

Сетевые настройки в CentOS 7

Обновлено: 18.10.2016 Написано: 23.08.2015 CentOS, Linux 21 комментариев 138,634 Views

В данном материале я затрону актуальную тему, с которой начинается практически любая первоначальная работа с сервером. Настройка основных параметров сети в CentOS — 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:

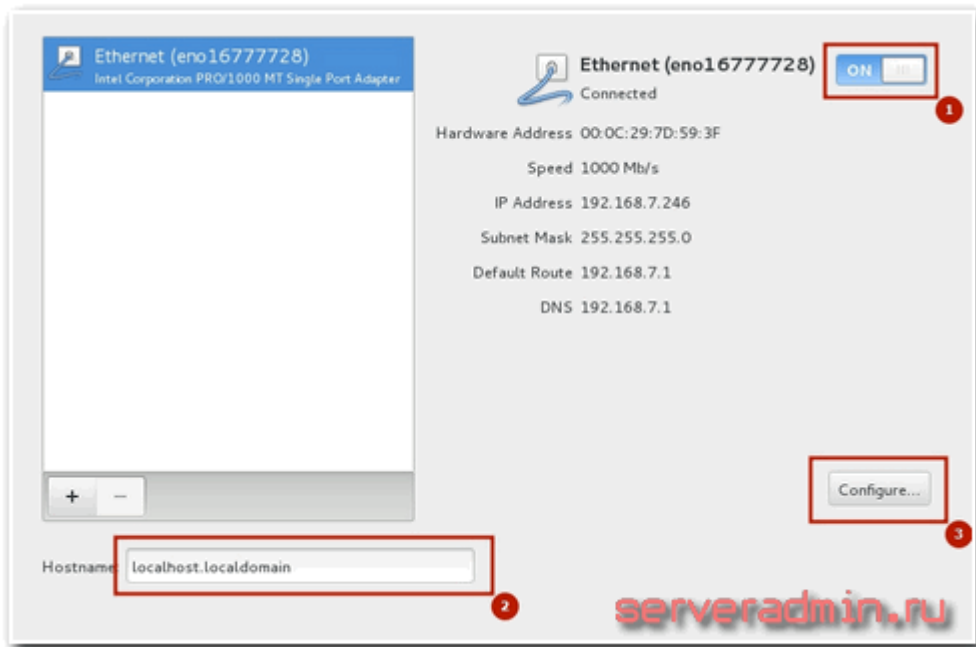
- 1 Сетевые настройки на сервере CentOS 7
- 2 Как получить сетевые настройки по DHCP
- 3 Как настроить DNS в CentOS 7
- 4 Как отключить ipv6 в CentOS 7
- 5 Как изменить hostname в CentOS 7
- 6 Установить шлюз по-умолчанию в CentOS 7
- 7 Network Manager в CentOS 7
- 8 System config network tui в CentOS 7
- 9 Как добавить статический маршрут в CentOS 7
- 10 Как настроить 2 IP адреса на одном интерфейсе
- 11 Как сделать перезапуск сети в CentOS 7
- 12 Как узнать IP адрес в CentOS 7
- 13 Что делать, если CentOS не видит сетевую карту?
- 14 Что делать, если сеть недоступна в CentOS?
- 15 Использование сетевых утилит traceroute, dig в CentOS
- 16 Настройка 802.1Q VLAN в CentOS 7
- 17 Дополнительные материалы по CentOS

