2.

In a browser go to *http://servername/phpmyadmin*, replacing *servername* with the server's actual hostname. At the login, page enter *root* for the *username*, or another MySQL user, if you have any setup, and enter the MySQL user's password.

Как только вы авторизуетесь, вы сможете при необходимости сменить пароль пользователя *root*, создавать пользователей, создавать/удалять базы данных, таблицы и прочее.

4.2. Конфигурация

Конфигурационные файлы для phpMyAdmin размещаются в `/etc/phpmyadmin`. Главный конфигурационный файл — это `/etc/phpmyadmin/config.inc.php`. Этот файл содержит конфигурационные опции, которые глобально применяются к phpMyAdmin.

Чтобы использовать phpMyAdmin для администрирования базы данных MySQL, расположенной на другом сервере, измените следующее в `/etc/phpmyadmin/config.inc.php`:

```
$cfg['Servers'][$i]['host'] = 'db_server';
```

Замените *db_server* на имя или IP-адрес реального сервера удалённой базы данных. Убедитесь также, что хост phpMyAdmin имеет права доступа к удалённой базе данных.

После настройки выйдите из phpMyAdmin и зайдите снова, и вы получите доступ к новому серверу.

Файлы `config.header.inc.php` и `config.footer.inc.php` используются для добавления верхнего и нижнего HTML-заголовков для phpMyAdmin.

Другим важным конфигурационным файлом является `/etc/phpmyadmin/apache.conf`, который является символьной ссылкой на `/etc/apache2/conf.d/phpmyadmin.conf` и используется для настройки Apache2 по обслуживанию сайта phpMyAdmin. Файл содержит настройки по загрузке PHP, правам доступа к каталогу и прочее. Для получения дополнительной информации о настройке Apache2 смотрите раздел *Раздел 1, «HTTPD - веб сервер Apache2» [213]*.

4.3. Ссылки

- Документация по phpMyAdmin устанавливается из пакета и доступна по ссылке *phpMyAdmin Documentation* (в виде знака вопроса в обрамлении) под логотипом phpMyAdmin. Официальная документация также доступна на сайте *phpMyAdmin*[7].
- Также хороший ресурс *Mastering phpMyAdmin*[8].
- Ещё один ресурс — это страница *phpMyAdmin Ubuntu Wiki*[9].

[7] http://www.phpmyadmin.net/home_page/docs.php
[8] http://www.packtpub.com/phpmyadmin-3rd-edition/book
[9] https://help.ubuntu.com/community/phpMyAdmin

5. WordPress

Wordpress is a blog tool, publishing platform and CMS implemented in PHP and licensed under the GNU GPLv2.

5.1. Установка

Чтобы установить WordPress, выполните следующую команду в терминале:

```
sudo apt-get install wordpress
```

Следует также установить веб-сервер apache2 и сервер mysql. Для установки веб-сервера apache2 обратитесь к подразделу *Раздел 1.1, «Установка» [213]* раздела *Раздел 1, «HTTPD - веб сервер Apache2» [213]*. Для установки сервера mysql обратитесь к подразделу *Раздел 1.1, «Установка» [237]* раздела *Раздел 1, «MySQL» [237]*.

5.2. Конфигурация

For configuring your first WordPress application, configure an apache site. Open `/etc/apache2/sites-available/wordpress.conf` and write the following lines:

```
Alias /blog /usr/share/wordpress
<Directory /usr/share/wordpress>
    Options FollowSymLinks
    AllowOverride Limit Options FileInfo
    DirectoryIndex index.php
    Order allow,deny
    Allow from all
</Directory>
<Directory /usr/share/wordpress/wp-content>
    Options FollowSymLinks
    Order allow,deny
    Allow from all
</Directory>
```

Enable this new WordPress site

```
sudo a2ensite wordpress
```

После того, как вы настроили веб-сервер apache2 и подготовили его для вашего приложения WordPress, нужно перезагрузить его. Можете выполнить следующую команду для перезапуска веб-сервера apache2:

```
sudo service apache2 restart
```

To facilitate multiple WordPress installations, the name of this configuration file is based on the Host header of the HTTP request. This means that you can have a configuration per VirtualHost by simply matching the hostname portion of this configuration with your Apache Virtual Host. e.g. /etc/wordpress/config-10.211.55.50.php, /etc/wordpress/config-hostalias1.php, etc. These instructions assume you can access Apache via the localhost hostname (perhaps by using an ssh tunnel) if not, replace /etc/wordpress/config-localhost.php with /etc/wordpress/config-NAME_OF_YOUR_VIRTUAL_HOST.php.

Once the configuration file is written, it is up to you to choose a convention for username and password to mysql for each WordPress database instance. This documentation shows only one, localhost, example.

Теперь настройте WordPress для использования базы данных MySQL. Откройте файл `/etc/wordpress/config-localhost.php` и вставьте следующие строки:

```
<?php
define('DB_NAME', 'wordpress');
define('DB_USER', 'wordpress');
define('DB_PASSWORD', 'yourpasswordhere');
define('DB_HOST', 'localhost');
define('WP_CONTENT_DIR', '/usr/share/wordpress/wp-content');
?>
```

Теперь создайте эту базу данных MySQL. Откройте временный файл с командами MySQL `wordpress.sql` и вставьте следующие строки:

```
CREATE DATABASE wordpress;
GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP,ALTER
ON wordpress.*
TO wordpress@localhost
IDENTIFIED BY 'yourpasswordhere';
FLUSH PRIVILEGES;
```

Выполните эти команды.

```
cat wordpress.sql | sudo mysql --defaults-extra-file=/etc/mysql/debian.cnf
```

Your new WordPress can now be configured by visiting *http://localhost/blog/wp-admin/install.php*. (Or *http://NAME_OF_YOUR_VIRTUAL_HOST/blog/wp-admin/install.php* if your server has no GUI and you are completing WordPress configuration via a web browser running on another computer.) Fill out the Site Title, username, password, and E-mail and click Install WordPress.

Note the generated password (if applicable) and click the login password. Your WordPress is now ready for use.

5.3. Ссылки

- *WordPress.org Codex*[10]
- *Ubuntu Wiki WordPress*[11]

[10] https://codex.wordpress.org/
[11] https://help.ubuntu.com/community/WordPress

Глава 14. Файл-серверы

Если у вас в одной сети более одного компьютера, то в какой-то момент вы наверняка захотите обмениваться файлами между ними. В этой секции рассмотрена установка и настройка серверов FTP, NFS и CUPS.

1. FTP-сервер

Протокол передачи файлов (FTP) — это протокол TCP для передачи файлов между компьютерами. В прошлом он использовался также для загрузки файлов на сервер в интернете, но, поскольку этот метод не использует шифрование, пользовательские данные и содержимое файлов передаются в открытую и легко перехватываются. Поэтому, если вы здесь ищете способ безопасно передавать и загружать файлы, лучше обратитесь к статье по OpenSSH в разделе *Глава 6, Удалённое администрирование [92]*.

FTP работает на основе модели клиент/сервер. Серверный компонент называется *сервисом FTP*. Он постоянно слушает FTP-запросы от удалённых клиентов. При получении запроса он управляет входом и установкой соединения. На протяжении сессии он выполняет любые команды, переданные клиентом FTP.

Доступ к FTP-серверу может быть установлен двумя путями:

- Анонимный
- Авторизованный

В анонимном режиме удалённый клиент может получить доступ к FTP-серверу, используя учётную запись пользователя по умолчанию с именем «anonymous» или «ftp» и передав адрес электронной почты в качестве пароля. В авторизованном режиме пользователь должен иметь учётное имя и пароль. Этот последний вариант крайне небезопасный и не должен использоваться за исключением специальных обстоятельств. Если вы хотите передавать файлы безопасно, смотрите SFTP в разделе по OpenSSH серверу. Пользовательский доступ к каталогам и файлам FTP сервера зависит от прав доступа пользователя, указанного при входе. Как правило, сервис FTP скрывает корневой каталог FTP сервера, подменяя его на домашний каталог FTP. Это скрывает корень файловой системы от удалённых сессий.

1.1. vsftpd — установка FTP-сервера

vsftpd — это демон FTP, доступный в Ubuntu. Его легко устанавливать, настраивать и поддерживать. Для установки vsftpd вы можете выполнить следующую команду:

```
sudo apt-get install vsftpd
```

1.2. Настройка анонимного доступа по FTP

Настройка vsftpd по умолчанию *не разрешает* анонимную загрузку. Если вы хотите разрешить анонимную загрузку, измените в `/etc/vsftpd.conf` следующее:

```
anonymous_enable=Yes
```

В процессе установки создается пользователь *ftp* с домашним каталогом `/srv/ftp`. Это каталог по умолчанию для FTP.

Если вы желаете поменять его расположение, например, на `/srv/files/ftp`, просто создайте новый каталог и измените домашний каталог пользователя *ftp*:

```
sudo mkdir /srv/files/ftp
sudo usermod -d /srv/files/ftp ftp
```

После изменений перезапустите vsftpd:

```
sudo restart vsftpd
```

Под конец скопируйте все файлы и каталоги, которые вы хотите сделать доступными для анонимного FTP, в `/srv/files/ftp`, или `/srv/ftp`, если вы хотите оставить настройки по умолчанию.

1.3. Настройка авторизованного доступа по FTP

По умолчанию vsftpd настроен на аутентификацию системных пользователей с возможностью скачивать файлы. Если вы хотите разрешить пользователям загружать файлы на сервер, измените в `/etc/vsftpd.conf`:

```
write_enable=YES
```

после чего перезагрузите vsftpd:

```
sudo restart vsftpd
```

Теперь при входе системных пользователей по FTP они будут попадать в свои *домашние* каталоги, где они смогут скачивать и загружать файлы, создавать каталоги и т.д.

Аналогично, по умолчанию анонимный пользователь не имеет возможности загружать файлы на FTP-сервер. Для изменения этой настройки уберите комментарий на следующей строке и перезапустите vsftpd:

```
anon_upload_enable=YES
```

Разрешение анонимному пользователю загружать файлы на сервер может оказаться серьёзной угрозой безопасности. Лучше не разрешать анонимную загрузку файлов на серверы с прямым доступом из интернета.

Конфигурационный файл содержит много параметров настройки. Информация по каждому параметру доступна в этом же файле. В качестве альтернативы вы можете посмотреть системное руководство по команде **man 5 vsftpd.conf** для уточнения деталей по каждому параметру.

1.4. Защита FTP

В `/etc/vsftpd.conf` существуют опции, помогающие сделать vsftpd более безопасным. Например, пользователи могут быть ограничены своими домашними каталогами, если раскомментировать:

```
chroot_local_user=YES
```

Вы также можете определить список пользователей, имеющих доступ только в домашний каталог:

```
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd.chroot_list
```

После снятия комментариев с этих опций, создайте `/etc/vsftpd.chroot_list`, содержащий список пользователей по одному на строку. Затем перезапустите vsftpd:

```
sudo restart vsftpd
```

Аналогично, файл `/etc/ftpusers` содержит список пользователей, которым *запрещён* доступ по FTP. По умолчанию он включает root, daemon, nobody и т.п. Для запрета доступа по FTP для дополнительных пользователей, просто добавьте их в этот список.

FTP может быть зашифрованным при использовании *FTPS*. В отличие от *SFTP*, *FTPS* — это FTP поверх SSL. *SFTP* — это сессия, подобная FTP, по зашифрованному *SSH* соединению. Основное отличие заключается в том, что пользователи SFTP должны иметь учётную запись с собственным окружением *shell* вместо оболочки *nologin*. Предоставление всем пользователям доступа к оболочке может оказаться не лучшим решением для некоторых систем, таких как веб-сервер общего доступа. Однако

есть возможность ограничить такие учетные записи только SFTP и запретить взаимодействие с оболочкой. Смотрите раздел по OpenSSH для дополнительной информации.

Для настройки *FTPS*, добавьте в конец файла `/etc/vsftpd.conf` следующее:

```
ssl_enable=Yes
```

Также обратите внимание на опции сертификата и ключа:

```
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
```

По умолчанию эти опции установлены в значения, предоставленные пакетом ssl-cert. Для рабочей среды они должны быть заменены на сертификат и ключ, созданные для определённого компьютера. Для дополнительной информации смотрите раздел *Раздел 5, «Сертификаты» [194]*.

Теперь перегрузите vsftpd и неанонимные пользователи будут использовать *FTPS*:

```
sudo restart vsftpd
```

Чтобы позволить пользователям с оболочкой `/usr/sbin/nologin` получить доступ к FTP, но не предоставлять shell доступ, отредактируйте `/etc/shells`, добавив к оболочке *nologin*:

```
# /etc/shells: valid login shells
/bin/csh
/bin/sh
/usr/bin/es
/usr/bin/ksh
/bin/ksh
/usr/bin/rc
/usr/bin/tcsh
/bin/tcsh
/usr/bin/esh
/bin/dash
/bin/bash
/bin/rbash
/usr/bin/screen
/usr/sbin/nologin
```

Это необходимо, поскольку по умолчанию vsftpd использует аутентификацию PAM, а файл настроек `/etc/pam.d/vsftpd` содержит:

```
auth    required        pam_shells.so
```

Модуль PAM *shells* ограничивает доступ к оболочкам, перечисленным в файле `/etc/shells`.

Наиболее популярные клиенты FTP могут быть настроены на использование FTPS. FTP-клиент командной строки lftp также имеет возможность использовать FTPS.

1.5. Ссылки

- Обратитесь к сайту *vsftpd website*[1] для дополнительной информации.
- For detailed `/etc/vsftpd.conf` options see the *vsftpd.conf man page*[2].

[1] http://vsftpd.beasts.org/vsftpd_conf.html
[2] http://manpages.ubuntu.com/manpages/trusty/en/man5/vsftpd.conf.5.html

2. Сетевая файловая система (NFS)

NFS позволяет системе предоставлять в общий сетевой доступ каталоги и файлы. Посредством NFS, пользователи и программы могут получать доступ к файлам на удаленных машинах так же легко, как будто это файлы на их локальном компьютере.

Некоторые из преимуществ, которые может обеспечить NFS:

- Рабочие станции используют меньше локального дискового пространства, так как общие данные могут содержаться на одной машине и оставаться доступными по сети для всех остальных.
- У пользователей отпадает необходимость в использовании отдельных домашних каталогов на каждой машине, подключенной в сеть. Можно разместить домашние каталоги пользователей на сервере NFS и сделать их доступными с помощью сети.
- Устройства хранения информации, такие как флоппи-дисководы, приводы компакт-дисков и USB-диски, могут использоваться другими компьютерами в сети. Это может уменьшить общее число накопителелей со сменными носителями в сети.

2.1. Установка

Введите следующую команду в терминале для установки NFS сервера:

```
sudo apt-get install nfs-kernel-server
```

2.2. Конфигурация

Вы можете настроить каталоги для экспорта, добавляя их в файл `/etc/exports`. Например:

```
/ubuntu *(ro,sync,no_root_squash)
/home *(rw,sync,no_root_squash)
```

Вы можете заменить * одним из форматов записи имени хоста. Сделайте объявление хоста настолько необычным, насколько это возможно, чтобы нежеланные системы не могли получить доступа к монтированию NFS.

Для запуска NFS сервера выполните следующую команду в терминале:

```
sudo service nfs-kernel-server start
```

2.3. Настройка клиента NFS

Используйте команду mount для монтирования каталога NFS, доступ к которому открыт на другом компьютере. Наберите в терминале команду, схожую со следующим примером.

```
sudo mount example.hostname.com:/ubuntu /local/ubuntu
```

Точка монтирования `/local/ubuntu` должна существовать. В каталоге `/local/ubuntu` не должно быть никаких файлов или подкаталогов.

Другой способ монтирования ресурса NFS, открытого на другом компьютере, состоит в добавлении соответствующей строчки в файл `/etc/fstab`. Строчка должна содержать имя хоста NFS-сервера, название каталога, открытого на сервере, и название каталога на локальном компьютере, куда будет монтироваться совместно используемый ресурс NFS.

Общий синтаксис строки файла `/etc/fstab` следующий:

```
example.hostname.com:/ubuntu /local/ubuntu nfs rsize=8192,wsize=8192,timeo=14,intr
```

Если вы испытываете сложности с монтированием NFS-ресурса, убедитесь, что пакет nfs-common установлен на вашем клиенте. Для установки пакета nfs-common введите следующую команду в терминале:

```
sudo apt-get install nfs-common
```

2.4. Ссылки

Линукс NFS FAQ[3]

Ubuntu Wiki NFS Howto[4]

[3] http://nfs.sourceforge.net/
[4] https://help.ubuntu.com/community/NFSv4Howto

3. iSCSI-инициатор

iSCSI (Internet Small Computer System Interface) — это протокол, который разрешает передавать команды SCSI по сети. Обычно iSCSI реализуется для сетевых дисковых массивов (Storage Area Network — SAN), чтобы позволить серверам иметь доступ к большим объемам дискового пространства. Протокол iSCSI считает клиентов *инициаторами*, а сервера iSCSI — *целью*.

Ubuntu Server can be configured as both an iSCSI initiator and a target. This guide provides commands and configuration options to setup an iSCSI initiator. It is assumed that you already have an iSCSI target on your local network and have the appropriate rights to connect to it. The instructions for setting up a target vary greatly between hardware providers, so consult your vendor documentation to configure your specific iSCSI target.

3.1. Установка инициатора iSCSI

Для настройки сервера Ubuntu в качестве инициатора iSCSI установите пакет open-iscsi. Введите в терминале:

```
sudo apt-get install open-iscsi
```

3.2. Настройка инициатора iSCSI

Как только пакет open-iscsi установлен, отредактируйте `/etc/iscsi/iscsid.conf`, изменив следующее:

```
node.startup = automatic
```

Вы можете определить, какие целевые объекты вам доступны, с помощью утилиты iscsiadm. Введите следующую команду в терминале:

```
sudo iscsiadm -m discovery -t st -p 192.168.0.10
```

- *-m:* определяет режим, в котором работает iscsiadm.
- *-t:* определяет тип поиска.
- *-p:* опция, определяющая IP-адрес целевого объекта.

Замените *192.168.0.10* в примере на IP-адрес вашего объекта в сети.

Если целевой объект доступен, вы увидите вывод, подобный следующему:

```
192.168.0.10:3260,1 iqn.1992-05.com.emc:sl7b92030000520000-2
```

Номер *iqn* и IP-адрес могут быть другими, в зависимости от вашего оборудования.

Теперь вы можете соединиться с iSCSI сервером и, в зависимости от его настроек, вам, возможно, придётся ввести данные учетной записи пользователя. Подключитесь к узлу iSCSI:

```
sudo iscsiadm -m node --login
```

Убедитесь, что новый диск определяется с помощью dmesg:

```
dmesg | grep sd
```

```
[    4.322384] sd 2:0:0:0: Attached scsi generic sg1 type 0
[    4.322797] sd 2:0:0:0: [sda] 41943040 512-byte logical blocks: (21.4 GB/20.0 GiB)
[    4.322843] sd 2:0:0:0: [sda] Write Protect is off
[    4.322846] sd 2:0:0:0: [sda] Mode Sense: 03 00 00 00
[    4.322896] sd 2:0:0:0: [sda] Cache data unavailable
[    4.322899] sd 2:0:0:0: [sda] Assuming drive cache: write through
[    4.323230] sd 2:0:0:0: [sda] Cache data unavailable
[    4.323233] sd 2:0:0:0: [sda] Assuming drive cache: write through
[    4.325312]  sda: sda1 sda2 < sda5 >
[    4.325729] sd 2:0:0:0: [sda] Cache data unavailable
[    4.325732] sd 2:0:0:0: [sda] Assuming drive cache: write through
[    4.325735] sd 2:0:0:0: [sda] Attached SCSI disk
[ 2486.941805] sd 4:0:0:3: Attached scsi generic sg3 type 0
[ 2486.952093] sd 4:0:0:3: [sdb] 1126400000 512-byte logical blocks: (576 GB/537 GiB)
[ 2486.954195] sd 4:0:0:3: [sdb] Write Protect is off
[ 2486.954200] sd 4:0:0:3: [sdb] Mode Sense: 8f 00 00 08
[ 2486.954692] sd 4:0:0:3: [sdb] Write cache: disabled, read cache: enabled, doesn't
 support DPO or FUA
[ 2486.960577]  sdb: sdb1
[ 2486.964862] sd 4:0:0:3: [sdb] Attached SCSI disk
```

В приведённом выводе *sdb* — это новый iSCSI диск. Помните, что это всего лишь пример; вывод на вашем экране может сильно отличаться.

Далее создадим раздел, отформатируем файловую систему и подсоединим новый iSCSI диск. Введите в терминале:

```
sudo fdisk /dev/sdb
n
p
enter
w
```

Команды, приведённые выше, выполняются внутри утилиты fdisk; смотрите **man fdisk** для дополнительных подробностей. Также утилита cfdisk иногда более дружелюбна к пользователям.

Теперь форматируем файловую систему и монтируем её, например, в `/srv`:

```
sudo mkfs.ext4 /dev/sdb1
sudo mount /dev/sdb1 /srv
```

Наконец добавим запись в `/etc/fstab` для монтирования iSCSI устройства в процесе загрузки:

```
/dev/sdb1       /srv        ext4    defaults,auto,_netdev 0 0
```

Хорошей идеей будет убедиться, что всё работает как надо, перегрузив сервер.

3.3. Ссылки

Сайт Open-iSCSI[5]

Страница Debian Open-iSCSI[6]

[5] http://www.open-iscsi.org/
[6] http://wiki.debian.org/SAN/iSCSI/open-iscsi

4. CUPS — сервер печати

Основным механизмом печати в Ubuntu служит **Common UNIX Printing System** (CUPS). Эта система печати — свободно доступный, переносимый уровень абстракции печати, который является стандартом печати для большинства дистрибутивов Linux.

CUPS управляет заданиями на печать и очередями, а также обеспечивает печать по сети, используя стандартный протокол печати интернета, (Internet Printing Protocol, IPP). В то же время он поддерживает большое количество принтеров, от матричных до лазерных. CUPS также поддерживает файлы описания принтеров PostScript (PostScript Printer Description, PPD) и авто-определение сетевых принтеров, и имеет простой веб-ориентированный инструмент настройки и администрирования.

4.1. Установка

Для того, чтобы установить CUPS на ваш компьютер, используйте sudo с командой apt-get указав в качестве первого параметра название пакета для установки. Полная установка CUPS зависит от множества пакетов, но все они могут быть указаны в той же командной строке. Для установки CUPS наберите в командной строке следующее:

```
sudo apt-get install cups
```

После аутентификации вас по паролю, пакеты должны загрузиться и установиться без ошибок. В заключении установки сервер CUPS будет запущен автоматически.

При необходимости решения проблем, вы можете получить доступ к ошибкам сервера CUPS через файл журнала: `/var/log/cups/error_log`. Если журнал ошибок не даёт достаточно информации для решения какой-либо проблемы, уровень журналирования CUPS можно повысить изменением директивы **LogLevel** в файле настроек (описывается ниже) до debug или даже debug2 со стандартного info, что будет сохранять в журнал абсолютно всё. Если вы провёдете такое изменение, не забудьте вернуть всё обратно после решения проблемы, чтобы избежать излишнего разрастания файла журнала.

4.2. Конфигурация

Поведение сервера CUPS настраивается с помощью инструкций, содержащихся в файле `/etc/cups/cupsd.conf`. Файл настроек CUPS использует такой же синтаксис, как и основной файл настроек HTTP сервера Apache,

то есть пользователи, знакомые с модификацией файлов настроек Apache, должны спокойно ориентироваться при работе с настройками CUPS. Примеры некоторых настроек, которые вы, возможно, захотите изменить с самого начала, будут представлены здесь.

Перед изменением конфигурационного файла сделайте копию с оригинала и защитите её от записи, чтобы использовать файл оригинальных настроек в качестве справки, а также иметь возможность использовать его снова.

Скопируйте файл `/etc/cups/cupsd.conf` и защитите копию от записи с помощью следующих команд, выполненных в командной строке терминала:

```
sudo cp /etc/cups/cupsd.conf /etc/cups/cupsd.conf.original
sudo chmod a-w /etc/cups/cupsd.conf.original
```

- **ServerAdmin**: Чтобы настроить адрес электронной почты для назначенного администратора сервера CUPS, просто откройте файл `/etc/cups/cupsd.conf` в своём любимом текстовом редакторе и добавьте или измените строку *ServerAdmin* соответствующим образом. Например, если вы администратор сервера CUPS и ваш почтовый адрес 'bjoy@somebigco.com', вам следует изменить строку ServerAdmin следующим образом:

  ```
  ServerAdmin bjoy@somebigco.com
  ```

- **Listen**: В Ubuntu по умолчанию установленный сервер CUPS слушает только интерфейс обратной петли по адресу *127.0.0.1*.Чтобы заставить сервер CUPS прослушивать актуальный IP-адрес сетевого адаптера, вы должны указать сетевое имя или пару IP-адрес/порт добавочной директивой Listen. Например, если ваш сервер находится в локальной сети с IP-адресом *192.168.10.250*, и вы хотите сделать его доступным для других систем в этой подсети, отредактируйте `/etc/cups/cupsd.conf`, добавив директиву Listen, как показано ниже:

  ```
  Listen 127.0.0.1:631 # существующий Listen интерфейса loopback
  Listen /var/run/cups/cups.sock # существующий Listen для сокетов
  Listen 192.168.10.250:631 # Listen на интерфейсе LAN, Порт 631 (IPP)
  ```

  В вышепривёденном примере вы можете закомментировать или удалить ссылки на loopback-адрес (127.0.0.1), если желаете, чтобы cupsd вместо этого интерфейса использовал только Ethernet-интерфейсы локальной сети. Для разрешения использования всех интерфейсов, включая

loopback, к которым привязано определенное имя хоста, создайте запись Listen для имени хоста *socrates* следующим образом:

```
Listen socrates:631  # Listen on all interfaces for the hostname 'socrates'
```

или опустив директиву Listen и используя вместо неё *Port*, как в

```
Port 631  # Listen on port 631 on all interfaces
```

Если вам необходимо большее количество примеров директив конфигурационного файла сервера CUPS, обратитесь к соответствующей странице руководства системы, введя следующую команду в терминале:

```
man cupsd.conf
```

Если вы внесёте изменения в файл конфигурации `/etc/cups/cupsd.conf`, вам будет необходимо перезапустить CUPS сервер, выполнив следующую команду в терминале:

```
sudo service cups restart
```

4.3. Веб-интерфейс

Настраивать CUPS и отслеживать его состояние можно через веб-интерфейс, который по умолчанию доступен по адресу *http://localhost:631/admin*. Веб-интерфейс можно использовать для выполнения любых задач управления принтером.

Чтобы выполнить административную задачу через веб-интерфейс, вы должны либо разрешить учётную запись root на своем сервере, либо авторизоваться как пользователь из группы *lpadmin*. По соображениям безопасности CUPS не авторизует пользователей с пустыми паролями.

Чтобы добавить пользователя в группу *lpadmin*, выполните в терминале следующую команду:

```
sudo usermod -aG lpadmin username
```

Дальнейшая документация доступна через закладку *Documentation/Help* веб-интерфейса.

[7] http://www.cups.org/

4.4. Ссылки

Сайт CUPS[7]

Страница Debian Open-iSCSI[8]

[8] http://wiki.debian.org/SAN/iSCSI/open-iscsi

Глава 15. Сервисы электронной почты

Процесс доставки электронных писем от одного человека к другому через локальную сеть или Интернет включает в себя взаимодействие множества систем. Каждая из этих систем должна быть правильно настроена, чтобы выполнять свою работу. Оправитель использует *почтовый агент пользователя* (Mail User Agent, MUA) или клиент электронной почты, чтобы отправлять сообщения через один или несколько *агентов передачи почты* (Mail Transfer Agents, MTA), последний из которых передаст сообщение *агенту доставки почты* (Mail Delivery Agent, MDA) для доставки почты в почтовый ящик получателя, откуда оно может быть доставлено получателю с помощью его почтового клиента, обычно через сервер POP3 или IMAP.

1. Postfix

В Ubuntu агент передачи почты (Mail Transfer Agent (MTA)) по умолчанию — Postfix. Он считается безопасным, быстрым и лёгким в администрировании. Он совместим с MTA sendmail. Данный раздел объяснит, как установить и настроить postfix. Так же будет описано, как настроить SMTP-сервер с использованием безопасного соединения (для безопасной передачи почты).

Это руководство не рассматривает настройку *виртуальных доменов* postfix. Для получения информации по виртуальным доменам и другим расширенным настройкам смотрите *Раздел 1.7.4, «Ссылки» [282]*.

1.1. Установка

Чтобы установить postfix, запустите следующую команду:

```
sudo apt-get install postfix
```

Просто нажимайте ввод, когда процесс установки задает вопросы, более детальная настройка будет выполнена на следующем этапе.

1.2. Базовая конфигурация

Чтобы настроить postfix, выполните следующую команду:

```
sudo dpkg-reconfigure postfix
```

Будет запущен пользовательский интерфейс. На каждом экране выбирайте следующие значения:

- Сайт в интернете
- mail.example.com
- steve
- mail.example.com, localhost.localdomain, localhost
- No
- 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.0.0/24
- 0
- +
- все

Замените mail.example.com на домен, для которого вы настраиваете email, 192.168.0.0/24 на актуальную подсеть и маску для вашего почтового сервера и steve на соответствующее имя пользователя.

Теперь самое время решить, какой формат почтового ящика вы хотите использовать. По умолчанию, Postfix будет использовать **mbox**, как формат почтового ящика. Вместо прямого редактирования конфигурационного файла, вы можете использовать команду **postconf** для настройки всех параметров postfix. Параметры конфигурации будут храниться в файле `/etc/postfix/main.cf`. Позже, если вы захотите перенастроить отдельный параметр, вы можете либо запустить команду, либо изменить его в файле вручную.

Для настройки формата почтового ящика в **Maildir:**

```
sudo postconf -e 'home_mailbox = Maildir/'
```

Это поместит новую почту в /home/*username*/Maildir, поэтому вам потребуется настроить вашего агента доставки почты (MDA) на использование этого же каталога.

1.3. Аутентификация SMTP

SMTP-AUTH позволяет клиенту идентифицировать себя через механизм аутентификации (SASL). Транспортный уровень безопасности (TLS) будет использоваться для шифрования процесса аутентификации. После аутентификации SMTP сервер позволит клиенту передавать почту.

