После установки нового сервера приходится выполнять один и тот же набор стандартных настроек. Сегодня мы займемся базовой настройкой сервера под управлением операционной системы **Debian**. Я приведу практические советы по небольшому увеличению безопасности и удобству администрирования, основанные на моем личном опыте.

Содержание:

Введение

Любая работа с сервером после установки чаще всего начинается со стандартных обязательных действий, без которых либо не получится продвинуться дальше, либо будет неудобно работать. Например, вам в любом случае необходимо выполнить сетевые настройки, желательно обновить систему и установить часовой пояс. Рекомендуется сразу настроить автообновление времени, подрихтовать параметры sshd, установить midnight commander и

выполнить другие настройки.

Об этом я хочу рассказать в статье. Я буду делиться своим реальным опытом работы. Это не значит, что нужно делать так, как я. Я могу в чем-то ошибаться, что-то делать не так удобно, как можно было бы сделать. Это просто советы, которые кому-то помогут узнать что-то новое, а кто-то возможно поделится со мной чем-то новым для меня, либо укажет на мои ошибки. Мне бы хотелось, чтобы это было так. Своими материалами я не только делюсь с вами знаниями, но и сам узнаю что-то новое в том числе и из комментариев и писем на почту.

Указываем сетевые параметры

Итак, у нас в наличии только что установленная система. Узнать или проверить ее версию можно командами:

```
# uname -a
Linux debian 3.16.0-4-amd64 #1 SMP Debian 3.16.7-ckt20-1+deb8u3 (2016-01-17) x86_64 GNU/Linux
# lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description: Debian GNU/Linux 8.3 (jessie)
Release: 8.3
Codename: jessie
```

Очень подробно про настройку сети в Debian я написал в отдельной статье. Рекомендую с ней ознакомиться. Здесь же кратко выполним основное. Для настройки сети, необходимо отредактировать файл */etc/network/interfaces.* Сделаем это:

```
# nano /etc/network/interfaces
```

Для получения IP адреса по dhcp достаточно будет следующего содержания:

```
allow-hotplug eth0
iface eth0 inet dhcp
```

Если у вас статический адрес, то его настроить можно следующими параметрами в файле:

```
allow-hotplug eth0
iface eth0 inet static
address 192.168.1.24
netmask 255.255.255.0
gateway 192.168.1.1
dns-nameservers 192.168.1.1
```

Сохраняем файл. Теперь нужно выполнить перезапуск сети. В Debian это делается командой:

```
# /etc/init.d/networking restart
[ ok ] Restarting networking (via systemctl): networking.service.
```

На этом настройка сети закончена.

Обновление системы

Сеть настроили, теперь можно обновить систему и пакеты. В Debian это делается достаточно просто. Воспользуемся несколькими командами. Сначала обновим локальный индекс пакетов до последних изменений в репозиториях:

```
# apt-get update
```

Теперь выполним обновление всех пакетов системы:

```
# apt-get upgrade
```

На этом обновление системы закончено. Если вы хотите обновить версию релиза, например Debian 7 wheezy обновить до Debian 8 Jessie, то читайте отдельный материал.

Настройка ssh

Теперь внесем некоторые изменения в настройки сервера **ssh**. Я рекомендую его запускать на нестандартном порту для исключения лишних общений с ботами, которые регулярно сканят инет и подбирают пароли пользователей по словарям. По-умолчанию в Debian, впрочем как и в любом другом дистрибутиве Linux, ssh сервер работает на 22 порту. Изменим этот порт, к примеру, на 23331. Так же я еще изменяю конфигурацию для разрешения подключения по ssh пользователя root с использованием пароля. В Debian из коробки пользователь root по ssh паролем авторизовываться не может. Изменим и это. Открываем файл настроек:

```
# nano /etc/ssh/sshd_config
```

И изменяем там следующие строки. Приводим их к виду:

```
Port 23331
PermitRootLogin yes
```

Сохраняем изменения и перезапускаем сервер ssh следующей командой:

```
# service sshd restart
```

Проверяем изменения:

```
# netstat -tulnp | grep ssh

tcp 0 0 0.0.0.0:23331 0.0.0.0:* LISTEN 925/sshd
tcp6 0 0 :::23331 :::* LISTEN 925/sshd
```

Все в порядке, сервер слушает 23331 порт. Теперь новое подключение будет осуществлено только по порту 23331. При этом, после перезапуска ssh, старое подключение не будет разорвано.

