du – disk usage (в конкретной директории)

df – по всем файлам файловой системы

Файловая система резервирует 1% памяти под inode

df -i посмотреть inode

sudo dmidecode – информация о системе(железо)

Важно уметь смотреть информацию об интерфейсах(добавить/включить, заставить работать 2 интерфйеса вместе, режим работы, макс. скорость)

ip a список интерфейсов

sudo ethtool посмотреть инфу про интерфейс

duplex: Full – режим при котором пакеты бегут одновременно в 2 стороны

при обычном режиме пакеты бегут сначала в одну, потом в другую сторону(это долго)

auto-negotiation: on – выбор скорости режиме дуплекса(сетевые карты сами согласуют режим)

link detected: yes – подключен кабель или нет(тут да)

Потоки. Их перенаправление.

В Линукс все файл. При запуске процесс пишет логи в какой-то файл или на терминал. Процесс обращается к файлу по файловому дескриптору(абстракция). Зарезервированные файловые дескрипторы: 0, 1, 2 stdin stdout stderr. Зарезервированные потому что каждому процессу, зачастую, нужен ввод, процесс должен записать свой сгенерированный вывод.

cat – цикл считывания с stdin данных, ctrl+d завершить ввод, затем с помощью stdout происходит перенавпрление этих данных в консоль.

Ошибки перенаправляются в файловый дескриптор 2 (stderr), не смотря на то что мы видим ошибку в консоли.

ls –la/proc/[id]/fd – используемые процессом файловые дескрипторы.

tail log.txt – посмотреть последние 10 записей в файле.

Мы можем перенаправить вывод одного процесса в файл.

ls > file.txt (если файла нет, то он создается)

При повторном перенаправлении в тот же файл, данные перезапишутся

Чтобы данные добавлялись в конец файла:

ls >> file.txt

ls test123 > file.txt (файла test123 не существует, появилась ошибка, в file.txt ничего не записалось)

ls test123 2> file.txt (цифрой 2 мы указали номер дескриптора с которого перенаправляем вывод)

ls test123 &>> std (& - перенаправить stdout и stderr)

ls test123 >> std 2>&1 (2 поток соединить с 1 и все это дописать в конец файла)

Трубы (pipe).

cat log.txt | grep http – pipe (|) передает вывод программы слева на ввод программы справа.

Пакеты.

Бинари, конфиги, библиотеки, логи программы - все это лежит в разных местах(а еще все это может использовать другие зависимости). Такое сложность отслеживать, обновлять и тп. Для этого в Линукс есть пакеты. Пакет – это архив, в котором в определенной иерархии расположены файлы и есть определенные инструкции по тому, как нужно установить программу.

Пакетный менеджер – программа, которая устанавливает, удаляет и проводит различные манипуляции с пакетами(программами).

dpkg – нужна для установки, удаление пакетов, посмотреть зависимости.

supo apt download htop (htop – программа которую скачиваем)

sido dpkg –i file.deb – установить программу (file.deb – архив с бинарями, конфигами и тп)

dpkg –L htop – листнг пакета(что установилось и куда)

dpkg –r htop - удалить программу(но оставить конфиги)

dpkg –P htop - удалить программу полностью

dpkg не решает проблему с зависимостями, вернет ошибку если нет какой-то библиотеки.

dpkg –force – принудительно установить программу, если нет необходимой библиотеки в системе.

apt – устанавливает пакеты с систему. Два отличия: она ставит пакеты из сети, а не с диска и разрешает зависимости.

sudo apt install nginx

sudo apt policy nginx – откуда будут скачиваться пакеты.

Так же есть возможность прописать другие репозитории для скачивания, но там еще надо обновить базу, подписывать ключ и тп, это все на сайте с пакетом обычно описано.

sudo apt search nginx– поиск пакетов по имени

sudo apt show nginx – описание (название пакета, версия и тп).

sudo apt clean – удалить скачанные пакеты (качаем с сервера перед установкой в кэш)

sudo apt install nginx - можно обновить определенный пакет, если он уже установлен

sudp depends nginx – зависимости

Парсинг.

sudo head log.txt – первые 10 строк файла.

sudo head -5 log.txt – первые 5 строк файла.

(тоже самое верно для tail)

sudo grep 'HTTP/1.1" 404' log.txt – строки с 'HTTP/1.1" 404'

sudo grep -Pvi 'HTTP/1.\d+" 200' log.txt – убираем все строки, где статус 200, ключ –P это значит используем паттерн(регулярку), -v убираем где статус 200, -i не чувствителен к регистру.

sudo grep -Po '\d+\.\d+\.\d+\.\d+' log.txt – выбираем только ip адреса. Ключ –o выбирает только то, что указано в паттерне.

sudo awk '{print $1, $2}' log.txt – печатает 1 и 2 столбец лога(по умолчанию столбцы разделены пробелом)

history – история команд

ctrl+r – поиск в истории команд

sudo awk '{print "bad:", $2, "url: ", $8}' log.txt – информативный вывод с 2-ой и 8-ой колонки

sudo awk '{print $2}' log.txt | sort | uniq – сортируем ip адреса по возрастанию и выбираем уникальные.

sudo awk '{print $2}' log.txt | sort | uniq -c | sort -k 1 - подсчитываем кол-во обращений с адреса и сортируем это кол-во

sudo awk –F ‘delimiter’ – можно изменить разделитель столбцов.