Сетевые настройки на сервере CentOS 7

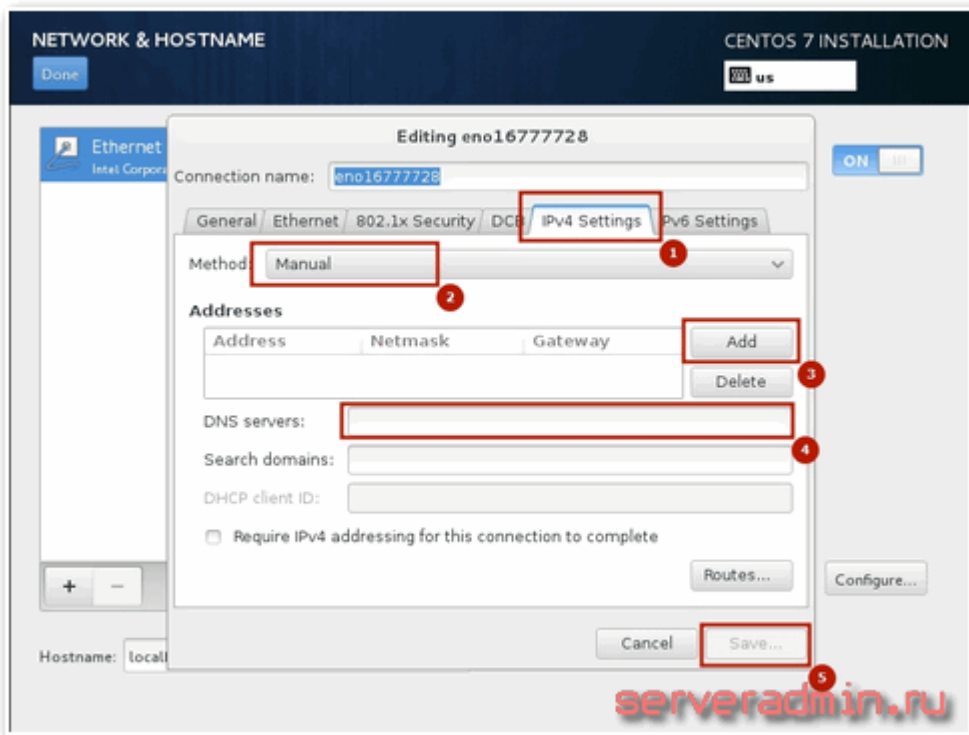
Первый раз с сетевыми настройками сервера CentOS мы сталкиваемся, когда производим установку. На экране первоначальной настройки есть отдельный пункт, касающийся настройки сетевых интерфейсов:



Зайдя в него мы видим список подключенных сетевых карт. Каждую из них можно включить соответствующим ползунком (пункт 1 на картинке). При активировании интерфейса он автоматически получает настройки по dhcp. Результат работы dhcp можно посмотреть тут же. Если вас не устраивают эти настройки, их можно отредактировать, нажав **configure** (пункт 3 на картинке). Здесь же можно задать **hostname** (пункт 2 на картинке):



Открыв окно дополнительных настроек Ethernet, вы сможете изменить имя сетевого интерфейса, указать настройки IP (пункт 1 на картинке), выбрать **ручные настройки** (пункт 2 на картинке), назначить **ip адрес** (пункт 3 на картинке), установить **dns сервер** (пункт 4 на картинке) и сохранить сетевые настройки (пункт 5 на картинке):



После выполнения остальных настроек начнется установка. После установки у вас будет сервер с указанными вами сетевыми настройками.

Теперь рассмотрим другую ситуацию. Сервер, а соответственно и конфигурацию сети, производили не вы, а теперь вам надо ее посмотреть либо изменить. В вашем распоряжении консоль сервера, в ней и будем работать. Если у вас установка производилась с дистрибутива **minimal**, то при попытке посмотреть сетевые настройки с помощью команды **ifconfig** в консоли вы увидите следующее:

```
-bash: ifconfig: command not found
```

или в русской версии:

```
-bash: ifconfig команда не найдена
```

Для работы с **ifconfig** и прочими сетевыми утилитами необходимо установить пакет **net-tools**. Сделаем это:

```
# yum -y install net-tools.x86_64
```

Теперь можно увидеть настройки сети:

```
# ifconfig

eno16777728: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.159.129 netmask 255.255.255.0 broadcast 192.168.159.255
inet6 fe80::20c:29ff:fe7d:593f prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:7d:59:3f txqueuelen 1000 (Ethernet)
RX packets 319 bytes 36709 (35.8 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 256 bytes 148817 (145.3 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0x10<host>
loop txqueuelen 0 (Local Loopback)
RX packets 6 bytes 624 (624.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 6 bytes 624 (624.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Если у вас нет желания устанавливать дополнительный пакет, то можно воспользоваться более простой командой **ip** с параметрами:

```
# ip addr

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

inet 127.0.0.1/8 scope host lo

valid_lft forever preferred_lft forever

inet6 ::1/128 scope host

valid_lft forever preferred_lft forever

2: eno16777728: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000

link/ether 00:0c:29:7d:59:3f brd ff:ff:ff:ff:ff:ff

inet 192.168.159.129/24 brd 192.168.159.255 scope global dynamic eno16777728
valid_lft 1709sec preferred_lft 1709sec
inet6 fe80::20c:29ff:fe7d:593f/64 scope link
valid_lft forever preferred_lft forever
```

Мы увидели конфигурацию сети, теперь давайте ее отредактируем. Допустим, нам нужно сменить ip адрес. Для этого идем в директорию `/etc/sysconfig/network-scripts` и открываем на редактирование файл **ifcfg-eth0**. Этот файл имеет примерно следующее содержание:

```
TYPE="Ethernet"
BOOTPROTO="dhcp"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
NAME="eth0"
UUID="e6ba24a1-bf73-4f85-9619-2347602e85b7"
ONBOOT="yes"
HWADDR="00:0C:29:7D:59:3F"
PEERDNS="no"
PEERROUTES="yes"
IPV6_PEERDNS="yes"
IPV6_PEERROUTES="yes"
```

По настройкам из этого файла мы получаем ip адрес по dhcp. Чтобы вручную прописать статический ip, приводим файл к следующему содержанию:

```
TYPE="Ethernet"
BOOTPROTO="none"
DNS1="192.168.159.2"
IPADDR0=192.168.159.129
PREFIX0=24
GATEWAY0=192.168.159.2
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
NAME="eth0"
UUID="e6ba24a1-bf73-4f85-9619-2347602e85b7"
ONBOOT="yes"
HWADDR="00:0C:29:7D:59:3F"
PEERDNS="no"
PEERROUTES="yes"
IPV6_PEERDNS="yes"
IPV6_PEERROUTES="yes"
```

Мы изменили параметры:

BOOTPROTO	с dhcp на none
DNS1	указали dns сервер
IPADDR0	настроили статический ip адрес
PREFIX0	указали маску подсети
GATEWAY0	настроили шлюз по-умолчанию

Чтобы изменения вступили в силу, необходимо перечитать сетевые настройки:

```
# /etc/init.d/network restart

Restarting network (via systemctl): [ OK ]
```

Проверяем, применилась ли новая конфигурация сети:

```
# ifconfig:

eno16777728: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500

inet 192.168.159.129 netmask 255.255.255.0 broadcast 192.168.159.255
inet6 fe80::20c:29ff:fe7d:593f prefixlen 64 scopeid 0x20<link>
ether 00:0c:29:7d:59:3f txqueuelen 1000 (Ethernet)
RX packets 672 bytes 71841 (70.1 kib)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 572 bytes 290861 (284.0 kib)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Все в порядке, новые настройки сетевого интерфейса установлены.

Как получить сетевые настройки по DHCP

Теперь рассмотрим обратную ситуацию. Допустим, у вас сетевая карта имеет какие-то настройки, установленные вручную. Но вы хотите, чтобы ваш компьютер получал настройки сети по dhcp в качестве клиента. Для этого вам нужно произвести операцию, обратную той, что мы делали раньше. То есть открываем файл `/etc/sysconfig/network-scripts/ifcfg-eth0` и удаляем там строки с параметрами `DNS`, `IPADDR`, `PREFIX`, `GATEWAY` а в параметре `BOOTPROTO` указываем значение **«dhcp»**. Сохраняем файл и перезапускаем сеть:

```
# /etc/init.d/network restart
```

Затем проверяем, получил ли наш client по dhcp настройки.

Как настроить DNS в CentOS 7

Текущие настройки dns сервера в CentOS можно посмотреть в двух местах:

1. В файле с настройками сетевой карты ifcfg-eth0, которым мы ранее неоднократно редактировали.
2. В файле /etc/resolv.conf

Зачем они сейчас в двух местах, я не знаю, но раньше настройки dns сервера в каких-то дистрибутивах, не помню уже точно каких, указывались только в resolv.conf, но в какой-то момент это изменилось. И все сетевые настройки стали храниться в одном файле вместе с адресом, шлюзом, маской и прочим. Если сейчас отредактировать файл resolv.conf и внести туда какие-то dns сервера, то после перезагрузки они будут заменены на значения из файла ifcfg-eth0.

Так что для того, чтобы установить параметры dns сервера, нужно отредактировать файл сетевых настроек ifcfg-eth0, добавив туда столько серверов, сколько требуется. Например так:

```
DNS1="192.168.159.2"
```

```
DNS2="8.8.8.8"
```

```
DNS3="8.8.4.4"
```

Для применения настроек сохраняем файл и перезапускаем сеть, все как обычно. После перезагрузки сервера настройки dns будут записаны в файл resolv.conf

```
# cat /etc/resolv.conf
```

```
# Generated by NetworkManager
```

```
nameserver 192.168.159.2
```

```
nameserver 8.8.8.8
```

```
nameserver 8.8.4.4
```