Я знаю, что многие возражают против подключения рутом к серверу. Якобы это небезопасно и т.д. и т.п. Мне эти доводы кажутся не убедительными. Не понимаю, в чем может быть проблема, если у меня нормальный сложный пароль на root, который не получится подобрать или сбрутить. Ни разу за всю

мою работу системным администратором у меня не возникло проблем с этим моментом. А вот работать так значительно удобнее, особенно, когда необходимо оперативно куда-то подключиться по форс мажорным обстоятельствам.

Установка утилит mc, htop, iftop

Следующим шагом я настраиваю некоторые полезные утилиты, которыми регулярно пользуюсь в повседневной работе. Первая из них это всем известный двухпанельный файловый менеджер Midnight Commander. Установим **mc** на наш сервер:

```
# apt-get -y install mc
```

Я сразу же ставлю редактором по-умолчанию **mcedit**. Для этого просто выбираю его из меню при первом редактировании какого-нибудь файла. Если у вас такое меню не появляется, можете вызвать его сами и выбрать необходимый редактор по-умолчанию:

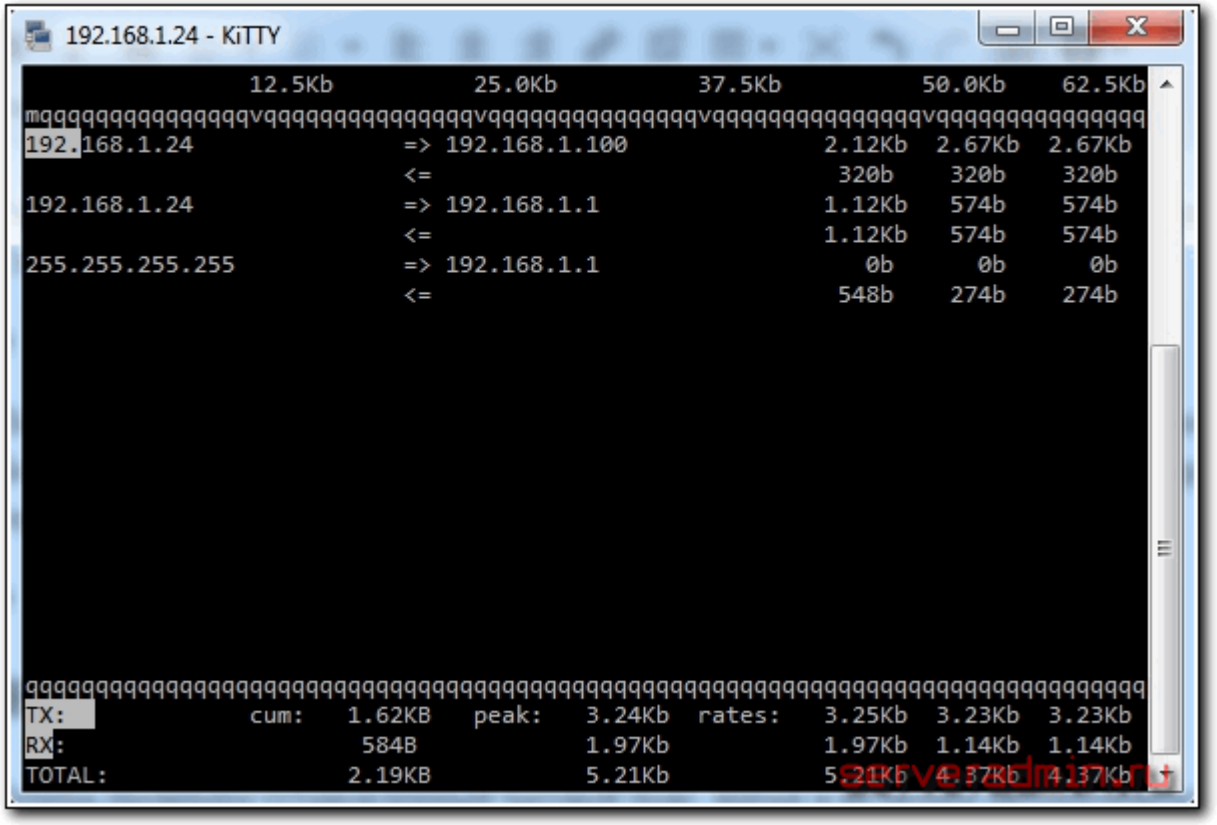
```
# select-editor

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano <---- easiest
 2. /usr/bin/mcedit
 3. /usr/bin/vim.tiny

Choose 1-3 [1]: 2
```

