В текущей статье я расскажу об очень популярной теме, с которой начинается практически любая первоначальная работа с сервером. Настройка основных параметров сети в Debian — ip адрес, dhcp, ipv6, dns, hostname, статические маршруты, сетевые карты и другие network параметры. Двигаться по теме будем шаг за шагом от простого к сложному, разбирая все нюансы по порядку и отвечая на наиболее популярные вопросы.

Содержание:

Сетевые настройки на сервере Debian

Вопрос настройки сети я уже кратко поднимал в теме начальной настройки Debian. Я рекомендую с ней ознакомиться и выполнить некоторые подготовительные действия, чтобы было удобнее работать далее. Сейчас мы подробно разберем все наиболее значимые нюансы сетевых настроек, которые могут пригодиться в повседневной работе.

Первоначальная настройка сети начинается во время установки сервера. Если у вас есть сетевой интерфейс и dhcp сервер в сети, то сеть сконфигурируется автоматически на основе полученных настроек и будет готова к работе. В последствии вы можете выполнить настройку сети в Debian через консоль с помощью программ **ip** или **ifconfig**. Наиболее популярным и современным средством на текущий момент является ip, поэтому в дальнейшем рассмотрим вопрос конфигурации сетевых интерфейсов с ее помощью. Про ifconfig тоже не забудем. Рассмотрим ее позже отдельно.

Отдельно стоит такой инструмент управления сетевыми подключениями как **Network manager**. Он используется в сочетании с графическими оболочками, которых на сервере обычно нет, поэтому вопрос его настройки я не буду рассматривать. Мне просто не на чем это делать, да и не вижу смысла.

Есть 2 различные возможности изменить сеть в Debian:

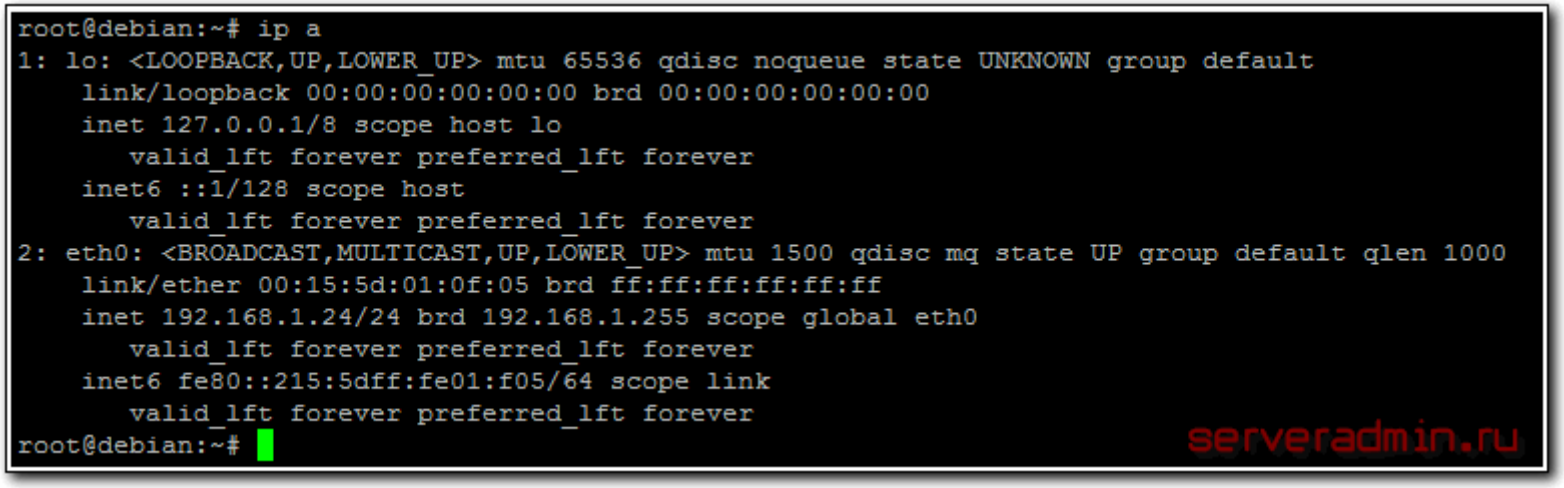
1. Настройка сети из консоли с помощью указанных ранее консольных программ.
2. С помощью редактирования конфигурационного файла сетевых интерфейсов */etc/network/interfaces*.

Мы рассмотрим оба этих варианта. Вводная теоретическая часть окончена, приступаем к практике.

Настройка статического IP

Вы установили сервер и во время установки указали какие-то сетевые параметры, или не указали, не важно. Но сейчас вы решили, что вам нужно назначить статический ip (static ip) адрес. Воспользуемся для этого утилитой **ip**. Сначала посмотрим список всех сетевых интерфейсов:

```
# ip a
```

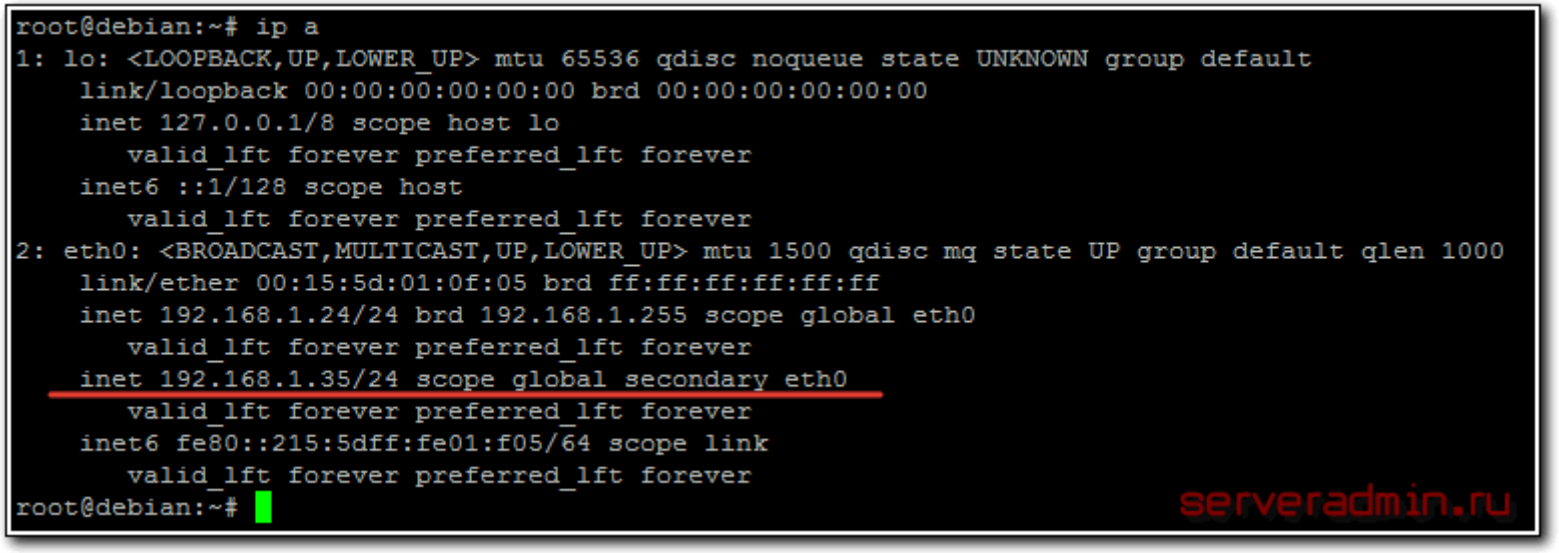


У меня в системе 1 сетевой интерфейс **eth0** и он каким-то образом уже сконфигурирован. Назначим ему еще один статический адрес:

```
# ip addr add 192.168.1.35/24 dev eth0
```