Как отключить ipv6 в CentOS 7

В настоящее время активного использования протокола **ipv6** нет и в обычной работе он не нужен. Хотя нас уже много лет пугают, что свободных ip адресов уже практически не осталось, но на деле пока еще всем хватает. Так что с точки зрения практических соображений ipv6 в настоящее время на сервере не нужен и его можно отключить. Перед отключением ipv6 необходимо на всякий случай проверить, какие программы его используют в своей работе. Это нужно для того, чтобы избежать ошибок в их работе, предварительно отключив ipv6 в конфигурациях. Для того, чтобы увидеть, какие программы висят на ipv6 интерфейсе воспользуемся командой netstat:

```
# netstat -tulnp
```

```
Active Internet connections (only servers)
```

```
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
```

```
tcp 0 0 127.0.0.1:25 0.0.0.0:* LISTEN 2317/master
```

```
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 1333/sshd
```

```
tcp6 0 0 :::25 :::* LISTEN 2317/master
```

```
tcp6 0 0 :::22 :::* LISTEN 1333/sshd
```

```
udp 0 0 0.0.0.0:49252 0.0.0.0:* 694/avahi-daemon: r
```

```
udp 0 0 0.0.0.0:123 0.0.0.0:* 715/chronyd
udp 0 0 0.0.0.0:5353 0.0.0.0:* 694/avahi-daemon: r
udp 0 0 127.0.0.1:323 0.0.0.0:* 715/chronyd
udp6 0 0 :::123 :::* 715/chronyd
udp6 0 0 ::1:323 :::* 715/chronyd
```

Все строки с ::: это ipv6 протокол. В моем случае это sshd, postfix и chronyd. Отключим им ipv6 и оставим только ipv4.

Начнем с sshd. Открываем файл настроек /etc/ssh/ssh_config и находим строки:

```
#AddressFamily any
#ListenAddress 0.0.0.0
```

Раскомментируем их и изменим. Должно получиться вот так:

```
AddressFamily inet
ListenAddress 0.0.0.0
```

Теперь открываем файл настроек постфикс /etc/postfix/main.cf. Ищем там строку:

```
#inet_protocols = all
```

Меняем на:

```
inet_protocols = ipv4
```

Отключаем ipv6 в chronyd. Для этого создаем файл /etc/sysconfig/**chronyd** и добавляем строку:

```
OPTIONS=-4
```

Теперь отключаем ipv6 в CentOS. Открываем файл /etc/**sysctl.conf** и добавляем туда строки:

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

Редактируем файл /etc/sysconfig/**network**, добавляя туда:

```
NETWORKING_IPV6=no
IPV6INIT=no
```

Перезагружаемся и проверяем результат:

```
# reboot
# ifconfig

eno16777728: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.159.129 netmask 255.255.255.0 broadcast 192.168.159.255
ether 00:0c:29:7d:59:3f txqueuelen 1000 (Ethernet)
RX packets 2301 bytes 243024 (237.3 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
```

```
TX packets 2138 bytes 1327955 (1.2 MiB)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536

inet 127.0.0.1 netmask 255.0.0.0

Loop txqueuelen 0 (Local Loopback)

RX packets 0 bytes 0 (0.0 B)

RX errors 0 dropped 0 overruns 0 frame 0

TX packets 0 bytes 0 (0.0 B)

TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Нигде нет упоминания про inet6 и адреса формата ipv6. Значит все в порядке, мы отключили ipv6 в CentOS. Теперь проверим список открытых портов:

```
# netstat -tulnp

Active Internet connections (only servers)

Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name

tcp 0 0 127.0.0.1:25 0.0.0.0:* LISTEN 2291/master

tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 1322/sshd

udp 0 0 0.0.0.0:123 0.0.0.0:* 2453/chronyd

udp 0 0 0.0.0.0:5353 0.0.0.0:* 697/avahi-daemon: r

udp 0 0 127.0.0.1:323 0.0.0.0:* 2453/chronyd

udp 0 0 0.0.0.0:57259 0.0.0.0:* 697/avahi-daemon: r
```

Все порты ipv4. Все в порядке, наша задача выполнена.

Как изменить hostname в CentOS 7

По-умолчанию, во время установки CentOS ставит имя хоста **localhost.localdomain**. Если вы его не поменяли, то можно это сделать позже. Для начала давайте проверим, какое имя хоста у вас установлено. Делается это с помощью команды в консоли **hostname**, либо с помощью **uname**:

```
# hostname

localhost.localdomain

# uname -n

localhost.localdomain
```

Для того, чтобы изменить имя хоста в CentOS, необходимо отредактировать файл **/etc/hostname**. Проверим его содержимое:

```
# cat /etc/hostname
```

```
localhost.localdomain
```

Отредактируем этот файл, чтобы изменить hostname:

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

```
centos.localhost
```

Сохраняем файл и проверяем:

```
# hostname
```

```
centos.localhost
```

Все в порядке, мы изменили имя хоста на centos.localhost

Установить шлюз по-умолчанию в CentOS 7

Если по каким-то причинам при настройке сетевых параметров у вас не установился шлюз по-умолчанию, то сделать это можно вручную. Для начала проверим, какой шлюз по-умолчанию установлен в системе в данный момент:

```
# netstat -nr
```

```
Kernel IP routing table
```

```
Destination Gateway Genmask Flags MSS window irtt Iface
```

```
0.0.0.0 149.154.71.254 0.0.0.0 UG 0 0 0 eth0
10.8.0.0 10.8.0.2 255.255.255.0 UG 0 0 0 tun0
10.8.0.2 0.0.0.0 255.255.255.255 UN 0 0 0 tun0
149.154.70.0 0.0.0.0 255.255.254.0 U 0 0 0 eth0
192.168.1.0 10.8.0.2 255.255.255.0 UG 0 0 0 tun0
```

Строка с Destination 0.0.0.0 определяет адрес шлюза. Если у вас ее нет, либо в поле Gateway установлен неверный шлюз, то можно это изменить. Устанавливаем шлюз по-умолчанию:

```
route add default gw 192.168.0.1
```

Network Manager в CentOS 7

В CentOS по-умолчанию имеется служба, которая управляет всеми сетевыми подключениями — **NetworkManager**. Она постоянно контролирует сетевые настройки и с помощью демона по управлению конфигурациями вносит соответствующие изменения в активные сетевые устройства. Она поддерживает стандартные файлы конфигураций ifcfg.

Список сетевых утилит и приложений:

ПРИЛОЖЕНИЕ	ОПИСАНИЕ
NetworkManager	Стандартный networking daemon
nmtui	Простой текстовый интерфейс (TUI) для NetworkManager
nmcli	Утилита, работающая в командной строке, которая позволяет пользователям и скриптам взаимодействовать с NetworkManager
control-center	Утилита с графическим интерфейсом оболочки GNOME

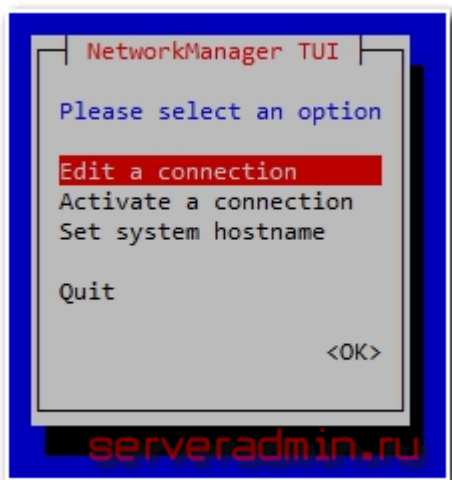
ПРИЛОЖЕНИЕ	ОПИСАНИЕ
nm-connection-editor	GTK+ 3 приложения, необходимые для некоторых задач, не поддерживаемых control

Пользователи не взаимодействуют с NetworkManager в CentOS напрямую, для этого используются графические и утилиты командной строки. Одной из таких утилит является **system config network tui**.

System config network tui в CentOS 7

Для управления сетевыми настройками в CentOS можно воспользоваться графической утилитой **nmtui**. Проверить ее наличие в системе очень просто. Достаточно запустить ее в консоли:

```
# nmtui
```



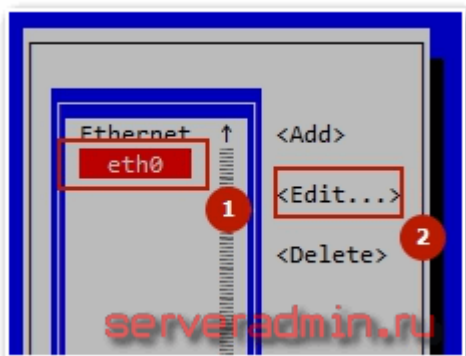
Если она у вас не установлена, то исправить это очень просто. Устанавливаем в CentOS system config network tui:

```
# yum install NetworkManager-tui
```

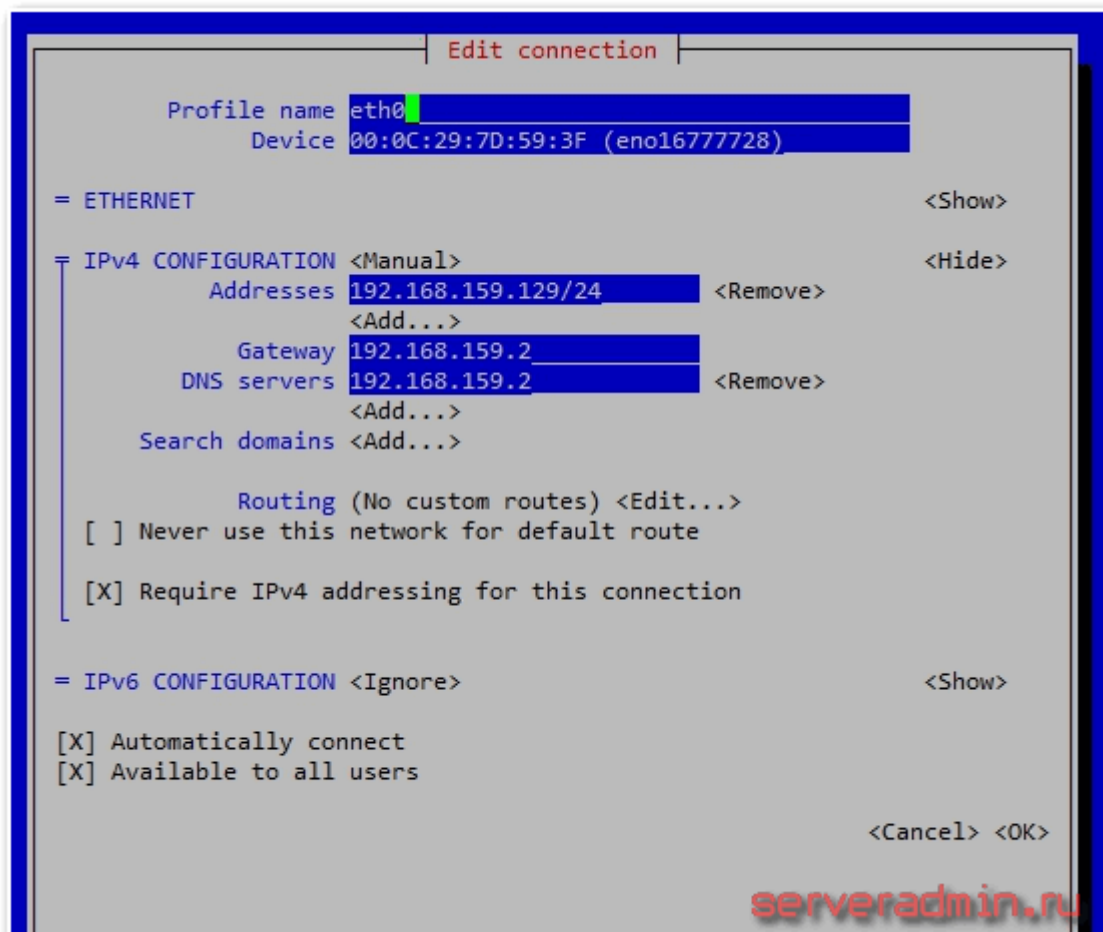
С помощью **tui** можно указать любые сетевые настройки, которые мы делали раньше через командную строку и редактирование конфигурационных файлов. Давайте сделаем это. Вызываем программу:

```
# nmtui
```

Выбираем первый пункт **Edit a connection**, затем выбираем сетевой интерфейс и ждем «Edit»:



Здесь мы можем изменить имя сетевой карты, mac адрес, указать тип сетевых настроек — ручной или dhcp, указать вручную ip адрес, адрес шлюза, dns сервера, добавить маршруты и некоторые другие настройки:



После завершения редактирования сохраняем настройки, нажимая **OK**. Если в первом экране утилиты выбрать пункт **Set System Hostname**, то можно быстро указать имя хоста. Результат будет такой же, как мы делали раньше в консоли.

Как добавить статический маршрут в CentOS 7

Для управления маршрутизацией в CentOS может понадобиться добавить статический маршрут. Сделать это достаточно просто с помощью консольной команды. Для начала проверим существующие маршруты, используя **netstat**:

```
# netstat -nr

Kernel IP routing table

Destination Gateway Genmask Flags MSS window irtt Iface
0.0.0.0 192.168.159.2 0.0.0.0 UG 0 0 0 eno16777728
192.168.159.0 0.0.0.0 255.255.255.0 U 0 0 0 eno16777728
```

В данном случае у нас один маршрут для адреса 0.0.0.0/0.0.0.0 шлюз используется 192.168.159.2, он же шлюз по умолчанию. То есть по сути, статических маршрутов никаких нет. Добавим один из них.

Допустим, у нас есть подсеть 192.168.8.0 маска 255.255.255.0, трафик в эту подсеть маршрутизирует шлюз 192.168.159.5 Добавляем маршрут:

```
# route add -net 192.168.8.0/24 gw 192.168.159.5
```

Проверяем, появился ли добавленный маршрут в **таблицу маршрутизации**:

```
# netstat -nr

Kernel IP routing table

Destination Gateway Genmask Flags MSS window irtt Iface
```

```
0.0.0.0 192.168.159.2 0.0.0.0 UG 0 0 0 eno16777728
192.168.8.0 192.168.159.5 255.255.255.0 UG 0 0 0 eno16777728
192.168.159.0 0.0.0.0 255.255.255.0 U 0 0 0 eno16777728
```

Все в порядке, маршрут добавлен. Но после перезагрузки этот статический маршрут будет удален. Чтобы этого не произошло и добавленные маршруты сохранялись, необходимо их записать в специальный файл. В папке `/etc/sysconfig/network-scripts` создаем файл с именем **route-eth0** следующего содержания:

```
# mcedit /etc/sysconfig/network-scripts/route-eth0

192.168.8.0/24 via 192.168.159.5
```

Перезагружаемся и проверяем, на месте ли маршрут:

```
# reboot

# netstat -nr

Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
0.0.0.0 192.168.159.2 0.0.0.0 UG 0 0 0 eno16777728
192.168.8.0 192.168.159.5 255.255.255.0 UG 0 0 0 eno16777728
192.168.159.0 0.0.0.0 255.255.255.0 U 0 0 0 eno16777728
```

Все в порядке, статический маршрут добавлен.

Как настроить 2 IP адреса на одном интерфейсе

Если у вас появилась необходимость настроить **2 IP** адреса на одном интерфейсе в CentOS, то сделать это достаточно просто. Воспользуемся командой `ifconfig`. Для начала проверим список сетевых интерфейсов:

```
# ifconfig

eno16777728: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.159.129 netmask 255.255.255.0 broadcast 192.168.159.255
ether 00:0c:29:7d:59:3f txqueuelen 1000 (Ethernet)
RX packets 254 bytes 30173 (29.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 204 bytes 27658 (27.0 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 0 (Local Loopback)
RX packets 11 bytes 940 (940.0 B)
```

```
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 11 bytes 940 (940.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Добавим к интерфейсу eno16777728 еще один ip адрес 192.168.159.120:

```
# ifconfig eno16777728:1 192.168.159.120 up
```

Проверим, что получилось:

```
# ifconfig
eno16777728: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.159.129 netmask 255.255.255.0 broadcast 192.168.159.255
ether 00:0c:29:7d:59:3f txqueuelen 1000 (Ethernet)
RX packets 254 bytes 30173 (29.4 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 204 bytes 27658 (27.0 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eno16777728:1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 192.168.159.120 netmask 255.255.255.0 broadcast 192.168.159.255
ether 00:0c:29:7d:59:3f txqueuelen 1000 (Ethernet)

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 0 (Local Loopback)
RX packets 11 bytes 940 (940.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 11 bytes 940 (940.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Все в порядке, мы добавили второй ip адрес на один и тот же интерфейс. Но после перезагрузки дополнительный адрес не сохранится. Чтобы его сохранить, необходимо создать файл настроек интерфейса в папке /etc/sysconfig/network-scripts

```
# mcedit /etc/sysconfig/network-scripts/ifcfg-eno16777728:1
DEVICE=eno16777728:1
BOOTPROTO=static
IPADDR=192.168.159.120
NETMASK=255.255.255.0
ONBOOT=yes
```

Сохраняем файл, перезагружаемся и проверяем, что получилось. Второй ip адрес должен быть на месте.

Как сделать перезапуск сети в CentOS 7

Ранее я уже касался этого вопроса, но на всякий случай повторим отдельно. Допустим, вы внесли некоторые изменения в конфигурацию сети. Как применить эти настройки, не перезагружая сервер? Очень просто. Для перезапуска сети в CentOS достаточно воспользоваться командой:

```
# /etc/init.d/network restart
```

Служба NetworkManager перечитывает все сетевые настройки и применит изменения.

Как узнать IP адрес в CentOS 7

Для того, чтобы быстро узнать текущий IP адрес в CentOS необходимо воспользоваться следующими командами:

```
# ifconfig | grep inet
```

```
inet 192.168.159.129 netmask 255.255.255.0 broadcast 192.168.159.255
inet 127.0.0.1 netmask 255.0.0.0
```

Либо второй вариант определения локального ip адреса:

```
# ip addr | grep inet
```

```
inet 127.0.0.1/8 scope host lo
```

```
inet 192.168.159.129/24 brd 192.168.159.255 scope global eno16777728
```

Обе команды позволяют быстро узнать свой ip адрес.

Что делать, если CentOS не видит сетевую карту?

Вы установили сервер, загрузились и обнаружили, что в системе нет ни одной сетевой карты. Что в таком случае делать? Первым делом посмотрите вывод команды `dmesg` и поищите там поминание о своей карте. Возможно, она в системе есть, просто не активирована. Активировать ее можно с помощью **nmtui**, а котором я рассказывал выше. Там есть пункт меню **Activate connection**, нужно в него зайти и активировать вашу сетевую карту. После этого ее можно будет настраивать.



Если же вашей сетевой карты нет в системе, то нужно поискать в интернете по модели информацию об этой сетевой карте. Возможно в репозиториях будут драйвера для нее. Это достаточно распространенная ситуация. Чаще всего драйвера найдутся и их необходимо будет правильно установить.

Есть еще вероятность, что вы не увидите своей карточки при выводе команды `ifconfig`, если в эту карту не воткнут сетевой провод. Чтобы наверняка посмотреть все интерфейсы, необходимо использовать ключ `-a`:

```
# ifconfig -a
```

Есть еще один способ поискать сетевую карту в системе. Установите пакет **pciutils**:

```
# yum -y install pciutils
```

И посмотрите вывод команды:

```
# lspci | grep Ethernet
```

Если сетевая карта видится системой, то должно быть что-то в этом роде:

```
02:00.0 Ethernet controller: Intel Corporation 82545EM Gigabit Ethernet Controller (Copper) (rev 01)
```

Если в выводе пусто, значит сетевая карта не определена.

Что делать, если сеть недоступна в CentOS?

Допустим, вы пытаетесь пинговать какой-то адрес, например 8.8.8.8 и получаете ответ, что сеть не доступна:

```
# ping 8.8.8.8
```

А в ответ получаете:

```
connect: Network is unreachable
```

Такая ситуация может возникнуть, если у вас не установлен шлюз по-умолчанию. Добавить его можно с помощью команды:

```
# route add default gw 192.168.0.1
```

Если дело не в шлюзе, то нужно проверить настройки сетевых адаптеров. Если нет ни одного активного адаптера, то вы так же будете получать сообщение о том, что сеть не работает. Необходимо, чтобы в системе был правильно настроен хотя бы один сетевой адаптер. Как это сделать, написано выше.

Использование сетевых утилит traceroute, dig в CentOS

Для диагностики сетевых подключений в CentOS полезно использовать специальные утилиты. Но если вы использовали установку `minimal`, то их скорее всего в системе не будет. К примеру, популярная утилита **traceroute** при попытке ее запуска выдаст сообщение:

```
# traceroute ya.ru
```

```
bash: traceroute: command not found
```

Ее нужно установить отдельно из репозитория:

```
# yum -y install traceroute
```

То же самое с популярной программой **dig** для работы с dns серверами и записями:

```
# dig ya.ru
```

```
bash: dig: command not found
```

Чтобы эта сетевая утилита заработала, необходимо установить пакет **bind-utils**:

```
# yum -y install bind-utils
```

Настройка 802.1Q VLAN в CentOS 7

Для поднятия тегированного интерфейса на CentOS нужно в первую очередь проверить поддержку ядром **8021q**:

```
# modprobe 8021q
```

Если сообщений об ошибке нет, значит все в порядке, модуль загрузился. Если же модуль не найден, необходимо пересобрать модули ядра, включив поддержку необходимого модуля. Проверим на всякий случай, загрузился ли модуль:

```
# lsmod | grep 8021q

8021q 29022 0
garp 14384 1 8021q
mrp 18542 1 8021q
```

Все в порядке, модуль 8021q загружен, добавим его в автозагрузку:

```
# echo 8021q >> /etc/modules-load.d/8021q.conf
```

Теперь создаем файл конфигурации для vlan в /etc/sysconfig/network-scripts:

```
# mcedit ifcfg-eth0.2000

VLAN=yes
DEVICE=eth0.2000
BOOTPROTO=static
ONBOOT=yes
TYPE=Vlan
IPADDR=192.168.100.2
NETMASK=255.255.255.0
```

Обращаю внимание на выделенное жирным. Во всех инструкциях в интернете, что мне попались, этот параметр был указан как **TYPE=Ethernet**, но с такой настройкой интерфейс с vlan не поднимался, появлялась ошибка:

```
Error: no device found for connection 'system eth0.2000'.
```

Только после того, как я исправил, все заработало как надо. Так что сохраняем и активируем интерфейс:

```
# ifup eth0.2000
```

```
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/7)
```

Проверяем наш vlan:

```
# ip l ls

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT qlen 1000
    link/ether 00:15:5d:01:0f:06 brd ff:ff:ff:ff:ff:ff
3: eth0.2000@eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT
    link/ether 00:15:5d:01:0f:06 brd ff:ff:ff:ff:ff:ff
```

Все в порядке, тегированный интерфейс поднят. По аналогии можно создать еще несколько подобных интерфейсов, не забывая изменять им имена и адреса.

На этом мой объемный материал на тему настройки сети в CentOS закончен. Буду очень рад комментариям, замечаниям, поправкам. Они наверняка будут. Создавая подобные материалы я в первую очередь учусь сам и подтягиваю свои знания в предметной области. Тут могут быть где-то ошибки и описки, хотя я и проверяю все на живых системах во время написания руководств, но все равно возможны неточности и опечатки.