**Сетевые интерфейсы**

* Сетевой интерфейс — физическое или виртуальное устройство, предназначенное для передачи данных между программами через компьютерную сеть
* Интерфейсы создаются автоматически для каждого обнаруженного сетевого устройства при загрузке ядра ОС. Однако в некоторых случаях интерфейсы создаются пользовательскими программами
* Если устройств, обеспечивающих одинаковый тип интерфейса, несколько, их автоматическое определение не производится
* При использовании нескольких устройств одного типа нужно произвести их ручную настройку, то есть явным образом назначить интерфейс каждому из них
* Вновь созданный интерфейс является ненастроенным: он выключен и к нему не привязан никакой IP-адрес. Для того чтобы ввести интерфейс в работу, нужно провести его настройку и включить (поднять) его при помощи, например, команды ifconfig
* Не путайте **интерфейсы** и **устройства** системы. Интерфейсам не соответствуют никакие специальные файлы в каталоге /dev

**Примеры сетевых интерфейсов**

* Петлевые интерфейсы для обмена данными между процессами на одном компьютере или управляемом сетевом устройстве. Для них выделена специальная подсеть 127.0.0.0/8 (**lo)**. Этот интерфейс используется для связи программ IP-клиентов с IP-серверами, запущенными на той же машине, так что его необходимо настроить даже в том случае, если вы вообще не подключаете никаких сетевых устройств. Он связывает Вашу машину саму с собой. Его можно использовать для разных целей. Например, Вы можете захотеть протестировать какое-либо сетевое программное обеспечение, не вмешиваясь в работу других машин в сети. Для этого интерфейса специально зарезервирован адрес 127.0.0.1. Поэтому, запустив telnet на адрес 127.0.0.1 на любой машине, получите соединение с этой же машиной. Все TCP/IP реализации поддерживают loopback механизмы, которые реализуют виртуальный сетевой интерфейс исключительно программно и не связаны с каким-либо оборудованием, но при этом полностью интегрированы во внутреннюю сетевую инфраструктуру компьютерной системы. Любой трафик, который посылается компьютерной программой на интерфейс loopback, тут же получается тем же интерфейсом.
* Физические интерфейсы сетевых карт и телекоммуникационных **(ethX, wlanX и т.д.)**
* Туннели — для инкапсуляции протокола того же или более низкого уровня в другой протокол **(tunX, tapX)**
* Интерфейсы виртуальных сетей (**vlanX либо ethX.Y**). Утилита для настройки **vconfig**

**Loopback**

**ifconfig lo** 127.0.0.1 – настроить локальный интерфейс

**ping** 127.0.0.1 – проверить интерфейс на работоспособность, послав на него пакеты

**Ethernet**

Интерфейсы Ethernet обозначаются в системе как **ethX/enpXsY**, где **X/Y** является числом. Первый интерфейс Ethernet обычно обозначается **eth0(enp2s0)**, второй **eth1(enp3s0)**, и так далее в порядке возрастания чисел.

Для быстрого определения всех доступных Ethernet интерфейсов вы можете использовать команду

**ifconfig -a**

**Другие сетевые интерфейсы:**

* **wlan** - cетевой интерфейс wi-fi адаптеров
* **fddi** - cетевой интерфейс к карте FDDI
* **tr** - cетевой интерфейс к карте Token Ring
* **ppp** - cетевой интерфейс к каналу PPP (Point-to-Point Protocol)
* **sl** - cетевой интерфейс к каналу SLIP (Serial Line IP)
* **plip** - cетевой интерфейс к каналу PLIP (Parallel Line IP)
* **hci** - Bluetooth (утилита для настройки hciconfig)

**TUN , TAP**

В терминологии компьютерных сетей, **TUN**и **TAP**— виртуальные сетевые драйверы ядра системы. Они представляют собой программные сетевые устройства, которые отличаются от обычных аппаратных сетевых карт.

**TAP**эмулирует Ethernet устройство и работает на канальном уровне модели OSI, оперируя кадрами Ethernet.

**TUN**(сетевой туннель) работает на сетевом уровне модели OSI, оперируя IP пакетами.

TAP используется для создания сетевого моста, тогда как TUN для маршрутизации.

Пакет, посылаемый операционной системой через TUN/TAP устройство, обрабатывается программой, которая контролирует это устройство (например openVPN). Сама программа также может отправлять пакеты через TUN/TAP устройство. В таком случае TUN/TAP устройство доставляет (или «внедряет») такой пакет в сетевой стек операционной системы, эмулируя таким образом доставку пакета с внешнего устройства.