1. Настройте Postfix на SMTP-AUTH с использованием SASL (Dovecot SASL):

```
sudo postconf -e 'smtpd_sasl_type = dovecot'
sudo postconf -e 'smtpd_sasl_path = private/auth-client'
sudo postconf -e 'smtpd_sasl_local_domain ='
sudo postconf -e 'smtpd_sasl_security_options = noanonymous'
sudo postconf -e 'broken_sasl_auth_clients = yes'
sudo postconf -e 'smtpd_sasl_auth_enable = yes'
sudo postconf -e 'smtpd_recipient_restrictions = \
permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination'
```

Настройка *smtpd_sasl_path* является путём, относительным к каталогу запросов Postfix.

2. Далее создайте или получите цифровой сертификат для TLS. Смотрите подробности в разделе *Раздел 5, «Сертификаты» [194]*. Этот пример также использует Центр сертификации (CA). Для информации по созданию сертификатов CA смотрите раздел *Раздел 5.5, «Центр Сертификации» [197]*.

Почтовым агентам пользователей (MUA), подключающимся к вашему почтовому серверу через TLS, потребуется распознать

сертификат, используемый для TLS. Это может быть сделано либо с использованием сертификата от коммерческого центра сертификации, либо с самоподписанным сертификатом, который пользователи установили вручную. Для TLS между MTA (агентами передачи почты) сертификаты никогда не подтверждаются без дополнительного соглашения с контролирующими организациями. Поэтому для таких соединений, если локальные политики этого не требуют, нет резона не использовать самоподписанные сертификаты. Смотрите подробности в разделе *Раздел 5.3, «Создание сертификата со своей подписью» [197]*.

3. Как только у вас появился сертификат, настройте Postfix на использование TLS-шифрования как для входящей, так и для исходящей почты:

```
sudo postconf -e 'smtp_tls_security_level = may'
sudo postconf -e 'smtpd_tls_security_level = may'
sudo postconf -e 'smtp_tls_note_starttls_offer = yes'
sudo postconf -e 'smtpd_tls_key_file = /etc/ssl/private/server.key'
sudo postconf -e 'smtpd_tls_cert_file = /etc/ssl/certs/server.crt'
sudo postconf -e 'smtpd_tls_loglevel = 1'
sudo postconf -e 'smtpd_tls_received_header = yes'
sudo postconf -e 'myhostname = mail.example.com'
```

4. Если вы используете *собственный Центр сертификации*, для подписи сертификата введите:

```
sudo postconf -e 'smtpd_tls_CAfile = /etc/ssl/certs/cacert.pem'
```

Опять же, для подробностей смотрите раздел *Раздел 5, «Сертификаты» [194]*.

После выполнения всех команд Postfix настроен на SMTP-AUTH и самоподписанный сертификат создан для TLS шифрования.

Теперь файл `/etc/postfix/main.cf` должен выглядеть примерно так:

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete
# version

smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
```

```
#delay_warning_time = 4h

myhostname = server1.example.com
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = server1.example.com, localhost.example.com, localhost
relayhost =
mynetworks = 127.0.0.0/8
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
smtpd_sasl_local_domain =
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions =
permit_sasl_authenticated,permit_mynetworks,reject _unauth_destination
smtpd_tls_auth_only = no
smtp_tls_security_level = may
smtpd_tls_security_level = may
smtp_tls_note_starttls_offer = yes
smtpd_tls_key_file = /etc/ssl/private/smtpd.key
smtpd_tls_cert_file = /etc/ssl/certs/smtpd.crt
smtpd_tls_CAfile = /etc/ssl/certs/cacert.pem
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
smtpd_tls_session_cache_timeout = 3600s
tls_random_source = dev:/dev/urandom
```

Начальная настройка postfix закончена. Выполните следующую команду для перезапуска сервиса postfix:

```
sudo service postfix restart
```

Postfix поддерживает SMTP-AUTH как описано в *RFC2554*[1]. Он основан на *SASL*[2]. Однако все-таки необходимо настроить аутентификацию перед тем, как вы сможете использовать SMTP-AUTH.

1.4. Настройка SASL

Postfix поддерживает две реализации SASL: Cyrus SASL и Dovecot SASL. Чтобы разрешить Dovecot SASL, требуется установить пакет dovecot-common. Для этого из терминала введите следующее:

```
sudo apt-get install dovecot-common
```

[1] http://www.ietf.org/rfc/rfc2554.txt
[2] http://www.ietf.org/rfc/rfc2222.txt

Теперь нужно отредактировать `/etc/dovecot/conf.d/10-master.conf`. Измените следующее:

```
service auth {
  # auth_socket_path points to this userdb socket by default. It's typically
  # used by dovecot-lda, doveadm, possibly imap process, etc. Its default
  # permissions make it readable only by root, but you may need to relax these
  # permissions. Users that have access to this socket are able to get a list
  # of all usernames and get results of everyone's userdb lookups.
  unix_listener auth-userdb {
    #mode = 0600
    #user =
    #group =
  }

  # Postfix smtp-auth
  unix_listener /var/spool/postfix/private/auth {
    mode = 0660
    user = postfix
    group = postfix
  }
```

Чтобы позволить клиентам Outlook использовать SMTP-AUTH, в секции *authentication mechanisms* файла /etc/dovecot/conf.d/10-auth.conf замените эту строку:

```
auth_mechanisms = plain
```

На следующую:

```
auth_mechanisms = plain login
```

После того, как Dovecot настроен, перезапустите его:

```
sudo service dovecot restart
```

1.5. Почтовый стек доставки

Другой опцией настройки Postfix для SMTP-AUTH является использование пакета mail-stack-delivery (ранее он назывался dovecot-postfix). Этот пакет установит Dovecot и настроит Postfix для его использования совместно с SASL аутентификацией и как агента доставки почты (MDA). Пакет также настроит Dovecot для IMAP, IMAPS, POP3 и POP3S.

Вы можете захотеть или не захотеть использовать IMAP, IMAPS, POP3, или POP3S на вашем почтовом сервере. Например, если вы настраиваете свой сервер в качестве почтового шлюза, фильтра

спама и вирусов и т.п. В этом случае возможно будет проще использовать вышеприведенные команды для настройки Postfix на SMTP_AUTH.

Чтобы установить пакет, введите в терминале:

```
sudo apt-get install mail-stack-delivery
```

У вас теперь рабочий почтовый сервер, но существует несколько опций, которые вы, возможно, захотите изменить в дальнейшем. Например, пакет использует сертификат и ключ от ssl-cert пакета, и в рабочей среде, вы должны использовать сертификат и ключ, сгенерированный для хоста. Смотрите радел *Раздел 5, «Сертификаты» [194]* для дополнительных деталей.

После того, как вы получили заказанный сертификат для сервера, замените следующую опцию в `/etc/postfix/main.cf`:

```
smtpd_tls_cert_file = /etc/ssl/certs/ssl-mail.pem
smtpd_tls_key_file = /etc/ssl/private/ssl-mail.key
```

Перезапустите Postfix:

```
sudo service postfix restart
```

1.6. Тестирование

Настройка SMTP-AUTH завершена. Теперь самое время проверить настройки.

Чтобы убедиться, что SMTP-AUTH и TLS работают правильно, выполните следующую команду:

```
telnet mail.example.com 25
```

После установления соединения с почтовым сервером postfix введите:

```
ehlo mail.example.com
```

Если среди прочего вы увидите следующие строки, всё работает замечательно. Введите **quit** для выхода.

```
250-STARTTLS
250-AUTH LOGIN PLAIN
```

```
250-AUTH=LOGIN PLAIN
250 8BITMIME
```

1.7. Устранение проблем

Этот раздел описывает несколько общих способов определения причин возникающих проблем.

1.7.1. Отказ от режима chroot

Пакет postfix в Ubuntu по умолчанию устанавливается в окружении *chroot* из соображений безопасности. Это может дополнительно усложнить процесс поиска решения проблем.

Для отключения функционирования chroot, найдите следующую строку в файле настроек `/etc/postfix/master.cf`:

```
smtp inet n - - - - smtpd
```

и измените на:

```
smtp inet n - n - - smtpd
```

После этого вам придётся перезапустить Postfix для использования новых настроек. В терминале введите следующее:

```
sudo service postfix restart
```

1.7.2. Smtps

Если вам нужен smtps, отредактируйте `/etc/postfix/master.cf` и раскомментируйте следующую строку:

```
smtps     inet  n       -       -       -       -       smtpd
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
```

1.7.3. Файлы журналов

Postfix посылает все сообщения в журнал `/var/log/mail.log`. Однако сообщения об ошибках и предупреждения могут иногда теряться в нормальном журнале, поэтому они отдельно сохраняются в `/var/log/mail.err` и `/var/log/mail.warn`, соответственно.

Для просмотра сообщений журнала в режиме реального времени вы можете использовать команду tail -f:

```
tail -f /var/log/mail.err
```

Количество деталей, записываемых в журнал, может быть увеличено. Ниже приведено несколько опций настройки для увеличения уровня детализации некоторых областей, описанных выше.

- Для увеличения *TLS* активности журнала, установите опции *smtpd_tls_loglevel* значение от 1 до 4.

```
sudo postconf -e 'smtpd_tls_loglevel = 4'
```

- Если вы испытываете трудности с отправкой или приёмом почты из отдельного домена, вы можете включить его в параметр *debug_peer_list*.

```
sudo postconf -e 'debug_peer_list = problem.domain'
```

- Вы можете увеличить детализацию любого сервиса Postfix редактированием `/etc/postfix/master.cf`, добавив *-v* после соответствующей записи. Для примера изменим запись *smtp*:

```
smtp      unix  -       -       -       -       -       smtp -v
```

Важно отметить, что после одного из регистрации изменений выше процесс Postfix должен быть перезагружен для того, чтобы признать новую конфигурацию: **sudo service postfix reload**

- Чтобы увеличить количество информации, записываемой в журнал при устранении проблем с *SASL*, вы можете задать следующие опции в `/etc/dovecot/conf.d/10-logging.conf`

```
auth_debug=yes
auth_debug_passwords=yes
```

Так же, как и в случае с Postfix, если вы изменили конфигурацию Dovecot, то процесс должен быть перезагружен: **sudo service dovecot reload**.

Некоторые опции выше могут серьёзно увеличить объем информации, передаваемой в файлы журналов. Не забывайте возвращать уровень детализации журналов к нормальному значению после решения проблем. Затем перезапустите соответствующий сервис, чтобы изменения настройки вступили в силу.

1.7.4. Ссылки

Администрирование сервера Postfix может быть очень сложной задачей. В какой-то момент вам может потребоваться обратиться к сообществу Ubuntu для более квалифицированной помощи.

Хорошее место, чтобы задать вопрос по сопровождению Postfix и влиться в сообщество Ubuntu Server community — это IRC-канал *#ubuntu-server* на *freenode*[3]. Вы также можете отправить сообщение на один из *веб-форумов*[4].

Для всесторонней информации Postfix разработчики Ubuntu очень рекомендуют прочитать *The Book of Postfix*[5].

Наконец, *веб-сайт Postfix*[6] также содержит много информации по всем возможным опциям настройки.

Also, the *Ubuntu Wiki Postfix*[7] page has more information.

[3] http://freenode.net
[4] http://www.ubuntu.com/support/community/webforums
[5] http://www.postfix-book.com/
[6] http://www.postfix.org/documentation.html
[7] https://help.ubuntu.com/community/Postfix

2. Exim4

Exim4 — это почтовый транспортный агент (MTA), разработанный в университете Кембриджа для использования в системах Unix, подключённых к Интернету. Exim можно установить вместо sendmail, хотя процедура настройки exim сильно отличается от настройки sendmail.

2.1. Установка

Чтобы установить exim4, выполните следующую команду:

```
sudo apt-get install exim4
```

2.2. Конфигурация

Для настройки Exim4 выполните следующую команду:

```
sudo dpkg-reconfigure exim4-config
```

Появится пользовательский интерфейс. Этот интерфейс позволит вам настроить множество параметров. Например, в Exim4 файлы настроек разделены между различными файлами. Если вы решите объединить их в один файл, вы можете настроить это в данном пользовательском интерфейсе.

Все параметры, которые вы настроите в пользовательском интерфейсе будут сохранены в файле `/etc/exim4/update-exim4.conf`. Если вы захотите что-то перенастроить, то либо перезапустите мастера настройки, либо вручную поправьте данный файл любым редактором. После настройки вам потребуется выполнить следующую команду для создания главного файла настроек:

```
sudo update-exim4.conf
```

Главный файл настроек будет создан и сохранён в `/var/lib/exim4/config.autogenerated`.

Вы не должны, ни при каких обстоятельствах, редактировать вручную основной файл настроек `/var/lib/exim4/config.autogenerated`. Он обновляется автоматически каждый раз, когда вы запускаете команду **update-exim4.conf**

Вы можете воспользоваться следующей командой для запуска сервиса Exim4.

```
sudo service exim4 start
```

2.3. Аутентификация SMTP

Этот раздел раскрывает, как настроить Exim4 для использования SMTP-AUTH с TLS и SASL.

Первым шагом будет создание сертификата для использования TLS. Введите следующее в терминале:

```
sudo /usr/share/doc/exim4-base/examples/exim-gencert
```

Теперь Exim4 нуждается в настройке TLS. Отредактируйте `/etc/exim4/conf.d/main/03_exim4-config_tlsoptions`, добавив следующее:

```
MAIN_TLS_ENABLE = yes
```

Далее вам потребуется настроить Exim4 на использование saslauthd для аутентификации. Вызовите на редактирование `/etc/exim4/conf.d/auth/30_exim4-config_examples` и раскомментируйте секции *plain_saslauthd_server* и *login_saslauthd_server*:

```
 plain_saslauthd_server:
   driver = plaintext
   public_name = PLAIN
   server_condition = ${if saslauthd{{$auth2}{$auth3}}{1}{0}}
   server_set_id = $auth2
   server_prompts = :
   .ifndef AUTH_SERVER_ALLOW_NOTLS_PASSWORDS
   server_advertise_condition = ${if eq{$tls_cipher}{}{}{*}}
   .endif
#
 login_saslauthd_server:
   driver = plaintext
   public_name = LOGIN
   server_prompts = "Username:: : Password::"
   # don't send system passwords over unencrypted connections
   server_condition = ${if saslauthd{{$auth1}{$auth2}}{1}{0}}
   server_set_id = $auth1
   .ifndef AUTH_SERVER_ALLOW_NOTLS_PASSWORDS
   server_advertise_condition = ${if eq{$tls_cipher}{}{}{*}}
   .endif
```

Дополнительно, чтобы внешний почтовый клиент имел возможность соединиться с вашим новым сервером exim, требуется добавить нового пользователя в exim, используя следующие команды:

```
sudo /usr/share/doc/exim4-base/examples/exim-adduser
```

Новый файл паролей должен быть защищён от пользователей с помощью следующих команд:

```
sudo chown root:Debian-exim /etc/exim4/passwd
sudo chmod 640 /etc/exim4/passwd
```

В конце обновите настройки Exim4 и перезапустите сервис:

```
sudo update-exim4.conf
sudo service exim4 restart
```

2.4. Настройка SASL

В этом разделе раскрываются подробности настройки saslauthd, чтобы обеспечить аутентификацию для Exim4.

Для начала установим пакет sasl2-bin. В терминале введите следующее:

```
sudo apt-get install sasl2-bin
```

Чтобы настроить saslauthd, отредактируйте файл настройки */etc/default/saslauthd* и замените *START=no* на:

```
START=yes
```

Далее пользователя *Debian-exim* требуется включить в группу *sasl*, чтобы Exim4 мог использовать сервис saslauthd:

```
sudo adduser Debian-exim sasl
```

Теперь запустите сервис saslauthd:

```
sudo service saslauthd start
```

Теперь Exim4 настроен на SMTP-AUTH с использованием TLS и SASL аутентификации.

2.5. Ссылки

- Смотрите *exim.org*[8] для дополнительной информации.
- Также доступна книга *Exim4 Book*[9].

[8] http://www.exim.org/
[9] http://www.uit.co.uk/content/exim-smtp-mail-server

- Ещё один ресурс — страница *Exim4 Ubuntu Wiki* [10].

[10] https://help.ubuntu.com/community/Exim4

3. Dovecot Server

Dovecot — это агент доставки почты, написанный с упором на безопасность. Он поддерживает основные форматы почтовых ящиков: mbox или Maildir. Этот раздел рассказывает о том, как настроить его в качестве сервера imap или pop3.

3.1. Установка

Для установки dovecot выполните следующую команду в терминале:

```
sudo apt-get install dovecot-imapd dovecot-pop3d
```

3.2. Конфигурация

Чтобы настроить dovecot, вам потребуется отредактировать файл `/etc/dovecot/dovecot.conf`. Вы можете выбрать, какой протокол использовать. Это может быть pop3, pop3s (безопасный pop3), imap или imaps (безопасный imap). Описание этих протоколов находится за пределами вопросов, рассматриваемых в данном руководстве. Для дополнительной информации обратитесь к статьям Википедии по *POP3*[11] и *IMAP*[12].

IMAPS и POP3S более безопасны, чем обычные IMAP и POP3, поскольку используют SSL-шифрование для соединения. Как только вы выберете протокол, исправьте следующую строку в файле `/etc/dovecot/dovecot.conf`:

```
protocols = pop3 pop3s imap imaps
```

Далее выберите почтовый ящик, который вы хотите использовать. Dovecot поддерживает форматы **maildir** и **mbox**. Это наиболее часто используемые форматы почтовых ящиков. Оба имеют свои преимущества, и их обсуждение можно найти на *веб-сайте Dovecot*[13].

После выбора типа почтового ящика отредактируйте файл `/etc/dovecot/conf.d/10-mail.conf`, изменив следующую строку:

```
mail_location = maildir:~/Maildir # (for maildir)
or
mail_location = mbox:~/mail:INBOX=/var/spool/mail/%u # (for mbox)
```

[11] http://en.wikipedia.org/wiki/POP3
[12] http://en.wikipedia.org/wiki/Internet_Message_Access_Protocol
[13] http://wiki2.dovecot.org/MailboxFormat

Вы должны настроить свой почтовый транспортный агент (MTA, Mail Transport Agent) для передачи входящей почты на почтовый ящик этого типа, если он отличен от того, который вы уже настроили.

Настроив dovecot, перезапустите сервис dovecot, чтобы проверить свои установки:

```
sudo service dovecot restart
```

Если вы разрешили imap или pop3, вы можете попробовать подключиться с помощью команд **telnet localhost pop3** или **telnet localhost imap2**. Если вы увидите что-то, похожее на следующий код, установка успешно завершена:

```
bhuvan@rainbow:~$ telnet localhost pop3
Пытаемся 127.0.0.1...
Соединился с localhost.localdomain.
Клавиша возврата '^]'.
+OK Dovecot готов.
```

3.3. Dovecot: Настройка SSL

Чтобы настроить dovecot на использование SSL, можете отредактировать файл `/etc/dovecot/conf.d/10-ssl.conf`, внеся изменения в следующие строки:

```
ssl = yes
ssl_cert = </etc/ssl/certs/dovecot.pem
ssl_key = </etc/ssl/private/dovecot.pem
```

You can get the SSL certificate from a Certificate Issuing Authority or you can create self signed SSL certificate. The latter is a good option for email, because SMTP clients rarely complain about "self-signed certificates". Please refer to *Раздел 5, «Сертификаты» [194]* for details about how to create self signed SSL certificate. Once you create the certificate, you will have a key file and a certificate file. Please copy them to the location pointed in the `/etc/dovecot/conf.d/10-ssl.conf` configuration file.

3.4. Настройка брандмауэра для почтового сервера

Для доступа к вашему почтовому серверу с другого компьютера вы должны настроить брандмауер на разрешение соединений по необходимым портам.

- IMAP - 143
- IMAPS - 993

- POP3 - 110
- POP3S - 995

3.5. Ссылки

- Смотрите *Dovecot website*[14] для дополнительной информации.
- Также страница *Dovecot Ubuntu Wiki*[15] содержит много подробностей.

[14] http://www.dovecot.org/
[15] https://help.ubuntu.com/community/Dovecot

4. Mailman

Mailman — это программа с открытыми кодами для управления дискуссиями, ведущимися через электронную почту, и рассылками электронных новостных сообщений. Многие открытые списки рассылок (включая все на *Ubuntu mailing lists*[16]) используют Mailman в качестве программы управления почтовыми списками. Это мощное приложение, при этом его легко установить и поддерживать.

4.1. Установка

Mailman обеспечивает веб-интерфейс для администраторов и пользователей, использующих внешний почтовый сервер для отправки и приема почты. Он великолепно работает со следующими почтовыми серверами:

- Postfix
- Exim
- Sendmail
- Qmail

Мы рассмотрим, как установить и настроить Mailman с веб-сервером Apache, а также с почтовым сервером Postfix или Exim. Если вы собираетесь устанавливать Mailman с другим почтовым сервером, обратитесь, пожалуйста, к разделу Ссылки.

Вам потребуется установить только один почтовый сервер и Postfix для Ubuntu является вариантом по умолчанию.

4.1.1. Apache2

Чтобы установить apache2, обратитесь к соответствующему разделу *Раздел 1.1, «Установка» [213]*.

4.1.2. Postfix

Для инструкций по установке и настройке Postfix смотрите раздел *Раздел 1, «Postfix» [274]*

4.1.3. Exim4

Для установки Exim4 обратитесь к разделу *Раздел 2, «Exim4» [283]*.

Когда exim4 устанавливается, его файлы настроек сохраняются в каталоге `/etc/exim4`. In B Ubuntu по умолчанию файлы настройки exim4 разделены

[16] http://lists.ubuntu.com

на несколько файлов. Вы можете это поменять, изменив следующую переменную в файле `/etc/exim4/update-exim4.conf`:

```
dc_use_split_config='true'
```

4.1.4. Mailman

Чтобы установить Mailman, выполните следующую команду в терминале:

```
sudo apt-get install mailman
```

Она скопирует установочные файлы в каталог /var/lib/mailman. Она также установит CGI-сценарии в каталог /usr/lib/cgi-bin/mailman, создаст пользователя Linux *list* и группу *list*. Процесс mailman будет управляться этим пользователем.

4.2. Конфигурация

В этом разделе предполагается, что у вас удачно установлены mailman, apache2, и postfix или exim4. Теперь вам требуется только их настроить.

4.2.1. Apache2

Примерный файл настройки Apache идет вместе с Mailman и помещён в `/etc/mailman/apache.conf`. Чтобы Apache смог его использовать, требуется переместить его в `/etc/apache2/sites-available`:

```
sudo cp /etc/mailman/apache.conf /etc/apache2/sites-available/mailman.conf
```

Это установит новый *VirtualHost* в Apache для административного сайта Mailman. Теперь разрешим новую конфигурацию и перезагрузим Apache:

```
sudo a2ensite mailman.conf
sudo service apache2 restart
```

Mailman использует apache2 для обработки CGI-сценариев. CGI-сценарии Mailman устанавливаются в каталог /usr/lib/cgi-bin/mailman. Поэтому адрес mailman будет http://hostname/cgi-bin/mailman/. Вы можете внести изменения в файл `/etc/apache2/sites-available/mailman.conf`, если вы решили изменить такой адрес.

4.2.2. Postfix

Для интеграции с Postfix мы ассоциируем домен lists.example.com со списком рассылки. Пожалуйста, замените *lists.example.com* на домен по вашему выбору.

Вы можете использовать команду postconf для добавления необходимых настроек в `/etc/postfix/main.cf`:

```
sudo postconf -e 'relay_domains = lists.example.com'
sudo postconf -e 'transport_maps = hash:/etc/postfix/transport'
sudo postconf -e 'mailman_destination_recipient_limit = 1'
```

В `/etc/postfix/master.cf` дважды проверьте, что у вас указан следующий транспорт:

```
mailman   unix  -       n       n       -       -       pipe
  flags=FR user=list argv=/usr/lib/mailman/bin/postfix-to-mailman.py
  ${nexthop} ${user}
```

Он вызывает сценарий *postfix-to-mailman.py*, когда почта доставлена по списку.

Ассоциируем домен lists.example.com с транспортом Mailman с помощью транспортной карты. Отредактируйте файл `/etc/postfix/transport`:

```
lists.example.com     mailman:
```

Теперь дадим Postfix построить транспортную карту, введя следующее в терминале:

```
sudo postmap -v /etc/postfix/transport
```

Затем перезапустите Postfix, чтобы разрешить новые настройки:

```
sudo service postfix restart
```

4.2.3. Exim4

Когда Exim4 установлен, вы можете запустить сервер Exim, используя следующую команду из терминала:

```
sudo service exim4 start
```

Чтобы mailman мог работать с Exim4, вам потребуется настроить Exim4. Как было замечено ранее, по умолчанию Exim4 использует множество файлов настроек различного типа. Для подробностей смотрите сайт *Exim*[17]. Чтобы запустить mailman, нам придётся добавить новый файл настройки к следующим типам настроек:

[17] http://www.exim.org

- Основное
- Передача почты
- Маршрутизатор

Exim создаёт главный файл настройки, сортируя все эти миниатюрные конфигурационные файлы. Поэтому последовательность этих конфигурационных файлов очень важна.

4.2.4. Основное

Все настроечные файлы основного типа хранятся в каталоге `/etc/exim4/conf.d/main/`. Создайте новый файл `04_exim4-config_mailman` и добавьте в него следующее:

```
# start
# Home dir for your Mailman installation -- aka Mailman's prefix
# directory.
# On Ubuntu this should be "/var/lib/mailman"
# This is normally the same as ~mailman
MM_HOME=/var/lib/mailman
#
# User and group for Mailman, should match your --with-mail-gid
# switch to Mailman's configure script. Value is normally "mailman"
MM_UID=list
MM_GID=list
#
# Domains that your lists are in - colon separated list
# you may wish to add these into local_domains as well
domainlist mm_domains=hostname.com
#
# -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
#
# These values are derived from the ones above and should not need
# editing unless you have munged your mailman installation
#
# The path of the Mailman mail wrapper script
MM_WRAP=MM_HOME/mail/mailman
#
# The path of the list config file (used as a required file when
# verifying list addresses)
MM_LISTCHK=MM_HOME/lists/${lc::$local_part}/config.pck
# end
```

4.2.5. Передача почты

Все настроечные файлы, принадлежащие к типу транспортировка, хранятся в каталоге `/etc/exim4/conf.d/transport/`. Создайте новый файл `40_exim4-config_mailman` и добавьте в него следующее:

```
mailman_transport:
 driver = pipe
 command = MM_WRAP \
             '${if def:local_part_suffix \
                   {${sg{$local_part_suffix}{-(\\w+)(\\+.*)?}{\$1}}} \
                   {post}}' \
             $local_part
  current_directory = MM_HOME
  home_directory = MM_HOME
  user = MM_UID
  group = MM_GID
```

4.2.6. Маршрутизатор

Все настроечные файлы, принадлежащие к типу роутер, хранятся в каталоге `/etc/exim4/conf.d/router/`. Создайте новый файл `101_exim4-config_mailman` и добавьте в него следующее:

```
mailman_router
 driver = accept
 require_files = MM_HOME/lists/$local_part/config.pck
 local_part_suffix_optional
 local_part_suffix = -bounces : -bounces+* : \
                     -confirm+* : -join : -leave : \
                     -owner : -request : -admin
 transport = mailman_transport
```

Порядок основных и транспортных файлов настроек не важен. Однако, порядок файлов настроек роутера должен быть сохранён. Конкретно этот файл по порядку должен быть до файла 200_exim4-config_primary. Оба этих файла содержат одинаковый тип информации. Первый из них будет определен как предшественник. Для получкения более полной информации, обратитесь к разделу ссылок.

4.2.7. Mailman

После того как установлен mailman, вы можете использовать следующую команду:

```
sudo service mailman start
```

Поскольку mailman установлен, вы можете создать список рассылки по умолчанию. Выполните следующую команду, чтобы создать список рассылки:

```
sudo /usr/sbin/newlist mailman
```

```
Enter the email address of the person running the list: bhuvan at ubuntu.com
Initial mailman password:
To finish creating your mailing list, you must edit your /etc/aliases (or
equivalent) file by adding the following lines, and possibly running the
`newaliases' program:
mailman: "|/var/lib/mailman/mail/mailman post mailman"
mailman-admin: "|/var/lib/mailman/mail/mailman admin mailman"
mailman-bounces: "|/var/lib/mailman/mail/mailman bounces mailman"
mailman-confirm: "|/var/lib/mailman/mail/mailman confirm mailman"
mailman-join: "|/var/lib/mailman/mail/mailman join mailman"
mailman-leave: "|/var/lib/mailman/mail/mailman leave mailman"
mailman-owner: "|/var/lib/mailman/mail/mailman owner mailman"
mailman-request: "|/var/lib/mailman/mail/mailman request mailman"
mailman-subscribe: "|/var/lib/mailman/mail/mailman subscribe mailman"
mailman-unsubscribe: "|/var/lib/mailman/mail/mailman unsubscribe mailman"

Hit enter to notify mailman owner...