Этот адрес будет добавлен к уже существующему адресу. Проверим это:

```
# ip a
```



Теперь сервер будет доступен по обоим адресам. Текущая настройка сохранится только до перезагрузки сервера. Чтобы она применялась во время загрузки, нужно либо каким-то образом добавить команду на изменение настроек в автозагрузку, например в */etc/rc.local*, но это будет костыль. Правильнее отредактировать файл, который отвечает за сетевые настройки в debian — */etc/network/interfaces*. Для того, чтобы назначить постоянный статический ip адрес, его нужно привести к следующему виду:

```
# mcedit /etc/network/interfaces
```

```
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
```

```
address 192.168.1.35
gateway 192.168.1.1
netmask 255.255.255.0
```

auto	указанный интерфейс необходимо запускать автоматически при загрузке системы
iface	интерфейс eth0 находится в диапазоне адресов IPv4 со статическим ip
address	статический ip адрес
gateway	шлюз по-умолчанию
netmask	маска сети

Для проверки перезагружаем сервер и смотрим, все ли в порядке с настройкой статического ip адреса. Если вам нужно сменить ip адрес в debian, то вы можете временно изменить его с помощью команды ip, либо постоянно, отредактировав параметр **address** в файле сетевых настроек *interfaces*.

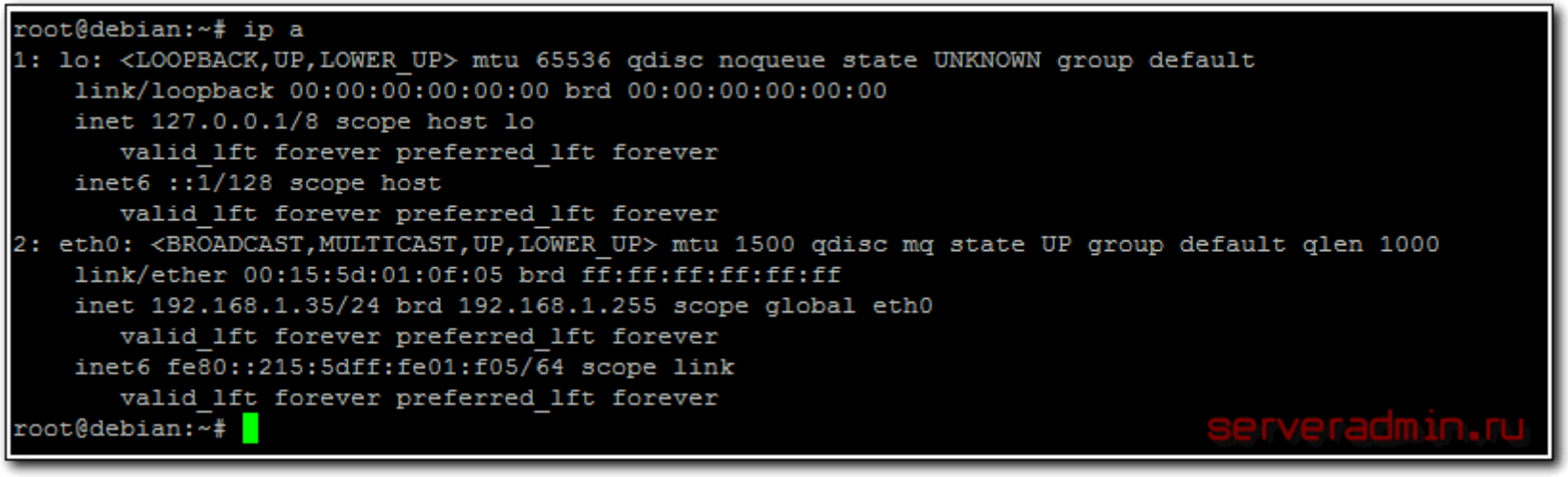
Получение сетевых настроек по DHCP

Теперь рассмотри вариант, когда вам необходимо получить динамический ip адрес в Debian. Здесь по аналогии с предыдущими настройками можно пойди двумя путями:

1. Получить ip адрес по dhcp в консоли с помощью программы **dhclient**, который будет работать до перезагрузки.
2. Отредактировать файл конфигурации сетевых интерфейсов.

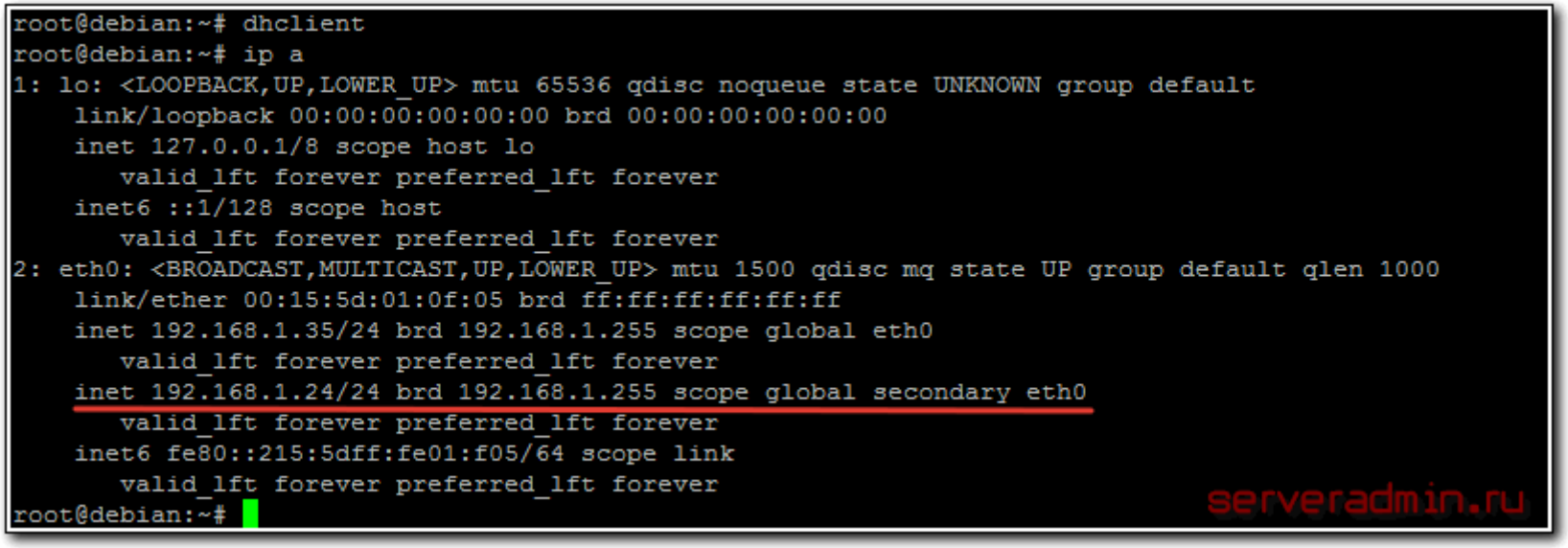
Смотрим снова на текущую конфигурацию сети:

```
# ip a
```