**Используется для виртуальных частных сетей и виртуальных машин**

**Параметры интерфейса**

Каждый интерфейс характеризуется определёнными параметрами, необходимыми для обеспечения его нормального функционирования, и в частности для сетевого обмена данными по протоколу IP.

* **IP-адрес**  
  Адрес IP, соответствующий данному сетевому интерфейсу. Пакеты, отправленные по этому адресу, поступят на соответствующий интерфейс
* **Маска подсети**  
  Битовая маска, необходимая для вычисления маршрута передачи IP-пакета
* **Широковещательный адрес**  
  Адрес, используемый при широковещательной рассылке пакетов через интерфейс.
* **Метрика**  
  Условная характеристика интерфейса соответствующая уровню затрат при передаче информации через него. Используется при маршрутизации пакетов, для выбора оптимального маршрута.
* **MTU**  
  Maximum Transfer Unit. Максимальный размер блока данных обрабатываемого интерфейсом. Наибольшее значение MTU определяется типом интерфейса (например, для Ethernet MTU=1500), но может быть искусственно снижено.
* **MAC-адрес**  
  Аппаратный адрес сетевого устройства, соответствующего интерфейсу (для которых это имеет смысл).

Кроме этих параметров интерфейс характеризуется ещё:

* **флагами**, которые определяют состояния устройства, например, такие как: включен ли интерфейс (Up/Down), находится ли он в неразборчивом режиме (promiscuous/nonpromiscuous)
* **аппаратными характеристиками***,*такими как адрес памяти, номер IRQ, DMA, порт ввода/вывода;
* **статистической информацией***,*характеризующей различные аспекты работы интерфейса. Например, количество переданных/полученных байтов/пакетов, число переполнений, коллизий и др. с момента создания интерфейса.

**Порядок работы с интерфейсом**

1. Вновь созданный интерфейс является ненастроенным
2. Он выключен и к нему не привязан никакой IP-адрес (и никакие другие параметры)
3. Настроить интерфейс
4. Поднять его

При настройке интерфейса обычно настраиваются следующие параметры:

* ***IP-адрес***должен быть указан обязательно, поскольку без него использование интерфейса неосуществимо
* ***Сетевая маска***должна указываться в том случае, если она отличается от той, которая соответствует классу IP-адреса
* ***Широковещательный адрес***указывается в том случае, если он отличается от широковещательного адреса, вычисляемого на основе значений IP-адреса и сетевой маски

**Конфигурация интерфейса**

**ifconfig** – просмотр конфигурации сети

* **флаги, метрика и MTU**  
  список установленных флагов интерфейса: включён UP; принимает широковещательные пакеты BROADCAST; принимает групповые пакеты MULTICAST. Среди списка установленных флагов может присутствовать слово PROMISC, означающее, что интерфейс работает в неразборчивом режиме. Установленный размер максимального блока, передаваемого через интерфейс MTU, и метрика интерфейса Metric.
* **характеристики сетевого уровня IPv4, IPv6**  
  ip-адрес интерфейса inet; маска netmask, широковещательный адрес broadcast
* **характеристики канального уровня**  
  канальный уровень ether - Аппаратный MAC-адрес устройства
* **информация о полученных пакетах RX**  
  число пакетов packets, ошибок errors, отброшенных пакетов dropped, переполнений overruns. Такое назначение полей соответствует только сетям Ethernet. В других сетях, смысл может отличаться.
* **информация об отправленных пакетах TX**  
  число пакетов packets, ошибок errors, отброшенных пакетов dropped, переполнений overruns, потерь несущей carrier, коллизий collisions: объем буфера передачи txqueuelen. Такое назначение полей соответствует только с етям Ethernet. В других сетях, смысл может отличаться.
* **объем переданных данных**  
  количество байтов, полученных RX bytes и отправленных TX bytes через интерфейс
* **аппаратные параметры**  
  номер линии IRQ Interrupt и адрес памяти Base address

**Канальный уровень:**

**ip link show** – показать все сетевые интерфейсы

**ip link show up** – показывает статус текущих сетевых интерфейсов

**sudo lspci** – список pci устройств, среди которых есть и сетевые карты

**cat /proc/net/dev** – показывает все сетевые интерфейсы, подключенные к системе

**sudo lshw -class network** – просмотр конфигурации сети. Показывает Ethernet интерфейс с логическим именем ethX вместе с информацией по шине, сведениями о драйвере и всеми поддерживаемыми возможностями.