#
```

Мы настроили как Postfix, так и Exim4 на распознавание всех почтовых сообщений от mailman. Поэтому нет необходимости создавать новые записи в `/etc/aliases`. Если вы делаете какие-либо изменения в конфигурационных файлах, не забывайте выполнять перезапуск соответствующих сервисов до перехода к следующей части.

Exim4 не использует псевдонимов (aliases) для перенаправления почты для Mailman, поскольку он использует метод перебора *discover*.Чтобы подавить использование псевдонимов при создании списка, вам потребуется добавить строку *MTA=None* в конфигурационный файл `/etc/mailman/mm_cfg.py`.

4.3. Администрирование

Мы предполагаем, что у вас установка по умолчанию. Файлы CGI-сценариев mailman'a расположены в каталоге /usr/lib/cgi-bin/mailman/. Mailman предоставляет возможность администрирования с помощью веб-интерфейса. Для доступа к этой странице откройте в своём браузере следующий адрес:

http://hostname/cgi-bin/mailman/admin

На этом экране появится список рассылок по умолчанию, *mailman*. Если вы щёлкнете мышкой на имени списка рассылки, у вас будет запрошен пароль. Если вы введёте правильный пароль, у вас появится доступ к изменению административных настроек списка рассылки. Вы можете создать новый список рассылки с помощью утилиты командной строки

(**/usr/sbin/newlist**). Также вы можете создать новый список рассылки с помощью веб-интерфейса.

4.4. Пользователи

Mailman предоставляет пользователю веб-интерфейс. Для доступа к этой странице, перейдите в браузере на следующий URL:

http://hostname/cgi-bin/mailman/listinfo

На этом экране появится созданный при установке список рассылки "*mailman*". Если щёлкнуть на названии списка рассылки, появится форма регистрации. Для подписки на этот список можно ввести daш почтовый адрес, имя (не обязательно) и пароль. После этого dam будет отправленно электронной почтой приглашение. Чтобы подписаться на список рассылки, следуйте инструкциям, содержащимся в этом приглашении.

4.5. Ссылки

GNU Mailman — руководство по установке[18]

HOWTO — Совместное использование Exim 4 и Mailman 2.1[19]

Также смотрите страницу *Mailman Ubuntu Wiki*[20].

[18] http://www.list.org/mailman-install/index.html
[19] http://www.exim.org/howto/mailman21.html
[20] https://help.ubuntu.com/community/Mailman

5. Фильтрация почты

Одной из больших проблем с электронной почтой является проблема массовой незатребованной почты (Unsolicited Bulk Email — UBE). Такие сообщения, более известные как СПАМ, могут к тому же содержать вирусы и другие виды вредоносных программ. Согласно некоторым отчётам, эти сообщения составляют подавляющую часть от всего трафика почтовых сообщений в интернете.

В этом разделе рассматривается интеграция Amavisd-new, Spamassassin и ClamAV с транспортным почтовым агентом (MTA) Postfix. Postfix Postfix может также проверять легальность почты с помощью передачи её внешним фильтрам содержания. Эти фильтры могут иногда определить, что сообщение является спамом без необходимости передачи его более ресурсоёмким приложениям. Пара таких фильтров — это opendkim и python-policyd-spf.

- Amavisd-new — это программа-обёртка, которая может вызывать любое количество программ фильтрации контента для обнаружения спама, антивирус и т.п.
- Spamassassin использует множество механизмов фильтрования почты на основе содержимого сообщений.
- ClamAV — антивирусное приложение с открытым кодом.
- opendkim является почтовым фильтром Sendmail для стандарта DKIM (почты, заверенной доменными ключами).
- python-policyd-spf обеспечивает проверку SPF (структуры политики отправителя) с Postfix.

А здесь то, как эти части работают вместе:

- Почтовое сообщение принимается Postfix.
- Это сообщение проходит через некоторые внешние фильтры, в том числе opendkim и python-policyd-spf.
- Затем сообщение обрабатывается Amavisd-new.
- ClamAV используется для проверки сообщения. Если сообщение содержит вирус, Postfix сбросит сообщение.
- Чистые сообщения затем будут проверены Spamassassin на принадлежность к спаму. Spamassassin затем добавит строки X-Header, позволяющие в дальнейшем Amavisd-new управлять сообщением.

Например, если сообщение содержит уровень спама более пятидесяти, оно будет автоматически выброшено из очереди, чтобы не беспокоить

получателя. В качестве альтернативы помеченное сообщение доставляется до почтового агента пользователя (MUA) чтобы пользователь сам определил насколько оно легальное.

5.1. Установка

Смотрите раздел*Раздел 1, «Postfix» [274]* для установки и настройки Postfix.

Чтобы установить основные приложения, введите следующее в терминале:

```
sudo apt-get install amavisd-new spamassassin clamav-daemon
sudo apt-get install opendkim postfix-policyd-spf-python
```

Существуют некоторые общие пакеты, подключаемые к Spamassassin для лучшего определения спама:

```
sudo apt-get install pyzor razor
```

Поскольку основным фильтрующим приложениям требуются утилиты архивации для обработки прикрепленных файлов:

```
sudo apt-get install arj cabextract cpio lha nomarch pax rar unrar unzip zip
```

Если какие-то пакеты не были найдены, проверьте, что хранилище *multiverse* разрешено в `/etc/apt/sources.list`

Если вы внесли изменения в этот файл, убедитесь, что выполнили **sudo apt-get update** перед повторной попыткой установки.

5.2. Конфигурация

Теперь настроим, чтобы всё работало и фильтровало почту.

5.2.1. ClamAV

Стандартное поведение ClamAV вполне подходит для наших нужд. Для дополнительных опций настройки смотрите конфигурационные файлы в `/etc/clamav`.

Добавьте пользователя *clamav* в группу *amavis*, чтобы Amavisd-new имел соответствующие права доступа для сканирования файлов:

```
sudo adduser clamav amavis
sudo adduser amavis clamav
```

5.2.2. Spamassassin

Spamassassin автоматически определяет общие компоненты и использует их, если они присутствуют. Это означает, что нет необходимости настраивать pyzor и razor.

Отредактируйте `/etc/default/spamassassin` для активации сервиса Spamassassin. Измените *ENABLED=0* на:

```
ENABLED=1
```

Теперь запустим сервис:

```
sudo service spamassassin start
```

5.2.3. Amavisd-new

Сначала активируем проверку на спам и вирусы в Amavisd-new, отредактировав `/etc/amavis/conf.d/15-content_filter_mode`:

```
use strict;

# You can modify this file to re-enable SPAM checking through spamassassin
# and to re-enable antivirus checking.

#
# Default antivirus checking mode
# Uncomment the two lines below to enable it
#

@bypass_virus_checks_maps = (
   \%bypass_virus_checks, \@bypass_virus_checks_acl, \$bypass_virus_checks_re);


#
# Default SPAM checking mode
# Uncomment the two lines below to enable it
#

@bypass_spam_checks_maps = (
   \%bypass_spam_checks, \@bypass_spam_checks_acl, \$bypass_spam_checks_re);

1;  # insure a defined return
```

Возврат спама может быть плохой идеей, поскольку обратный адрес часто неверный. Подумайте над тем, чтобы изменить в `/etc/amavis/conf.d/20-debian_defaults` установку *$final_spam_destiny* на D_DISCARD вместо D_BOUNCE, как показано ниже:

```
$final_spam_destiny       = D_DISCARD;
```

Дополнительно вы можете захотеть установить следующие опциональные флаги для отметки большего количества сообщений как спам:

```
$sa_tag_level_deflt = -999; # add spam info headers if at, or above that level
$sa_tag2_level_deflt = 6.0; # add 'spam detected' headers at that level
$sa_kill_level_deflt = 21.0; # triggers spam evasive actions
$sa_dsn_cutoff_level = 4; # spam level beyond which a DSN is not sent
```

Если *сетевое имя сервера (hostname)* отличается от MX-записи домена, вам может потребоваться установить вручную опцию *$myhostname*. Также, если сервер принимает почту для нескольких доменов, опцию *@local_domains_acl* потребуется изменить. Отредактируйте файл `/etc/amavis/conf.d/50-user`:

```
$myhostname = 'mail.example.com';
@local_domains_acl = ( "example.com", "example.org" );
```

Если вы хотите покрывать несколько доменов, вы можете использовать следующее в файле `/etc/amavis/conf.d/50-user`

```
@local_domains_acl = qw(.);
```

После настройки Amavisd-new требуется перезапустить:

```
sudo service amavis restart
```

5.2.3.1. Белые списки DKIM

Amavisd-new может быть настроен на автоматическое занесение адресов в *Whitelist* из доменов с действительными доменными ключами. Есть несколько предварительно настроенных доменов в `/etc/amavis/conf.d/40-policy_banks`.

Существует несколько вариантов настройки белого списка для домена:

- '*example.com*' => '*WHITELIST*',: будет помещён в белый список любой адрес домена "example.com".
- '*.example.com*' => '*WHITELIST*',: будет помещён в белый список любой адрес любого *поддомена* "example.com", который имеет действительную подпись.
- '*.example.com/@example.com*' => '*WHITELIST*',: будут помещены в белый список поддомены "example.com", которые используют подпись родительского домена *example.com*.

- '*./@example.com*' => '*WHITELIST*',: добавляет адреса, которые имеют действительную подпись от "example.com". Это обычно используется для дискуссионных групп, которые подписывают свои сообщения.

Домен может иметь несколько настроек белого списка. После редактирования файла перезапустите amavisd-new:

```
sudo service amavis restart
```

В этом контексте, если домен добавлен в белый список, сообщение не будет передано каким-либо антивирусным или спам фильтрам. Это может быть как желательным, так и нежелательным поведением для вашего домена.

5.2.4. Postfix

Для интеграции Postfix, введите следующее в терминале:

```
sudo postconf -e 'content_filter = smtp-amavis:[127.0.0.1]:10024'
```

Далее отредактируйте `/etc/postfix/master.cf`, добавив следующее в конец файла:

```
smtp-amavis     unix    -       -       -       -       2       smtp
        -o smtp_data_done_timeout=1200
        -o smtp_send_xforward_command=yes
        -o disable_dns_lookups=yes
        -o max_use=20

127.0.0.1:10025 inet    n       -       -       -       -       smtpd
        -o content_filter=
        -o local_recipient_maps=
        -o relay_recipient_maps=
        -o smtpd_restriction_classes=
        -o smtpd_delay_reject=no
        -o smtpd_client_restrictions=permit_mynetworks,reject
        -o smtpd_helo_restrictions=
        -o smtpd_sender_restrictions=
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o smtpd_data_restrictions=reject_unauth_pipelining
        -o smtpd_end_of_data_restrictions=
        -o mynetworks=127.0.0.0/8
        -o smtpd_error_sleep_time=0
        -o smtpd_soft_error_limit=1001
        -o smtpd_hard_error_limit=1000
        -o smtpd_client_connection_count_limit=0
        -o smtpd_client_connection_rate_limit=0
        -o receive_override_options=no_header_body_checks,no_unknown_recipient_checks,no_milters
```

Также добавьте следующие две строки непосредственно после транспортного сервиса "*pickup*":

```
        -o content_filter=
        -o receive_override_options=no_header_body_checks
```

Это предотвратит от попадания в спам сообщений, созданных в качестве отчётов о спаме.

Теперь перезапустите Postfix:

```
sudo service postfix restart
```

Фильтрация содержимого с поиском спама и вирусов теперь включена.

5.2.5. Amavisd-new и Spamassassin

При интеграции Amavisd-new со Spamassassin, если вы решили заблокировать фильтр Байеса, отредактировав `/etc/spamassassin/local.cf`, и использовать cron для обновления ночных правил, то это может привести к ситуации, когда большой объём сообщений об ошибках будет посылаться пользователю *amavis* через задание cron amavisd-new.

Существует несколько способов справиться с этой ситуацией:

- Настройте ваш MDA на фильтрацию сообщений, которые вы не желаете видеть.
- Измените `/usr/sbin/amavisd-new-cronjob` на проверку, что *use_bayes 0*. Например, отредактируйте `/usr/sbin/amavisd-new-cronjob`, добавив следующее в начало до строк *проверки*:

```
egrep -q "^[ \t]*use_bayes[ \t]*0" /etc/spamassassin/local.cf && exit 0
```

5.3. Тестирование

Для начала проверьте, что Amavisd-new SMTP активен:

```
telnet localhost 10024
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 [127.0.0.1] ESMTP amavisd-new service ready
^]
```

В заголовке сообщения, которое проходит через фильтр контента, вы должны увидеть:

```
X-Spam-Level:
X-Virus-Scanned: Debian amavisd-new at example.com
X-Spam-Status: No, hits=-2.3 tagged_above=-1000.0 required=5.0 tests=AWL, BAYES_00
X-Spam-Level:
```

В вашем случае вывод может отличаться, но важно то, что здесь есть записи *X-Virus-Scanned* и *X-Spam-Status*.

5.4. Устранение проблем

Лучший способ узнать, почему что-то пошло не так — проверить журнальные файлы.

- Для инструкций по журналам Postfix смотрите раздел *Раздел 1.7, «Устранение проблем» [280]*.
- Amavisd-new использует Syslog для отправки сообщений в `/var/log/mail.log`. Количество деталей можно увеличить, добавив опцию *$log_level* в `/etc/amavis/conf.d/50-user` и задав её значение в диапазоне от 1 до 5.

  ```
  $log_level = 2;
  ```

  

  Когда вывод журнала Amavisd-new увеличивается, то вывод журнала Spamassassin также увеличивается.
- Уровень журналирования ClamAV может быть увеличен редактированием`/etc/clamav/clamd.conf` и установкой следующей опции:

  ```
  LogVerbose true
  ```

  По умолчанию ClamAV отправляет сообщения журнала в `/var/log/clamav/clamav.log`.

После изменения уровня журналирования не забывайте перезапускать сервис для активации новых настроек. Также после установления причины проблем будет хорошей идеей вернуть уровень журналирования к нормальному значению.

5.5. Ссылки

Для дополнительной информации о фильтрации почты смотрите следующие ссылки:

- *Документация Amavisd-new*[21]
- *Документация ClamAV*[22] и *ClamAV Wiki*[23]

[21] http://www.ijs.si/software/amavisd/amavisd-new-docs.html
[22] http://www.clamav.net/doc/latest/html/
[23] http://wiki.clamav.net/Main/WebHome

- *Spamassassin Wiki*[24]
- *Домашняя страница Pyzor*[25]
- *Домашняя страница Razor*[26]
- *DKIM.org*[27]
- *Postfix Amavis New*[28]

Также не стесняйтесь задавать вопросы в *#ubuntu-server* канале IRC на *freenode*[29].

[24] http://wiki.apache.org/spamassassin/
[25] http://sourceforge.net/apps/trac/pyzor/
[26] http://razor.sourceforge.net/
[27] http://dkim.org/
[28] https://help.ubuntu.com/community/PostfixAmavisNew
[29] http://freenode.net

Глава 16. Приложения для чата

1. Обзор

В этом разделе мы объясним, как установить и настроить IRC-сервер ircd-irc2. Мы также обсудим установку и настройку сервера обмена мгновенным сообщениями Jabber.

2. IRC-сервер

В репозиториях Ubuntu находится много серверов IRC (Internet Relay Chat). В этой секции будет рассмотрено, как установить и настроить оригинальный IRC-сервер ircd-irc2.

2.1. Установка

Для установки ircd-irc2, выполните команду в окне терминала:

```
sudo apt-get install ircd-irc2
```

Конфигурационные файлы хранятся в каталоге `/etc/ircd`. Документация доступна в каталоге `/usr/share/doc/ircd-irc2`.

2.2. Конфигурация

Установки IRC могут быть сделаны в конфигурационном файле `/etc/ircd/ircd.conf`. Вы можете задать сетевое имя сервера в этом файле, изменив следующую строку:

```
M:irc.localhost::Debian ircd default configuration::000A
```

Пожалуйста, убедитесь, что вы добавили записи DNS для сетевого имени IRC-сервера. Например, если вы установили в качестве сетевого имени IRC irc.livecipher.com, убедитесь, что irc.livecipher.com разрешается вашим DNS-сервером. Сетевое имя IRC не обязательно должно совпадать с общим сетевым именем.

Информация об администраторе IRC может быть настроена редактированием следующей строки:

```
A:Organization, IRC dept.:Daemon <ircd@example.irc.org>:Client Server::IRCnet:
```

Вы можете добавить отдельные строки для настройки списка прослушиваемых портов IRC, для настройки информации по Оператору, для настройки аутентификации клиентов и т.д. За подробностями обратитесь к примеру файла настройки `/usr/share/doc/ircd-irc2/ircd.conf.example.gz`.

Заголовок IRC, который будет показан на клиенте IRC при соединении с сервером, может быть установлен в файле`/etc/ircd/ircd.motd`.

После внесения необходимых изменений в файл настройки, вам надо перезапустить сервер IRC, используя следующую команду:

```
sudo service ircd-irc2 restart
```

2.3. Ссылки

Вам также может быть будет интересно посмотреть на другие IRC-серверы, доступные в репозитории Ubuntu. Среди них ircd-ircu и ircd-hybrid.

- Смотрите также *IRCD FAQ*[1] для подробностей об IRC-сервере.

[1] http://www.irc.org/tech_docs/ircnet/faq.html

3. Сервер мгновенных сообщений Jabber

Jabber — это популярный протокол мгновенных сообщений, основанный на XMPP, открытом стандарте мгновенных сообщений, и используется многими популярными приложениями. Этот раздел посвящен установке сервера *Jabberd 2* для локальной сети. Эта конфигурация может быть также адаптирована для предоставления сервиса сообщений для пользователей через интернет.

3.1. Установка

Для установки jabberd2 введите в терминале:

```
sudo apt-get install jabberd2
```

3.2. Конфигурация

Пара конфигурационных XML файлов будет использована для настройки jabberd2 для аутентификации пользователей с использованием *Berkeley DB*. Однако jabberd2 может быть настроен на использование LDAP, MySQL, PostgreSQL и др. для аутентификации пользователей.

Сначала отредактируем `/etc/jabberd2/sm.xml`, изменив следующее:

```
<id>jabber.example.com</id>
```

Замените *jabber.example.com* на сетевое имя или другой идентификатор вашего сервера.

Теперь в секции <storage> замените <driver> на:

```
<driver>db</driver>
```

Далее редактируем в `/etc/jabberd2/c2s.xml` секцию *<local>*:

```
<id>jabber.example.com</id>
```

А в секции <authreg> устанавливаем секцию <module>:

```
<module>db</module>
```

Наконец перезагружаем jabberd2 для подключения новых настроек:

```
sudo service jabberd2 restart
```

Теперь вы можете соединиться с сервером, используя Jabber-клиент, например, Pidgin.

Преимущество использования Berkeley DB заключается в том, что после настройки не требуется никакого дополнительного управления. Если вам требуется больше контроля над учётными записями пользователей и их правами, предпочтительнее использовать другой метод аутентификации.

3.3. Ссылки

- *Jabberd2 Web Site*[2] содержит больше информации по настройкам Jabberd2.
- Для дополнительных опций аутентификации смотрите *Jabberd2 Install Guide*[3].
- Также страница *Setting Up Jabber Server Ubuntu Wiki*[4] содержит дополнительную информацию.

[2] http://codex.xiaoka.com/wiki/jabberd2:start
[3] http://www.jabberdoc.org/
[4] https://help.ubuntu.com/community/SettingUpJabberServer

Глава 17. Система контроля версий

Контроль версий — это искусство управления изменениями в информации. Данный инструмент издавна был важен для программистов, которые обычно вносят небольшие изменения в программы, а затем, на следующий день, отменяют эти изменения. Однако польза от систем контроля версий простирается далеко за границы мира разработки программного обеспечения. Место для систем контроля версий есть везде, где люди используют компьютеры для управления часто изменяющейся информацией

1. Bazaar

Bazaar — это новая система контроля версий, финансируемая Canonical, коммерческой компанией, стоящей за Ubuntu. В отличие от Subversion и CVS, которые поддерживают только централизованную модель хранения, Bazaar также поддерживает *распределённый контроль версий*, давая людям возможность взаимодействовать эффективнее. В сущности, Bazaar разработан для увеличения уровня участия сообщества в проектах с открытым исходным кодом.

1.1. Установка

Чтобы установить bzr, введите в консоли следующую команду:

```
sudo apt-get install bzr
```

1.2. Конфигурация

Чтобы представиться bzr, используйте команду *whoami* таким образом:

```
$ bzr whoami 'Joe Doe <joe.doe@gmail.com>'
```

1.3. Изучение Bazaar

Bazaar поставляется со встроенной документацией, устанавливаемой по умолчанию в /usr/share/doc/bzr/html. "Руководство" — это наилучший момент для начала работы. Команда bzr также имеет встроенную справку:

```
$ bzr help
```

Чтобы узнать больше о команде *foo*:

```
$ bzr help foo
```

1.4. Взаимодействие с Launchpad

Будучи крайне полезным, как одиночная система, Bazaar имеет хорошую опциональную возможность интеграции с *Launchpad*[1], системой совместной разработки, используемой Canonical и широким сообществом открытого программного обеспечения, чтобы координировать разработку и улучшать Ubuntu. Информацию о том, как Bazaar может быть использован совместно

[1] https://launchpad.net/

с Launchpad для взаимодействия в рамках проектов с открытым исходным кодом, смотрите здесь: *http://bazaar-vcs.org/LaunchpadIntegration*[2].

[2] http://bazaar-vcs.org/LaunchpadIntegration/

2. Git

Git is an open source distributed version control system originally developped by Linus Torvalds to support the development of the linux kernel. Every Git working directory is a full-fledged repository with complete history and full version tracking capabilities, not dependent on network access or a central server.

2.1. Установка

Система контроля версий git устанавливается следующей командой

```
sudo apt-get install git-core
```

2.2. Конфигурация

Каждый пользователь git сначала должен представиться git, выполнив следующие две команды:

```
git config --global user.email "ваша@почта.com"
git config --global user.name "Ваше имя"
```

2.3. Основы использования

Приведённого выше обычно достаточно для использования git распределённым и безопасным образом, при условии, что пользователи имеют доступ к компьютеру, выполняющему роль сервера, через SSH. На сервере создать новый репозиторий можно командой

```
git init --bare /path/to/repository
```

This creates a bare repository, that cannot be used to edit files directly. If you would rather have a working copy of the contents of the repository on the server, ommit the *--bare* option.

Любой клиент с доступом к компьютеру по SSH затем может клонировать репозиторий командой

```
git clone username@hostname:/path/to/repository
```

Once cloned to the client's machine, the client can edit files, then commit and share them with:

```
cd /path/to/repository
#(edit some files
```

```
git commit -a # Commit all changes to the local version of the repository
git push origin master # Push changes to the server's version of the repository
```

2.4. Установка сервера gitolite

Хотя изложенного выше достаточно для создания, клонирования и редактирования репозиториев, пользователи, которые хотят установить git на свой сервер, скорее всего пожелают, чтобы git работал подобно более традиционному серверу системы контроля исходного кода, с многопользовательским доступом и управлением правами доступа. Предлагаемое решение — установить gitolite следующей командой:

```
sudo apt-get install gitolite
```

2.5. Конфигурация Gitolite

Конфигурация сервера gitolite немного отличается от большинства других серверов в Unix-подобных системах. В отличие от традиционных конфигурационных файлов в /etc/, gitolite хранит свою конфигурацию в git-репозитории. Поэтому в первую очередь для настройки вновь установленного сервера необходимо разрешить доступ к конфигурационному репозиторию.

Прежде всего, давайте создадим пользователя для доступа к gitolite.

```
sudo adduser --system --shell /bin/bash --group --disabled-password --home /home/git git
```

Теперь нам нужно предоставить gitolite сведения об открытом ключе SSH администратора репозитория. Это подразумевает, что текущий пользователь является администратором репозитория.

```
cp ~/.ssh/id_rsa.pub /tmp/$(whoami).pub
```

Let's switch to the git user and import the administrator's key into gitolite.

```
sudo su - git
gl-setup /tmp/*.pub
```

Gitolite will allow you to make initial changes to its configuration file during the setup process. You can now clone and modify the gitolite configuration repository from your administrator user (the user whose public SSH key you imported). Switch back to that user, then clone the configuration repository:

```
exit
```

```
git clone git@$IP_ADDRESS:gitolite-admin.git
cd gitolite-admin
```

gitolite-admin содержит два подкаталога "conf" и "keydir". Конфигурационные файлы находятся в каталоге conf, а каталог keydir содержит список открытых ключей SSH пользователя.

2.6. Управление пользователями gitolite и репозиториями

Adding new users to gitolite is simple: just obtain their public SSH key and add it to the keydir directory as $DESIRED_USER_NAME.pub. Note that the gitolite usernames don't have to match the system usernames - they are only used in the gitolite configuration file to manage access control. Similarly, users are deleted by deleting their public key file. After each change, do not forget to commit the changes to git, and push the changes back to the server with

```
git commit -a
git push origin master
```

Управление репозиториями осуществляется путём редактирования файла conf/gitolite.conf. Синтаксические элементы разделяются пробелами и представляют собой просто список репозиториев с указанием некоторых прав доступа. Вот стандартный пример

```
repo    gitolite-admin
        RW+     =   admin
        R       =   alice

repo    project1
        RW+     =   alice
        RW      =   bob
        R       =   denise
```

2.7. Using your server

To use the newly created server, users have to have the gitolite admin import their public key into the gitolite configuration repository, they can then access any project they have access to with the following command:

```
git clone git@$SERVER_IP:$PROJECT_NAME.git
```

Or add the server's project as a remote for an existing git repository:

```
git remote add gitolite git@$SERVER_IP:$PROJECT_NAME.git
```

3. Subversion

Subversion — это система контроля версий с открытым исходным кодом. Используя Subversion, вы можете сохранять историю изменений файлов и документов. Дерево файлов и папок хранится в центральном репозитории, похожем на обыкновенный файловый архив, за исключением того, что сохраняются любые их модификации.

3.1. Установка

Для доступа к репозиторию Subversion посредством протокола HTTP вы должны установить и настроить веб-сервер. Apache2 гарантированно работает с Subversion. Для установки и настройки сервера Apache2 обратитесь к подразделу HTTP раздела Apache2. Для доступа к репозиторию Subversion посредством протокола HTTPS вы должны установить и настроить цифровой сертификат в веб-сервере Apache2. Для установки и настройки цифрового сертификата обратитесь к подразделу HTTPS раздела Apache2.

Для установки Subversion выполните следующую команду в терминале:

```
sudo apt-get install subversion apache2 libapache2-svn
```

3.2. Настройка сервера

Данный шаг подразумевает, что вы установили в систему пакеты, отмеченные выше. Этот раздел объясняет, как создать репозиторий Subversion и получить доступ к проекту

3.2.1. Создание репозитория Subversion

Репозиторий Subversion можно создать, введя в терминале:

```
svnadmin create /path/to/repos/project
```

3.2.2. Импорт файлов

Как только вы создадите репозиторий, вы сможете *импортировать* в него файлы. Для импорта каталога введите следующую строку в терминале:

```
svn import /путь/к/импортируемой/папке file:///путь/к/репозиторию/проект
```

3.3. Методы доступа

Доступ к репозиториям Subversion можно получить многими различными способами — через локальный диск или с помощью различных сетевых

протоколов. Однако адрес репозитория всегда URL. Таблица показывает, как различные схемы URL соответствуют доступным способам доступа.

Таблица 17.1. Методы доступа

Схема	Метод доступа
file://	Прямой доступ к репозиторию (на локальном диске)
http://	Доступ по протоколу WebDAV к вебсерверу Apache2, умеющему работать с системой Subversion
https://	То же самое, что и http://, но с SSL шифрованием
svn://	Доступ через выборочный протокол к серверу svnserve
svn+ssh://	То же самое, что и svn://, но через SSH тунель

В этом разделе объясняется, как настроить Subversion для всех этих методов доступа. Здесь мы описываем основы. Для более детального описания, обратитесь к *книге svn*[3].

3.3.1. Прямой доступ к репозиторию (file://)

Это самый простой из всех методов доступа. Он не требует запуска никакого процесса сервера Subversion. Этот метод доступа используется для доступа к Subversion с той же машины. Синтакс команды, введенной в строке терминала, следующий:

```
svn co file:///path/to/repos/project
```

или

```
svn co file://localhost/path/to/repos/project
```

Если вы не указали имя хоста, используйте три слэша (///) — два для протокола (в данном случае — файл), плюс первый слэш в пути. Если вы указали имя хоста, используйте два слеша (//).

Права доступа к репозиторию зависят от прав доступа к файловой системе. Если пользователь обладает правами на чтение/запись — он может производить отладку и вносить изменения в репозиторий

3.3.2. Доступ через протокол WebDAV (http://)

Для доступа к хранилищу Subversion через протокол WebDAV вам потребуется настроить сервер Apache2. Добавьте следующий фрагмент между элементами emphasis><VirtualHost>

[3] http://svnbook.red-bean.com/

```
<Location /svn>
 DAV svn
 SVNParentPath /path/to/repos
 AuthType Basic
 AuthName "Your repository name"
 AuthUserFile /etc/subversion/passwd
 Require valid-user
</Location>
```

The above configuration snippet assumes that Subversion repositories are created under `/path/to/repos` directory using **svnadmin** command and that the HTTP user has sufficent access rights to the files (see below). They can be accessible using **http://hostname/svn/repos_name** url.

Изменения в конфигурации apache, подобные показанным выше, требуют перезапуска службы с помощью следующей команды

```
sudo service apache2 reload
```

To import or commit files to your Subversion repository over HTTP, the repository should be owned by the HTTP user. In Ubuntu systems, the HTTP user is **www-data**. To change the ownership of the repository files enter the following command from terminal prompt:

```
sudo chown -R www-data:www-data /путь/к/репозиторию
```

Меняя владельца репозитория на **www-data**, вы потеряете способность импортировать или фиксировать файлы в нем, используя команду **svn import file:///** любым, отличным от **www-data**, пользователем.

Далее, вы должны создать файл `/etc/subversion/passwd`, который будет содержать данные для аутентификации пользователя. Для создания файла выполните в командной строке следующую команду (которая создаст файл и добавит первого пользователя):

```
sudo htpasswd -c /etc/subversion/passwd имя_пользователя
```

Для добавления дополнительных пользователей не задавайте опцию "*-c*", так как она заменяет старый файл на новый. Вместо этого используйте:

```
sudo htpasswd /etc/subversion/passwd user_name
```

Команда запросит ввести пароль. Как только пароль будет введён — пользователь будет добавлен. Теперь, чтобы получить доступ к репозиторию, вам необходимо выполнить эту команду:

```
svn co http://имясервера/svn
```

Передача пароля происходит открытым текстом. Если вы не хотите, чтобы пароль был перехвачен, используйте шифрование трафика с применением SSL. Дополнительные сведения вы можете найти в следующем разделе.

3.3.3. Доступ к протоколу WebDAV с применением SSL (https://)

Доступ к хранилищу Subversion по протоколу WebDAV с SSL шифрованием (https://) аналогичен http://, за исключением того, что вы должны установить и настроить цифровой сертификат для вашего сервера Apache2. Для использования SSL с Subversion добавьте конфигурацию, приведенную выше, в файл `/etc/apache2/sites-available/default-ssl`. Для дополнительной информации по установке Apache2 с SSL смотрите *Раздел 1.3, «Настройка HTTPS» [220]*.

Вы можете установить цифровой сертификат, выпущенный центром сертификации. В качестве альтернативы можно использовать самоподписанный сертификат.

Этот шаг подразумевает, что у вас есть установленный и сконфигурированный цифровой сертификат в веб-сервере Apache 2. Для доступа к репозиторию Subversion обязательно ознакомьтесь с предыдущим разделом! Спобобы доступа такие же, за исключением протокола. Необходимо использовать https:// для доступа к репозиторию Subversion.

3.3.4. Доступ с использованием своего протокола (svn://)

Как только репозиторий Subversion будет создан, можно будет сконфигурировать контроль доступа. Для изменения контроля доступа измените файл `/путь/к/репозиторию/проект/conf/svnserve.conf`. Например, для включения аутентификации уберите комментарий на следующих строчках:

```
# [general]
# password-db = passwd
```

Как только вы раскомментируете вышеуказанные строки, вы можете использовать список пользователей из файла passwd. Итак, отредактируйте файл `passwd` , находящийся в том же каталоге, и добавьте нового пользователя.