Выполним команду на получение ip адреса по dhcp и проверим сеть:

```
# dhclient
# ip a
```



В качестве второго ip мы получили адрес от dhcp сервера. Если у вас несколько сетевых интерфейсов, то необходимо добавлять название после команды, например вот так:

```
# dhclient eth0
```

Для того, чтобы сбросить адрес, нужно воспользоваться следующим ключом:

```
# dhclient -r
```

Обращаю внимание, что эта команда сбросит все ip адреса интерфейса, в том числе и статические. Чтобы снова запросить адрес, нужно выполнить предыдущую команду на получение ip с указанием сетевого интерфейса eth0.

Все изменения, сделанные в консоли после перезагрузки исчезнут. Чтобы их сохранить, приведем файл */etc/network/interfaces* к следующему виду:

```
# mcedit /etc/network/interfaces
```

```
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp
```

Установка шлюза по-умолчанию (default gateway)

Теперь разберемся со шлюзом по-умолчанию. В предыдущих примерах со статическим ip адресом и настройками по dhcp у нас не было необходимости указывать отдельно default gateway. Мы его устанавливали вместе с остальными настройками. Чтобы посмотреть установленный по-умолчанию шлюз в debian, можно воспользоваться следующей командой в консоли:

```
# ip r sh
```

```
default via 192.168.1.1 dev eth0
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.24
```

Это шлюз по-умолчанию (default gateway). Можно воспользоваться другими, более популярными и привычными командами:

```
# route
# netstat -nr
```

Все они показывают одни и то же. Если нам нужно сменить default gateway, то сначала надо удалить текущий шлюз, а потом назначить новый.

```
# route del default gw 192.168.1.1
```

```
# route add default gw 192.168.1.50
```

Проверяем, что получилось:

```
# ip r sh
```

```
default via 192.168.1.50 dev eth0
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.24
```

Все получилось. Эта настройка будет действовать до перезагрузки. Чтобы ее сохранить, либо меняйте конфигурационный файл *interfaces*, либо при необходимости придумывайте что-то еще.

Как указать DNS сервер

Остался последний из основных сетевых параметров — **dns** сервер. С ним в debian и ubuntu есть определенная путаница. Традиционно в linux для установки dns серверов используется файл */etc/resolv.conf*. Но в какой-то момент в этих дистрибутивах появилась программа **resolvconf**, которая стала управлять настройками dns в системе. В итоге, файл *resolv.conf* постоянно перезаписывается этой программой. Нужна она в первую очередь для систем, где dns сервера постоянно меняются. Она следит за их изменением и корректно передает информацию об изменении программам, для которых это важно. Я лично не знаю таких программ и мне не приходилось сталкиваться с ситуацией, когда это было необходимо.

Если я не ошибаюсь, то в минимальной конфигурации debian программа resolvconf не устанавливается, а вот в ubuntu она стоит. Проверить наличие программы очень просто:

```
# dpkg -l | grep resolvconf
```

Если в выводе пусто, значит ее нет. Тогда все очень просто. Для того, чтобы указать dns сервер, достаточно его записать в файл */etc/resolv.conf* в следующем виде:

```
# mcedit /etc/resolv.conf
```

```
nameserver 192.168.1.1
nameserver 77.88.8.1
nameserver 8.8.8.8
```

192.168.1.1	локальный dns сервер
77.88.8.1	публичный сервер Яндекса
8.8.8.8	публичный сервер Гугла

Я на всякий случай указал локальный сервер и 2 внешних. Если у вас стоит resolvconf, то в случае ненадобности, удалите его командой:

```
# apt-get remove resolvconf
```

После этого сервер надо перезагрузить и удалить сломавшуюся символьную ссылку /etc/resolv.conf, а вместо нее создать файл с нужным содержанием, которое я привел выше.

Если же вам по какой-то причине необходима указанная выше программа, она у вас стоит и вы не хотите ее удалять, то адрес dns сервера необходимо указать в файле */etc/network/interfaces*, добавив к параметрам интерфейса еще один:

```
dns-nameservers 192.168.1.1 77.88.8.1 8.8.8.8
```

Этот параметр нужно установить сразу после указания шлюза gateway. Несколько адресов разделяются пробелом.

Изменить hostname (имя хоста)

Во время установки debian вы указывали имя хоста. Посмотреть его текущее значение можно в консоли:

```
# hostname
debian
```

Это значение записано в файле */etc/hostname*. Есть 2 способа изменить hostname в debian:

1. Простой и быстрый с помощью консольной команды. Результат работает только до перезагрузки компьютера. Потом вернется старое имя.
2. С помощью изменения конфигурационного файла результат сохраняется и после перезагрузки. Чтобы сразу применить изменение, потребуется выполнить системный скрипт.

Для первого способа достаточно в консоли ввести команду:

```
# hostname debian8
```

Теперь проверим, что получилось:

```
# hostname
debian8
```

Имя хоста изменилось, но в файле */etc/hostname* по-прежнему указано прошлое значение. После перезагрузки hostname снова примет старое значение debian. Чтобы сделать постоянное изменение, необходимо ввести новое значение в файл вручную:

```
# mcedit /etc/hostname
debian8
```

Для применения изменения без перезагрузки сервера, выполните системный скрипт:

```
/etc/init.d/hostname.sh
```

Теперь проверьте текущее значение имени хоста. Оно изменится на то, что записано в файле.

2 и более IP на одном интерфейсе

Достаточно часто возникают ситуации, когда необходимо назначить несколько ip на одном интерфейсе. Сделать это очень просто. В самом начале я показал, как быстро через консоль в Debian можно назначить несколько ip с помощью программы. Теперь сделаем так, чтобы эти настройки сохранялись после перезагрузки. Для этого редактируем наш любимый и ненаглядный */etc/network/interfaces*:

```
# mcedit /etc/network/interfaces
```

```
source /etc/network/interfaces.d/*

auto lo
iface lo inet loopback

auto eth0
iface eth0 inet dhcp

auto eth0:1
iface eth0:1 inet static
address 192.168.1.35
netmask 255.255.255.0
```

С такими настройками я получу 2 ip адреса на сетевом интерфейсе:

1. Первый от dhcp сервера.
2. Второй адрес на этой же сетевой карте будет указан статически.