Так же я рекомендую очень удобный диспетчер задач — **htop**. Мне он помог, к примеру, решить проблему Взлома сервера CentOS. Ставим его на сервер:

```
# apt-get -y install htop
```

Полезной утилитой, позволяющей смотреть сетевую загрузку в режиме реального времени, является **iftop**. Очень рекомендую. Более простого и удобного инструмента мне не попадалось, хотя я много перепробовал подобных вещей. Устанавливаем iftop на сервер:

```
# apt-get -y install iftop
```



Настройка и обновление времени в Debian

Теперь проверим установленный часовой пояс, время и включим автоматическую синхронизацию времени с удаленного сервера. Узнать дату, время, часовой пояс можно командой **date**:

```
# date
```

```
Fri Jan 29 00:01:37 MSK 2016
```

Если все указано верно, то менять ничего не нужно. Если же у вас неправильное время или указан часовой пояс не соответствующий вашему, то настроить это можно следующим образом. Сначала обновим часовые пояса:

```
# apt-get -y install tzdata
```

Теперь выберем правильный часовой пояс с помощью команды:

```
# dpkg-reconfigure tzdata
```

Выбирая соответствующие пункты визарда, указываете свой часовой пояс.

Дальше синхронизируем время с сервером времени в интернете. Для разовой или ручной синхронизации понадобится отдельная утилита**.** Установим **ntpdate** на сервер:

```
# apt-get install ntpdate
```

И синхронизируем время:

```
# ntpdate-debian

29 Jan 00:02:30 ntpdate[3172]: adjust time server 85.114.26.194 offset -0.001045 sec
```

Если получаете ошибку:

```
29 Jan 00:02:09 ntpdate[3123]: the NTP socket is in use, exiting
```

Значит у вас уже работает служба ntp. Ее нужно остановить и обновить время вручную. Хотя если она работает, то у вас и так должно быть все в порядке.

Для того, чтобы время автоматически синхронизировалось без вашего участия с определенной периодичностью, используется инструмент **ntp**. Установим его:

```
# apt-get -y install ntp
```

После установки он сам запустится и будет автоматически синхронизировать часы сервера. Проверим, запустился ли сервис ntpd:

```
# netstat -tulnp | grep ntp

udp 0 0 192.168.1.24:123 0.0.0.0:* 3119/ntpd
udp 0 0 127.0.0.1:123 0.0.0.0:* 3119/ntpd
udp 0 0 0.0.0.0:123 0.0.0.0:* 3119/ntpd
udp6 0 0 ::1:123 :::* 3119/ntpd
udp6 0 0 fe80::215:5dff:fe01:123 :::* 3119/ntpd
udp6 0 0 :::123 :::* 3119/ntpd
```

Настройка firewall (iptables) в Debian 8

В качестве firewall по-умолчанию используется **iptables**, его и будем настраивать. Изначально фаервол полностью открыт и пропускает весь трафик. Проверить список правил iptables можно следующей командой:

```
# iptables -L -v -n

Chain INPUT (policy ACCEPT 5851 packets, 7522K bytes)
pkts bytes target prot opt in out source destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination

Chain OUTPUT (policy ACCEPT 3328 packets, 452K bytes)
```

```
pkts bytes target prot opt in out source destination
```

Создадим файл с правилами iptables:

```
# mcedit /etc/iptables.sh
```

Очень подробно вопрос настройки iptables я рассмотрел отдельно, очень рекомендую ознакомиться. Хотя в примере другая ОС linux, принципиальной разницы нет, настройки iptables абсолютно одинаковые.