```
username = password
```

Чтобы получить больше информации, посмотрите файл.

Теперь, чтобы получить доступ к Subversion через протокол svn:// с того же или с другого компьютера, вы можете запустить сервер Subversion, используя комманду svnserve. Синтаксис:

```
$ svnserve -d --foreground -r /путь/к/репозиторию
# -d -- daemon режим сервиса (невидимый)
# --foreground -- запустить на консоль (полезно для отладки)
# -r -- корень репозитория

Для подробного описания использования команды выполните команду:
$ svnserve --help
```

После запуска этой команды Subversion будет запущен на порту 3690. Для того, что бы сменить репозиторий, необходимо выполнить команду:

```
svn co svn://имяхотса/проект проект --username имя_пользователя
```

В зависимости настроек сервера, может быть запрошен пароль. После аутентификации будет проверен код из репозитория Subversion. Для синхронизации локальной копии и репозитория проекта можно выполнить подкоманду **update**. Синтакс введённой команды следующий:

```
cd каталог_проекта ; svn update
```

Вы можете обратиться к инструкции пользователя, если вас интересует детали использования каждой подкоманды Subversion. На пример, что бы узнать больше про команду "co", введите в терминале следующее:

```
svn co help
```

3.3.5. Access via custom protocol with SSH encryption (svn+ssh://)

Конфигурация и процесс сервера такие же как и в случае с svn://. Более подробно описано в предыдущей секции. На этом этапе подразумевается, что вы выполнили предыдущие шаги и запустили сервер Subversion, используя команду svnserve

Также подразумевается, что на том же компьютере запущен сервер SSH и на него разрешены входящие соединения. Чтобы проверить, попробуйте подключиться к этому компьютеру, используя SSH. Если вы зашли в этот компьютер, значит всё замечательно. Если вы не можете войти в этот компьютер, решите эту проблему перед тем, как приступать к дальнейшим шагам.

Протокол svn+ssh:// применяется, если необходимо подключиться к репозиторию Subversion, используя SSL. В этом случае все передаваемые

данные будут зашифрованы. Для доступа к репозиторию проекта необходимо использовать следующую команду:

```
svn co svn+ssh://ssh_username@hostname/path/to/repos/project
```

Чтобы получить доступ к репозиторию Subversion, используя этот метод, необходимо ввести полный путь (/путь/к/репозиторию/проекту).

Если в указано в настройках, будет запрошен пароль. Необходимо ввести пароль, используемый при подключении через SSH. Если пароль верный, будет проверен код из репозитория Subversion.

4. Ссылки

- *Домашняя страница Bazaar*[4]
- *Launchpad*[5]
- *Домашняя страница Git*[6]
- *Gitolite*[7]
- *Домашняя страница Subversion*[8]
- *Книга Subversion*[9]
- *Easy Bazaar в Ubuntu Wiki*[11][10]
- *Subversion на Ubuntu Wiki*[13][12]

[4] http://bazaar.canonical.com/en/
[5] https://launchpad.net/
[6] http://git-scm.com
[7] https://github.com/sitaramc/gitolite
[8] http://subversion.tigris.org/
[9] http://svnbook.red-bean.com/
[11] https://help.ubuntu.com/community/EasyBazaar
[10] https://help.ubuntu.com/community/EasyBazaar
[13] https://help.ubuntu.com/community/Subversion
[12] https://help.ubuntu.com/community/Subversion

Глава 18. Samba

Компьютерные сети часто состоят из разнородных систем. В то время как управление сетью, полностью состоящей из рабочих станций и серверов Ubuntu — лёгкая задача, некоторые сети должны объединять системы на основе Ubuntu и Microsoft®Windows®, гармонично сосуществующие друг с другом. Эта часть Руководства рассматривает принципы и инструменты, используемые в настройке вашего сервера Ubuntu для взаимодействия с Windows-компьютерами.

1. Введение

Успешное сетевое взаимодействие вашей системы Ubuntu с Windows-клиентами включает в себя обеспечение и интеграцию со службами, распространёнными в окружении Windows. Такие службы поддерживают совместное использование данных и информации о компьютерах и пользователях сети, и могут относиться к трём основным функциональным категориям:

- **Службы доступа к файлам и принтерам**. Использование протокола блока серверных сообщений (SMB) для обеспечения совместного использования файлов, папок, томов, а также общего доступа к принтерам через сеть.
- **Службы каталогов**. Распределение важной информации о компьютерах и пользователях сети с использованием таких технологий, как облегченный протокол доступа к каталогам (LDAP) и Microsoft Active Directory®.
- **Идентификация и доступ**. Установление подлинности компьютера или пользователя сети и определение информации, доступ к которой разрешается компьютеру или пользователю при помощи прав доступа к файлам, групповых политик и службы удостоверения Kerberos.

К счастью, ваша система Ubuntu может предоставить все эти возможности для клиентов Windows и открыть им общий доступ к сетевым ресурсам. Одним из основных компонентов Ubuntu, предназначенных для работы с сетями Windows, является пакет Samba, состоящий из серверных приложений и инструментов SMB.

Этот раздел Руководства по Ubuntu Server познакомит вас с общими принципами работы с Samba, покажет, как установить и настроить необходимые пакеты. Дополнительная, более подробная, информация по Samba может быть найдена на *Сайте проекта Samba*[1].

[1] http://www.samba.org

2. File Server

Один из наиболее часто встречающихся способов объединения в сеть компьютеров под управлением Ubuntu и Windows — настройка Samba в качестве файлового сервера. Этот раздел охватывает настройку сервера Samba для предоставления доступа к файлам для Windows-клиентов.

Сервер будет настроен для предоставления доступа к файлам любому клиенту сети без запроса пароля. Если вам требуется более строгий контроль доступа, смотрите *Раздел 4, «Защита файлового сервера и сервера печати» [331]*

2.1. Установка

Первый шаг — установка пакета samba. Наберите в терминале:

```
sudo apt-get install samba
```

Эта команда установит всё необходимое. Теперь вы готовы к настройке Samba для предоставления доступа к файлам.

2.2. Конфигурация

Главный файл настройки Samba находится здесь: `/etc/samba/smb.conf`. Настройки по умолчанию содержат значительное количество комментариев, описывающих различные варианты настройки.

Не все доступные опции включены в файл настроек по умолчанию. Смотрите страницу руководства man для файла `smb.conf` или посетите *Коллекцию Samba HOWTO*[2] для получения дополнительной информации.

1. Для начала отредактируйте нижеследующие пары ключ/значение в секции *[global]* файла `/etc/samba/smb.conf`:

   ```
   workgroup = EXAMPLE
   ...
   security = user
   ```

   Параметр *security* находится почти в самом низу секции [global] и по умолчанию закомментирован. Для большего соответствия реальной ситуации измените название *EXAMPLE*.

[2] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/

2. Для предоставления доступа к файлам создайте новую секцию в конце файла или раскомментируйте один из примеров:

```
[share]
    comment = Ubuntu File Server Share
    path = /srv/samba/share
    browsable = yes
    guest ok = yes
    read only = no
    create mask = 0755
```

- *comment:* — краткое описание ресурса. Измените для своих нужд.
- *path:* — путь к каталогу, к которому будет открыт доступ

  Этот пример использует `/srv/samba/sharename`, поскольку в соответствии со *Стандартом иерархии файловой системы (FHS)* папка */srv*[3] предназначена для хранения данных, относящихся к сайту. Технически ресурсы Samba могут располагаться в любом месте файловой системы, если для них выставлены правильные права доступа, но всё-таки рекомендуется придерживаться стандартов.
- *browsable:* позволяет Windows-клиентам просматривать каталог ресурса с помощью Windows Explorer.
- *guest ok:* позволяет клиентам подключаться к ресурсу без ввода пароля.
- *read only:* определяет, доступен ли ресурс только для чтения, или же предоставлены привилегии на запись. Привилегии на запись доступны только тогда, когда выставлено значение *no*, как показано в примере. Если значение равно *yes*, то ресурс может быть доступен только для чтения (read only).
- *create mask:* определяет права для вновь создаваемых файлов.

3. Теперь, когда Samba настроена, необходимо создать каталог и изменить права доступа. Введите в терминале:

```
sudo mkdir -p /srv/samba/share
sudo chown nobody.nogroup /srv/samba/share/
```

Параметр *-p* указывает mkdir на создание полного дерева папок, если оно не существует.

4. Наконец, перезапустите сервис samba, чтобы применить новую конфигурацию.

[3] http://www.pathname.com/fhs/pub/fhs-2.3.html#SRVDATAFORSERVICESPROVIDEDBYSYSTEM

```
sudo restart smbd
sudo restart nmbd
```

Повторим ещё раз, приведенная выше конфигурация даёт полный доступ любому клиенту в локальной сети. Если вам нужна более защищённая конфигурация, смотрите *Раздел 4, «Защита файлового сервера и сервера печати» [331]*.

Из Windows-клиента у вас теперь есть возможность просматривать совместно используемые папки Ubuntu Server. Если ваш клиент не отображает общую папку автоматически, попробуйте получить доступ к серверу по IP-адресу (например, \\192.168.1.1) в окне проводника Windows. Чтобы проверить, что всё работает, попробуйте создать папку из под Windows.

Для создания ещё одного ресурса просто создайте новую секцию *[dir]* в файле `/etc/samba/smb.conf` и перезапустите *Samba*. Перед этим убедитесь, что каталог, к которому вы хотите открыть доступ, существует и имеет правильные права доступа.

Общий каталог "*[share]*" и путь `/srv/samba/share` — это просто примеры. Измените имя совместно используемого ресурса и путь так, как вам необходимо. Будет хорошей идеей использовать для ресурса то же имя, что и у соответствующего каталога в файловой системе. Ещё одним примером может быть имя ресурса *[qa]* с путём `/srv/samba/qa`.

2.3. Ресурсы

- Если вам нужны более сложные примеры конфигураций Samba, смотрите *Samba HOWTO Collection*[4].
- Данное руководство также доступно в *печатном виде*[5].
- Книга O'Reilly *Использование Samba*[6] — ещё один хороший источник знаний.
- Страница *Ubuntu Wiki Samba* [7].

[4] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/
[5] http://www.amazon.com/exec/obidos/tg/detail/-/0131882228
[6] http://www.oreilly.com/catalog/9780596007690/
[7] https://help.ubuntu.com/community/Samba

3. Сервер печати

Ещё одной распространенной сферой применения Samba является предоставление доступа к принтерам, установленным на сервере Ubuntu локально или в сети. Так же, как и *Раздел 2, «File Server» [326]*, этот раздел опишет процесс настройки Samba, позволяющий любому клиенту локальной сети использовать установленный принтер без необходимости вводить имя пользователя и пароль.

Если вам нужна более защищённая конфигурация, смотрите *Раздел 4, «Защита файлового сервера и сервера печати» [331]*.

3.1. Установка

Перед установкой и настройкой Samba неплохо бы иметь уже рабочую систему CUPS. Для разъяснений обратитесь к разделу *Раздел 4, «CUPS — сервер печати» [269]*

Для установки пакета samba введите в терминале:

```
sudo apt-get install samba
```

3.2. Конфигурация

После установки Samba отредактируйте `/etc/samba/smb.conf`. Измените рабочую группу *workgroup* согласно вашей сети, и измените *security* на *user*:

```
workgroup = EXAMPLE
...
security = user
```

В разделе *[printers]* измените опцию *guest ok* на *yes*:

```
browsable = yes
guest ok = yes
```

После редактирования `smb.conf` перезапустите Samba:

```
sudo restart smbd
sudo restart nmbd
```

По умолчанию Samba будет публиковать любые настроенные принтеры. Просто установите принтер локально на ваших клиентских системах Windows.

3.3. Ресурсы

- Если вам нужны более сложные примеры конфигураций Samba, смотрите *Samba HOWTO Collection*[8].
- Данное руководство также доступно в *печатном виде*[9].
- Книга O'Reilly *Использование Samba*[10] — ещё один хороший источник знаний.
- Также посетите *Веб-сайт CUPS*[11], где вы сможете найти больше информации о настройке CUPS.
- Страница *Ubuntu Wiki Samba* [12].

[8] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/
[9] http://www.amazon.com/exec/obidos/tg/detail/-/0131882228
[10] http://www.oreilly.com/catalog/9780596007690/
[11] http://www.cups.org/
[12] https://help.ubuntu.com/community/Samba

4. Защита файлового сервера и сервера печати

4.1. Режимы безопасности Samba

В протоколе CIFS (Common Internet Filesystem) доступно два уровня безопасности — *уровень пользователей* и *уровень ресурсов*. Реализация *режима безопасности* в Samba обеспечивает большую гибкость, поскольку в ней доступно четыре способа установки защиты на уровне пользователей и один способ установки защиты на уровне ресурсов:

- *security = user:* будет требовать от клиентов указания имени пользователя и пароля при подключении к ресурсам. Учётные записи Samba хранятся отдельно от системных учётных записей, но пакет libpam-smbpass позволит синхронизировать системных пользователей и их пароли с базой данных пользователей Samba.
- *security = domain:*. Этот режим позволяет серверу Samba представляться Windows-клиентам как первичный контроллер домена (PDC), резервный контроллер домена (BDC) или сервер-участник домена (DMS). Для подробной информации смотрите *Раздел 5, «As a Domain Controller» [337]*.
- *security = ADS:* позволяет серверу Samba присоединиться к сомену Active Directory как полноправный участник. Более подробно читайте *Раздел 6, «Active Directory Integration» [342]*.
- *security = server:*. Этот режим остался с тех пор, когда Samba могла быть сервером-участником, и по некоторым причинам безопасности не должн использоваться. Смотрите раздел *Безопасность сервера*[13] руководства Samba для дальнейшей информации.
- *security = share:* позволяет клиентам подключаться к ресурсу без указания имени пользователя и пароля.

Выберите тот режим безопасности, который лучше подходит к вашей ситуации в зависимости от задач, которые выполняет ваш сервер Samba.

4.2. Security = User

В этом разделе мы настроим файловый сервер и сервер печати на Samba, описанный в *Раздел 2, «File Server» [326]* и *Раздел 3, «Сервер печати» [329]*, таким образом, чтобы он требовал аутентификации.

[13] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/ServerType.html#id349531

Для начала установите пакет libpam-smbpass, который позволит синхронизировать пользователей системы с базой данных пользователей Samba.

```
sudo apt-get install libpam-smbpass
```

Если вы выберете задачу *Samba Server* в процессе установки, то libpam-smbpass будет установлен автоматически.

Отредактируйте `/etc/samba/smb.conf`, изменив в разделе *[share]*:

```
    guest ok = no
```

Наконец, перезапустите Samba, чтобы новые параметры вступили в силу:

```
sudo restart smbd
sudo restart nmbd
```

Теперь, при подключении к общим папкам и принтерам вам потребуется ввести имя пользователя и пароль.

Если вы выбрали подключение сетевого диска к ресурсу, то можете выбрать флажок «Подключать наново при входе в систему», который потребует от вас ввести имя пользователя и пароль один раз, дол тех пор, пока пароль не будет сменен.

4.3. Безопасность ресурсов

Существует несколько вариантов увеличения безопасности индивидуальных ресурсов. В этом разделе будет рассмотрены самые распространённые из них, на примере *[share]*.

4.3.1. Группы

Группы определяют набор компьютеров или пользователей, имеющих одинаковый уровень доступа к определенным сетевым ресурсам, и предоставляют средство для структурирования контроля доступа к ресурсам. Например, если группа *qa* определена и в нее входят пользователи *freda*, *danika* и *rob*, а в другую существующую группу *support* входят *danika*, *jeremy* и *vincent*, тогда определенный сетевой ресурс, настроенный для разрешения доступа группе *qa*, будет доступен для пользователей freda, danika, и rob, но не для jeremy или vincent. Так как пользователь *danika* входит в обе группы, *qa* и *support*, она будет иметь доступ к ресурсам, настроенным для доступа обеих групп, в то же время все другие пользователи будут иметь доступ только к тем ресурсам, которые непосредственно доступны для группы, в которую они входят.

По умолчанию Samba просматривает локальные системные группы, описанные в файле `/etc/group`, чтобы определить, какой пользователь к какой группе принадлежит. Для более полной информации о том, как добавить или удалить пользователя из группы, смотрите *Раздел 1.2, «Добавление и удаление пользователей» [173]*.

При определении групп в файле настроек Samba `/etc/samba/smb.conf` имя группы должно начинаться с символа "@". Например, если вы хотите определить группу *sysadmin* в определённом разделе файла `/etc/samba/smb.conf`, имя группы необходимо указать как **@sysadmin**.

4.3.2. Права доступа к файлам

Разрешения на файл определяют явно заданные права компьютера или пользователя использовать определённый каталог, файл или набор файлов. Для определения этих разрешений необходимо отредактировать файл `/etc/samba/smb.conf` и указать конкретные разрешения определенного ресурса.

Например, если у вас определён ресурс Samba под именем *share*, и вы хотите дать разрешения *только-чтение* группе пользователей *qa*, но при этом хотите разрешить запись группе *sysadmin* и пользователю *vincent*, то для этого вы можете отредактировать файл `/etc/samba/smb.conf`, добавив в него следующие ниже строки *[share]*:

```
read list = @qa
write list = @sysadmin, vincent
```

Другими возможными разрешениями Samba являются *административные* разрешения, котрые могут быть объявлены для определённого ресурса. Пользователи с административными правами могут читать, записывать или изменять информацию, хранящуюся на ресурсе, для которого определены административные права пользователя.

Например, если вы захотите предоставить пользователю *melissa* права администратора на ресурс *share*, отредактируйте файл `/etc/samba/smb.conf` и добавьте следующую строку в секции *[share]*:

```
admin users = melissa
```

После изменения `/etc/samba/smb.conf`, перезапустите Samba, чтобы изменения вступили в силу:

```
sudo restart smbd
sudo restart nmbd
```

Для того, чтобы работали *списки чтения* и *списки записи*, режим безопасности Samba *не* должен быть установлен в *security = share*

Теперь, когда Samba настроена на ограничение доступа групп пользователей к ресурсу, необходимо обновить разрешения файловой системы.

Традиционные разрешения на файлы в Linux не совсем совпадают со списками контроля доступа (ACL) Windows NT. К счастью, ACL POSIX, также присутствующие в сервере Ubuntu, обеспечивают более надёжный контроль. Например, для включения ACL для файла `/srv` на файловой системе EXT3, отредактируйте файл `/etc/fstab`, добавив опцию *acl*:

```
UUID=66bcdd2e-8861-4fb0-b7e4-e61c569fe17d /srv  ext3    noatime,relatime,acl 0       1
```

После чего перемонтируйте раздел:

```
sudo mount -v -o remount /srv
```

В вышеприведённом примере предполагается, что `/srv` находится на отдельном разделе. Если же `/srv` или ваш собственный путь к ресурсу входит в корневой раздел `/`, может потребоваться перезагрузка компьютера.

Для соответствия вышеуказанной конфигурации Samba, группе *sysadmin* будут предоставлены права на чтение, запись и выполнение в `/srv/samba/share`, группе *qa* — на чтение и выполнение, а владельцем файлов будет являться пользователь *melissa*. Введите в терминале следующие команды:

```
sudo chown -R melissa /srv/samba/share/
sudo chgrp -R sysadmin /srv/samba/share/
sudo setfacl -R -m g:qa:rx /srv/samba/share/
```

Команда setfacl даёт права на *выполнение* всех файлов в каталоге `/srv/samba/share`, что может потребоваться для вашей ситуации или нет.

Теперь, работая в среде Windows-клиента, вы можете заметить, что к файлам применились новые разрешения. Смотрите страницы руководства man программ acl и setfacl для получения большей информации о POSIX ACL.

4.4. Профиль Samba для AppArmor

Ubuntu поставляется с модулем безопасности для AppArmor, который обеспечивает мандатный контроль доступа. Встроенный профиль AppArmor

для Samba должен буть адаптирован под вашу конфигурацию. Для получения большей информации по использованию AppArmor смотрите *Раздел 4, «AppArmor» [189]*.

Есть встроенные профили AppArmor для файлов `/usr/sbin/smbd` и `/usr/sbin/nmbd`, бинарных файлов демонов Samba. Они содержатся в пакете apparmor-profiles. Для установки этого пакета введите в терминале:

```
sudo apt-get install apparmor-profiles apparmor-utils
```

Этот пакет содержит профили для нескольких остальных бинарных файлов.

По умолчанию, профили для smbd и nmbd находятся в режиме *жалоб (complain)*, позволяя Samba работать без изменения профиля, записывая отчёты об ошибках. Для перевода профиля для smbd в *принудительный (enforce)* режим, чтобы заставить Samba работать так, как и ожидалось, профиль должен быть изменён, чтобы это отразилось на любом каталоге ресурса.

Отредактируйте файл `/etc/apparmor.d/usr.sbin.smbd`, добавив информацию для *[share]* из примера файлового сервера:

```
  /srv/samba/share/ r,
  /srv/samba/share/** rwkix,
```

Теперь переведите профиль в *принудительный* режим и перезагрузите его:

```
sudo aa-enforce /usr/sbin/smbd
cat /etc/apparmor.d/usr.sbin.smbd | sudo apparmor_parser -r
```

Теперь вы сможете читать, записывать и исполнять файлы в разделяемом каталоге как в обычном, и у приложения smbd binary будет доступ только к файлам и каталогам. Обязательно добавьте запись для каждого каталога, настроенного для совместного доступа через Samba. Любые ошибки будут записываться в `/var/log/syslog`.

4.5. Ресурсы

- Если вам нужны более сложные примеры конфигураций Samba, смотрите *Samba HOWTO Collection*[14].
- Данное руководство также доступно в *печатном виде*[15].

[14] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/
[15] http://www.amazon.com/exec/obidos/tg/detail/-/0131882228

- Книга O'Reilly *Using Samba*[16] — хорошее подспорье.
- *Глава 18*[17] коллекции HOWTO по Samba посвящена безопасности.
- Для получения большей информации по Samba и ACL смотрите *Страница ACL в Samba*[18].
- Страница *Ubuntu Wiki Samba* [19].

[16] http://www.oreilly.com/catalog/9780596007690/
[17] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/securing-samba.html
[18] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/AccessControls.html#id397568
[19] https://help.ubuntu.com/community/Samba

5. As a Domain Controller

Несмотря на то, что Samba не может выступать в качестве первичного контроллера домена Active Directory, сервер может быть сконфигурирован как контроллер домена Windows NT4. Главным преимуществом этой конфигурации является возможность централизовать мандаты пользователей и компьютеров. Samba может хранить информацию о пользователе во множестве драйверов (бэкэндов).

5.1. Первичный контроллер домена

Этот раздел охватывает настройку Samba в качестве первичного контроллера домена с использованием встроенного драйвера smbpasswd.

1. Для начала установите Samba, а также libpam-smbpass для синхронизации бюджетов пользователей, введя в терминале:

   ```
   sudo apt-get install samba libpam-smbpass
   ```

2. Затем настройте Samba, отредактировав файл `/etc/samba/smb.conf`. Режим *security* должен быть установлен в *user*, а строка *workgroup* должна соответствовать вашей организации:

   ```
   workgroup = EXAMPLE
   ...
   security = user
   ```

3. В закомментированной секции «Domains» добавьте или раскомментируйте следующее (последняя строка была разделена, чтобы соответствовать формату этого документа):

   ```
   domain logons = yes
   logon path = \\%N\%U\profile
   logon drive = H:
   logon home = \\%N\%U
   logon script = logon.cmd
   add machine script = sudo /usr/sbin/useradd -N -g machines -c Machine -d
         /var/lib/samba -s /bin/false %u
   ```

   

   Если вы не хотите использовать *Перемещаемые профили*, оставьте опции *logon home* и *logon path* закомментированными.

   - *domain logons:* предоставляет сервис netlogon, заставляющий Samba работать как контроллер домена.

- *logon path:* указывает на расположение профиля пользователя Windows в его домашнем каталоге. Также возможна настройка секции *[profiles]* для хранения всех профилей в одном каталоге.
- *logon drive:* определяет локальный путь к домашнему каталогу.
- *logon home:* определяет расположение домашнего каталога.
- *logon script:* определяет скрипт, который будет выполняться локально один раз при входе пользователя в систему. Скрипт должен быть расположен в секции *[netlogon]*.
- *add machine script:* скрипт, который автоматически будет создавать *Machine Trust Account*, без которого рабочая станция не может вступить в домен.

  В этом примере группа *machines* должна быть создана с использованием утилиты addgroup. Детальнее расписано тут: *Раздел 1.2, «Добавление и удаление пользователей» [173]*.

4. Раскомментируйте секцию *[homes]*, чтобы разрешить подключать *logon home*.

```
[homes]
   comment = Home Directories
   browseable = no
   read only = no
   create mask = 0700
   directory mask = 0700
   valid users = %S
```

5. Для настройки контроллера домена необходимо настроить секцию *[netlogon]*. Для того, чтобы определить ресурс, раскомментируйте:

```
[netlogon]
   comment = Network Logon Service
   path = /srv/samba/netlogon
   guest ok = yes
   read only = yes
   share modes = no
```

Оригинальный путь к ресурсу *netlogon* — `/home/samba/netlogon`, но, согласно Стандарту иерархии файловой системы (FHS), правильным местом расположения информации, предоставляемой системой для сайта, является */srv*[20].

6. Теперь создайте каталог `netlogon` и пустой (пока) файл скрипта `logon.cmd`:

[20] http://www.pathname.com/fhs/pub/fhs-2.3.html#SRVDATAFORSERVICESPROVIDEDBYSYSTEM

```
sudo mkdir -p /srv/samba/netlogon
sudo touch /srv/samba/netlogon/logon.cmd
```

Вы можете ввести любые скриптовые команды Windows в файле `logon.cmd` для настройки окружения клиента.

7. Перезапустим Samba, чтобы запустить контроллер нового домена:

```
sudo restart smbd
sudo restart nmbd
```

8. Наконец, есть несколько дополнительных команд, необходимых для настройки соответствующих прав.

   Поскольку пользователь *root* по умолчанию отключен, для вступления в домен системная группа должна быть отражена на группу *Domain Admins* в Windows. Сделайте это с помощью утилиты *Domain Admins*, введя в терминале:

```
sudo net groupmap add ntgroup="Domain Admins" unixgroup=sysadmin rid=512 type=d
```

   Измените *sysadmin* на любую группу, которую вы предпочитаете. Кроме того, пользователь, который будет добавлять компьютер в домен, должен быть членом группы *sysadmin* и членом системной группы *admin*. Группа *admin* позволяет использовать sudo.

   Если пользователь всё ещё не имеет полномочий Samba, вы можете добавить их с помощью утилиты smbpasswd, изменяя, соответственно, имя пользователя *sysadmin*:

```
sudo smbpasswd -a sysadmin
```

   Также исключительные права должны быть предоставлены группе *Domain Admins* для того, чтобы работал сценарий *add machine script* (и другие функции администрирования). Это может быть достигнуто выполнением:

```
net rpc rights grant -U sysadmin "EXAMPLE\Domain Admins" SeMachineAccountPrivilege \ SePrintOpe
```

9. Теперь вы можете добавить Windows-клиентов в домен так же, как добавляли их в домен NT4 под управлением сервера Windows.

5.2. Резервный контроллер домена

Наряду с первичным контроллером домена (PDC) полезно иметь и резервный контроллер (BDC). Это позволит клиентам проходить аутентификацию, даже если PDC недоступен.

При настройке Samba в качестве BDC вам нужен механизм синхронизации информации об учетных записях с PDC. Существует много способов решить эту проблему: scp, rsync или использование LDAP в качестве *драйвера passdb*.

Использование LDAP — это самый разумный способ синхронизации информации об учетных записях, поскольку оба контроллера домена могут использовать одну и ту же информацию в реальном времени. Однако, настройка сервера LDAP может быть более сложной для небольшого количества учетных записей пользователей и компьютеров. Для более подробной информации смотрите *Раздел 2, «Samba и LDAP» [134]*.

1. Для начала установите samba и libpam-smbpass. Введите в терминале:

   ```
   sudo apt-get install samba libpam-smbpass
   ```

2. Теперь отредактируйте файл `/etc/samba/smb.conf` и раскомментируйте нижеследующее в секции *[global]*:

   ```
   workgroup = EXAMPLE
   ...
   security = user
   ```

3. В закомментированной секции *Domains* раскомментируйте или добавьте:

   ```
   domain logons = yes
   domain master = no
   ```

4. Убедитесь, что пользователь имеет права читать файлы в `/var/lib/samba`. Например, для того, чтобы разрешить пользователям в группе *admin* выполнять команду scp для файлов, введите:

   ```
   sudo chgrp -R admin /var/lib/samba
   ```

5. Затем синхронизируйте учетные записи пользователей, используя scp, чтобы скопировать каталог `/var/lib/samba` с PDC:

   ```
   sudo scp -r username@pdc:/var/lib/samba /var/lib
   ```

Замените *username* на действительное имя пользователя и *pdc* на имя компьютера или IP-адрес вашего PDC.

6. Наконец, перезапустите samba:

```
sudo restart smbd
sudo restart nmbd
```

Вы можете проверить работу резервного контроллера домена, остановив демон Samba на PDC, а затем попробовав войти в систему на Windows-клиенте, входящем в состав домена.

Ещё одна вещь, на которую стоит обратить внимание: если вы настроили опцию *logon home* как каталог на PDC, то при недоступном PDC доступ к пользовательскому диску *Home* также будет невозможен. Поэтому лучше всего настраивать *logon home* на отдельном сервере, а не на PDC или BDC.

5.3. Ресурсы

- Если вам нужны более сложные примеры конфигураций Samba, смотрите *Samba HOWTO Collection*[21].
- Данное руководство также доступно в *печатном виде*[22].
- Книга O'Reilly *Using Samba*[23] — хорошее подспорье.
- *Глава 4*[24] коллекции HOWTO Samba описывает настройку первичного контроллера домена.
- *Глава 5*[25] коллекции HOWTO Samba описывает настройку резервного контроллера домена.
- Страница *Ubuntu Wiki Samba* [26].

[21] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/
[22] http://www.amazon.com/exec/obidos/tg/detail/-/0131882228
[23] http://www.oreilly.com/catalog/9780596007690/
[24] http://samba.org/samba/docs/man/Samba-HOWTO-Collection/samba-pdc.html
[25] http://us3.samba.org/samba/docs/man/Samba-HOWTO-Collection/samba-bdc.html
[26] https://help.ubuntu.com/community/Samba

6. Active Directory Integration

6.1. Доступ к ресурсу Samba

Ещё одной сферой применения Samba является интеграция в существующую сеть Windows. Как часть домена Active Directory, Samba может быть файловым сервером и сервером печати.

Самым простым способом присоединиться к домену AD является использование Likewise-open. Подробные инструкции смотрите в *документации Likewise Open*[27].

Будучи частью домена Active Directory, введите следующие команды в терминале:

```
sudo apt-get install samba smbfs smbclient
```

Затем отредактируйте файл `/etc/samba/smb.conf`, изменив:

```
workgroup = EXAMPLE
...
security = ads
realm = EXAMPLE.COM
...
idmap backend = lwopen
idmap uid = 50-9999999999
idmap gid = 50-9999999999
```

Перезапустите samba, чтобы применить новые настройки:

```
sudo restart smbd
sudo restart nmbd
```

Теперь у вас должен появиться доступ к любому ресурсу Samba с Windows-клиента. Однако убедитесь, что вы дали соответствующие права пользователям и группам из AD на доступ к ресурсу. Для более детальной информации смотрите *Раздел 4, «Защита файлового сервера и сервера печати» [331]*.

6.2. Доступ к ресурсу Windows

Теперь, когда сервер Samba является частью домена Active Directory, вы можете получить доступ к любому ресурсу сервера Windows.

[27] http://www.beyondtrust.com/Technical-Support/Downloads/files/pbiso/Manuals/ubuntu-active-directory.html

- Чтобы примонтировать файловый ресурс Windows, введите в терминале:

```
mount.cifs //fs01.example.com/share mount_point
```

  Возможен доступ к ресурсам и с компьютеров, не являющихся частью домена AD, но в этом случае будет запрашиваться имя пользователя и пароль.

- Для монтирования ресурса в процессе загрузки поместите строчку в файл `/etc/fstab`, например:

```
//192.168.0.5/share /mnt/windows cifs auto,username=steve,password=secret,rw 0        0
```

- Другим способом скопировать файлы с сервера Windows является использование утилиты smbclient. Чтобы получить список файлов ресурса Windows, введите в терминале:

```
smbclient //fs01.example.com/share -k -c "ls"
```

- Чтобы скопировать файлы с ресурса, введите:

```
smbclient //fs01.example.com/share -k -c "get file.txt"
```

  Это скопирует файл `file.txt` в текущий каталог.

- И чтобы скопировать файл на ресурс:

```
smbclient //fs01.example.com/share -k -c "put /etc/hosts hosts"
```

  `/etc/hosts` будет скопирован в `//fs01.example.com/share/hosts`.

- Опция *-c*, используемая в примере вверху, позволяет выполнять сразу все команды приложения smbclient. Это полезно для написания скриптов и второстепенных файловых операций. Для получения строки *smb: \>*, в которой вы можете выполнять команды по работе с файлами и каталогами, как и в FTP, просто введите:

```
smbclient //fs01.example.com/share -k
```

Замените все вхождения *fs01.example.com/share*, *//192.168.0.5/share*, *username=steve,password=secret* и *file.txt* на IP-адрес вашего сервера, имя компьютера, имя файла и реальное имя пользователя/пароль, соответствующие пользователю с правами доступа к ресурсу.

6.3. Ресурсы

For more smbclient options see the man page: **man smbclient**, also available *online*[28].

The mount.cifs *man page*[29] is also useful for more detailed information.

Страница *Ubuntu Wiki Samba* [30].

[28] http://manpages.ubuntu.com/manpages/trusty/en/man1/smbclient.1.html
[29] http://manpages.ubuntu.com/manpages/trusty/en/man8/mount.cifs.8.html
[30] https://help.ubuntu.com/community/Samba

Глава 19. Резервное копирование

Существует много способов сделать резервную копию системы Ubuntu. Наиболее важным в резервном копировании является выработка *плана резервного копирования*, отражающего что нужно копировать, куда это нужно копировать и как это восстанавливать.

Следующие разделы рассматривают различные способы решения этих задач.

1. Сценарии Shell

Один из простейших вариантов резервного копирования системы — использование *shell script*. Например, сценарий может быть использован для настройки, какие каталоги требуют резервного копирования, и для передачи этих каталогов в качестве аргументов утилите tar, которая создаёт архивные файлы. Архивный файл может быть затем перемещён или скопирован в другое место. Архив также может быть создан на удалённой файловой системе, такой как *NFS*.

Утилита tar создаёт один архивный файл из множества файлов и каталогов.tar может также пропускать файлы через утилиты сжатия, уменьшая таким образом размер архивного файла.

1.1. Простой Shell сценарий

Следующий shell сценарий использует tar для создания архивного файла на удалённо смонтированной файловой системе. Имя архива определяется с помощью дополнительных утилит командной строки.