Вы таким образом можете добавить сколько вам необходимо адресов, используя различные параметры.

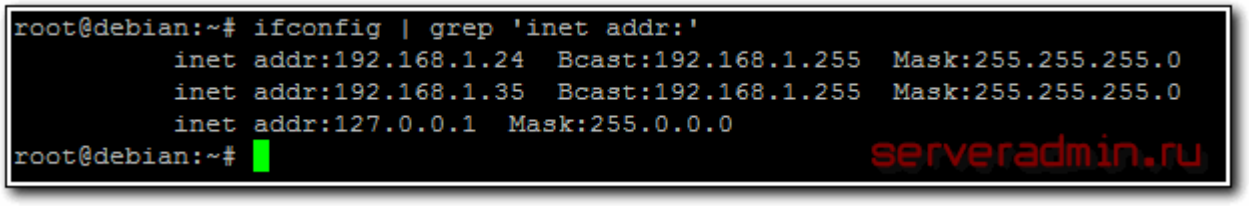
Как быстро узнать ip адрес сервера в Debian

Часто возникает вопрос, как в debian быстро узнать или проверить ip адреса, назначенные серверу. Выше я уже приводил пример простой команды, которая показывает все сетевые параметры интерфейса. Сейчас рассмотрим несколько вариантов, которые наглядно покажут только ip адреса, без лишней информации. Для начала воспользуемся уже известной командой ip, немного подрезав ее вывод:

```
# ip a | grep inet | awk '{print $2}'
```

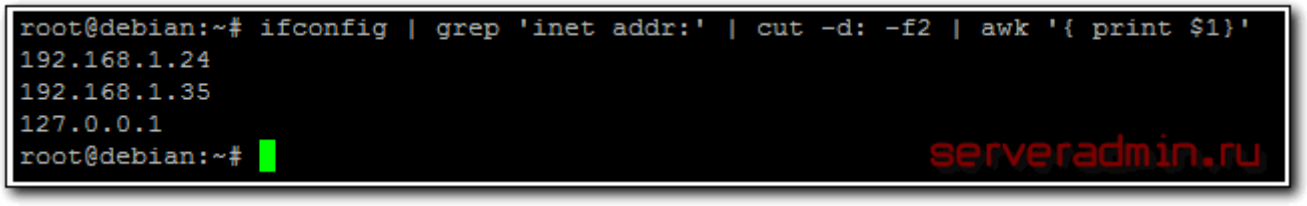
Вы узнаете все ip адреса сервера, в том числе и ipv6. То же самое, только без ipv6 можно увидеть другой командой:

```
# ifconfig | grep 'inet addr:'
```



А если совсем заморочиться и обрезать все лишнее, то можно вывести только список ip адресов:

```
# ifconfig | grep 'inet addr:' | cut -d: -f2 | awk '{ print $1}'
```

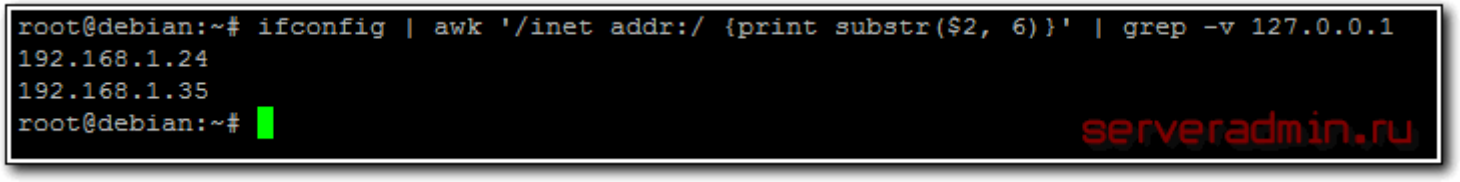


Аналогичный вывод будет и в таком случае:

```
# ifconfig | awk '/inet addr:/ {print substr($2, 6)}'
```

Можно убрать локальный интерфейс, чтобы глаза не мозолил, тогда вообще все наглядно:

```
# ifconfig | awk '/inet addr:/ {print substr($2, 6)}' | grep -v 127.0.0.1
```



Думаю, этих вариантов достаточно, чтобы проверить все ip адреса, отрезав лишнее.

Static routes (статические маршруты)

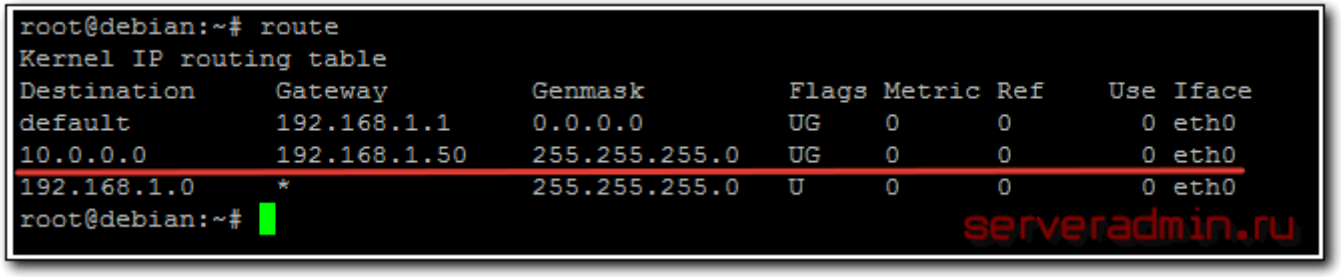
Следующим важным элементом настройки сети является управление статическими маршрутами (static routes): добавление (route add) и удаление (route del). Вновь воспользуемся консольной командой ip. Для того, чтобы добавить маршрут в debian, достаточно ввести в консоли:

```
# ip route add 10.0.0.0/24 via 192.168.1.50
```

10.0.0.0	адрес подсети, для которой создаем отдельный маршрут в обход шлюза по-умолчанию
/24	маска подсети
192.168.1.50	адрес шлюза, который будет роутить трафик в указанную подсеть

Проверяем таблицу маршрутов:

```
# route
```



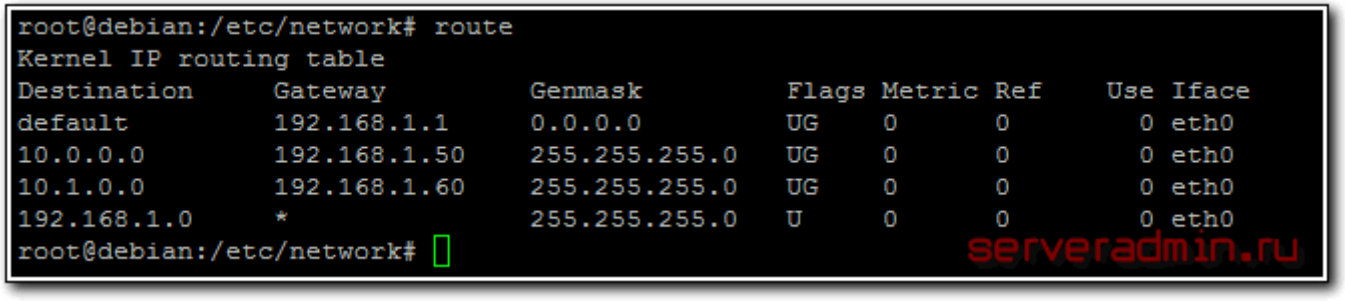
Чтобы удалить маршрут, выполняем команду:

```
# ip route delete 10.0.0.0/24
```