**sudo ip** **link set** **dev** существующийEthИнтерфейс **up –** Поднять интерфейс

**sudo ip link set** **dev** существующийEthИнтерфейс **down –** отключить интерфейс

**ifconfig** интерфейс **up** – поднять интерфейс

**ifconfig** интерфейс **down** – отключить интерфейс

**sudo ifconfig** существующийEthИнтерфейс:новыйНомерВиртуальногоИнтерейса ip\_address **netmask** маска **broadcast** ip\_address **up**– создание временного виртуального сетевого интерфейса + назначение ему ip адреса, маски сети, широковещательного адреса и поднятие интерфейса



**sudo ifconfig** существующийEthИнтерфейс ip\_address **netmask** маска **broadcast** ip\_address– назначение ip адреса, маски сети, широковещательного адреса уже существующему интерфейсу

**Сетевой уровень:**

**ip address** **show** - показывает статус всех сетевых интерфейсов + IP адреса

**sudo ip** **address add** ip\_address/mask **broadcast** ip\_address **dev** существующийEthИнтерфейс **–** добавление нового IP адреса, маски сети, широковещательного адреса

**sudo ip address add** ip\_address/mask **broadcast** ip\_address **dev** существующийEthИнтерфейс **label** существующийEthИнтерфейс: новыйНомерВиртуальногоИнтерейса – создание нового интерфейса

**sudo ip** **address del** ip\_address/mask **dev** существующийEthИнтерфейс **–** удаление IP адреса

**Другое:**

**sudo ip neighbour** – просмотреть таблицу соседей (**arp таблица**). Когда компьютер с некоторым IP-адресом решает отправить пакет другому компьютеру, он выясняет, принадлежит ли адресат той же локальной сети, что и отправитель (т. е. подключены ли они к одной среде передачи данных). Делается это так: на IP-адрес получателя накладывается сетевая маска, и таким образом вычисляется адрес сети, которой принадлежит получатель. Если этот адрес совпадает с адресом сети отправителя, значит, оба находятся в одной локальной сети. Это, в свою очередь, означает, что аппаратный адрес (MAC) получателя должен быть отправителю известен. MAC-адреса компьютеров локальной сети хранятся в специальной таблице ядра, называемой «таблицей соседей» или, как это принято в протоколе IP, **ARP-таблицей.**

**ifconfig** интерфейс **arp** – включить этот интерфейс в ARP таблицу

**ifconfig** интерфейс **-arp** – удалить интерфейс из ARP таблицы

**ifconfig** интерфейс **promisc** – включить promiscuous mode («неразборчивый» режим, в котором сетевая плата позволяет принимать все пакеты независимо от того, кому они адресованы.)

**ifconfig** интерфейс **-promisc** – отключить promiscuous mode

* ifconfig, ip позволяют изменить настройки, которые будут применены мгновенно, но они не будут постоянными и будут утеряны после перезагрузки
* Обычно, люди настраивают сеть вручную, чтобы проверить как скажутся новые параметры на работе сервера
* Если вы хотите, чтобы сетевой интерфейс работал постоянно, необходимо модифицировать конфигурационные файлы в соответствии с требованиями вашего дистрибутива Linux

**Сетевые конфиги**

Начиная с релиза Ubuntu 17.10, для управления конфигурацией сети используется утилита Netplan - /etc/netplan. Раньше для этих целей применялся скрипт ifupdown, конфигурационные файлы которого находились в папке /etc/network/interfaces. Недостатком такого подхода было то, что файлы настройки сети были разбросаны по всей системе, частью настроек мог управлять NetworkManager, частью systemd-networkd, а часть вообще делалась с помощью ifupdown.

Новая система конфигурации позволяет таки сложить все яйца в одну корзину. Вы выполняете настройку в формате YAML, а Netplan автоматически преобразовывает её при запуске в конфигурацию той системы, которую вы выбрали.

**Синтаксис**

**поле0:**  
**поле1: значение**  
**поле2: [ элемент1, элемент2 ]**

**поле3:**  
**- элемент1**  
**- элемент2**  
**- элемент3**

Имя поля и его значение разделяется двоеточием. В качестве значения поля можно передавать не только текстовое или числовое значение, но и другое поле, несколько полей или список значений. При передаче списка каждый новый элемент списка должен начинаться с дефиса или быть заключены в квадратные скобки и перечисляться через запятую. Табуляции использовать нельзя. Отступы используются для указания структуры. Например, из примера видно, что **поле1** , **поле2** , **поле 3** относятся к **полю0**.

**Netplan**

**network:**  
**version: 2**  
**renderer: программа\_бэкенд**  
**вид\_интерфейса:**  
**имя\_интерфейса:**  
**параметр: значение**

Первые две строчки конфигурации стандартны. Первая указывает, что мы будем иметь дело с сетью, а вторая указывает версию стандарта конфигурации, которая будет использоваться. Их лучше не трогать.