```
#!/bin/sh
###################################
#
# Backup to NFS mount script.
#
###################################

# What to backup.
backup_files="/home /var/spool/mail /etc /root /boot /opt"

# Where to backup to.
dest="/mnt/backup"

# Create archive filename.
day=$(date +%A)
hostname=$(hostname -s)
archive_file="$hostname-$day.tgz"

# Print start status message.
echo "Backing up $backup_files to $dest/$archive_file"
date
echo

# Backup the files using tar.
tar czf $dest/$archive_file $backup_files

# Print end status message.
echo
```

```
echo "Backup finished"
date

# Long listing of files in $dest to check file sizes.
ls -lh $dest
```

- *$backup_files:* переменная для перечисления, какие каталоги вы желаете сохранять. Список может быть изменён под ваши требования.
- *$day:* переменная, содержащая день недели. Она используется для создания архивных файлов на каждый день недели, обеспечивая историю резервного копирования на семь дней. Существуют иные способы получения такого результата, включая использование утилиты date.
- *$hostname:* переменная, содержащая *короткое* имя системы. Использование сетевого имени в имени архива позволяет вам помещать ежедневные архивы от разных систем в один каталог.
- *$archive_file:* полное имя архива.
- *$dest:* место расположения архивного файла. Каталог должен быть создан и в данном случае *смонтирован* до выполнения сценария резервного копирования. Смотрите раздел *Раздел 2, «Сетевая файловая система (NFS)» [264]* для уточнения деталей использования *NFS*.
- *status messages:* необязательные сообщения, выводимые в консоль с использованием утилиты echo.
- *tar czf $dest/$archive_file $backup_files:* команда tar, используемая для создания архивного файла.
  - *c:* создание архива.
  - *z:* пропускание архива через утилиту сжатия gzip.
  - *f:* вывод в архивный файл. В противном случае tar будет посылать результат в STDOUT.
- *ls -lh $dest:* необязательный оператор, выводящий *-l* длинный вариант списка в *-h* читаемом виде целевого каталога. Это удобно для быстрой проверки размера архивного файла. Эта проверка не заменяет тестирования целостности архивного файла.

Это простой пример сценария резервного копирования, однако есть много опций, которые можно включить в такой сценарий. Смотрите *Раздел 1.4, «Ссылки» [350]* для указаний на ресурсы, предоставляющие более глубокое описание shell сценариев.

1.2. Выполнение сценария

1.2.1. Выполнение из терминала

Простейший способ выполнить приведенный выше сценарий — это скопировать его содержимое в файл. Например, `backup.sh`. Затем ввести в терминале:

```
sudo bash backup.sh
```

Это прекрасный способ проверить сценарий, чтобы убедиться, что всё работает как задумывалось.

1.2.2. Выполнение с помощью cron

Утилита cron может быть использована для автоматизации выполнения сценария. Сервис cron позволяет выполнять сценарии или команды в определенное время.

cron настраивается через записи в файле `crontab`. Файлы `crontab` разделяются на поля:

```
# m h dom mon dow   command
```

- *m:* минуты запуска команды, от 0 до 59.
- *h:* час запуска команды, от 0 до 23.
- *dom:* день месяца для выполнения команды.
- *mon:* месяц даты выполнения команды.
- *dow:* день недели для выполнения команды, от 0 до 7. Воскресенье может быть обозначено как 0 так и 7, оба значения допустимы.
- *command:* выполняемая команда.

Для добавления или изменения записей в файле `crontab` используется команда crontab -e. Кроме того, содержимое файла `crontab` можно просмотреть с помощью команды crontab -l.

Для выполнения приведенного выше сценария backup.sh с помощью cron, введите следующее в терминале:

```
sudo crontab -e
```

Использование sudo для выполнения команды crontab -e изменяет файл пользователя *root*. Это требуется для резервного копирования каталогов, доступ к которым разрешен только root.

Добавьте следующую запись в файл crontab: `crontab`:

```
# m h dom mon dow   command
0 0 * * * bash /usr/local/bin/backup.sh
```

Сценарий backup.sh будет теперь выполняться каждый день в полночь.

Сценарий backup.sh требуется скопировать в каталог `/usr/local/bin/`, чтобы данная запись выполнялась правильно. Сценарий можно разместить где угодно в файловой системе, просто соответственно измените путь к сценарию в crontab.

Для более глубокого изучения опций crontab смотрите секцию *Раздел 1.4, «Ссылки» [350]*.

1.3. Восстановление из архива

Как только архив создан, важно проверить его. Архив может быть проверен выводом списка файлов, которые в нем находятся, но лучшей проверкой будет *восстановление* файлов из архива.

- Чтобы посмотреть содержимое архива, наберите в терминале:

  ```
  tar -tzvf /mnt/backup/host-Monday.tgz
  ```

- Чтобы восстановить файл из архива в другой каталог, введите:

  ```
  tar -xzvf /mnt/backup/host-Monday.tgz -C /tmp etc/hosts
  ```

  Параметр *-C* команды tar перенаправляет извлекаемые файлы в указанный каталог. Приведённый пример извлечёт файл `/etc/hosts` в `/tmp/etc/hosts`. tar создаёт заново структуру каталогов для извлекаемых файлов.

  Также обратите внимание на отсутствие лидирующего "/" в пути извлекаемого файла.

- Чтобы восстановить все файлы из архива, введите следующее:

  ```
  cd /
  sudo tar -xzvf /mnt/backup/host-Monday.tgz
  ```

Это перепишет все файлы, находящиеся в файловой системе.

1.4. Ссылки

- Для дополнительной информации по shell сценариям смотрите руководство *Advanced Bash-Scripting Guide*[1]
- Книга *Teach Yourself Shell Programming in 24 Hours*[2] доступна в сети и является замечательным ресурсом для создания shell сценариев.
- Страница *CronHowto Wiki Page*[3] содержит подробности по дополнительным опциям cron.
- Смотрите руководство *GNU tar Manual*[4] для дополнительных параметров tar.
- Статья *Backup Rotation Scheme*[5] содержит информацию по другим схемам ротации архивов.
- Shell сценарий использует tar для создания архива, но существует много других утилит, которые можно использовать. Например:
  - *cpio*[6]: используется для копирования файлов в и из архива.
  - *dd*[7]: часть пакета coreutils. Утилита нижнего уровня, которая может копировать данные из одного формата в другой.
  - *rsnapshot*[8]: утилита получения снимка файловой системы, используемая для получения копий всей файловой системы.
  - *rsync*[9]: гибкая утилита, используемая для копирования изменённых частей файлов (инкрементное копирование).

[1] http://tldp.org/LDP/abs/html/
[2] http://safari.samspublishing.com/0672323583
[3] https://help.ubuntu.com/community/CronHowto
[4] http://www.gnu.org/software/tar/manual/index.html
[5] http://en.wikipedia.org/wiki/Backup_rotation_scheme
[6] http://www.gnu.org/software/cpio/
[7] http://www.gnu.org/software/coreutils/
[8] http://www.rsnapshot.org/
[9] http://www.samba.org/ftp/rsync/rsync.html

2. Ротация архивов

Shell сценарий в разделе *Раздел 1, «Сценарии Shell» [346]* позволяет создавать только 7 различных архивов. Для сервера, данные на котором меняются нечасто, этого может быть достаточно. Если же сервер содержит большой объем данных, требуется использовать более комплексную схему ротации архивов.

2.1. Ротация NFS архивов

В этой секции наш shell сценарий будет немного модифицирован с целью осуществления схемы ротации 'дед-отец-сын' (ежемесячно-еженедельно-ежедневно):

- ротация будет выполнять *ежедневное* резервное копирование с воскресенья по пятницу.
- в субботу будет *еженедельное* копирование, обеспечивая четыре недельных архива в месяц.
- *ежемесячное* копирование выполняется в первый день месяца, обеспечивая ротацию двух ежемесячных архивов, на основе чётности месяца.

Вот новый сценарий:

```
#!/bin/bash
#####################################
#
# Backup to NFS mount script with
# grandfather-father-son rotation.
#
#####################################

# What to backup.
backup_files="/home /var/spool/mail /etc /root /boot /opt"

# Where to backup to.
dest="/mnt/backup"

# Setup variables for the archive filename.
day=$(date +%A)
hostname=$(hostname -s)

# Find which week of the month 1-4 it is.
day_num=$(date +%d)
if (( $day_num <= 7 )); then
        week_file="$hostname-week1.tgz"
elif (( $day_num > 7 && $day_num <= 14 )); then
        week_file="$hostname-week2.tgz"
```

```
elif (( $day_num > 14 && $day_num <= 21 )); then
        week_file="$hostname-week3.tgz"
elif (( $day_num > 21 && $day_num < 32 )); then
        week_file="$hostname-week4.tgz"
fi

# Find if the Month is odd or even.
month_num=$(date +%m)
month=$(expr $month_num % 2)
if [ $month -eq 0 ]; then
        month_file="$hostname-month2.tgz"
else
        month_file="$hostname-month1.tgz"
fi

# Create archive filename.
if [ $day_num == 1 ]; then
 archive_file=$month_file
elif [ $day != "Saturday" ]; then
        archive_file="$hostname-$day.tgz"
else
 archive_file=$week_file
fi

# Print start status message.
echo "Backing up $backup_files to $dest/$archive_file"
date
echo

# Backup the files using tar.
tar czf $dest/$archive_file $backup_files

# Print end status message.
echo
echo "Backup finished"
date

# Long listing of files in $dest to check file sizes.
ls -lh $dest/
```

Сценарий может запускаться так же, как описано в секции *Раздел 1.2, «Выполнение сценария» [348]*.

Хорошей практикой является выделять для резервного копирования удалённый носитель на случай физического уничтожения сервера. В примере shell сценария носителем для резервной копии является NFS-ресурс на другом сервере. По всей вероятности, перенос NFS-сервера в другое место не является целесообразным. В зависимости от скорости связи, подходящим вариантом может быть копирование архивного файла через соединение по внешней сети (WAN) на сервер, находящийся на другой территории.

Другим вариантом может быть копирование архивного файла на внешний жёсткий диск, который может отключаться и храниться отдельно. Поскольку цены на внешние диски продолжают снижаться, использование пары дисков для каждого уровня архивов может оказаться эффективным вариантом с точки зрения стоимости. Это позволит вам подключать один диск к серверу резервного копирования, а второй хранить отдельно.

2.2. Устройства на магнитной ленте

Устройство на магнитной ленте, подключенное к серверу, может использоваться вместо ресурса NFS. Использование ленточного устройства упрощает ротацию архивов, а также решает проблему хранения резервного носителя отдельно от сервера.

При использовании ленточного устройства часть сценария, касающаяся имени файла, становится ненужной, поскольку данные посылаются непосредственно на устройство. Однако требуются некоторые команды для управления лентой. Это достигается использованием утилиты управления магнитной лентой mt, являющейся частью пакета cpio.

Здесь приведён сценарий, изменённый для использования с ленточным устройством:

```
#!/bin/bash
###################################
#
# Backup to tape drive script.
#
###################################

# What to backup.
backup_files="/home /var/spool/mail /etc /root /boot /opt"

# Where to backup to.
dest="/dev/st0"

# Print start status message.
echo "Backing up $backup_files to $dest"
date
echo

# Make sure the tape is rewound.
mt -f $dest rewind

# Backup the files using tar.
tar czf $dest $backup_files

# Rewind and eject the tape.
```

```
mt -f $dest rewoffl

# Print end status message.
echo
echo "Backup finished"
date
```

По умолчанию именем ленточного SCSI устройства является `/dev/st0`. Используйте подходящий путь к устройству для вашей системы.

Восстановление с ленточного устройства в основном такое же, как и из файла. Просто перемотайте ленту и используйте путь к устройству вместо пути к файлу. Например, для восстановления файла `/etc/hosts` в `/tmp/etc/hosts` используйте следующее:

```
mt -f /dev/st0 rewind
tar -xzf /dev/st0 -C /tmp etc/hosts
```

3. Bacula

Bacula — это программа резервного копирования, позволяющая вам сохранять, восстанавливать и проверять данные через вашу сеть. Для Bacula существуют клиенты под Linux, Windows и Mac OS X, что превращает её в кросс-платформенное сетевое решение.

3.1. Обзор

Bacula Bacula состоит из нескольких компонентов и сервисов для управления тем, какие файлы сохранять и где хранить резервные копии:

- Bacula Director: сервис, который управляет всеми операциями резервного копирования, восстановления, проверки и архивации.
- Bacula Console: приложение, позволяющее взаимодействовать с Director. Существует три версии Console:
  - текстовая версия, основанная на командной строке.
  - графический пользовательский интерфейс (GUI) для Gnome, основанный на GTK+.
  - графический интерфейс на базе wxWidgets.
- Bacula File: программа, известная также под названием Bacula Client. Это приложение устанавливается на компьютерах, на которых производится резервное копирование, и оно отвечает на данные, отправленные по запросу Director.
- Bacula Storage: программа, которая выполняет хранение и восстановление данных на физических носителях.
- Bacula Catalog: отвечает за поддержку файловых индексов и томов баз данных для всех сохраняемых фалов, допуская быстрое нахождение и восстановление сохраненных файлов. Catalog поддерживает три различных базы данных: MySQL, PostgreSQL и SQLite.
- Bacula Monitor: позволяет отслеживать работу Director и сервисов File и Storage. На данный момент Monitor доступен только в виде GTK+ GUI приложения.

Эти сервисы и приложения могут быть запущены на разных серверах и клиентах или их можно установить на одной машине, если требуется резервное копирование одного диска или тома.

3.2. Установка

При использовании MySQL или PostgreSQL в качестве базы данных, вам уже должны быть доступны эти сервисы.Bacula не будет устанавливать их для вас.

Существует несколько пакетов, содержащих различные компоненты Bacula. Для установки Bacula введите в терминале:

```
sudo apt-get install bacula
```

При установке по умолчанию пакет bacula будет использовать базу данных MySQL для Catalog. Если вы хотите использовать SQLite или PostgreSQL, установите соответственно пакет bacula-director-sqlite3 или bacula-director-pgsql.

В процессе установки у вас спросят данные об *администраторе* базы данных и *владельце* базы данных *bacula*. Администратор базы данных требуется для получения необходимых прав на создание базы данных. Дополнительную информацию смотрите в разделе *Раздел 1, «MySQL» [237]*.

3.3. Конфигурация

Файлы настройки Bacula форматированы на основе *ресурсов*, включающих*directives*, обрамлённые фигурными скобками «{}».Каждый компонент Bacula имеет индивидуальный файл в каталоге `/etc/bacula`.

Различные компоненты Bacula должны авторизовывать себя друг для друга. Это решается использованием директивы *password*. Например, пароль в ресурсе *Storage* файла `/etc/bacula/bacula-dir.conf`должен соответствовать паролю ресурса *Director* файла `/etc/bacula/bacula-sd.conf`.

По умолчанию настраивается задание резервного копирования *Client1* для архивирования Bacula. Если вы планируете использовать сервер для резервного копирования более чем на одном клиенте, вам потребуется изменить имя этого задания на что-то более осмысленное. Для переименования отредактируйте файл `/etc/bacula/bacula-dir.conf`:

```
#
# Define the main nightly save backup job
#   By default, this job will back up to disk in
Job {
  Name = "BackupServer"
  JobDefs = "DefaultJob"
  Write Bootstrap = "/var/lib/bacula/Client1.bsr"
}
```

В примере имя задания изменено на *BackupServer* в соответствии с сетевым именем машины. Можете заменить «BackupServer» на соответствующее сетевое имя вашего сервера или другое описательное название.

Требуется использовать *Console* для запросов к *Director* по поводу заданий, но чтобы *non-root* мог использовать Console, он должен быть включён в группу *bacula*. Чтобы добавить пользователя в группу bacula, введите следующую команду в терминале:

```
sudo adduser $username bacula
```

Замените *$username* на актуальное имя пользователя. Также, если вы добавили в группу текущего пользователя, вам придется выйти из системы и зайти снова, чтобы применились новые права доступа.

3.4. Создание резервной копии локального сервера

Данная секция описывает процесс создания архивной копии единственного сервера на магнитной ленте.

- Для начала требуется настроить *устройство хранения*. Отредактируйте `/etc/bacula/bacula-sd.conf`, добавив:

```
Device {
  Name = "Tape Drive"
  Device Type = tape
  Media Type = DDS-4
  Archive Device = /dev/st0
  Hardware end of medium = No;
  AutomaticMount = yes;               # when device opened, read it
  AlwaysOpen = Yes;
  RemovableMedia = yes;
  RandomAccess = no;
  Alert Command = "sh -c 'tapeinfo -f %c | grep TapeAlert'"
}
```

  Этот пример для ленточного устройства *DDS-4*. Измените «Media Type» и «Archive Device» в соответствии с вашим оборудованием.

  Вы также можете раскомментировать в файле один из нескольких других примеров.

- После редактирования файла `/etc/bacula/bacula-sd.conf` сервис Storage требуется перезагрузить:

```
sudo service bacula-sd restart
```

- Теперь добавьте *Storage* ресурс в `/etc/bacula/bacula-dir.conf` чтобы использовать новые устройства:

```
# Definition of "Tape Drive" storage device
Storage {
```

```
  Name = TapeDrive
  # Do not use "localhost" here
  Address = backupserver                # N.B. Use a fully qualified name here
  SDPort = 9103
  Password = "Cv70F6pf1t6pBopT4vQOnigDrR0v3LT3Cgkiyjc"
  Device = "Tape Drive"
  Media Type = tape
}
```

Директива *Address* должна быть полностью квалифицированным доменным именем (FQDN) сервера. Замените *backupserver* на актуальное сетевое имя.

Так же, убедитесь, что *Пароль* директивы соответствует паролю строки в `/etc/bacula/bacula-sd.conf`.

- Создайте новый *Набор файлов*, который определит какие директории добавить для резерного хранения:

```
# Набор файлов для архивации локального сервера.
FileSet {
  Name = "LocalhostFiles"
  Include {
    Options {
      signature = MD5
      compression=GZIP
    }
    File = /etc
    File = /home
  }
}
```

Этот *FileSet* задает резервное копирование для каталогов `/etc` и `/home`. Директивы ресурса *Options* настраивают FileSet на создание контрольных сумм MD5 для каждого сохраненного файла и сжатие файлов с использованием GZIP.

- Далее создайте новое *расписание* для задачи резервирования

```
# Расписание архивации локального сервера -- Ежедневно.
Schedule {
  Name = "LocalhostDaily"
  Run = Full daily at 00:01
}
```

Задача будет запускаться каждый день в 00:01 или 12:01. Доступно ещё много опций расписания

- Наконец, создайте *Задачу*:

```
# Архивация локального сервера.
Job {
  Name = "LocalhostBackup"
  JobDefs = "DefaultJob"
  Enabled = yes
  Level = Full
  FileSet = "LocalhostFiles"
  Schedule = "LocalhostDaily"
  Storage = TapeDrive
  Write Bootstrap = "/var/lib/bacula/LocalhostBackup.bsr"
}
```

Задача будет делать *полное* резервное копирование каждый день на ленточный накопитель.

- На каждую ленту следует установить *Метку*. Если текущая лента не имеет метки, Bacula известит вас по электронной почте. Для того, чтобы пометить ленту, используя Console, введите следующее в терминале:

```
bconsole
```

- В командной строке Bacula введите:

```
метка
```

- Вам предложат выбрать один из ресурсов *Storage*:

```
Автоматически выбран каталог: MyCatalog
Используется каталог "MyCatalog"
Автоматически выбран каталог: MyCatalog
Using Catalog "MyCatalog"
Возможные средства резервного сохранения:
1: Файл
2: Стриммер
Выберите средство резервного сохранения (1-2):2
```

- Введите новое название *Volume*:

```
Введите имя нового тома: Воскресенье
Defined Pools:
     1: Default
     2: Scratch
```

Замените *Sunday* желаемой меткой.

- Теперь выберите *Pool*:

```
Select the Pool (1-2): 1
Connecting to Storage daemon TapeDrive at backupserver:9103 ...
Sending label command for Volume "Sunday" Slot 0 ...
```

Поздравляем, вы настроили *Bacula* для резервного копирования localhost на стриммер.

3.5. Ресурсы

- Подробнее о опциях настройки *Bacula* смотрите *Руководство пользователя Bacula*[10]
- На *домашней странице Bacula*[11] находятся последние новости и разработки о проекте Bacula.
- Кроме того, смотрите на этой странице *Bacula Ubuntu Wiki*[12].

[10] http://www.bacula.org/en/rel-manual/index.html
[11] http://www.bacula.org/
[12] https://help.ubuntu.com/community/Bacula

Глава 20. Виртуализация

Виртуализация подходит для множества разных сред и ситуаций. Если вы разработчик, виртуализация может дать вам изолированную среду, где можно спокойно вести практически любые разработки, не боясь разрушить вашу основную рабочую среду. Если вы системный администратор, вы можете использовать виртуализацию для более лёгкого разделения служб и перемещать их повсюду, когда потребуется.

По умолчанию в Ubuntu поддерживается технология виртуализации KVM. KVM требует наличия процессора с поддержкой технологий аппаратной виртуализации от Intel или AMD. Xen также поддерживается в Ubuntu. Xen может использовать преимущества аппаратной виртуализации, если она доступна, но также может использоваться на оборудовании без аппаратной виртуализации. Qemu — ещё одно популярное решение для оборудования без аппаратной виртуализации.

1. Виртуальная библиотека

libvirt — библиотека, используемая как интерфейс к разным технологиям виртуализации. Прежде чем начать использовать libvirt, стоит узнать, поддерживает ли ваше оборудование расширения виртуализации для KVM. Введите следующую команду в консоли:

```
kvm-ok
```

Будет выведено сообщение о том, *поддерживает* или *не поддерживает* ваш процессор аппаратную виртуализацию.

На многих компьютерах, процессоры которых поддерживают аппаратную виртуализацию, для её активации необходимо включить соответствующую опцию в BIOS.

1.1. Виртуальная сеть

Есть несколько способов дать виртуальной машине доступ к внешней сети. По умолчанию конфигурация виртуальной сети включает *bridging* и *iptables* -правила работы с *usermode* сети, которая использует протокол SLIRP. Трафик NAT-ифицируется через интерфейс хоста во внешнюю сеть.

To enable external hosts to directly access services on virtual machines a different type of *bridge* than the default needs to be configured. This allows the virtual interfaces to connect to the outside network through the physical interface, making them appear as normal hosts to the rest of the network.

1.2. Установка

Чтобы установить требующиеся пакеты, введите в терминале:

```
sudo apt-get install qemu-kvm libvirt-bin
```

После установки libvirt-bin потребуется добавить пользователя, управляющего виртуальными машинами, в группу *libvirtd*. Это предоставит пользователю доступ к расширенным сетевым настройкам.

В консоли введите:

```
sudo adduser $USER libvirtd
```

Если выбранный пользователь - текущий, потребуется выйти из системы и войти снова, чтобы новое членство в группе возымело эффект.

Теперь вы готовы к установке *гостевой* операционной системы. Установка на виртуальную машину производится так же, как установка операционной системы на реальном аппаратном обеспечении. Вам потребуется либо способ автоматизации установки, либо будут нужны клавиатура и монитор, подключённые к физической машине.

В случае виртуальной машины графический пользовательский интерфейс (GUI) аналогичным образом использует физические клавиатуру и мышь. Вместо установки GUI для подключения к консоли виртуальной машины с помощью VNC может использоваться приложение virt-viewer. Смотрите *Раздел 1.6, «Средство просмотра виртуальных машин» [366]* для большей информации.

There are several ways to automate the Ubuntu installation process, for example using preseeds, kickstart, etc. Refer to the *Ubuntu Installation Guide*[1] for details.

Yet another way to install an Ubuntu virtual machine is to use uvtool. This application, available as of 14.04 allows you to set up specific VM options, execute custom post-install scripts, etc. For details see *Раздел 2, «Облачные образы и uvtool» [368]*

Libvirt также может быть настроен для работы с Xen. Подробные сведения смотрите на странице сообщества Xen в Ubuntu, доступной по приведённой ниже ссылке.

1.3. virt-install

virt-install является частью пакета virtinst. Для его установки введите в приглашении терминала:

```
sudo apt-get install virtinst
```

Существует несколько настроек, доступных при использовании virt-install. Например:

```
sudo virt-install -n web_devel -r 256 \
--disk path=/var/lib/libvirt/images/web_devel.img,bus=virtio,size=4 -c \
ubuntu-14.04-server-i386.iso --network network=default,model=virtio \
--graphics vnc,listen=0.0.0.0 --noautoconsole -v
```

- *-n web_devel:* имя новой виртуальной машины в этом примере будет *web_devel*
- *-r 256:* указывает объём памяти, выделяемый виртуальной машине, в мегабайтах.

[1] https://help.ubuntu.com/14.04/installation-guide/

- *--disk path=/var/lib/libvirt/images/web_devel.img,size=4:* указывает путь к виртуальному диску, который может быть файлом, разделом или логическим томом. В этом примере файл называется `web_devel.img`, расположен в каталоге /var/lib/libvirt/images/, имеет размер 4 гигабайта и использует *virtio* в качестве шины передачи данных.
- *-c ubuntu-14.04-server-i386.iso:* file to be used as a virtual CDROM. The file can be either an ISO file or the path to the host's CDROM device.
- *--network* содержит сведения, относящиеся к виртуальному сетевому интерфейсу. Здесь используется сеть *default*, и модель интерфейса настроена для *virtio*.
- *--graphics vnc,listen=0.0.0.0:* экспорт виртуальной гостевой консоли с помощью VNC и всех хост-интерфейсов. Обычно серверы не имеют графического интерфейса, поэтому другой графический интерфейс компьютера в локальной сети (LAN) можно подключить с помощью VNC, чтобы завершить установку.
- *--noautoconsole:* автоматически не подключит к консоли виртуальной машины.
- *-v:* создаёт полностью виртуализированный гостевой аккаунт.

After launching virt-install you can connect to the virtual machine's console either locally using a GUI (if your server has a GUI), or via a remote VNC client from a GUI based computer.

1.4. virt-clone

Приложение virt-clone может использоваться для копирования одной виртуальной машины в другую. Например:

```
sudo virt-clone -o web_devel -n database_devel -f /path/to/database_devel.img \ --connect=qemu:///s
```

- *-o:* оригинал виртуальной машины.
- *-n:* имя новой виртуальной машины.
- *-f:* путь к файлу, логическому диску или разделу для использования виртуальной машиной.
- *--connect:* указывает гипервизор для подключения.

Также используйте опции *-d* или *--debug* для помощи в решении проблем с virt-clone.

Замените *web_devel* и *database_devel* на подходящие имена виртуальных машин.

1.5. Управление виртуальными машинами

1.5.1. virsh

Существует несколько утилит, предназначенных для управления виртуальными машинами и libvirt. Утилита virsh может использоваться из командной строки. Некоторые примеры:

- Список запущенных виртуальных машин:

```
virsh -c qemu:///system list
```

- Для запуска виртуальной машины:

```
virsh -c qemu:///system start web_devel
```

- Подобным образом, для запуска виртуальной машины при загрузке:

```
virsh -c qemu:///system autostart web_devel
```

- Перезагрузка виртуальной машины:

```
virsh -c qemu:///system reboot web_devel
```

- *Состояние* виртуальных машин может быть сохранено в файл с возможностью дальнейшего восстановления. Следующая команда сохранит состояние виртуальной машины в файл, названный в соответствии с датой:

```
virsh -c qemu:///system save web_devel web_devel-022708.state
```

  После сохранения виртуальная машина не будет больше запущена.

- Сохранённая виртуальная машина может быть восстановлена с использованием:

```
virsh -c qemu:///system restore web_devel-022708.state
```

- Чтобы закрыть виртуальную машину, выполните:

```
virsh -c qemu:///system shutdown web_devel
```

- Устройство CD-ROM может быть примонтировано к вируальной машине следующей командой:

```
virsh -c qemu:///system attach-disk web_devel /dev/cdrom /media/cdrom
```

В предыдущих примерах замените *web_devel* требуемым именем виртуальной машины, а `web_devel-022708.state` понятным именем файла.

1.5.2. Менеджер виртуальных машин

Пакет virt-manager содержит графическую утилиту для управления локальными и удаленными виртуальными машинами. Для установки virt-manager в консоли введите:

```
sudo apt-get install virt-manager
```

Так как virt-manager требует среду пользовательского графического интерфейса (GUI), рекомендуется устанавливать его на рабочую станцию или тестовую машину, вместо готового сервера. Чтобы подключиться к локальному сервису libvirt введите:

```
virt-manager -c qemu:///system
```

Можно подключиться к сервису libvirt, запущенному на другом хосте, введя в терминале:

```
virt-manager -c qemu+ssh://virtnode1.mydomain.com/system
```

В предыдущем примере предполагается, что связь SSH между управляющей системой и virtnode1.mydomain.com уже была настроена и использует ключи SSH для аутентификации. *Ключи* SSH необходимы потому, что libvirt посылает запрос пароля другому процессу. Детально о настройке SSH смотрите *Раздел 1, «Сервер OpenSSH» [93]*

1.6. Средство просмотра виртуальных машин

Приложение virt-viewer позволяет вам подключаться к консоли виртуальной машины. Для взаимодействия с виртуальной машиной virt-viewer требуется графический пользовательский интерфейс.

Чтобы установить virt-viewer, введите в консоли:

```
sudo apt-get install virt-viewer
```

Когда виртуальная машина установлена и запущена, вы можете подключаться к консоли виртуальной машины, используя:

```
virt-viewer -c qemu:///system web_devel
```

Аналогично virt-manager, virt-viewer может подключаться к удалённому хосту, используя *SSH* с ключами аутентификации, как:

```
virt-viewer -c qemu+ssh://virtnode1.mydomain.com/system web_devel
```

Замените *web_devel* именем требуемой виртуальной машины.

If configured to use a *bridged* network interface you can also setup SSH access to the virtual machine.

1.7. Ресурсы

- Дополнительную информацию смотрите на домашней странице *KVM*[2].
- Для детальной информации по libvirt смотрите *домашнюю страницу libvirt*[3]
- На сайте *Менеджер виртуальной машины*[4] есть больше информации по разработке virt-manager.
- Также, заходите на IRC канал *#ubuntu-virt* на *freenode*[5] чтобы обсудить технологии виртуализации в Ubuntu.
- Ещё один хороший ресурс — это страница: *Ubuntu Wiki KVM*[6].
- Для получения информации о Xen, в том числе с помощью Xen с libvirt, пожалуйста, посмотрите страницу *Ubuntu Wiki Xen*[7].

[2] http://www.linux-kvm.org/
[3] http://libvirt.org/
[4] http://virt-manager.et.redhat.com/
[5] http://freenode.net/
[6] https://help.ubuntu.com/community/KVM
[7] https://help.ubuntu.com/community/Xen

2. Облачные образы и uvtool

2.1. Введение

With Ubuntu being one of the most used operating systems on most of the cloud platforms, the availability of stable and secure cloud images has become very important. As of 12.04 the utilization of cloud images outside of a cloud infrastructure has been improved. It is now possible to use those images to create a virtual machine without the need of a complete installation.

2.2. Создание виртуальных машин с помощью uvtool

Starting with 14.04 LTS, a tool called uvtool greatly facilitates the task of generating virtual machines (VM) using the cloud images. uvtool provides a simple mechanism to to synchronize cloud-images locally and use them to create new VMs in minutes.

2.2.1. Пакеты Uvtool

Для использования uvtool необходимы следующие пакеты и их зависимости:

- uvtool
- uvtool-libvirt

Установить uvtool можно так же, как и любое другое приложение, с помощью apt-get:

```
$ apt-get -y install uvtool
```

При этом будут установлены основные команды uvtool:

- uvt-simplestreams-libvirt
- uvt-kvm

2.2.2. Получение облачного образа Ubuntu с помощью uvt-simplestreams-libvirt

This is one of the major simplifications that uvtool brings. It is aware of where to find the cloud images so only one command is required to get a new cloud image. For instance, if you want to synchronize all cloud images for the amd64 architecture, the uvtool command would be:

```
$ uvt-simplestreams-libvirt sync arch=amd64
```

After an amount of time required to download all the images from the internet, you will have a complete set of cloud images stored locally. To see what has been downloaded use the following command:

```
$ uvt-simplestreams-libvirt query
release=oneiric arch=amd64 label=release (20130509)
release=precise arch=amd64 label=release (20140227)
release=quantal arch=amd64 label=release (20140302)
release=saucy arch=amd64 label=release (20140226)
release=trusty arch=amd64 label=beta1 (20140226.1)
```

In the case where you want to synchronize only one specific cloud-image, you need to use the release= and arch= filters to identify which image needs to be synchronized.

```
$ uvt-simplestreams-libvirt sync release=precise arch=amd64
```

2.2.3. Создание виртуальной машины с помощью uvt-kvm

In order to be able to connect to the virtual machine once it has been created, it is necessary to have a valid SSH key available for the ubuntu user. If your environment does not have a ssh key, you can easily create one using the following command:

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ubuntu/.ssh/id_rsa.
Your public key has been saved in /home/ubuntu/.ssh/id_rsa.pub.
The key fingerprint is:
4d:ba:5d:57:c9:49:ef:b5:ab:71:14:56:6e:2b:ad:9b ubuntu@TrustyS
The key's randomart image is:
+--[ RSA 2048]----+
|                ..|
|               o.=|
|           .    **|
|         +    o+=|
|        S . ...=.|
|         o . .+ .|
|        . .  o o |
|              *  |
|             E   |
+-----------------+
```

Создать новую виртуальную машину с помощью uvtool очень просто. В простейшем случае понадобится лишь выполнить:

```
$ uvt-kvm create firsttest
```

This will create a VM named **firsttest** using the current LTS cloud image available locally. If you want to specify a release to be used to create the VM, you need to use the **release=** filter

```
$ uvt-kvm create secondtest release=trusty
```

The uvt-kvm wait {name} can be used to wait until the creation of the VM has completed

```
$ uvt-kvm wait secondttest --insecure
Warning: secure wait for boot-finished not yet implemented; use --insecure.
```

2.2.4. Connect to the running VM

Когда создание виртуальной машины завершится, вы сможете подключиться к ней с помощью ssh:

```
$ uvt-kvm ssh secondtest --insecure
```

For the time being, the **--insecure** is required so you should be using this mechanism to connect to your VM only if you completely trust your network infrastructure

You can also connect to your VM using a regular ssh session using the IP address of the VM. The address can be queried using the following command:

```
$ uvt-kvm ip secondtest
192.168.123.242
$ ssh -i ~/.ssh/id_rsa ubuntu@192.168.123.242
The authenticity of host '192.168.123.242 (192.168.123.242)' can't be established.
ECDSA key fingerprint is 3a:12:08:37:79:24:2f:58:aa:62:d3:9d:c0:99:66:8a.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.123.242' (ECDSA) to the list of known hosts.
Welcome to Ubuntu Trusty Tahr (development branch) (GNU/Linux 3.13.0-12-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

 System information disabled due to load higher than 1.0

  Get cloud support with Ubuntu Advantage Cloud Guest:
    http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.


Last login: Fri Mar 21 13:25:56 2014 from 192.168.123.1
```

2.2.5. Получение списка работающих виртуальных машин

Получить список виртуальных машин, запущенных в вашей системе, можно этой командой:

```
$ uvt-kvm list
```

```
secondtest
```

2.2.6. Destroy your VM

Once you are done with your VM, you can proceed to destroy it with:

```
$ uvt-kvm destroy secondtest
```

2.2.7. Другие параметры uvt-kvm

Следующие опции можно использовать для изменения некоторых характеристик создаваемой вами виртуальной машины

- --memory : объём RAM в мегабайтах. По умолчанию: 512
- --disk : объём диска в гигабайтах. По умолчанию: 8
- --cpu : число ядер процессора. По умолчанию: 1

Some other parameters will have an impact on the cloud-init configuration

- --password password : Allow login to the VM using the ubuntu account and this provided password
- --run-script-one script_file : Run script_file as root on the VM the first time it is booted, but never again.
- --packages package_list : Install the comma-separated packages specified in package_list on first boot.

Полное описание всех доступных модификаторов доступно в man-странице uvt-kvm

2.3. Ресурсы

Если вам интересно узнать больше, или у вас есть вопросы или предложения, свяжитесь с командой Ubuntu Server по адресу:

- IRC: #ubuntu-server on freenode
- Список рассылки: *ubuntu-server at lists.ubuntu.com*[8]

[8] https://lists.ubuntu.com/mailman/listinfo/ubuntu-server

3. Облако Ubuntu

Облачные вычисления (cloud computing) — это модель вычислений, которая позволяет использовать распределять по запросу ресурсы из большого общего объёма доступных ресурсов. Эти ресурсы, такие как хранилище, вычислительная мощность, сеть и программное обеспечение, являются абстрактными и могут предоставляться как сервис через Интернет в любом месте и в любое время. Оплата этих сервисов производится за использованное время, аналогично тому, как это делается для таких общедоступных услуг, как электроснабжение, водоснабжение и телефон. Облачная инфраструктура Ubuntu использует открытое программное обеспечение OpenStack для создания масштабируемых решений в области облачных вычислений как для открытых, так и для частных облаков.

3.1. Installation and Configuration

Due to the current high rate of development of this complex technology we refer the reader to *upstream documentation*[9] for all matters concerning installation and configuration.

3.2. Поддержка и устранение неисправностей

Поддержка сообщества

- *Список рассылки OpenStack*[10]
- *Поиск в Wiki OpenStack*[11]
- *Сообщения об ошибках на Launchpad*[12]
- Присоединитесь к каналу IRC #openstack на freenode.

3.3. Ресурсы

- *Облачные вычисления — сервисные модели.*[13]
- *Вычисления OpenStack*[14]
- *Сервис образов OpenStack*[15]
- *Руководство администратора OpenStack Object Storage*[16]
- *Установка OpenStack Object Storage на Ubuntu*[17]

[9] http://docs.openstack.org/havana/install-guide/install/apt/content/
[10] https://launchpad.net/~openstack
[11] http://wiki.openstack.org
[12] https://bugs.launchpad.net/nova
[13] http://en.wikipedia.org/wiki/Cloud_computing#Service_Models
[14] http://www.openstack.org/software/openstack-compute/
[15] http://docs.openstack.org/diablo/openstack-compute/starter/content/GlanceMS-d2s21.html
[16] http://docs.openstack.org/trunk/openstack-object-storage/admin/content/index.html
[17] http://docs.openstack.org/trunk/openstack-object-storage/admin/content/installing-openstack-object-storage-on-ubuntu.html

- *http://cloudglossary.com/*

4. LXC

Containers are a lightweight virtualization technology. They are more akin to an enhanced chroot than to full virtualization like Qemu or VMware, both because they do not emulate hardware and because containers share the same operating system as the host. Therefore containers are better compared to Solaris zones or BSD jails. Linux-vserver and OpenVZ are two pre-existing, independently developed implementations of containers-like functionality for Linux. In fact, containers came about as a result of the work to upstream the vserver and OpenVZ functionality.

Существует две реализации пользовательского пространства контейнеров, каждая из которых использует те же самые возможности ядра. Libvirt позволяет использовать контейнеры через драйвер LXC, подсоединяясь к 'lxc:///'. Это очень удобно, поскольку поддерживается то же использование, что и для других драйверов. Другая реализация, называемая просто 'LXC', несовместима с libvirt, но более гибкая с использованием дополнительных утилит пользовательского пространства. Есть возможность переключаться с одной на другую, хотя существуют особенности, которые могут привести в замешательство.

In this document we will mainly describe the lxc package. Use of libvirt-lxc is not generally recommended due to a lack of Apparmor protection for libvirt-lxc containers.

В этом документе имя контейнера будет указано, как CN, C1, или C2.

4.1. Установка

Пакет lxc может быть установлен так

```
sudo apt-get install lxc
```

This will pull in the required and recommended dependencies, as well as set up a network bridge for containers to use. If you wish to use unprivileged containers, you will need to ensure that users have sufficient allocated subuids and subgids, and will likely want to allow users to connect containers to a bridge (see *Раздел 4.2.3, «Basic unprivileged usage» [376]*).

4.2. Основы использования

LXC can be used in two distinct ways - privileged, by running the lxc commands as the root user; or unprivileged, by running the lxc commands as a non-root user. (The starting of unprivileged containers by the root user is possible, but

not described here.) Unprivileged containers are more limited, for instance being unable to create device nodes or mount block-backed filesystems. However they are less dangerous to the host, as the root userid in the container is mapped to a non-root userid on the host.

4.2.1. Basic privileged usage

To create a privileged container, you can simply to

```
sudo lxc-create --template download --name u1
```

or, abbreviated

```
sudo lxc-create -t download -n u1
```

This will interactively ask for a container root filesystem type to download - in particular the distribution, release, and architecture. To create the container non-interactively, you can specify these values on the command line:

```
sudo lxc-create -t download -n u1 -- --dist ubuntu --release trusty --arch amd64
```

or

```
sudo lxc-create -t download -n u1 -- -d ubuntu -r trusty -a amd64
```

You can now use **lxc-ls** to list containers, **lxc-info** to obtain detailed container information, **lxc-start** to start and **lxc-stop** to stop the container. **lxc-attach** and **lxc-console** allow you to enter a container, if ssh is not an option. **lxc-destroy** removes the container, including its rootfs. See the manual pages for more information on each command. An example session might look like:

```
sudo lxc-ls --fancy
sudo lxc-start --name u1 --daemon
sudo lxc-info --name u1
sudo lxc-stop --name u1
sudo lxc-destroy --name u1
```

4.2.2. User namespaces

Unprivileged containers allow users to create and administer containers without having any root privilege. The feature underpinning this is called user

namespaces. User namespaces are hierarchical, with privileged tasks in a parent namespace being able to map its ids into child namespaces. By default every task on the host runs in the initial user namespace, where the full range of ids is mapped onto the full range. This can be seen by looking at /proc/self/uid_map and /proc/self/gid_map, which both will show "0 0 4294967295" when read from the initial user namespace. As of Ubuntu 14.04, when new users are created they are by default offered a range of userids. The list of assigned ids can be seen in the files `/etc/subuid` and `/etc/subgid` See their respective manpages for more information. Subuids and subgids are by convention started at id 100000 to avoid conflicting with system users.

If a user was created on an earlier release, it can be granted a range of ids using **usermod**, as follows:

```
sudo usermod -v 100000-200000 -w 100000-200000 user1
```

The programs **newuidmap** and **newgidmap** are setuid-root programs in the `uidmap` package, which are used internally by lxc to map subuids and subgids from the host into the unprivileged container. They ensure that the user only maps ids which are authorized by the host configuration.

4.2.3. Basic unprivileged usage

To create unprivileged containers, a few first steps are needed. You will need to create a default container configuration file, specifying your desired id mappings and network setup, as well as configure the host to allow the unprivileged user to hook into the host network. The example below assumes that your mapped user and group id ranges are 100000-165536.

```
mkdir -p ~/.config/lxc
echo "lxc.id_map = u 0 100000 65536" > ~/.config/lxc/default.conf
echo "lxc.id_map = g 0 100000 65536" >> ~/.config/lxc/default.conf
echo "lxc.network.type = veth" >> ~/.config/lxc/default.conf
echo "lxc.network.link = lxcbr0" >> ~/.config/lxc/default.conf
echo "$USER veth lxcbr0 2" | sudo tee -a /etc/lxc/lxc-usernet
```

After this, you can create unprivileged containers the same way as privileged ones, simply without using sudo.

```
lxc-create -t download -n u1 -- -d ubuntu -r trusty -a amd64 lxc-start -n u1 -d lxc-attach -n u1 lx
```

4.2.4. Nesting

In order to run containers inside containers - referred to as nested containers - two lines must be present in the parent container configuration file:

```
lxc.mount.auto = cgroup
lxc.aa_profile = lxc-container-default-with-nesting
```

The first will cause the cgroup manager socket to be bound into the container, so that lxc inside the container is able to administer cgroups for its nested containers. The second causes the container to run in a looser Apparmor policy which allows the container to do the mounting required for starting containers. Note that this policy, when used with a privileged container, is much less safe than the regular policy or an unprivileged container. See *Раздел 4.9, «Apparmor» [381]* for more information.

4.3. Global configuration

The following configuration files are consulted by LXC. For privileged use, they are found under `/etc/lxc`, while for unprivileged use they are under `~/.config/lxc`.

- `lxc.conf` may optionally specify alternate values for several lxc settings, including the lxcpath, the default configuration, cgroups to use, a cgroup creation pattern, and storage backend settings for lvm and zfs.
- `default.conf` specifies configuration which every newly created container should contain. This usually contains at least a network section, and, for unprivileged users, an id mapping section
- `lxc-usernet.conf` specifies how unprivileged users may connect their containers to the host-owned network.

`lxc.conf` and `default.conf` are exist both under `/etc/lxc` and `$HOME/.config/lxc`, while `lxc-usernet.conf` is only host-wide.

By default, containers are located under /var/lib/lxc for the root user, and $HOME/.local/share/lxc otherwise. The location can be specified for all lxc commands using the "-P|--lxcpath" argument.

4.4. Работа в сети

By default LXC creates a private network namespace for each container, which includes a layer 2 networking stack. Containers usually connect to the outside world by either having a physical NIC or a veth tunnel endpoint passed into the container. LXC creates a NATed bridge, lxcbr0, at host startup. Containers created using the default configuration will have one veth NIC with the remote end plugged into the lxcbr0 bridge. A NIC can only exist in one namespace at a time, so a physical NIC passed into the container is not usable on the host.

It is possible to create a container without a private network namespace. In this case, the container will have access to the host networking like any other application. Note that this is particularly dangerous if the container is running a distribution with upstart, like Ubuntu, since programs which talk to init, like **shutdown**, will talk over the abstract Unix domain socket to the host's upstart, and shut down the host.

To give containers on lxcbr0 a persistent ip address based on domain name, you can write entries to `/etc/lxc/dnsmasq.conf` like:

```
dhcp-host=lxcmail,10.0.3.100
dhcp-host=ttrss,10.0.3.101
```

If it is desirable for the container to be publicly accessible, there are a few ways to go about it. One is to use **iptables** to forward host ports to the container, for instance

```
 iptables -t nat -A PREROUTING -p tcp -i eth0 --dport 587 -j DNAT \
  --to-destination 10.0.3.100:587
```

Another is to bridge the host's network interfaces (see the bridging section in *the Ubuntu Server Guide's Network Configuration chapter*[18]. Then, specify the host's bridge in the container configuration file in place of lxcbr0, for instance

```
lxc.network.type = veth
lxc.network.link = br0
```

Finally, you can ask LXC to use macvlan for the container's NIC. Note that this has limitations and depending on configuration may not allow the container to talk to the host itself. Therefore the other two options are preferred and more commonly used.

There are several ways to determine the ip address for a container. First, you can use **lxc-ls --fancy** which will print the ip addresses for all running containers, or **lxc-info -i -H -n C1** which will print C1's ip address. If dnsmasq is installedon the host, you can also add an entry to `/etc/dnsmasq.conf` as follows

```
server=/lxc/10.0.3.1
```

after which dnsmasq will resolve C1.lxc locally, so that you can do:

```
ping C1
```

[18] https://help.ubuntu.com/serverguide/network-configuration.html

```
ssh C1
```

For more information, see the lxc.conf manpage as well as the example network configurations under `/usr/share/doc/lxc/examples/`.

4.5. LXC startup

LXC does not have a long-running daemon. However it does have three upstart jobs.

- `/etc/init/lxc-net.conf`: is an optional job which only runs if `/etc/default/lxc-net` specifies USE_LXC_BRIDGE (true by default). It sets up a NATed bridge for containers to use.
- `/etc/init/lxc.conf` loads the lxc apparmor profiles and optionally starts any autostart containers. The autostart containers will be ignored if LXC_AUTO (true by default) is set to true in `/etc/default/lxc`. See the lxc-autostart manual page for more information on autostarted containers.
- `/etc/init/lxc-instance.conf`: is used by `/etc/init/lxc.conf` to autostart a container.

4.6. Резервные хранилища

LXC supports several backing stores for container root filesystems. The default is a simple directory backing store, because it requires no prior host customization, so long as the underlying filesystem is large enough. It also requires no root privilege to create the backing store, so that it is seamless for unprivileged use. The rootfs for a privileged directory backed container is located (by default) under `/var/lib/lxc/C1/rootfs`, while the rootfs for an unprivileged container is under `~/.local/share/lxc/C1/rootfs`. If a custom lxcpath is specified in lxc.system.com, then the container rootfs will be under `$lxcpath/C1/rootfs`.

A snapshot clone C2 of a a directory backed container C1 becomes an overlayfs backed container, with a rootfs called `overlayfs:/var/lib/lxc/C1/rootfs:/var/lib/lxc/C2/delta0`. Other backing store types include loop, btrfs, LVM and zfs.

A btrfs backed container mostly looks like a directory backed container, with its root filesystem in the same location. However, the root filesystem comprises a subvolume, so that a snapshot clone is created using a subvolume snapshot.

The root filesystem for an LVM backed container can be any separate LV. The default VG name can be specified in lxc.conf. The filesystem type and size are configurable per-container using lxc-create.

The rootfs for a zfs backed container is a separate zfs filesystem, mounted under the traditional `/var/lib/lxc/C1/rootfs` location. The zfsroot can be specified at lxc-create, and a default can be specified in lxc.system.conf.

More information on creating containers with the various backing stores can be found in the lxc-create manual page.

4.7. Templates

Creating a container generally involves creating a root filesystem for the container. **lxc-create** delegates this work to *templates*, which are generally per-distribution. The lxc templates shipped with lxc can be found under `/usr/share/lxc/templates`, and include templates to create Ubuntu, Debian, Fedora, Oracle, centos, and gentoo containers among others.

Creating distribution images in most cases requires the ability to create device nodes, often requires tools which are not available in other distributions, and usually is quite time-consuming. Therefore lxc comes with a special *download* template, which downloads pre-built container images from a central lxc server. The most important use case is to allow simple creation of unprivileged containers by non-root users, who could not for instance easily run the **debootstrap** command.

When running **lxc-create**, all options which come after **--** are passed to the template. In the following command, **--***name*, **--***template* and **--***bdev* are passed to **lxc-create**, while **--***release* is passed to the template:

```
lxc-create --template ubuntu --name c1 --bdev loop -- --release trusty
```

You can obtain help for the options supported by any particular container by passing **--***help* and the template name to **lxc-create**. For instance, for help with the download template,

```
lxc-create --template download --help
```

4.8. Autostart

LXC supports marking containers to be started at system boot. Prior to Ubuntu 14.04, this was done using symbolic links under the directory `/etc/lxc/auto`. Starting with Ubuntu 14.04, it is done through the container configuration files. An entry

```
lxc.start.auto = 1
```

```
lxc.start.delay = 5
```

would mean that the container should be started at boot, and the system should wait 5 seconds before starting the next container. LXC also supports ordering and grouping of containers, as well as reboot and shutdown by autostart groups. See the manual pages for lxc-autostart and lxc.container.conf for more information.

4.9. Apparmor

LXC ships with a default Apparmor profile intended to protect the host from accidental misuses of privilege inside the container. For instance, the container will not be able to write to `/proc/sysrq-trigger` or to most `/sys` files.

Профиль `usr.bin.lxc-start` используется при запуске **lxc-start**. Этот профиль в основном предотвращает монтирование **lxc-start** новых файловых систем вне корневой файловой системы контейнера. Перед инициализацией **init** контейнера, **LXC** запрашивает переключение на профиль контейнера. По умолчанию используется профиль `lxc-container-default`определенный в `/etc/apparmor.d/lxc/lxc-default`. Этот профиль запрещает контейнеру доступ к многим опасным каталогам и монтирование большинства файловых систем.

Programs in a container cannot be further confined - for instance, MySQL runs under the container profile (protecting the host) but will not be able to enter the MySQL profile (to protect the container).

lxc-execute не просматривает профиль Apparmor, но контейнер, который он порождает, будет ограничен.

4.9.1. Customizing container policies

Если вы обнаружили, что **lxc-start** падает из-за попытки легитимного доступа, перекрытого политикой Apparmor, вы можете отключить профиль lxc-start следующим образом:

```
sudo apparmor_parser -R /etc/apparmor.d/usr.bin.lxc-start
sudo ln -s /etc/apparmor.d/usr.bin.lxc-start /etc/apparmor.d/disabled/
```

Это позволит запускать **lxc-start** без ограничений, но продолжит ограничивать собственно контейнер. Если вы хотите также снять ограничения с контейнера, в дополнение к блокировке использования профиля `usr.bin.lxc-start`, вам потребуется в файл настроек контейнера добавить:

```
lxc.aa_profile = unconfined
```

to the container's configuration file.

LXC ships with a few alternate policies for containers. If you wish to run containers inside containers (nesting), then you can use the lxc-container-default-with-nesting profile by adding the following line to the container configuration file

```
lxc.aa_profile = lxc-container-default-with-nesting
```

If you wish to use libvirt inside containers, then you will need to edit that policy (which is defined in `/etc/apparmor.d/lxc/lxc-default-with-nesting`) to uncomment the following line

```
mount fstype=cgroup -> /sys/fs/cgroup/**,
```

and re-load the policy.

Note that the nesting policy with privileged containers is far less safe than the default policy, as it allows containers to re-mount `/sys` and `/proc` in nonstandard locations, bypassing apparmor protections. Unprivileged containers do not have this drawback since the container root cannot write to root-owned `proc` and `sys` files.

Another profile shipped with lxc allows containers to mount block filesystem types like ext4. This can be useful in some cases like maas provisioning, but is deemed generally unsafe since the superblock handlers in the kernel have not been audited for safe handling of untrusted input.

If you need to run a container in a custom profile, you can create a new profile under `/etc/apparmor.d/lxc/`. Its name must start with `lxc-` in order for **lxc-start** to be allowed to transition to that profile. The `lxc-default` profile includes the re-usable abstractions file `/etc/apparmor.d/abstractions/lxc/container-base`. An easy way to start a new profile therefore is to do the same, then add extra permissions at the bottom of your policy.

После создания политики загрузите её, используя команду:

```
sudo apparmor_parser -r /etc/apparmor.d/lxc-containers
```

Профиль автоматически загрузится после перезагрузки системы, поскольку его содержимое учтено в `/etc/apparmor.d/lxc-containers`. Наконец, чтобы

заставить контейнер `CN` использовать новый профиль `lxc-CN-profile`, добавьте следующие строки в его файл настройки:

```
lxc.aa_profile = lxc-CN-profile
```

4.10. Группы управления

Control groups (cgroups) are a kernel feature providing hierarchical task grouping and per-cgroup resource accounting and limits. They are used in containers to limit block and character device access and to freeze (suspend) containers. They can be further used to limit memory use and block i/o, guarantee minimum cpu shares, and to lock containers to specific cpus.

By default, a privileged container CN will be assigned a cgroup called `/lxc/CN`. In the case of name conflicts (which can occur when using custom lxcpaths) a suffix "-n", where n is an integer starting at 0, will be appended to the cgroup name.

By default, a privileged container CN will be assigned a cgroup called `CN` under the cgroup of the task which started the container, for instance `/usr/1000.user/1.session/CN`. The container root will be given group ownership of the directory (but not all files) so that it is allowed to create new child cgroups.

As of Ubuntu 14.04, LXC uses the cgroup manager (cgmanager) to administer cgroups. The cgroup manager receives D-Bus requests over the Unix socket `/sys/fs/cgroup/cgmanager/sock`. To fascilitate safe nested containers, the line

```
lxc.mount.auto = cgroup
```

can be added to the container configuration causing the `/sys/fs/cgroup/cgmanager` directory to be bind-mounted into the container. The container in turn should start the cgroup management proxy (done by default if the cgmanager package is installed in the container) which will move the `/sys/fs/cgroup/cgmanager` directory to `/sys/fs/cgroup/cgmanager.lower`, then start listening for requests to proxy on its own socket `/sys/fs/cgroup/cgmanager/sock`. The host cgmanager will ensure that nested containers cannot escape their assigned cgroups or make requests for which they are not authorized.

4.11. Клонирование

For rapid provisioning, you may wish to customize a canonical container according to your needs and then make multiple copies of it. This can be done with the **lxc-clone** program.

Clones are either snapshots or copies of another container. A copy is a new container copied from the original, and takes as much space on the host as the original. A snapshot exploits the underlying backing store's snapshotting ability to make a copy-on-write container referencing the first. Snapshots can be created from btrfs, LVM, zfs, and directory backed containers. Each backing store has its own peculiarities - for instance, LVM containers which are not thinpool-provisioned cannot support snapshots of snapshots; zfs containers with snapshots cannot be removed until all snapshots are released; LVM containers must be more carefully planned as the underlying filesystem may not support growing; btrfs does not suffer any of these shortcomings, but suffers from reduced fsync performance causing dpkg and apt-get to be slower.