После перезагрузки все маршруты, добавленные таким способом, исчезнут. Чтобы статический маршрут в Debian сохранялся после перезагрузки, опять редактируем *interfaces*, добавляя в описание того интерфейса, к которому будут относиться маршруты, следующие строки:

```
post-up route add -net 10.0.0.0 netmask 255.255.255.0 gw 192.168.1.50
post-up route add -net 10.1.0.0 netmask 255.255.255.0 gw 192.168.1.60
pre-down route del -net 10.0.0.0 netmask 255.255.255.0 gw 192.168.1.50
pre-down route del -net 10.1.0.0 netmask 255.255.255.0 gw 192.168.1.60
```

Мы добавили 2 статических маршрута. Перезагружаемся и проверяем.



Мой конфиг целиком стал выглядеть вот так:

```
source /etc/network/interfaces.d/*
auto lo
iface lo inet loopback
auto eth0
iface eth0 inet dhcp
post-up route add -net 10.0.0.0 netmask 255.255.255.0 gw 192.168.1.50
post-up route add -net 10.1.0.0 netmask 255.255.255.0 gw 192.168.1.60
pre-down route del -net 10.0.0.0 netmask 255.255.255.0 gw 192.168.1.50
pre-down route del -net 10.1.0.0 netmask 255.255.255.0 gw 192.168.1.60

auto eth0:1
iface eth0:1 inet static
address 192.168.1.35
netmask 255.255.255.0
gateway 192.168.1.35
```

С постоянными статическими маршрутами в debian разобрались, ничего сложного. Идем дальше.

Как выполнить перезапуск сети

Выше я везде предлагал перезагрузить сервер, чтобы проверить новые сетевые настройки. Не всегда это обязательно делать. Иногда достаточно просто перечитать сетевую конфигурацию. Для перезапуска сети в Debian можно воспользоваться командой:

```
# service networking restart
```

Она выполняется не мгновенно, обычно несколько секунд. Выполнить перезагрузку сети можно так же командой:

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

Она делает то же самое, но есть некая обратная связь о том, как все прошло. Если у вас все в порядке на сервере и нет сетевых ошибок, то рестарт сети можно спокойно делать удаленно по ssh. Вас даже не отключит от текущей сессии. Но на всякий случай я не рекомендую это делать, если у вас нет доступа к консоли сервера. Всякое может случиться, я бы не рисковал лишний раз. Настройку сети, как и фаервола, лучше не делать, не имея доступа к консоли сервера.

Настройка vlan в Debian

Для настройки vlan в Debian первым делом необходимо установить пакет vlan:

```
# apt-get -y install vlan
```

Теперь нужно разобраться с необходимым модулем ядра для vlan — **8021q**. Проверим, загружено ли оно в системе:

```
# lsmod | grep 8021q
```

Если в выводе пусто, а по-умолчанию там будет пусто, модуль не загружен. Загрузим его и проверим:

```
# modprobe 8021q
# lsmod | grep 8021q
8021q                  27844  0
garp                   13117  1 8021q
mrp                    17343  1 8021q
```

Все в порядке, vlan модуль загрузился. Добавим его в автозагрузку:

```
# echo 8021q >> /etc/modules
```

Теперь этот модуль будет загружаться при старте сервера. Создадим виртуальный интерфейс с vlan с id 1500:

```
# vconfig add eth0 1500
```

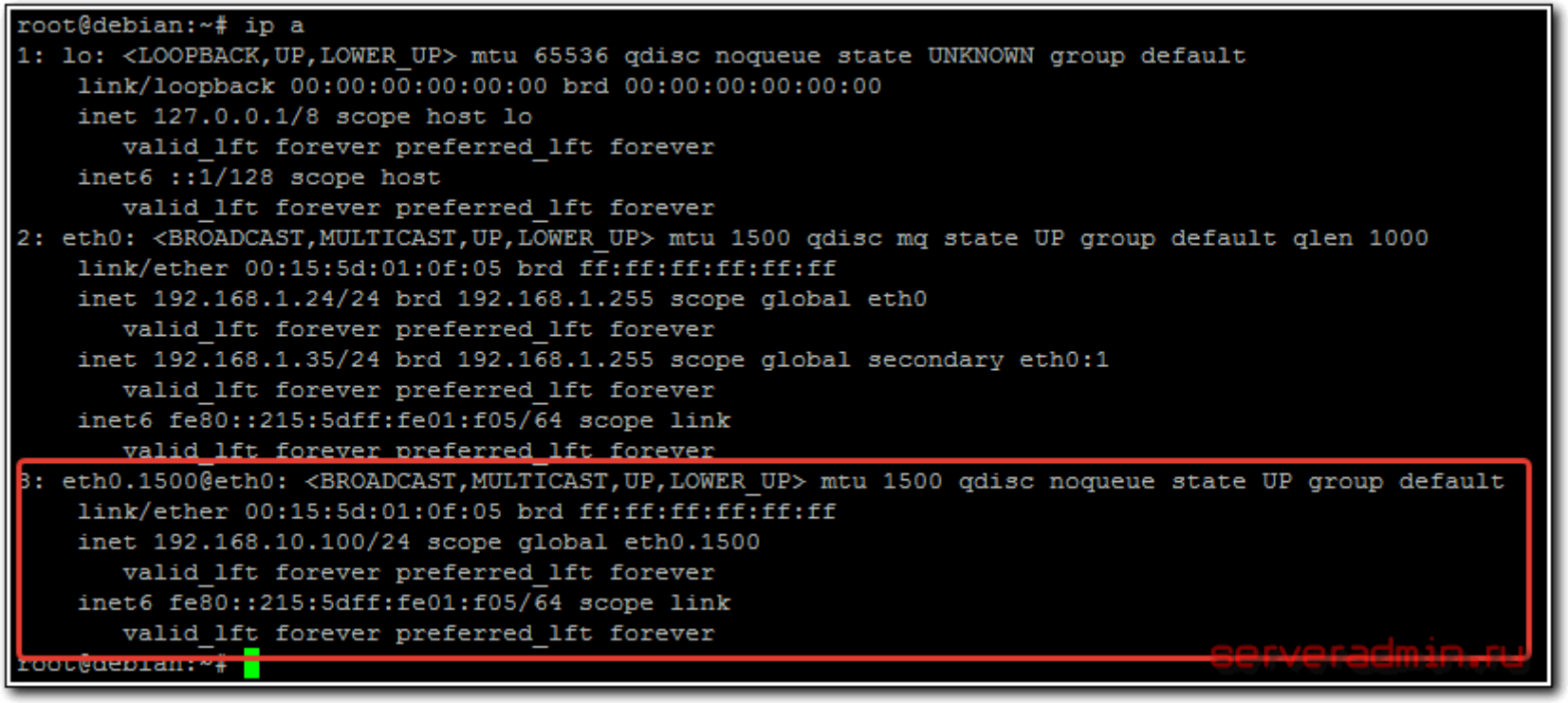
Назначим адрес новому интерфейсу и запустим его:

```
# ip addr add 192.168.10.100/24 dev eth0.1500
# ip link set eth0.1500 up
```