* **renderer** - указывает программу, для которой будут преобразоваться ваши настройки. На данный момент поддерживаются только network-manager и systemd-networkd. networkd является приемником network-manager и Ubuntu постепенно переходит на него;
* **вид\_интерфейса** - вид сетевых интерфейсов, которые вы будете настраивать в этой секции. Они делятся на физические: ethernets (проводные), wifis (беспроводные) и виртуальные: vlans , bonds, bridges.
* **имя\_интерфейса** - имя сетевого интерфейса в системе, например enp3s0 или eth0;
* **параметры** - настройки, с помощью которых указывается, как нужно подключаться к сети.
  + **renderer** - программа для обработки конфигурации;
  + **dhcp4** - получение IPv4 адреса по DHCP;
  + **dhcp6** - получение IPv6 адреса по DHCP;
  + **dhcp-identifier** - если передать значение "mac", то будет использоваться MAC-адрес в качестве идентификатора DHCP;
  + **addresses** - добавляет статические адреса к интерфейсу, можно несколько; (параметр ожидает список IP-адресов, которые нужно присвоить нашей сетевой карте, у нас только один адрес, однако, передавать его надо тоже в формате списка, иначе будет ошибка. В конце адреса указывается префикс маски - /24, который отвечает маске сети 255.255.255.0. Вы можете указать и другую маску, но только с помощью префикса, отдельного параметра для этого нет.
  + **gateway4** - указывает шлюз IPv4;
  + **gateway6** - указывает шлюз IPv6;
  + **nameservers** - указывает DNS-серверы;
  + **macaddress** - устанавливает новый MAC-адрес;
  + **routes** - позволяет настроить маршруты таблицы маршрутизации;
  + **routing-policy** - дополнительная настройка маршрутов, для IP или подсети;
  + **access-points** - список точек доступа для Wi-Fi;
  + **password** - пароль для точки доступа Wi-Fi;
  + **mode** - режим работы сетевой карты Wi-Fi.

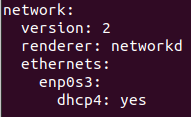
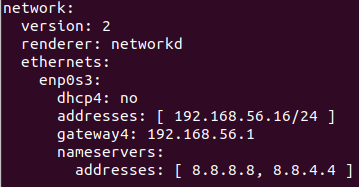
Синтаксис самой команды **netplan** очень прост:

**netplan опции команда**

В качестве команды можно передать одну из команд:

* **try** - попробовать применить конфигурацию с возможностью отмены;
* **apply** - применить конфигурацию;
* **generate** - проверка текущей конфигурации и запись на диск;
* **config** - записать текущую конфигурацию сети в YAML.

**Настройка динамического IP Настройка статического IP + Gateway + DNS**

nano /etc/netplan/01-network-manager-all.yaml nano /etc/netplan/01-network-manager-all.yaml 

**sudo netplan generate** – проверить конфигурацию **sudo netplan generate** – проверить конфигурацию

**sudo netplan apply** – применить конфигурацию **sudo netplan apply** – применить конфигурацию

**Маршрутизация**

**route -n** – открыть таблицу маршрутизации

**netstat -rn** – открыть таблицу маршрутизации

**ip route show** – открыть таблицу маршрутизации

**sudo route add default gw** ip\_шлюза интерфейс – добавить шлюз по умолчанию (где destination будет 0.0.0.0)

**sudo route add -net** ip\_address/mask **gw** ip\_шлюза – добавление маршрута через шлюз (нужно указывать ip адрес сети)

**sudo** **route add -net** ip\_address/mask **dev** интерфейс – добавление маршрута через интерфейс (нужно указывать ip адрес сети)

**sudo** **route add -host** ip\_address **gw** ip\_шлюза – маршрут до отдельного хоста (нужно указывать ip адрес хоста и не указывать маску сети)

**sudo** **route del default –** удаление маршрута по умолчанию

**sudo** **route del -net** ip\_address/mask – удаление маршрута к сети

**sudo** **route del -host** ip\_address/mask – удаление маршрута к отдельному хосту

**sudo** **ip route add default** **via** ip\_шлюза – маршрут по умолчанию для всех адресов

**sudo** **ip route add default** **dev** интерфейс – маршрут по умолчанию для всех адресов

**sudo** **ip route add** ip\_address/mask **via** ip\_шлюза – добавление маршрута через шлюз (нужно указывать ip адрес сети)