Snapshots of directory-packed containers are created using the overlay filesystem. For instance, a privileged directory-backed container C1 will have its root filesystem under `/var/lib/lxc/C1/rootfs`. A snapshot clone of C1 called C2 will be started with C1's rootfs mounted readonly under `/var/lib/lxc/C2/delta0`. Importantly, in this case C1 should not be allowed to run or be removed while C2 is running. It is advised instead to consider C1 a *canonical* base container, and to only use its snapshots.

Given an existing container called C1, a copy can be created using:

```
sudo lxc-clone -o C1 -n C2
```

A snapshot can be created using

```
sudo lxc-clone -s -o C1 -n C2
```

See the lxc-clone manpage for more information.

4.11.1. Snapshots

To more easily support the use of snapshot clones for iterative container development, LXC supports *snapshots*. When working on a container C1, before making a potentially dangerous or hard-to-revert change, you can create a snapshot

```
sudo lxc-snapshot -n C1
```

which is a snapshot-clone called 'snap0' under /var/lib/lxcsnaps or $HOME/.local/share/lxcsnaps. The next snapshot will be called 'snap1', etc. Existing snapshots can be listed using **lxc-snapshot -L -n C1**, and a snapshot can be restored -

erasing the current C1 container - using **lxc-snapshot -r snap1 -n C1**. After the restore command, the snap1 snapshot continues to exist, and the previous C1 is erased and replaced with the snap1 snapshot.

Snapshots are supported for btrfs, lvm, zfs, and overlayfs containers. If lxc-snapshot is called on a directory-backed container, an error will be logged and the snapshot will be created as a copy-clone. The reason for this is that if the user creates an overlayfs snapshot of a directory-backed container and then makes changes to the directory-backed container, then the original container changes will be partially reflected in the snapshot. If snapshots of a directory backed container C1 are desired, then an overlayfs clone of C1 should be created, C1 should not be touched again, and the overlayfs clone can be edited and snapshotted at will, as such

```
lxc-clone -s -o C1 -n C2
lxc-start -n C2 -d # make some changes
lxc-stop -n C2
lxc-snapshot -n C2
lxc-start -n C2 # etc
```

4.11.2. Ephemeral Containers

While snapshots are useful for longer-term incremental development of images, ephemeral containers utilize snapshots for quick, single-use throwaway containers. Given a base container C1, you can start an ephemeral container using

```
lxc-start-ephemeral -o C1
```

The container begins as a snapshot of C1. Instructions for logging into the container will be printed to the console. After shutdown, the ephemeral container will be destroyed. See the lxc-start-ephemeral manual page for more options.

4.12. Обработчики управления жизненным циклом

Начиная с Ubuntu 12.10, можно определять обработчики, которые будут выполняться в различных состояниях жизненного цикла контейнера:

- Предстартовые обработчики запускаются в пространстве имен хоста до запуска терминалов, консоли или операций монтирования. Любые операции монтирования, сделанные этим обработчиком, должны быть очищены обработчиком пост-остановки.

- Обработчики пред-монтирования запускаются в пространстве имен контейнера, но до того, как будет примонтирована корневая файловая система. Операции монтирования, сделанные этим обработчиком, будут автоматически очищены при остановке контейнера.
- Обработчики монтирования, запущенные после того, как файловые системы контейнера были примонтированы, но до вызова контейнером команды **pivot_root** для смены его корневой файловой системы.
- Обработчики запуска выполняются незамедлительно после инициализации контейнера. Поскольку они выполняются после перемещения в файловую систему контейнера, команды для выполнения должны быть скопированы в файловую систему контейнера.
- Обработчики пост-остановки выполняются после завершения работы контейнера.

Если обработчик возвращает ошибку, выполнение контейнера будет прервано. Любой *пост-стоп* обработчик все равно будет выполнен. Любой вывод, генерируемый скриптом будет записан при отладке приоритета.

Please see the lxc.container.conf manual page for the configuration file format with which to specify hooks. Some sample hooks are shipped with the lxc package to serve as an example of how to write and use such hooks.

4.13. Консоли

Контейнеры имеют настраиваемое количество консолей. Одна всегда существует в `/dev/console` контейнера. Она отображается в терминале, из которого вы запускаете **lxc-start**, если не указана опция *-d*. Вывод в `/dev/console` может быть перенаправлен в файл с помощью опции *-c console-file* команды **lxc-start**. Количество дополнительных консолей определяется переменной **lxc.tty**, и обычно равно 4. Эти консоли отображаются в `/dev/ttyN` (для 1 <= N <= 4). Для входа в консоль 3 с хоста используйте

```
sudo lxc-console -n container -t 3
```

в противном случае, если *-t N* не указано, будет автоматически выбрана неиспользуемая консоль. Для входа из консоли используйте последовательность Ctrl-a q. Обратите внимание, что последовательность не сработает в консоли при использовании **lxc-start** без опции *-d*.

Каждая консоль контейнера фактически является Unix98 pty смонтированной в pty основной (не гостевой) системы через связанное монтирование гостевых `/dev/ttyN` и `/dev/console`. Следовательно, если гостевая система размонтирует их или с другой стороны попытается

получить доступ к символьному устройству **4:N**, она не будет обслужена getty на LXC консолях. (При настройках по умолчанию контейнер не сможет получить доступ к этому символьному устройству и getty, соответственно, завершится с ошибкой). Это может легко случиться, когда загрузочный сценарий вслепую монтирует новые устройства в `/dev`.

4.14. Устранение проблем

4.14.1. Ведение журнала

If something goes wrong when starting a container, the first step should be to get full logging from LXC:

```
sudo lxc-start -n C1 -l trace -o debug.out
```

This will cause lxc to log at the most verbose level, `trace`, and to output log information to a file called 'debug.out'. If the file `debug.out` already exists, the new log information will be appended.

4.14.2. Отслеживание статуса контейнеров

Две команды доступны для отслеживания изменения статуса контейнера. **lxc-monitor** отслеживает один или более контейнеров на любые изменения статусов. Она как правило получает имя контейнера с помощью опции *-n*, однако в этом случае имя контейнера может быть регулярным выражением posix, чтобы позволять отслеживать желаемые наборы контейнеров. **lxc-monitor** продолжает выполнение пока выводит статусы контейнеров. Вторая команда **lxc-wait** ожидает специфического изменения статуса контейнера и затем завершается. Например,

```
sudo lxc-monitor -n cont[0-5]*
```

будет выводить все изменения статусов контейнеров с именами, попадающими под приведенное регулярное выражение, в то время как

```
sudo lxc-wait -n cont1 -s 'STOPPED|FROZEN'
```

будет ожидать, пока контейнер cont1 не войдет в состояния STOPPED или FROZEN и затем завершится.

4.14.3. Attach

As of Ubuntu 14.04, it is possible to attach to a container's namespaces. The simplest case is to simply do

```
sudo lxc-attach -n C1
```

which will start a shell attached to C1's namespaces, or, effectively inside the container. The attach functionality is very flexible, allowing attaching to a subset of the container's namespaces and security context. See the manual page for more information.

4.14.4. Container init verbosity

If LXC completes the container startup, but the container init fails to complete (for instance, no login prompt is shown), it can be useful to request additional verbosity from the init process. For an upstart container, this might be:

```
sudo lxc-start -n C1 /sbin/init loglevel=debug
```

You can also start an entirely different program in place of init, for instance

```
sudo lxc-start -n C1 /bin/bash
sudo lxc-start -n C1 /bin/sleep 100
sudo lxc-start -n C1 /bin/cat /proc/1/status
```

4.15. LXC API

Most of the LXC functionality can now be accessed through an API exported by `liblxc` for which bindings are available in several languages, including Python, lua, ruby, and go.

Below is an example using the python bindings (which are available in the python3-lxc package) which creates and starts a container, then waits until it has been shut down:

```
# sudo python3
Python 3.2.3 (default, Aug 28 2012, 08:26:03)
[GCC 4.7.1 20120814 (prerelease)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import lxc
__main__:1: Warning: The python-lxc API isn't yet stable and may change at any p
oint in the future.
>>> c=lxc.Container("C1")
>>> c.create("ubuntu")
True
```

```
>>> c.start()
True
>>> c.wait("STOPPED")
True
```

4.16. Защита

Пространство имен сопоставляет идентификаторы с ресурсами. Чтобы не предоставлять доступ контейнерам к любым идентификаторам (id), указывающим на ресурсы, ресурсы должны быть защищены. Это является основой некоторой безопасности, предоставляемой пользователям контейнеров. Например, пространство имён IPC (взаимодействия между процессами) полностью изолировано. Однако другие пространства имён имеют различные уязвимости, которые позволяют получать неправильно предоставленные привилегии из одного контейнера в другой или в основную систему.

By default, LXC containers are started under a Apparmor policy to restrict some actions. The details of AppArmor integration with lxc are in section *Раздел 4.9, «Apparmor» [381]*. Unprivileged containers go further by mapping root in the container to an unprivileged host userid. This prevents access to `/proc` and `/sys` files representing host resources, as well as any other files owned by root on the host.

4.16.1. Используемые системные вызовы

Это функция основного контейнера, который поделится ядром с хостовой системой. Поэтому, если ядро содержит любые эксплойдные системные вызывы, то контейнер может использовать и их. Как только контейнер управляет ядром, то он может полностью контролировать любой ресурс своей хостовой системой.

Начиная с Ubuntu 12.10 (Quantal), контейнер также может быть ограничен фильтром seccomp. Seccomp — это новая функция ядра, фильтрующая системные вызовы, которые могут использоваться задачей и её потомками. Хотя в ближайшем будущем ожидается улучшенное и упрощённое управление политиками, текущая политика состоит из простого белого листа номеров системных вызовов. Файл политики начинается с номера версии (который должен быть равен 1) в первой строке и типа политики (который должен иметь значение 'whitelist') во второй строке. Далее следует список номеров, по одному в строке.

In general to run a full distribution container a large number of system calls will be needed. However for application containers it may be possible to reduce the number of available system calls to only a few. Even for system

containers running a full distribution security gains may be had, for instance by removing the 32-bit compatibility system calls in a 64-bit container. See the lxc.container.conf manual page for details of how to configure a container to use seccomp. By default, no seccomp policy is loaded.

4.17. Ресурсы

- Статья в DeveloperWorks *LXC: Linux container tools*[19] является введением в использование контейнеров.
- *Secure Containers Cookbook*[20] демонстрирует использование модулей безопасности с целью сделать контейнеры более безопасными.
- Manual pages referenced above can be found at:

  capabilities[22][21]
  lxc.conf[24].[23]
- The upstream LXC project is hosted at *linuxcontainers.org*[25].
- Проблемы безопасности приведены и обсуждаются на странице *the LXC Security wiki page*[26]
- Подробнее пространствах имён в Linux читайте в книге: S. Bhattiprolu, E. W. Biederman, S. E. Hallyn, and D. Lezcano. Virtual Servers and Check- point/ Restart in Mainstream Linux. SIGOPS Operating Systems Review, 42(5), 2008.

[19] https://www.ibm.com/developerworks/linux/library/l-lxc-containers/
[20] http://www.ibm.com/developerworks/linux/library/l-lxc-security/index.html
[22] http://manpages.ubuntu.com/manpages/en/man7/capabilities.7.html
[21] http://manpages.ubuntu.com/manpages/en/man7/capabilities.7.html
[24] http://manpages.ubuntu.com/manpages/en/man5/lxc.conf.5.html
[23] http://manpages.ubuntu.com/manpages/en/man5/lxc.conf.5.html
[25] http://linuxcontainers.org
[26] http://wiki.ubuntu.com/LxcSecurity

Глава 21. Группы управления

Control groups (cgroups) are a kernel mechanism for grouping, tracking, and limiting the resource usage of tasks. The kernel-provided administration interface is through a virtual filesystem. Higher level cgroup administration tools have been developed, including libcgroup and lmctfy. Additionally, there is guidance at freedesktop.org for how applications can best cooperate using the cgroup filesystem interface (see Resources).

As of Ubuntu 14.04, the cgroup manager (cgmanager) is available as another cgroup administion interface. It's goal is to respond to dbus requests from any user, allowing him to administer only those cgroups which have been delegated to him.

Раздел 1, «Обзор» [392] will describe cgroups in more detail. *Раздел 2, «Filesystem» [393]* will describe the long-standing cgroups filesystem interface. *Раздел 4, «Manager» [395]* will describe the cgroup manager.

1. Обзор

Cgroups are the generalized feature for grouping tasks. The actual resource tracking and limits are implemented by subsystems. A hierarchy is a set of subsystems mounted together. For instance, if the memory and devices subsystems are mounted together under /sys/fs/cgroups/set1, then any task which is in "/child1" will be subject to the corresponding limits of both subsystems.

Each set of mounted subsystems consittutes a 'hierarchy'. With exceptions, cgroups which are children of "/child1" will be subject to all limits placed on "/child1", and their resource usage will be accounted to "/child1".

The existing subsystems include:

- *cpusets*: fascilitate assigning a set of CPUS and memory nodes to cgroups. Tasks in a cpuset cgroup may only be scheduled on CPUS assigned to that cpuset.
- *blkio* : limits per-cgroup block io.
- *cpuacct* : provides per-cgroup cpu usage accounting.
- *devices* : controls the ability of tasks to create or use devices nodes using either a blacklist or whitelist.
- *freezer* : provides a way to 'freeze' and 'thaw' whole cgroups. Tasks in the cgroup will not be scheduled while they are frozen.
- *hugetlb* : fascilitates limiting hugetlb usage per cgroup.
- *memory* : allows memory, kernel memory, and swap usage to be tracked and limited.
- *net_cls* : provides an interface for tagging packets based on the sender cgroup. These tags can then be used by tc (traffic controller) to assign priorities.
- *net_prio* : allows setting network traffic priority on a per-cgroup basis.
- *cpu* : enables setting of scheduling preferences on per-cgroup basis.
- *perf_event* : enables per-cpu mode to monitor only threads in certain cgroups.

In addition, named cgroups can be created with no bound subsystems for the sake of process tracking. As an example, systemd does this to track services and user sessions.

2. Filesystem

A hierarchy is created by mounting an instance of the cgroup filesystem with each of the desired subsystems listed as a mount option. For instance,

```
mount -t cgroup -o devices,memory,freezer cgroup /cgroup1
```

would instantiate a hierarchy with the devices and memory cgroups comounted. A child cgroup "child1" can be created using 'mkdir'

```
mkdir /cgroup1/child1
```

and tasks can be moved into the new child cgroup by writing their process ids into the 'tasks' or 'cgroup.procs' file:

```
sleep 100
echo $! > /cgroup1/child1/cgroup.procs
```

Other administration is done through files in the cgroup directories. For instance, to freeze all tasks in child1,

```
echo FROZEN > /cgroup1/child1/freezer.state
```

A great deal of information about cgroups and its subsystems can be found under the cgroups documentation directory in the kernel source tree (see Resources).

3. Delegation

Cgroup files and directories can be owned by non-root users, enabling delegation of cgroup administration. In general, the kernel enforces the hierarchical constraints on limits, so that for instance if devices cgroup `/child1` cannot access a disk drive, then `/child1/child2` cannot give itself those rights.

As of Ubuntu 14.04, users are automatically placed in a set of cgroups which they own, safely allowing them to contrain their own jobs using child cgroups. This feature is relied upon, for instance, for unprivileged container creation in lxc.

4. Manager

The cgroup manager (cgmanager) provides a D-Bus service allowing programs and users to administer cgroups without needing direct knowledge of or access to the cgroup filesystem. For requests from tasks in the same namespaces as the manager, the manager can directly perform the needed security checks to ensure that requests are legitimate. For other requests - such as those from a task in a container - enhanced D-Bus requests must be made, where process-, user- and group-ids are passed as SCM_CREDENTIALS, so that the kernel maps the identifiers to their global host values.

To fascilitate the use of simple D-Bus calls from all users, a 'cgroup manager proxy' (cgproxy) is automatically started when in a container. The proxy accepts standard D-Bus requests from tasks in the same namespaces as itself, and converts them to SCM-enhanced D-Bus requests which it passes on to the cgmanager.

A simple example of creating a new cgroup in which to run a cpu-intensive compile would look like:

```
cmg create cpuset build1
cgm movepid cpuset build1 $$
cgm setvalue cpuset build1 cpuset.cpus 1
make
```

5. Ресурсы

- Manual pages referenced above can be found at:

cgm[1]
cgconfig.conf[2]
cgmanager[3]
cgproxy[4]

- The upstream cgmanager project is hosted at *linuxcontainers.org*[5].
- The upstream kernel documentation page on cgroups can be seen *here* [6].
- The freedesktop.org control group usage guidelines can be seen *here*[7].

[1] http://manpages.ubuntu.com/manpages/en/man8/cgm.1.html
[2] http://manpages.ubuntu.com/manpages/en/man5/cgconfig.conf.5.html
[3] http://manpages.ubuntu.com/manpages/en/man8/cgmanager.8.html
[4] http://manpages.ubuntu.com/manpages/en/man8/cgproxy.8.html
[5] http://cgmanager.linuxcontainers.org
[6] https://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/tree/Documentation/cgroups
[7] http://www.freedesktop.org/wiki/Software/systemd/PaxControlGroups/

Глава 22. Кластеризация

1. DRBD

Распределённое копируемое блочное устройство (Distributed Replicated Block Device — DRBD) создаёт зеркало между блочными устройствами нескольких систем. Копирование незаметно (прозрачно) для других приложений на этих системах. Любые блочные устройства: жёсткие диски, разделы, RAID устройства, логические тома и т.п. могут зеркалироваться.

Перед началом использования drbd установим необходимые пакеты. Введите в терминале:

```
sudo apt-get install drbd8-utils
```

Если вы используете *виртуальное ядро* как часть виртуальной машины, вам потребуется вручную скомпилировать модуль drbd. Возможно, более простым способом окажется установка пакета linux-server внутри виртуальной машины.

В этом разделе рассматривается установка drbd для репликации отдельных `/srv` разделов с файловой системой ext3 между двумя системами. Размер разделов не имеет особого значения, но оба раздела должны иметь одинаковый размер.

1.1. Конфигурация

Две системы в этом примере будут называться *drbd01* и *drbd02*. Им потребуется разрешение имён, настроенное через DNS или с помощью файла `/etc/hosts`. Подробности смотрите в разделе *Глава 8, Служба доменных имён (DNS) [157]*.

- Для настройки drbd на первой системе отредактируйте `/etc/drbd.conf`:

```
global { usage-count no; }
common { syncer { rate 100M; } }
resource r0 {
        protocol C;
        startup {
                wfc-timeout  15;
                degr-wfc-timeout 60;
        }
        net {
                cram-hmac-alg sha1;
                shared-secret "secret";
        }
        on drbd01 {
                device /dev/drbd0;
```

```
                disk /dev/sdb1;
                address 192.168.0.1:7788;
                meta-disk internal;
        }
        on drbd02 {
                device /dev/drbd0;
                disk /dev/sdb1;
                address 192.168.0.2:7788;
                meta-disk internal;
        }
}
```

Существует множество других опций в `/etc/drbd.conf`, но для данного примера прекрасно подходят их значения по умолчанию.

- Теперь скопируем `/etc/drbd.conf` на вторую систему:

```
scp /etc/drbd.conf drbd02:~
```

- И на *drbd02* переместим файл в `/etc`:

```
sudo mv drbd.conf /etc/
```

- Теперь с помощью утилиты drbdadm инициализируем хранилище метаданных. На каждом сервере выполним:

```
sudo drbdadm create-md r0
```

- Далее на обеих системах запустим сервис drbd:

```
sudo service drbd start
```

- На *drbd01* или той системе, которую вы хотите сделать основной, введите следующее:

```
sudo drbdadm -- --overwrite-data-of-peer primary all
```

- После выполнения вышеприведённой команды данные начнут реплицироваться на вторую систему. Чтобы наблюдать за процессом, на *drbd02* введите следующее:

```
watch -n1 cat /proc/drbd
```

Для остановки просмотра нажмите *Ctrl+c*.

- Наконец, установите файловую систему на `/dev/drbd0` и смонтируйте ее:

```
sudo mkfs.ext3 /dev/drbd0
sudo mount /dev/drbd0 /srv
```

1.2. Тестирование

Чтобы убедиться, что данные действительно синхронизируются между системами, скопируйте несколько файлов на *drbd01* (основной системе) в каталог `/srv`:

```
sudo cp -r /etc/default /srv
```

Далее, отсоедините `/srv`:

```
sudo umount /srv
```

Установите *первичному* серверу роль *вторичного*:

```
sudo drbdadm secondary r0
```

Теперь установите *вторичному* серверу роль *первичного* role:

```
sudo drbdadm primary r0
```

Наконец, монтируем раздел:

```
sudo mount /dev/drbd0 /srv
```

Используя *ls* вы сможете увидеть `/srv/default`, скопированный с бывшего *первичного* сервера *drbd01*.

1.3. Ссылки

- Для дополнительной информации по DRBD посетите *DRBD web site*[1].
- The *drbd.conf man page*[2] contains details on the options not covered in this guide.
- Also, see the *drbdadm man page*[3].
- Дополнительную информацию также содержит страница *DRBD Ubuntu Wiki*[4].

[1] http://www.drbd.org/
[2] http://manpages.ubuntu.com/manpages/trusty/en/man5/drbd.conf.5.html
[3] http://manpages.ubuntu.com/manpages/trusty/en/man8/drbdadm.8.html
[4] https://help.ubuntu.com/community/DRBD

Глава 23. VPN

OpenVPN — гибкое, надёжное и безопасное решение для создания виртуальной частной сети (VPN), которое доступно в репозиториях Ubuntu. Оно относится к семейству стеков SSL/TLS VPN (отличается от IPSec VPN). Эта глава посвящена установке и настройкеOpenVPN для создания VPN.

1. OpenVPN

Если вы хотите больше, чем просто pre-shared ключи OpenVPN облегчает установку и использование инфраструктуры открытых ключей (PKI), использующих SSL/TLS сертификаты для аутентификации и обмена ключами между VPN-сервером и клиентами. OpenVPN может использоваться в маршрутизирующим или мостовым VPN режиме и может быть настроен для использования либо UDP или TCP. Номер порта может быть сконфигурирован как угодно, но официальный порт 1194. Один порт используется для всех коммуникаций. VPN сервер доступны практически для чего угодно, включая все дистрибутивы Linux, OS X, Windows и WLAN-маршрутизаторах на основе OpenWRT.

1.1. Установка сервера

Для установки openvpn наберите в терминале:

```
sudo apt-get install openvpn easy-rsa
```

1.2. Настройка инфраструктуры открытых ключей

Первый шаг в построении конфигурации OpenVPN является создание инфраструктуры открытых ключей (Public Key Infrastructure). PKI состоит из:

- отдельного сертификата (также известного как открытый ключ) и закрытого ключа для сервера и каждого клиента, и
- центра сертификации (CA), сертификата и ключа, который используется для входа каждого сервера и клиентских сертификатов.

OpenVPN поддерживает двунаправленную аутентификацию на основе сертификатов, что означает, что клиент должен проверить подлинность сертификата сервера и сервер должен проверить подлинность сертификата клиента перед установлением взаимного доверия.

И сервер и клиент аутентифицируют друг друга сначала проверяя, что представленный сертификат подписан главным сертификатом центра сертификатов, а затем проверяя информацию в заголовке свеже-аутентифицированного сертификата, такую как общее имя или тип сертификата (клиент или сервер).

1.2.1. Установка центра сертификации

Чтобы установить ваш собственный центр сертификации (CA) и сгенерировать сертификаты и ключи для OpenVPN сервера и нескольких клиентов необходимо сначала скопировать `easy-rsa` в каталог `/etc/openvpn`.

Это будет гарантировать, что любые изменения в сценарии не будут потеряны при обновлении пакета. В терминале зайдите под пользователем root и:

```
mkdir /etc/openvpn/easy-rsa/
cp -r /usr/share/easy-rsa/* /etc/openvpn/easy-rsa/
```

Затем отредактируйте `/etc/openvpn/easy-rsa/vars`, настроив следующее под свою рабочую среду:

```
export KEY_COUNTRY="US"
export KEY_PROVINCE="NC"
export KEY_CITY="Winston-Salem"
export KEY_ORG="Example Company"
export KEY_EMAIL="steve@example.com"
export KEY_CN=MyVPN
export KEY_NAME=MyVPN
export KEY_OU=MyVPN
```

Введите следующее для создания главного сертификата центра сертификации (CA) и ключа:

```
cd /etc/openvpn/easy-rsa
Переменные в исходном коде
./clean-all
./build-ca
```

1.2.2. Сертификаты сервера

Далее, мы будем генерировать сертификат и закрытый ключ для сервера:

```
./build-key-server myservername
```

Как и в предыдущем шаге, для большинства параметров можно оставить значения по умолчанию. Два других запроса требуют положительного ответа: "Sign the certificate? [y/n]" и "1 out of 1 certificate requests certified, commit? [y/n]".

Для сервера OpenVPN необходимо использовать алгори́тм Диффи — Хеллмана

```
./build-dh
```

Все сертификаты и ключи были сгенерированы в подкаталоге keys/. Обычно их копируют в /etc/openvpn/:

```
cd keys/
cp myservername.crt myservername.key ca.crt dh1024.pem /etc/openvpn/
```

1.2.3. Сертификаты клиента

Клиенту VPN также будет необходим сертификат для самоаутентификации на сервере. Обычно вы создаёте свой сертификат для каждого клиента. Для создания сертификата, введите следующую команду в терминале, будучи пользователем root:

```
cd /etc/openvpn/easy-rsa
Переменные в исходном коде
./build-key client1
```

Скопируйте следующие файлы на клиент с помощью безопасного метода:

- /etc/openvpn/ca.crt
- /etc/openvpn/easy-rsa/keys/client1.crt
- /etc/openvpn/easy-rsa/keys/client1.key

Все клиентские сертификаты и ключи необходимы только на клиентском компьютере, вы должны удалить их с сервера.

1.3. Простая конфигурация сервера

Вместе с установкой OpenVPN вы получили примеры этиих конфигурационных файлов (и многих других, если вы посмотрите):

```
root@server:/# ls -l /usr/share/doc/openvpn/examples/sample-config-files/
total 68
-rw-r--r-- 1 root root 3427 2011-07-04 15:09 client.conf
-rw-r--r-- 1 root root 4141 2011-07-04 15:09 server.conf.gz
```

Начните с копирования и распаковки server.conf.gz в /etc/openvpn/server.conf.

```
sudo cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/openvpn/
sudo gzip -d /etc/openvpn/server.conf.gz
```

Отредактируйте `/etc/openvpn/server.conf` так, чтобы убедиться, что следующие строки указывают на сертификаты и ключи, которые вы создали в предыдущем разделе.

```
ca ca.crt
cert myservername.crt
key myservername.key
```

```
dh dh1024.pem
```

Это минимальная настройка для получения рабочего сервера OpenVPN. Вы можете использовать все настройки по умолчанию в файле конфигурации server.conf. Теперь запустите сервер. Вы найдете общий отчет и сообщения об ошибках в вашем системном журнале syslog.

```
root@server:/etc/openvpn# service openvpn start
 * Starting virtual private network daemon(s)...
   *   Autostarting VPN 'server'                       [ OK ]
```

Теперь проверьте, что OpenVPN создал интерфейс tun0:

```
root@server:/etc/openvpn# ifconfig tun0
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.8.0.1  P-t-P:10.8.0.2  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
[...]
```

1.4. Простая конфигурация клиента

Существуют различные реализации клиента OpenVPN с GUI и без него. Вы можете прочитать больше о клиентах в одном из следующих разделов. Сейчас мы используем клиент OpenVPN для Ubuntu, который представляет собой тот же исполняемый файл, что и сервер. Поэтому вы должны снова установить пакет openvpn на клиентском компьютере:

```
sudo apt-get install openvpn
```

В этот раз скопируйте файл примера client.conf в каталог /etc/openvpn/:

```
sudo cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/openvpn/
```

Скопируйте ключи и сертификат CA клиента, созданный в предыдущем разделе, например, в /etc/openvpn/ и отредактируйте `/etc/openvpn/client.conf`, чтобы убедиться, что следующие строки указывают на эти файлы. Если ваши файлы находятся в /etc/openvpn/, можно не указывать путь.

```
ca ca.crt
cert client1.crt
key client1.key
```

И вы должны определить по крайней мере имя или адрес сервера OpenVPN. Убедитесь, что в конфигурационном файле присутствует ключевое слово client. Оно включает режим клиента.

```
client
remote vpnserver.example.com 1194
```

Also, make sure you specify the keyfile names you copied from the server

```
ca ca.crt
cert client1.crt
key client1.key
```

Затем запустите клиент OpenVPN:

```
root@client:/etc/openvpn# service openvpn start
 * Starting virtual private network daemon(s)...
   *   Autostarting VPN 'client'                          [ OK ]
```

Проверьте, создан ли интерфейс tun0:

```
root@client:/etc/openvpn# ifconfig tun0
tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.8.0.6  P-t-P:10.8.0.5  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
```

Проверьте командой ping доступность сервера OpenVPN:

```
root@client:/etc/openvpn# ping 10.8.0.1
PING 10.8.0.1 (10.8.0.1) 56(84) bytes of data.
64 bytes from 10.8.0.1: icmp_req=1 ttl=64 time=0.920 ms
```

Сервер OpenVPN всегда использует первый доступный адрес в сети клиента, и только этот IP адрес будет откликаться на ping. Например, если вы настроили сетевую маску клиента /24, то будет использован адрес, оканчивающийся на .1. Адрес P-t-P, который вы видите в выводе ifconfig выше, обычно не отвечает на запросы ping.

Проверьте свои маршруты:

```
root@client:/etc/openvpn# netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
10.8.0.5        0.0.0.0         255.255.255.255 UH        0 0          0 tun0
10.8.0.1        10.8.0.5        255.255.255.255 UGH       0 0          0 tun0
192.168.42.0    0.0.0.0         255.255.255.0   U         0 0          0 eth0
0.0.0.0         192.168.42.1    0.0.0.0         UG        0 0          0 eth0
```

1.5. Первые поиски и устранения неисправностей

Если что-то из того, что рассмотрено выше, у вас не работает, проверьте следующее:

- Проверьте ваш системный журнал, например, с помощью grep -i vpn /var/log/syslog
- Проверьте, правильно ли вы указали имена ключевых файлов в client.conf и server.conf.
- Может ли клиент подключиться к серверу? Может быть, брандмауэр блокирует доступ? Проверьте системный журнал на сервере.
- Клиент и сервер должны использовать тот же протокол и порт, например. UDP порт 1194, смотрите порт и опцию proto в config
- Клиент и сервер должны использовать один и тот же алгоритм сжатия, смотрите опцию comp-lzo в config
- Клиент и сервер должны использовать одну и ту же конфигурацию относительно режима мости режима маршрутизации, смотрите server и server-bridge в config

1.6. Расширенные настройки

1.6.1. Расширенная настройка VPN на сервере в режиме маршрутизации

Выше рассмотрена очень простая работающая настройка VPN. Клиент имеет доступ к сервисам на машине VPN сервера через зашифрованный канал. Если вы хотите получить доступ к большему количеству серверов или к чему-то в других сетях, добавьте несколько маршрутов на клиенте. Например, если сеть вашей компании в целом может быть описана как 192.168.0.0/16, вы можете добавить этот маршрут на клиенте. Но вам придется также изменить маршрут для обратного направления - ваши сервера должны знать как проложить маршрут до сети VPN клиента.