Можно в одну команду, с помощью ifconfig:

```
ifconfig eth0.1500 192.168.10.100 netmask 255.255.255.0 up
```

Смотрим, что получилось:



Мы выполнили настройку vlan интерфейса в Debian. Теперь сделаем так, чтобы после перезагрузки настройки сохранились. Для этого добавляем свойства vlan интерфейса в файл конфигурации сети, оставляя и не трогая то, что там уже есть:

```
# mcedit /etc/network/interfaces
```

```
auto eth0.1500
iface eth0.1500 inet static
address 192.168.10.100
netmask 255.255.255.0
```

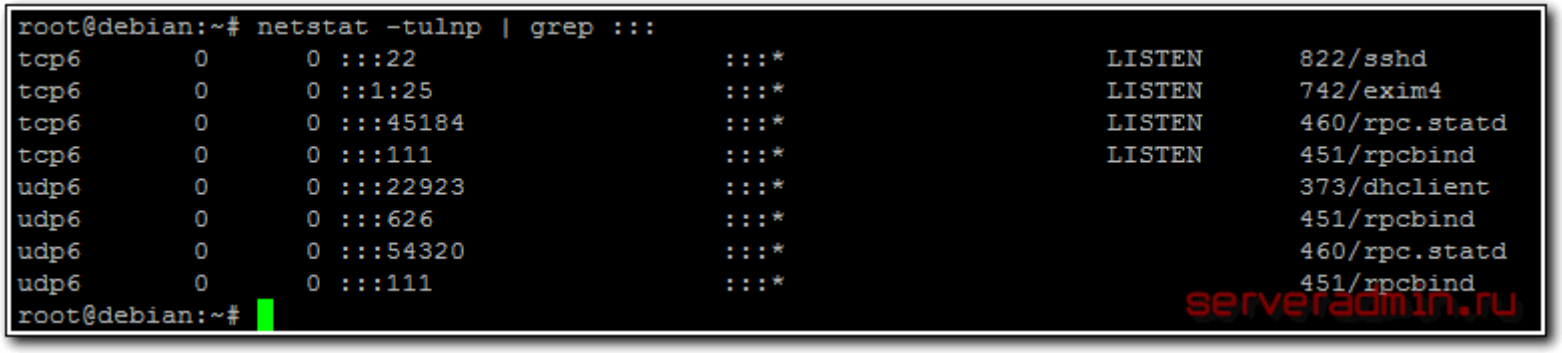
```
vlan_raw_device eth0
```

Перезагрузите сервер и проверьте, что получилось. Можно выполнить перезапуск сети, но если вы до этого все вручную через консоль сделали, то не поймете, сработали настройки из файла или нет. Таким образом очень просто и быстро настроить vlan на сервере с Debian.

Как отключить ipv6 в Debian

Пока еще новая версия протокола ip не получила широкого распространения, если он вам специально не нужен, ipv6 можно отключить. Прежде чем отключать, необходимо узнать, какие сервисы в настоящее время его используют и перенастроить их, отключив у них ipv6. Если этого не сделать, то в работе этих программ могут возникнуть ошибки. Скорее всего не критичные, но все равно, сделаем все аккуратно и правильно. Сначала проверим, что у нас работает на ipv6:

```
# netstat -tulnp | grep :::
```



На свежеустановленном сервере debian программы ssh, exim, dhclient и rpcbind используют ipv6. Отключим это. Начнем с ssh. Открываем файл */etc/ssh/sshd_config* и раскомментируем параметр ListenAddress:

```
# mcedit /etc/ssh/sshd_config
```

```
ListenAddress 0.0.0.0
```

Перезапускаем ssh:

```
# service sshd restart
```

Сделаем то же самое с exim4. Открываем файл конфигурации /etc/exim4/exim4.conf.template и в самом начале, после вступительных комментариев пишем:

```
# mcedit /etc/exim4/exim4.conf.template
```

```
disable_ipv6 = true
```

Перезапускаем exim:

```
# service exim4 restart
```

В dhclient для отключения ipv6 в конфиге убираем все параметры в запросе request, начинающиеся с dhcp6. Должно получиться вот так:

```
# mcedit /etc/dhcp/dhclient.conf
```

```
request subnet-mask, broadcast-address, time-offset, routers,
        domain-name, domain-name-servers, domain-search, host-name,
        netbios-name-servers, netbios-scope, interface-mtu,
        rfc3442-classless-static-routes, ntp-servers;
```

Перезапускаем сеть:

```
# service networking restart
```

Отключаем ipv6 в rpcbind. Открываем конфигурацию */etc/netconfig* и комментируем 2 строки с udp6 и tcp6:

```
# mcedit /etc/netconfig
```

```
#udp6       tpi_clts      v     inet6    udp     -       -
#tcp6       tpi_cots_ord  v     inet6    tcp     -       -
```

Перезапускаем службу rpcbind и nfs-common, которая от него зависит:

```
# service rpcbind restart
# service nfs-common restart
```

Проверяем, что у нас осталось:

```
# netstat -tulnp | grep :::
udp6       0      0 :::6909                 :::*                                1166/dhclient
```

dhclient почему-то остался висеть на ipv6 порту, но ладно, это не страшно, запрашивать по ipv6 он все равно ничего не будет. Теперь полностью отключаем ipv6 в Debian:

```
# mcedit /etc/sysctl.conf
```

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

Добавьте эти строки в любое место конфига, например, в самый конец. Перезапустим sysctl для применения настроек:

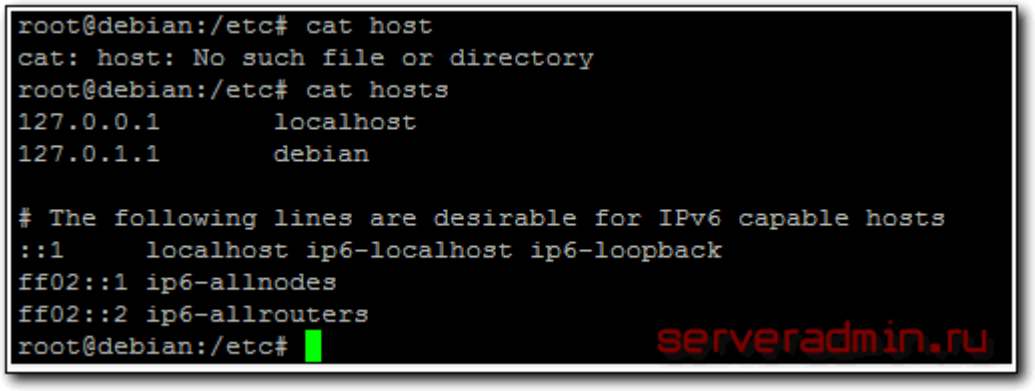
```
# sysctl -p
```

Проверяем свойства сетевых интерфейсов командами ifconfig и ip. Информации об ipv6 быть не должно, мы его полностью отключили.