Добавляем набор простых правил для базовой настройки. Все необходимое вы потом сможете сами открыть или закрыть по аналогии с существующими правилами:

```
#!/bin/bash
#
# Объявление переменных
export IPT="iptables"

# Активный сетевой интерфейс
export WAN=eth0
export WAN_IP=192.168.1.24

# Очистка всех цепочек iptables
$IPT -F
$IPT -F -t nat
$IPT -F -t mangle
$IPT -X
$IPT -t nat -X
$IPT -t mangle -X

# Установим политики по умолчанию для трафика, не соответствующего ни одному из правил
$IPT -P INPUT DROP
$IPT -P OUTPUT DROP
```

```
$IPT -P FORWARD DROP

# разрешаем локальный траффик для loopback
$IPT -A INPUT -i lo -j ACCEPT
$IPT -A OUTPUT -o lo -j ACCEPT

# разрешаем пинги
$IPT -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

# Разрешаем исходящие соединения самого сервера
$IPT -A OUTPUT -o $WAN -j ACCEPT

# Состояние ESTABLISHED говорит о том, что это не первый пакет в соединении.
# Пропускать все уже инициированные соединения, а также дочерние от них
$IPT -A INPUT -p all -m state --state ESTABLISHED,RELATED -j ACCEPT
# Пропускать новые, а так же уже инициированные и их дочерние соединения
$IPT -A OUTPUT -p all -m state --state ESTABLISHED,RELATED -j ACCEPT
# Разрешить форвардинг для уже инициированных и их дочерних соединений
$IPT -A FORWARD -p all -m state --state ESTABLISHED,RELATED -j ACCEPT

# Включаем фрагментацию пакетов. Необходимо из за разных значений MTU
$IPT -I FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --clamp-mss-to-pmtu

# Отбрасывать все пакеты, которые не могут быть идентифицированы
# и поэтому не могут иметь определенного статуса.
$IPT -A INPUT -m state --state INVALID -j DROP
$IPT -A FORWARD -m state --state INVALID -j DROP

# Приводит к связыванию системных ресурсов, так что реальный
# обмен данными становится не возможным, обрубаем
```

```
$IPT -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
$IPT -A OUTPUT -p tcp ! --syn -m state --state NEW -j DROP

# Открываем порт для ssh (!!!не забудьте указать свой порт, который вы изменили ранее!!!)
$IPT -A INPUT -i $WAN -p tcp --dport 22 -j ACCEPT
# Открываем порт для DNS
$IPT -A INPUT -i $WAN -p udp --dport 53 -j ACCEPT
# Открываем порт для NTP
$IPT -A INPUT -i $WAN -p udp --dport 123 -j ACCEPT

# Записываем правила в файл
/sbin/iptables-save > /etc/iptables_rules
```

Даем файлу права на запуск:

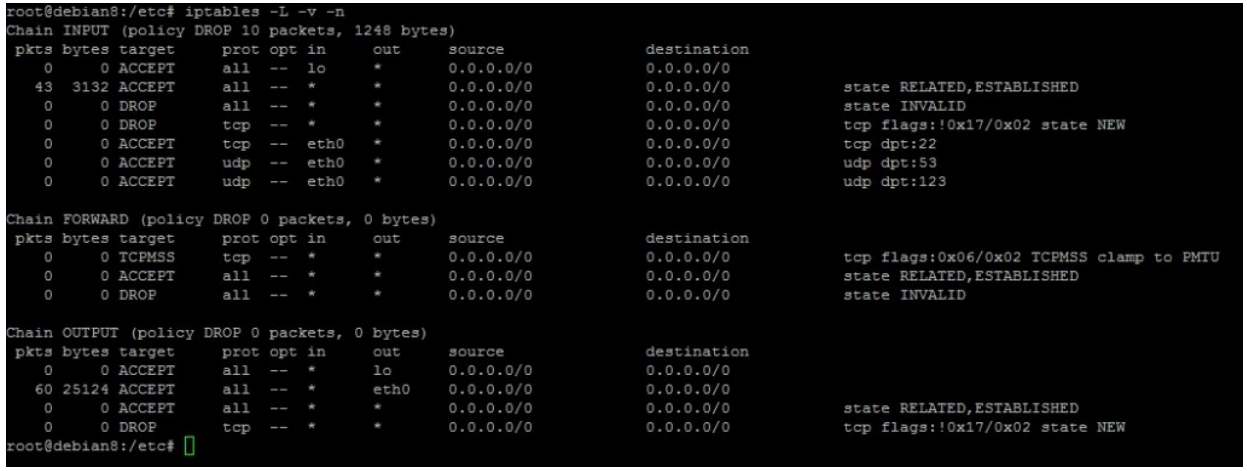
```
# chmod 0740 /etc/iptables.sh
```

Запускаем скрипт:

```
sh /etc/iptables.sh
```

Проверяем правила:

```
# iptables -L -v -n
```



Проверяем, что правила записались в файл */etc/iptables_rules*. Если их там нет, то записываем их вручную.

```
# /sbin/iptables-save > /etc/iptables_rules
```

Правила применились и произошла их запись в файл /etc/iptables_rules. Теперь нужно сделать так, чтобы они применялись при загрузке сервера. Для этого делаем следующее. Открываем файл /etc/network/interfaces и добавляем в него строку pre-up iptables-restore < /etc/iptables_rules Должно получиться вот так:

```
# cat /etc/network/interfaces

allow-hotplug eth0
iface eth0 inet dhcp
pre-up iptables-restore < /etc/iptables_rules
```

Настройка логов cron

По-умолчанию, в Debian нет отдельного лог файла для событий cron, они все сыпятся в общий лог. Лично мне это не очень нравится, я предпочитаю выводить эти события в отдельный файл. Об этом я написал отдельно — вывести логи cron в отдельный файл. Рекомендую пройти по ссылке и настроить, если вам это необходимо. Там очень кратко и только по делу, не буду сюда копировать эту информацию.