Или вы можете указать шлюз по умолчанию для всех клиентов чтобы посылать весь их трафик сначала на VPN сервер, а от него через защитный сервер (firewall) компании в интернет. В этом разделе вы увидите некоторые возможные варианты настроек.

Передать маршрут клиенту чтобы разрешить ему доступ к другим частным подсетям за сервером. Помните, что эти частные сети также должны знать как построить маршрут до диапазона адресов клиента OpenVPN (10.8.0.0/24), находящегося за OpenVPN сервером:

```
push "route 10.0.0.0 255.0.0.0"
```

If enabled, this directive will configure all clients to redirect their default network gateway through the VPN, causing all IP traffic such as web browsing and DNS lookups to go through the VPN (the OpenVPN server machine or your central

firewall may need to NAT the TUN/TAP interface to the internet in order for this to work properly).

```
push "redirect-gateway def1 bypass-dhcp"
```

Настройте режим сервера и предоставьте подсеть VPN, откуда OpenVPN будет брать адреса клиентов. Сервер возьмёт себе адрес 10.8.0.1, а остальные могут использоваться для предоставления клиентам. Каждый клиент будет иметь возможность подключиться к серверу по адресу 10.8.0.1. Закомментируйте эту строку, если используете режим сетевого моста:

```
server 10.8.0.0 255.255.255.0
```

Сохраните записи соответствий клиентов с их виртуальными IP-адресами в указанном файле. Если OpenVPN выключается или перегружается, повторно подключившиеся клиенты получат те же виртуальные IP-адреса, что и в прошлый раз:

```
ifconfig-pool-persist ipp.txt
```

Передать настройки на DNS сервера клиенту:

```
push "dhcp-option DNS 10.0.0.2"
push "dhcp-option DNS 10.1.0.2"
```

Разрешите связь между клиентами.

```
client-to-client
```

Включите сжатие в VPN канале.

```
comp-lzo
```

Директива keepalive обеспечивает отправку сообщений типа ping вперёд и назад через соединение для того, чтобы каждая сторона знала, когда другая сторона становится недоступна. Проверка выполняется раз в секунду; предполагается, что что удалённый узел не отвечает, если ответ не получен в течение 3 секунд.

```
keepalive 1 3
```

Неплохая идея — понизить привилегии демона OpenVPN после инициализации.

```
user nobody
group nogroup
```

OpenVPN 2.0 включает в себя функцию, которая позволяет серверу OpenVPN безопасным способом получить имя пользователя и пароль от подключаемого клиента, и использовать эту информацию в качестве основы для аутентификации клиента. Чтобы использовать этот метод аутентификации, сначала добавьте директиву auth-user-pass для настройки клиента. Это укажет клиенту OpenVPN запросить у пользователя "имя пользователя/пароль" для передачи на сервер по защищенному TLS-каналу.

```
# client config!
auth-user-pass
```

Это заставит сервер OpenVPN проверить имя пользователя/пароль, полученный от клиента, используя логин PAM-модуля. Полезно, если у вас есть централизованная аутентификация, например Kerberos.

```
plugin /usr/lib/openvpn/openvpn-plugin-auth-pam.so login
```

Пожалуйста, ознакомьтесь с дополнительными мерами безопасности в *руководстве по усиленной безопасности*[1] OpenVPN.

1.6.2. Расширенная настройка VPN на сервере в режиме сетевого моста

OpenVPN может быть установлен либо для маршрутизации либо в режиме моста VPN. Иногда это называют OSI Layer-2 по сравнению с Layer-3 VPN. В режиме моста VPN все кадры Layer-2, например все Ethernet кадры отправляются в VPN канал, а в режиме маршрутизации в VPN канал отправляются Layer-3 пакеты. В режиме моста весь трафик, включая трафик локальной сети, как Broadcast, DHCP-запросы, ARP запросы и т.д., направляются в VPN канал, тогда как в режиме маршрутизации он будет отфильтрован.

1.6.2.1. На сервере предварительно сконфигурируйте интерфейс для режима моста

Убедитесь, что у вас установлен пакет bridge-utils:

```
sudo apt-get install bridge-utils
```

Перед вами настройки для OpenVPN в режиме моста, вы должны изменить ваши настройки интерфейса. Давайте предположим, что ваш сервер имеет

[1] http://openvpn.net/index.php/open-source/documentation/howto.html#security

интерфейс eth0 подсоединен к Интернет и интерфейс eth1 подключен к локальной сети, вы хотите, чтобы он работал в режиме моста. Ваш файл конфигурации должен быть выгляжеть примерно так /etc/network/interfaces:

```
auto eth0
iface eth0 inet static
  address 1.2.3.4
  netmask 255.255.255.248
  default 1.2.3.1

auto eth1
iface eth1 inet static
  address 10.0.0.4
  netmask 255.255.255.0
```

Эта простая конфигурация должна быть изменена для режима моста, где конфигурация интерфейса eth1 переходит на новый интерфейс br0. Плюс мы настроим, что br0 будет привязан к интерфейсу eth1. Мы также должны убедиться, что интерфейс eth1 всегда в «неразборчивом» режиме, в котором сетевая плата позволяет принимать все пакеты независимо от того, кому они адресованы.

```
auto eth0
iface eth0 inet static
  address 1.2.3.4
  netmask 255.255.255.248
  default 1.2.3.1

auto eth1
iface eth1 inet manual
  up ip link set $IFACE up promisc on

auto br0
iface br0 inet static
  address 10.0.0.4
  netmask 255.255.255.0
  bridge_ports eth1
```

At this point you need to bring up the bridge. Be prepared that this might not work as expected and that you will lose remote connectivity. Make sure you can solve problems having local access.

```
sudo ifdown eth1 && sudo ifup -a
```

1.6.2.2. Подготовить конфигурацию сервера для режима моста

Отредактируйте `/etc/openvpn/server.conf` изменяя следующие настройки на:

```
;dev tun
dev tap
up "/etc/openvpn/up.sh br0 eth1"
;server 10.8.0.0 255.255.255.0
server-bridge 10.0.0.4 255.255.255.0 10.0.0.128 10.0.0.254
```

Далее, создайте вспомогательный скрипт, чтобы добавить *tap* интерфейс к мосту, и обеспечте, чтобы интерфейс eth1 был в «неразборчивом» режиме. Создайте `/etc/openvpn/up.sh`:

```
#!/bin/sh

BR=$1
ETHDEV=$2
TAPDEV=$3

/sbin/ip link set "$TAPDEV" up
/sbin/ip link set "$ETHDEV" promisc on
/sbin/brctl addif $BR $TAPDEV
```

Затем сделайте его исполняемым:

```
sudo chmod 755 /etc/openvpn/up.sh
```

После настройки сервера, перезапустите openvpn, введя:

```
sudo service openvpn restart
```

1.6.2.3. Настройка клиента

Сначала установите openvpn на стороне клиента:

```
sudo apt-get install openvpn
```

После окончания конфигурирования сервера и копирования клиентского сертификата в папку `/etc/openvpn/`, создайте файл конфигурации клиента, используя приведенный пример. В окне терминала на клиентской машине введите:

```
sudo cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/openvpn
```

Теперь отредактируйте `/etc/openvpn/client.conf`, изменив следующие параметры:

```
dev tap
```

```
;dev tun
ca ca.crt
cert client1.crt
key client1.key
```

Наконец, перезапустите openvpn:

```
sudo service openvpn restart
```

Теперь вам должна быть доступна возможность подсоединения к удалённой сети через VPN.

1.7. Реализации клиентского программного обеспечения

1.7.1. Графический интерфейс сетевого менеджера Linux для OpenVPN

Многие дистрибутивы Linux, включая варианты Ubuntu для настольных компьютеров, поставляются с Network Manager — удобным графическим интерфейсом для настройки параметров сети. Он также может управлять вашими VPN-соединениями. Убедитесь, что у вас установлен пакет network-manager-openvpn. Здесь вы видите, что устанавливаются также и другие необходимые пакеты:

```
root@client:~# apt-get install network-manager-openvpn
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  liblzo2-2 libpkcs11-helper1 network-manager-openvpn-gnome openvpn
Suggested packages:
  resolvconf
The following NEW packages will be installed:
  liblzo2-2 libpkcs11-helper1 network-manager-openvpn
  network-manager-openvpn-gnome openvpn
0 upgraded, 5 newly installed, 0 to remove and 631 not upgraded.
Need to get 700 kB of archives.
After this operation, 3,031 kB of additional disk space will be used.
Do you want to continue [Y/n]?
```

Чтобы информировать network-manager о новых установленных пакетах, вам придется перезапустить его:

```
root@client:~# restart network-manager
network-manager start/running, process 3078
```

Open the Network Manager GUI, select the VPN tab and then the 'Add' button. Select OpenVPN as the VPN type in the opening requester and press 'Create'. In

the next window add the OpenVPN's server name as the 'Gateway', set 'Type' to 'Certificates (TLS)', point 'User Certificate' to your user certificate, 'CA Certificate' to your CA certificate and 'Private Key' to your private key file. Use the advanced button to enable compression (e.g. comp-lzo), dev tap, or other special settings you set on the server. Now try to establish your VPN.

1.7.2. OpenVPN с GUI для Mac OS X: Tunnelblick

Tunnelblick — это отличная бесплатная, свободная реализации GUI OpenVPN для OS X. Домашняя страница проекта *http://code.google.com/p/tunnelblick/*. Скачайте оттуда последнюю версию для OS X и установите её. Затем поместите конфигурационный файл client.ovpn вместе с сертификатами и ключами в /Users/username/Library/Application Support/Tunnelblick/Configurations/ и запустите Tunnelblick

```
# sample client.ovpn for Tunnelblick
client
remote blue.example.com
port 1194
proto udp
dev tun
dev-type tun
ns-cert-type server
reneg-sec 86400
auth-user-pass
auth-nocache
auth-retry interact
comp-lzo yes
verb 3
ca ca.crt
cert client.crt
key client.key
```

1.7.3. OpenVPN с GUI для Win 7

First download and install the latest *OpenVPN Windows Installer*[2]. OpenVPN 2.3.2 was the latest when this was written. As of this writing, the management GUI is included with the Windows binary installer.

Вы должны запустить службу OpenVPN. Путь: Пуск> Компьютер> Управление> Службы и приложения> Службы. Найдите OpenVPN службу и запустить ее. Установите тип ее запуска автоматически. Когда вы запускаете OpenVPN GUI MI первый раз вам нужно запустите ее с правами администратора. Вы должны щелкнуть правой кнопкой мыши по нему и вы увидите дополнительные опции.

[2] http://www.openvpn.net/index.php/open-source/downloads.html

Вам придется написать свой конфигурационный файл OpenVPN в текстовом файле и поместить его в C:\Program Files\OpenVPN\config\client.ovpn вместе с сертификатами CA. Вы можете установить сертификат пользователя в его домашний каталог, как в следующем примере.

```
# C:\Program Files\OpenVPN\config\client.ovpn
client
remote server.example.com
port 1194
proto udp
dev tun
dev-type tun
ns-cert-type server
reneg-sec 86400
auth-user-pass
auth-retry interact
comp-lzo yes
verb 3
ca ca.crt
cert "C:\\Users\\username\\My Documents\\openvpn\\client.crt"
key "C:\\Users\\username\\My Documents\\openvpn\\client.key"
management 127.0.0.1 1194
management-hold
management-query-passwords
auth-retry interact
; Set the name of the Windows TAP network interface device here
dev-node MyTAP
```

Примечание: если вы не используете аутентификацию пользователей и/или хотите запустить службу без взаимодействия с пользователями, закомментируйте следующие опции:

```
auth-user-pass
auth-retry interact
management 127.0.0.1 1194
management-hold
management-query-passwords
```

You may want to set the Windows service to "automatic".

1.7.4. OpenVPN для OpenWRT

OpenWRT описывается в виде дистрибутива Linux для встраиваемых устройств, таких как WLAN маршрутизаторы. Есть определенные типы WLAN-маршрутизаторов, которые могут быть прошиты в OpenWRT. В зависимости от доступной памяти на вашем OpenWRT маршрутизаторе, вы можете запустить программное обеспечение, как OpenVPN и вы могли бы, например поставить в небольшой филиал недорогой маршрутизатор для

соединений по VPN в центральный офис. Больше информации о OpenVPN на OpenWRT *здесь*[3]. И вот проект OpenWRT-страницу: *http://openwrt.org*

Войдите в свой OpenWRT маршрутизатор и установите OpenVPN:

```
opkg update
opkg install openvpn
```

Check out /etc/config/openvpn and put your client config in there. Copy certificates and keys to /etc/openvpn/

```
config openvpn client1
        option enable 1
        option client 1
#       option dev tap
        option dev tun
        option proto udp
        option ca /etc/openvpn/ca.crt
        option cert /etc/openvpn/client.crt
        option key /etc/openvpn/client.key
        option comp_lzo 1
```

Перезапустите OpenVPN:

```
service openvpn restart
```

Вы должны видеть, если вам нужно настроить маршруты, маршрутизацию и правила брандмауэра.

1.8. Ссылки

- Обращайтесь на сайт *OpenVPN*[4] за дополнительной информацией.
- *Руководство усиленной по безопасности OpenVPN*[5]
- Также, хорошим подспорьем будет руководство издательства Pakt: *OpenVPN: Building and Integrating Virtual Private Networks*[6].

[3] http://wiki.openwrt.org/doc/howto/vpn.overview
[4] http://openvpn.net/
[5] http://openvpn.net/index.php/open-source/documentation/howto.html#security
[6] http://www.packtpub.com/openvpn/book

Глава 24. Другие полезные приложения

Существует множество очень полезных приложений, разработанных командой Ubuntu Server Team и другими разработчиками, интегрированных в Ubuntu Server Edition, но которые могут быть не очень хорошо известны. Эта глава демонстрирует несколько полезных приложений, которые делают управление сервером Ubuntu (или множеством Ubuntu серверов) более простым.

1. pam_motd

При входе на сервер Ubuntu вы можете заметить информативное сообщение дня (Informative Message Of The Day — MOTD). Эта информация собирается и отображается с использованием пары пакетов:

- *landscape-common:* provides the core libraries of landscape-client, which is needed to manage systems with *Landscape*[1] (proprietary). Yet the package also includes the landscape-sysinfo utility which is responsible for displaying core system data involving cpu, memory, disk space, etc. For instance:

```
System load:  0.0               Processes:           76
Usage of /:   30.2% of 3.11GB   Users logged in:     1
Memory usage: 20%               IP address for eth0: 10.153.107.115
Swap usage:   0%

Graph this data and manage this system at https://landscape.canonical.com/
```

You can run landscape-sysinfo manually at any time.

- *update-notifier-common:* provides information on available package updates, impending filesystem checks (fsck), and required reboots (e.g.: after a kernel upgrade).

pam_motd выполняет сценарии в `/etc/update-motd.d` в порядке чисел, предваряющих имена файлов. Вывод сценариев записывается в `/var/run/motd` с сохранением порядковых номеров и затем объединяется с `/etc/motd.tail`.

Вы можете добавить вашу собственную динамическую информацию в MOTD. Например, чтобы добавить информацию о местной погоде:

- Сначала установите пакет weather-util:

```
sudo apt-get install weather-util
```

- Утилита weather использует данные METAR из National Oceanic and Atmospheric Administration и прогнозы от National Weather Service. Чтобы найти информацию для вашего региона вам потребуется 4-хсимвольный индикатор местоположения ICAO. Он может быть определен при заходе на сайт *National Weather Service*[2] .

  Несмотря на то, что National Weather Service является правительственным агентством США, оно имеет доступ к данным от погодных станций

[1] http://landscape.canonical.com/
[2] http://www.weather.gov/tg/siteloc.shtml

по всему миру. Однако локальная информация о погоде может предоставляться не для всех мест за пределами США.

- Создайте файл `/usr/local/bin/local-weather`, простейший shell сценарий, использующий weather с вашим ICAO индикатором местоположения:

```
#!/bin/sh
#
#
# Prints the local weather information for the MOTD.
#
#

# Replace KINT with your local weather station.
# Local stations can be found here: http://www.weather.gov/tg/siteloc.shtml

echo
weather -i KINT
echo
```

- Сделайте сценарий исполняемым:

```
sudo chmod 755 /usr/local/bin/local-weather
```

- Создайте символьную ссылку в `/etc/update-motd.d/98-local-weather`:

```
sudo ln -s /usr/local/bin/local-weather /etc/update-motd.d/98-local-weather
```

- Наконец, выйдите из сервера и войдите повторно, чтобы увидеть новое сообщение MOTD.

Теперь вы будете получать приветствия с некоторой полезной информацией и информацией о погоде, которая может быть не такой полезной. Надеемся, пример с application>local-weather

2. etckeeper

etckeeper позволяет легко сохранять содержимое каталога `/etc` в репозиторий системы контроля версий (VCS). Он отслеживает когда apt автоматически сохраняет изменения в /etc при установке или обновлении пакетов. Помещение `/etc` под контроль версий сейчас рассматривается как лучшая практика в индустрии, и назначение etckeeper — сделать этот процесс безболезненным, насколько это возможно.

Установите etckeeper, введя следующую команду в терминале:

```
sudo apt-get install etckeeper
```

Основной файл конфигурации, `/etc/etckeeper/etckeeper.conf`, достаточно простой. Основной опцией является выбор какую VSC использовать. По умолчанию etckeeper настроен на использование в качестве системы контроля версий bzr. Хранилище автоматически инициализируется (и сохраняет начальное состояние) в процессе установки. Есть возможность отменить это, выполнив следующую команду:

```
sudo etckeeper uninit
```

По умолчанию etckeeper будет сохранять незафиксированные изменения в /etc ежедневно. Это может быть отменено использованием опции настройки AVOID_DAILY_AUTOCOMMITS. Он также будет автоматически сохранять изменения до и после установки пакетов. Для более точного отслеживания изменений рекомендуется фиксировать изменения вручную, добавляя описание фиксации следующим образом:

```
sudo etckeeper commit "..Reason for configuration change.."
```

С помощью команд VCS Вы можете просмотреть логи о файлах в `/etc`:

```
sudo bzr log /etc/passwd
```

Чтобы показать интеграцию с системой управления пакетами, установите postfix:

```
sudo apt-get install postfix
```

После завершения установки, все postfix конфигурационные файлы должны быть записаны в репозиторий:

```
Committing to: /etc/
added aliases.db
```

```
modified group
modified group-
modified gshadow
modified gshadow-
modified passwd
modified passwd-
added postfix
added resolvconf
added rsyslog.d
modified shadow
modified shadow-
added init.d/postfix
added network/if-down.d/postfix
added network/if-up.d/postfix
added postfix/dynamicmaps.cf
added postfix/main.cf
added postfix/master.cf
added postfix/post-install
added postfix/postfix-files
added postfix/postfix-script
added postfix/sasl
added ppp/ip-down.d
added ppp/ip-down.d/postfix
added ppp/ip-up.d/postfix
added rc0.d/K20postfix
added rc1.d/K20postfix
added rc2.d/S20postfix
added rc3.d/S20postfix
added rc4.d/S20postfix
added rc5.d/S20postfix
added rc6.d/K20postfix
added resolvconf/update-libc.d
added resolvconf/update-libc.d/postfix
added rsyslog.d/postfix.conf
added ufw/applications.d/postfix
Committed revision 2.
```

В качестве примера, как etckeeper отслеживает изменения вручную, добавьте новую систему в `/etc/hosts`. Используя `/etc/hosts` вы сможете увидеть какие файлы были изменены:

```
sudo bzr status /etc/
modified:
  hosts
```

Теперь сохраните изменения:

```
sudo etckeeper commit "new host"
```

Дополнительную информацию по bzr смотрите в разделе *Раздел 1, «Bazaar» [312]*.

3. Byobu

Наиболее используемым приложением любого системного администратора является screen. Оно позволяет выполнять несколько оболочек в одном терминале. Чтобы сделать некоторые расширенные возможности screen более дружественными и предоставить некоторую полезную информацию о системе, был создан пакет byobu.

При выполнении byobu нажатие клавиши F9 выдаст меню настройки. Это меню позволит вам:

- Посмотреть меню помощи
- Изменить цвет фона Byobu
- Изменить цвет переднего плана Byobu
- Изменить состояние статусных уведомлений
- Изменить набор связанных клавиш
- Изменить последовательность для выхода
- Создавать новые окна
- Управлять окнами по умолчанию
- Byobu не запускается при входе в систему (включить)

Связанные клавиши определяют такие вещи, как последовательность для выхода, открытие нового окна, изменение окна и т.д. Можно выбирать между двумя наборами связанных клавиш: *f-keys* и *screen-escape-keys*. Если вы собираетесь использовать оригинальные последовательности, выберите набор *none*.

byobu предоставляет меню, которое показывает версию Ubuntu, информацию о процессоре, о памяти, а также дату и время. Создается эффект меню рабочего стола.

Использование опции "*Byobu на данный момент не загружаться при входе (включить)*" позволит byobu выполняться каждый раз, когда открывается терминал. Изменения для byobu выполняются для каждого пользователя отдельно и не влияют на настройки других пользователей системы.

Одно из различий при использовании byobu является *режим прокрутки*. Нажмите клавишу *F7* для входа в режим прокрутки. Режим прокрутки позволяет вам передвигаться по последнему выводу с использованием команд, аналогичных *vi*. Здесь короткий список команд перемотки:

- *h* — перемещает курсор влево на один символ
- *j* — перемещает курсор вниз на одну строку

- *k* — перемещает курсор вверх на одну строку
- *l* — перемещает курсор вправо на один символ
- *0* — перемещает курсор в начало текущей строки
- *$* — перемещает курсор в конец текущей строки
- *G* — выполняет переход на заданную строку (по умолчанию в конец буфера)
- */* — поиск по тексту
- *?* — Поиск в обратном направлении
- *n* - Moves to the next match, either forward or backward

4. Ссылки

- See the *update-motd man page*[3] for more options available to update-motd.
- Статья из Debian Package of the Day *weather*[4] содержит дополнительные детали об использовании утилиты weather.
- Посетите сайт *etckeeper*[5] чтобы узнать подробности использования etckeeper.
- *etckeeper на Ubuntu Wiki*[6]
- Последние новости и информацию о bzr смотрите на *сайте bzr*[7]
- Дополнительная информация по screen доступна на *сайте screen*[8].
- Информация о screen на *Ubuntu Wiki*[9].
- Также посетите страницу *проекта byobu* [10] для дополнительной информации.

[3] http://manpages.ubuntu.com/manpages/trusty/en/man5/update-motd.5.html
[4] http://debaday.debian.net/2007/10/04/weather-check-weather-conditions-and-forecasts-on-the-command-line/
[5] http://kitenet.net/~joey/code/etckeeper/
[6] https://help.ubuntu.com/community/etckeeper
[7] http://bazaar-vcs.org/
[8] http://www.gnu.org/software/screen/
[9] https://help.ubuntu.com/community/Screen
[10] https://launchpad.net/byobu

Приложение А. Дополнение

1. Уведомление об ошибках в Ubuntu Server Edition

Хотя проект Ubuntu старается свести к минимуму количество ошибок в выпускаемым им программном обеспечении, ошибки всё равно встречаются. Вы можете помочь их исправить, отправляя проекту отчёты об обнаруженных ошибках. Проект Ubuntu использует *Launchpad*[1] для отслеживания отчётов о своих ошибках. Чтобы сообщить об ошибке в Ubuntu Server на Launchpad, вам потребуется *создать учётную запись*[2].

1.1. Уведомление об ошибках с помощью ubuntu-bug

Предпочтительным способом отправить отчет об ошибке является использование команды ubuntu-bug. Утилита ubuntu-bug собирает информацию о системе, полезную разработчикам при диагностике описываемой проблемы, которая затем будет включена в отчёт об ошибке, зарегистрированный на Launchpad. Отчёты об ошибках в Ubuntu требуют указания программного пакета, поэтому имя пакета, в котором произошла ошибка, должно быть передано ubuntu-bug:

```
ubuntu-bug ИМЯ_ПАКЕТА
```

Например, чтобы сообщить об ошибке в пакете openssh-server, нужно набрать:

```
ubuntu-bug openssh-server
```

Вы можете указать для ubuntu-bug как двоичный пакет, так и пакет исходного кода. Опять же, используя openssh-server в качестве примера, вы можете создать отчёт по пакету исходного кода для openssh-server, openssh:

```
ubuntu-bug openssh
```

Чтобы узнать больше о пакетах в Ubuntu, смотрите *Глава 3, Управление пакетами [24]*.

Команда ubuntu-bug собирает информацию об упомянутой системе, возможно, включая специфическую информацию для указанного пакета, и затем спрашивает что вы собираетесь делать с собранной информацией:

[1] https://launchpad.net/
[2] https://help.launchpad.net/YourAccount/NewAccount

```
ubuntu-bug postgresql

*** Collecting problem information

The collected information can be sent to the developers to improve the
application. This might take a few minutes.
..........

*** Send problem report to the developers?

After the problem report has been sent, please fill out the form in the
automatically opened web browser.

What would you like to do? Your options are:
  S: Send report (1.7 KiB)
  V: View report
  K: Keep report file for sending later or copying to somewhere else
  C: Cancel
Please choose (S/V/K/C):
```

Доступными вариантами являются:

- **Send Report** Selecting Send Report submits the collected information to Launchpad as part of the process of filing a bug report. You will be given the opportunity to describe the situation that led up to the occurrence of the bug.

  ```
  *** Uploading problem information

  The collected information is being sent to the bug tracking system.
  This might take a few minutes.
  91%

  *** To continue, you must visit the following URL:

  https://bugs.launchpad.net/ubuntu/+source/postgresql-8.4/+filebug/kc6eSnTLnLxF8u0t3e56EukFeqJ?

  You can launch a browser now, or copy this URL into a browser on another
  computer.

  Choices:
    1: Launch a browser now
    C: Cancel
  Please choose (1/C):
  ```

  Если вы выберете запуск браузера, по умолчанию будет запущен текстовый веб-браузер w3m для завершения регистрации отчёта об ошибке. В качестве альтернативы вы можете скопировать указанный URL в уже запущенный веб-браузер.
- **View Report.** Выбор просмотра отчёта приведёт к показу собранной информации в терминале для проверки.

```
Package: postgresql 8.4.2-2
PackageArchitecture: all
Tags: lucid
ProblemType: Bug
ProcEnviron:
  LANG=en_US.UTF-8
  SHELL=/bin/bash
Uname: Linux 2.6.32-16-server x86_64
Dependencies:
  adduser 3.112ubuntu1
  base-files 5.0.0ubuntu10
  base-passwd 3.5.22
  coreutils 7.4-2ubuntu2
...
```

После просмотра отчёта вы будете снова перенаправлены в меню с вопросом о том, что вы собираетесь делать с отчётом.

- **Keep Report File.** Выбор сохранения файла отчёта приведёт к записи в файл собранной информации. Этот файл может быть использован для дальнейшей регистрации отчёта об ошибке или передан для отправки отчёта в другую систему Ubuntu. Чтобы передать файл отчёта, просто укажите его в качестве аргумента команды ubuntu-bug:

```
What would you like to do? Your options are:
  S: Send report (1.7 KiB)
  V: View report
  K: Keep report file for sending later or copying to somewhere else
  C: Cancel
Please choose (S/V/K/C): k
Problem report file: /tmp/apport.postgresql.v4MQas.apport

ubuntu-bug /tmp/apport.postgresql.v4MQas.apport

*** Send problem report to the developers?
...
```

- **Cancel.** Выбор отмены приведёт к тому, что собранная информация будет сброшена.

1.2. Уведомление о сбоях приложений

Пакет программ, который предоставляет утилиту ubuntu-bug (apport), может быть настроен на срабатывание при падении приложений. По умолчанию это отключено, поскольку захват сбоев может быть достаточно ресурсоёмким в зависимости от количества памяти, которую использовало упавшее приложение, а apport захватывает и обрабатывает память ядра.

Настройка apport на захват информации о падении приложений требует выполнения пары шагов. Сначала требуется установить gdb; он не установлен по умолчанию в Ubuntu Server Edition.

```
sudo apt-get install gdb
```

Смотрите раздел *Глава 3, Управление пакетами [24]* для дополнительной информации об управлении пакетами в Ubuntu.

Как только вы убедитесь, что gdb установлен, откройте файл `/etc/default/apport` в вашем текстовом редакторе и измените настройку *enabled* в **1**, как показано ниже:

```
# set this to 0 to disable apport, or to 1 to enable it
# you can temporarily override this with
# sudo service apport start force_start=1
enabled=1

# set maximum core dump file size (default: 209715200 bytes == 200 MB)
maxsize=209715200
```

После того, как завершите редактирование `/etc/default/apport`, запустите службу apport:

```
sudo start apport
```

После падения приложения используйте команду apport-cli для поиска информации о сохраненном отчёте о сбое приложения:

```
apport-cli

*** dash closed unexpectedly on 2010-03-11 at 21:40:59.

If you were not doing anything confidential (entering passwords or other
private information), you can help to improve the application by
reporting
the problem.

What would you like to do? Your options are:
  R: Report Problem...
  I: Cancel and ignore future crashes of this program version
  C: Cancel
Please choose (R/I/C):
```

Выбор *Report Problem* (сообщить о проблеме) проведёт вас по шагам, аналогичным использованию ubuntu-bug. Одним важным отличием будет то, что отчёт о сбое будет помечен как частный (private) при регистрации

на Launchpad, то есть он будет виден только ограниченному количеству сортировщиков. Эти сортировщики просмотрят собранные данные на наличие частной информации перед тем, как отчёт об ошибке станет публично доступным.

1.3. Ресурсы

- Посетите wiki страницу *Reporting Bugs*[3].
- Также, страница *Apport*[4] содержит некоторую полезную информацию. Часть информации касается использования графического интерфейса.

[3] https://help.ubuntu.com/community/ReportingBugs
[4] https://wiki.ubuntu.com/Apport