Работа с файлом hosts

В папке */etc* любого linux дистрибутива, в том числе debian есть файл **hosts**. Разберемся немного что это за файл и для чего он нужен. По-умолчанию он выглядит следующим образом:

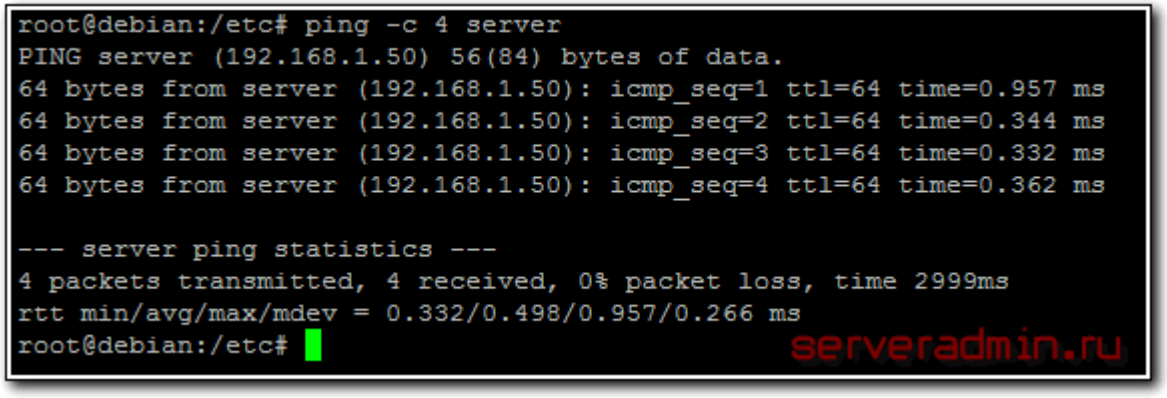
```
# cat /etc/hosts
```



Представим, что у нас в локальной сети есть некий сервер с именем server и ip 192.168.1.50 Мы хотим к нему обращаться по имени. Тогда добавляем запись в файл hosts:

```
192.168.1.50 server
```

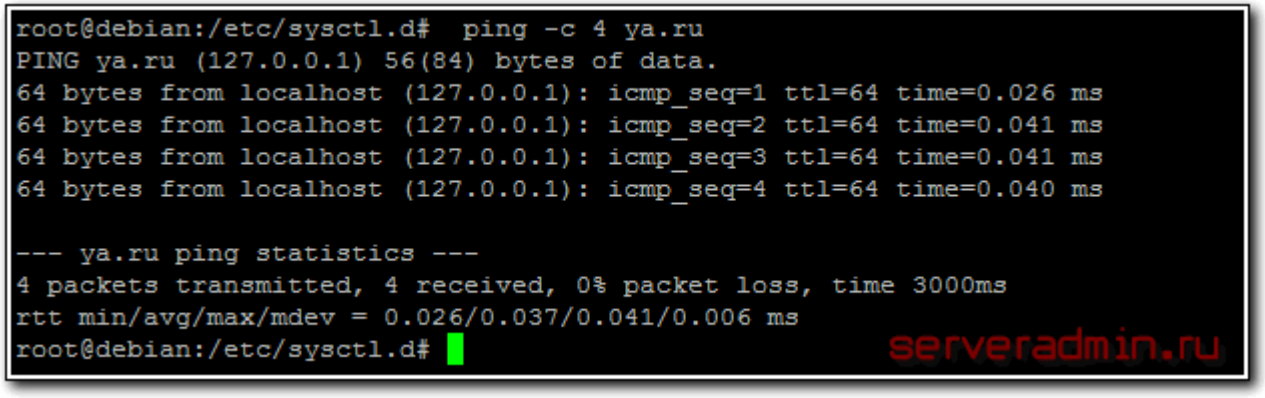
Теперь мы можем обращаться к серверу просто по имени server.



Сервер будет в первую очередь смотреть информацию в файле hosts, только потом в dns сервере. Например, если вы добавите в файл строку:

```
127.0.0.1 ya.ru
```

То обращаясь к адресу ya.ru будете попадать на локалхост:



Этот функционал может быть полезен для публикации в локальной сети каких-то внутренних сервисов, к которым доступ будет только из вашей сети. Вам нет необходимости использовать dns сервер, с файлом hosts настройка делается гораздо быстрее.

Настройка сетевой карты

Иногда возникают ситуации, когда необходимо настроить или изменить настройки сетевой карты. Сразу отмечу, что настраивать сетевую карту можно только на реальном железе. На виртуальном, скорее всего, ни одна из предложенных дальше команд не приведет к какому-нибудь результату. У виртуальных сетевых адаптеров просто нет настроек. Для начала посмотрим, какие сетевые карты есть на сервере:

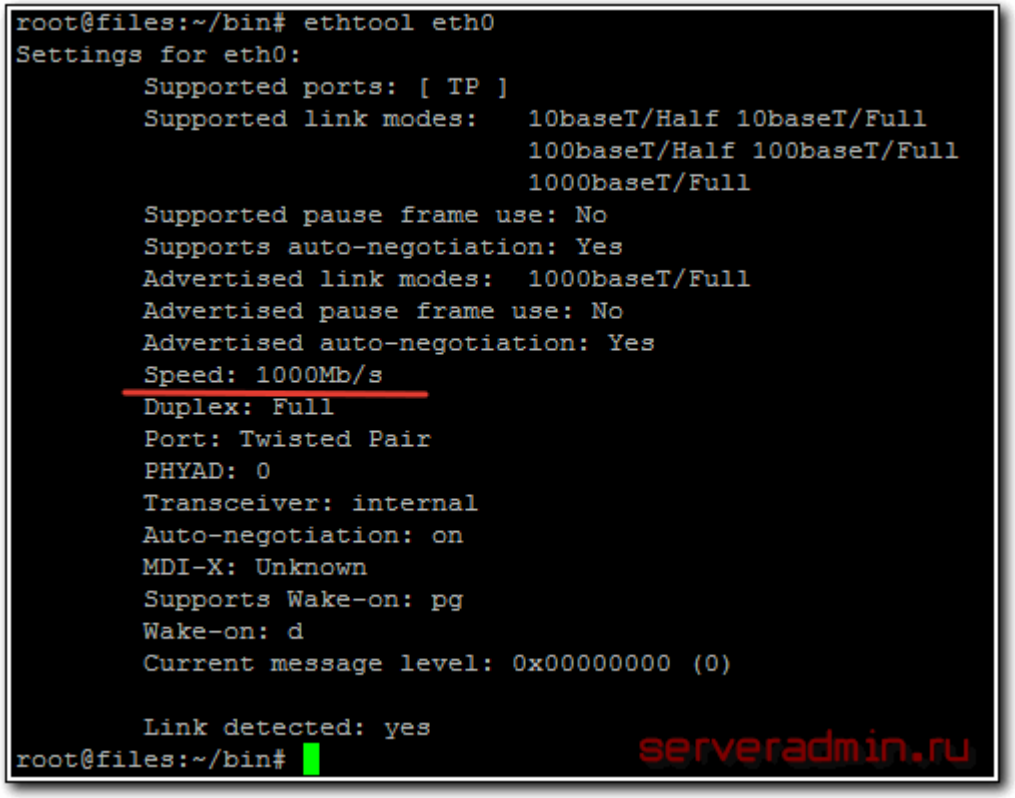
```
# lspci | grep Eth
01:00.0 Ethernet controller: Qualcomm Atheros AR8121/AR8113/AR8114 Gigabit or Fast Ethernet (rev b0)
```