Установка и настройка screen

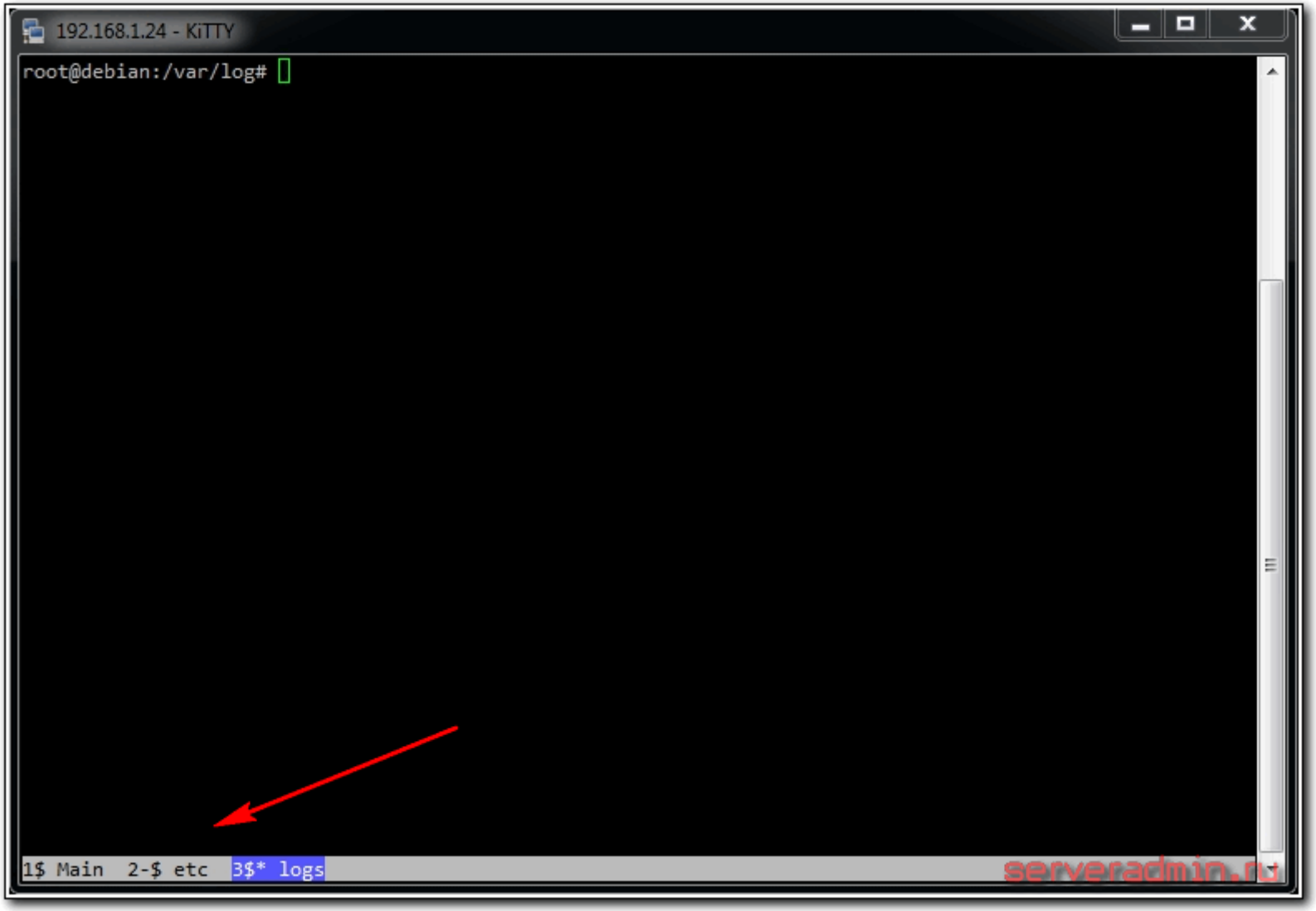
Я привык в своей работе пользоваться консольной утилитой screen. Изначально она задумывалась как инструмент, который позволяет запустить что-то удаленно в консоли, отключиться от сервера и при этом все, что выполняется в консоли продолжит свою работу. Вы сможете спокойно вернуться в ту же сессию и продолжить работу.

Первое время я именно так и использовал эту утилиту. Редко ее запускал, если не забывал, когда выполнялся какой-то длительный процесс, который жалко было прервать из-за случайного обрыва связи или необходимости отключить ноутбук от сети и куда-то переместиться.

Позже я решил подробнее ознакомиться с этим инструментом и обнаружил, что там есть несколько удобных моментов, которые можно использовать в ежедневной работе. Вот как использую утилиту screen я. При подключении к серверу у меня запускается screen с тремя окнами 1, 2, 3. Первое окно автоматически переходит в каталог /, второе в /etc, третье в /var/log. Я осмысленно назвал эти окна: Main, etc, logs соответственно. Внизу находится строка состояния, в которой отображен список всех открытых окон и подсвечено активное окно.

С помощью горячих клавиш я очень быстро переключаюсь между окнами в случае необходимости. Когда я не использовал screen, для быстрого перемещения между разными окнами я открывал несколько сессий ssh и через виндовый alt+tab переключался. Потом я стал использовать менеджер сессий mRemoteNG и там альтабаться уже не получалось. Долго страдал и переключал окна мышкой. Это было очень неудобно. Screen решил все мои проблемы по работе с разными сессиями на сервере. Теперь все реально в одной ssh сессии и получается очень быстро переключаться между окнами.