**sudo ip route add** ip\_address/mask **dev** интерфейс - добавление маршрута через интерфейс (нужно указывать ip адрес сети) (в Gateway проставляются 0.0.0.0)

**sudo** **ip route add** ip\_address/mask **dev** интерфейс **metric** метрика – маршрут с метрикой

**sudo ip route change** ip\_address/mask **via** ip\_шлюза – поменять gateway у маршрута

**sudo ip route del** ip\_address/mask – удаление маршрута

**sudo ip route del default** – удаление шлюза по умолчанию (маршрута во внешнюю сеть)

**sudo ip route flush table main** – очистить всю routing таблицу (после этого перезапускаем интерфейс, что бы восстановился дефолтный маршрут)

**Диагностика сети**

Проходя стек протоколов уровень за уровнем (вверх или вниз), проверяйте следующее:

* Есть ли физическое соединение и светится ли индикатор связи?
* Правильно ли сконфигурирован сетевой интерфейс?
* Отображаются ли в таблицах маршрутизации адреса других компьютеров?
* Есть ли брандмауэр на вашем локальном компьютере?
* Если брандмауэр есть, проходят ли через него ICMP-пакеты утилиты ping и от­веты?
* Находит ли команда ping узел localhost (127.0.0.1)?
* Находит ли команда ping другие компьютеры локальной сети по IР-адресам?
* Правильно ли сконфигурирована служба DNS?
* Находит ли команда ping другие компьютеры локальной сети по именам?
* Находит ли команда ping компьютеры в другой сети?
* Работают ли высокоуровневые сетевые утилиты, такие как telnet или ssh?
* Вы правда проверили брандмауэры?

Определив, на каком уровне возникает проблема, не спешите с выводами. Подумайте сначала, как отразятся последующие проверки и вносимые изменения на других сетевых службах и компьютерах. Далее используем **АККУРАТНО** следующие команды:

**ping** ip\_address – посылает ICMP пакеты для проверки доступности узла

**traceroute/tracepath** ip\_address – показывает путь пакета до адреса назначения

**mtr** ip\_address – утилита, которая объединяет ping + traceroute

**netstat** - инструмент командной строки для мониторинга как входящих, так и исходящих сетевых подключений, а также просмотра таблиц маршрутизации, статистики интерфейса и т.д Показывает какие порты открыты и какие программы их используют

**netstat -i** – Отображение транзакций пакетов сетевого интерфейса, включая как отправляемые, так и принимаемые пакеты с размером MTU.

**netstat -rn** – отображение таблицы IP маршрутизации

**netstat -an | grep listen** – показать открытые порты

**netstat -tunpl** – показать TCP, UDP, использовать численные IP адреса бес попытки определить его имя, показать PID и имя программы и сокет к которому она относится, показать только LISTENING сокеты

**netstat -tupn –** показать TCP,UDP, показать PID и имя программы и сокет к которому она относится, использовать численные IP адреса бес попытки определить его имя

**netstat -s** – отображает статистику по каждому протоколу (TCP, UDP, ICMP, IP)

**dig** domainName –(domain information groper) многофункциональный инструмент для опроса DNS-серверов. Она позволяет получить больше информации о конкретном домене, для того чтобы, например, узнать используемые им IP-адреса. Этот инструмент может оказаться полезным сетевым администраторам для выявления неисправностей DNS

**whois** [ip\_address][domain] – получение в текстовом виде регистрационных данных о владельцах IP адресов и доменных имен (главным образом, их контактной информации)

**netcat** host port – (аналог nc) утилита Unix, позволяющая устанавливать соединения TCP и UDP, принимать оттуда данные и передавать их

**nmap** - утилита, предназначенная для разнообразного настраиваемого сканирования IP-сетей с любым количеством объектов, определения состояния объектов сканируемой сети

**tcpdump –** утилита позволяющая перехватывать и анализировать сетевой трафик, проходящий через компьютер, на котором запущена данная программа.

**nslookup** – утилита, предоставляющая пользователю интерфейс командной строки для обращения к системе DNS. Позволяет задавать различные типы запросов и опрашивать произвольно указываемые сервера

**host** [domain][ip\_address] – утилита **CLI** для выполнения **DNS**-запросов, которые переводят имена доменов на **IP**-адреса и наоборот

**telnet** - это сетевая утилита, которая позволяет соединиться с удаленным портом любого компьютера и установить интерактивный канал связи, например, для передачи команд или получения информации. Используется, для тестирования сети, проверки портов, а также для взаимодействия с различными IoT устройствами и роутерами

**sar** – утилита, которая является частью Systat пакета, которую можно использовать для просмотра данных производительности на сервере