В моем случае это единственная сетевая карта фирмы Qualcomm. Теперь установим утилиту **ethtool** для настройки сетевой карты:

```
# apt-get install ethtool
```

Посмотрим информацию о сетевой карте:

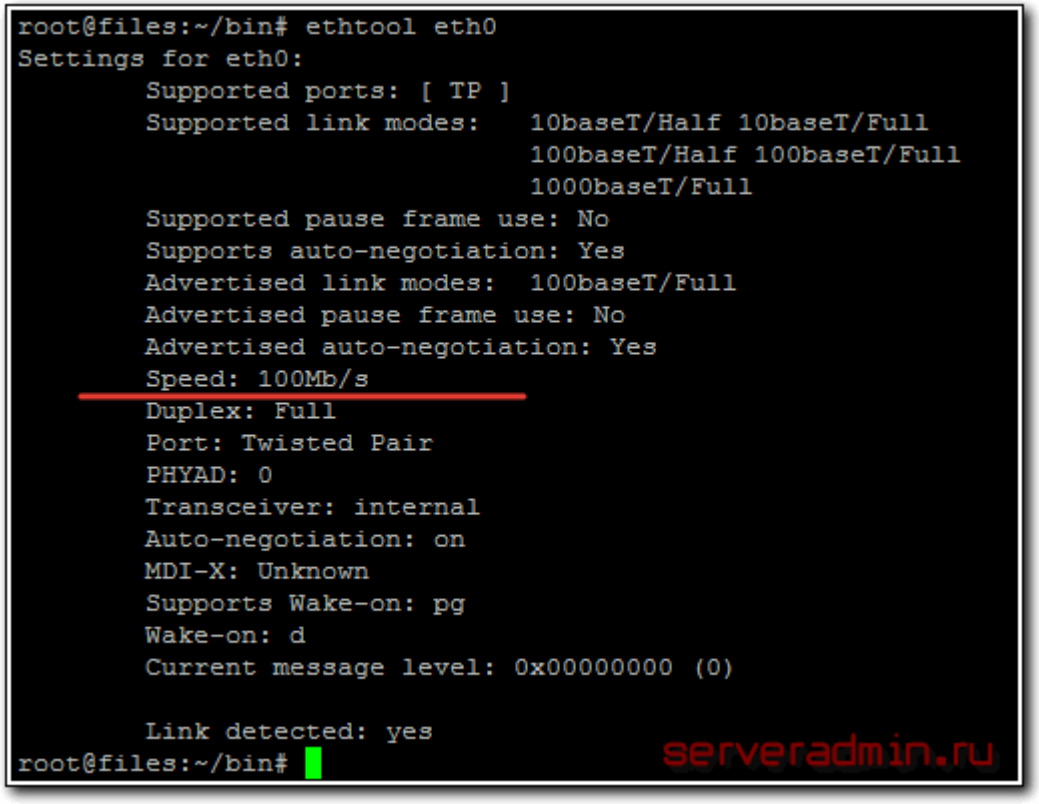
```
# ethtool eth0
```



Указана текущая скорость, на которой работает карточка. Ее можно сменить в случае необходимости:

```
# ethtool -s eth0 speed 100 duplex full
```

Этой командой можно изменить скорость сетевой карточки до 100Mb/s в случае, если там стояла другая скорость. Смотрим, что получилось:



Показал просто для примера, вряд ли кому-то понадобится уменьшать скорость. Чаще нужно выполнить обратное преобразование. У меня была ситуация, когда сетевая карта упорно не хотела работать на скорость 1Gb, хотя поддерживала такую работу, и свитч был гигабитный. Долго бился и пробовал различные утилиты для изменения скорости. Оказалось, что патч корд был 4-х жильный из комплекта какого-то роутера. Им воспользовались для подключения и даже не обратили внимание на то, что он не поддерживает работу по гигабиту.

У утилиты ethtool много параметров, с помощью которых можно настроить сетевую карту. Пример этих параметров можно посмотреть на сайте redhat.

Работа с утилитой ifconfig

В завершении сетевых настроек хотел немного поговорить об **ifconfig**. Ранее я упомянул, что более современным средством для настройки сети является утилита **ip**. В последнем релизе CentOS ifconfig вообще исключили из базовой установки, ее приходится устанавливать отдельно. IP объединяет в себе два функционала — настройка сетевых интерфейсов и маршрутизации. То есть по сути она заменяет ifconfig + route. В ней реализован функционал обоих программ.

Сам я привык к ifconfig, так как она есть не только в linux, но и в freebsd. Удобно использовать одно и то же средство во всех дистрибутивах. Но последнее время переучиваюсь на ip, так как надо идти в ногу со временем. Тенденция такова, что ip будут продвигать все сильнее и сильнее в силу его большей новизны и, наверное, удобства. Что касается удобства, лично я ничего не могу сказать, мне совершенно все равно, какую команду использовать:

```
ip addr add 192.168.1.35/24 dev eth0
```

или

```
ifconfig eth0 192.168.1.35 netmask 255.255.255.0
```

Делают они одно и то же. Остальные команды по конфигурированию сетевых интерфейсов тоже не сильно отличаются, просто пишутся немного по-разному. Вот пример работы с маршрутами:

```
ip route add 192.168.100.0/24 dev eth2
```

или

```
route add -net 192.168.100.0/24 dev eth2
```

В случае с ip мы работаем с одной командой, а не двумя. Чем пользоваться вам, выбирайте сами. Если не привыкли к чему-то конкретному, рекомендую использовать ip. Мне было бы любопытно узнать, кто, что использует. Оставьте свой комментарий на эту тему.

На этом я завершаю свой материал по теме настройки сети в debian. Я рассмотрел все наиболее значимые и необходимые параметры, с которыми

приходится сталкиваться во время конфигурирования сервера.

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 1С
• 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