Вот как выглядит мое рабочее окно ssh подключения:


192.168.1.24 - KiTTY
root@debian:/var/log#
1$ Main 2-$ etc 3$* logs
serveradmin.ru

Переключаюсь между окнами с помощью стандартных горячих клавиш screen: ctrl+a 1, ctrl+a 2, ctrl+a 3. Я специально изменил нумерацию, чтобы она начиналась не с 0 по-дефолту, а с 1. Так удобнее на клавиатуре переключать окна. Кнопка 0 находится слишком далеко от 1 и 2.

Чтобы настроить такую же работу screen, как у меня достаточно выполнить несколько простых действий. Сначала устанавливаем screen:

```
# apt-get install -y screen
```

Создаем в каталоге */root* конфигурационный файл *.screenrc* следующего содержания:

```
# mcedit /root/.screenrc
```

```
#Выводим строку состояния
hardstatus alwayslastline "%-Lw%{= BW}%50>%n%f* %t%{-}%+Lw%<"

# Добавляем некоторые настройки
startup_message off
defscrollback 1000
defutf8 on
shell -$SHELL

# Создаем несколько окон
chdir
screen -t Main 1
chdir /etc
screen -t etc 2
chdir /var/log
screen -t logs 3

# Активное первое окно после запуска
select 1
```

Для знакомства с настройками, горячими клавишами и вариантами применения утилиты screen можно по адресу http://itman.in/ssh-screen/ Мне помог этот материал. Написано кратко, по делу и доходчиво.

Заключение

Теперь можно перезагрузить сервер и проверить, все ли в порядке. У меня все в порядке, проверил :) На этом базовая настройка сервера debian окончена. Можно приступать к конфигурации различных сервисов, под которые он настраивался. Об этом я расскажу в отдельных статьях.

Дополнительные материалы по Debian

Рекомендую полезные материалы по Debian:

Настройки системы

• Установка
• **Базовая настройка**
• Настройка сети
• Обновление 8 до 9
• Обновление 7 до 8
• Включение логов cron

Подробная установка Debian 9 Stratch с помощью графического инсталлятора со скриншотами и пояснениями к каждому пункту установщика.
Базовая настройка сервера Debian. Приведены практические советы по улучшению безопасности и удобства администрирования.
Подробное описание настройки сети в Debian - задать ip адрес, dhcp, отключить ipv6, dns, hostname, статические маршруты и др.
Обновление предыдущей версии Debian 8 Jessie до последней Debian 9 Stratch. Подробная инструкция с описанием по каждому этапу обновления.
Обновление версии Debian 7 wheezy до Debian 8 Jessie. Подробная инструкция с описанием по каждому этапу обновления.
Включение записи логов cron в Debian в отдельный файл и настройка ротации этого файла. Отключение логов в syslog.

Настройка программных комплексов

• Proxmox
• Шлюз в интернет
• Установка Asterisk
• Asterisk+Freepbx
• PostgreSQL для 1C
• Настройка pptp

Подробное описание установки гипервизора proxmox на raid1 mdadm на базе операционной системы Debian 8. Приведены практические советы по настройке.
Настройка интернет шлюза на Debian. Включает в себя настройку iptables, nat, dhcp, dns, iftop.
Чистая установка Asterisk 13 на сервер под управлением Debian 8. Никаких дополнений и GUI, только vanilla asterisk.
Установка Freepbx 12 и Asterisk 13 на сервер под управлением Debian/Ubuntu. Подробное описание и разбор ошибок установки.
Рассказ об установке и небольшой настройке сервера бд postgresql для работы с базами 1С. Задача не сложная, но есть небольшие нюансы как по настройке, так и по выбору дистрибутива.
Описание установки и настройки pptp сервера в Debian с передачей статических маршрутов клиенту для организации доступа к ресурсам сети.

Разное

• Бэкап с помощью rsync
• Тюнинг postgresl для 1С

Подробное описание настройки бэкапа с помощью rsync на примере скрипта инкрементного архива на системе Centos, Debian, Ubuntu, Windows.
Ускорение работы 1С с postgresql и диагностика проблем производительности