sudo awk '{print $(NF-1)}' log.txt – печать предпоследнего столбца

ps -aufx | awk ' NR==1 || /root/' - вывести таблицу со строками, где есть ‘root’ и вывести подписи столбцов

sed -i 's/11.22.33.44/192.168.1.1/g' log.txt – заменить ип 11.22.33.44 на 192.168.1.1 Ключ –i означает, что надо изменить именно в файле, а не изменить вывод, флаг g заменяет все вхождения.

grep –rni 10.22.22.1 . – найти в директории . (текущая) строку 10.22.22.1

Ключ –r – делаем рекурсивно, -n выводим номер строки, -i не чувствителен к регистру.

ls -1 | xargs -I{} sed -i 's/11.22.33.44/192.168.1.1/g' {} – передаем в xargs массив с файлами и вызываем функцию sed для каждого файла из ls -1

ls -1 – получим список файлов в столбик, -I – указание массива, который мы приняли, {} – массив файлов, который принял xargs.

sudo grep –v ‘^#’ log.txt – убрать все строки, начинающиеся с #

Сети

Модель OSI – взаимодействие сетевых протоколов друг с другом. Их 7 уровней.
1 уровень физический, кабели, бегают биты байты.
2 уровень канальный, там где бегают mac’и, работает поверх физического уровня.

Сетевой мост – сетевое устройство канального уровня, которое предназначено для объединения сегментов компьютерной сети в единую сеть. (например объединить 2 локальных сети.)

NAT – преобразование сетевых адресов. Это механизм, позволяющий преобразовывать локальный ip адрес во внешний ip при запросе из локальной сети в сеть Интернет и наоборот.

Список интерфейсов – ip address

Менеджер сетевых настроек – netplan, ifupdown

dhcp – протокол, который позволяет автоматически получать от маршрутизатора сетевые настройки.

LOOPBACK – интерфейс, который заворачивает весь трафик, который приходит на локальный ip 127.0.0.1, обратно в сервер

Параметры из ip addr:

qdisc – какой алгоритм обрабатывает поток трафика, state - физически подключен интерфейс или нет, qlen - длина очереди, если больше указанной, пакеты будут уничтожаться.

link/ether – протокол подключения (у нас Ethernet)

inet – обозначает ipv4, дальше указывается сам ip

brd – широковещательный адрес.

netplan и ifupdown

конфиги:

```
sudo vim /etc/netplan/00
sudo vim /etc/network/interface

sudo netplan try
sudo netplan apply
```

reboot – для проверки настройки ifupdown

cat /lib/systemd/system/networking.service – файл, где прописан запуск ifupdown

l2 – означает что маршрутизация всего трафика осуществляется на уровне mac адреса.

MAC-адрес — уникальный идентификатор, присваиваемый каждой единице активного оборудования или некоторым их интерфейсам в компьютерных сетях Ethernet.

Сетевой коммутатор (network switch) – это устройство, используемое в сетях передачи пакетов, предназначенное для объединения нескольких сегментов. В отличие от маршрутизатора (router) коммутатор работает на канальном уровне модели OSI, что и определяет главные различия между ними.

switch смотрит с какого mac адреса и на какой mac адрес отправлен пакет, ip не смотрит.

Address Resolution Protocol (ARP- Протокол определения адреса) - протокол канального уровня, использующийся в основном для преобразования IP-адреса в MAC-адрес.

ip neigh – таблица соответствия ip и mac-адреса

Так как у нас может быть много интерфейсов эти интерфейсы могут вести в разное оборудование, в разные сети, нужна таблица маршрутизации, чтобы пакет дошел в ту сеть, куда нам нужно (на какой интерфейс послать, на какой ip послать и с какого ip послать)

ip route – таблица маршрутизации

Все пакеты по умолчанию буду отправлены через default маршрут, если не совпадают с перечисленными.

Сетевой шлюз (англ. *Gateway*) — аппаратный маршрутизатор для сопряжения компьютерных сетей, использующих разные протоколы (например, локальной и глобальной).

адрес gateway узнается так: посылается на brd(широковещательный адрес) arp-запрос(его принимают все участники сети), и роутер посылает свой mac.

sudo ip link set up enp0s8 – опустить интерфейс. Может понадобится, если на интерфейсе ошибки или надо перенастроить.

Маска – размер сети. Чем маска больше, тем меньше хостов в сети. и наоборот.

sudo ip a add 192.168.1.187/32 dev lo – добавить адрес в интерфейс loopback

Вешать на loopback лучше, потому что он всегда поднят.

TCP – адресация на уровне портов адресов. (откуда порт, кому порт и сами данные)

IP Packet адресация на уровне ip

Ethernet frame – адресация на уровне mac адреса (откуда, куда, данные)

ip route – main таблица маршрутизации

ip r show table local – таблица маршрутизации loopback

Здесь принимается решение через какой интерфейс, маршрутизатор отправлять трафик, исходя из условия куда отправляется трафик.

Чем metric меньше, тем приоритетнее маршрут

host vk.com - узнать ip домена

ip route get 87.240.190.78 – получить маршрут по ip адресу

sudo ip route add 87.240.139.194 dev enp0s8 – перенаправлять указанный адрес через интерфейс enp0s8

Что такое интернет?

Это большая локальная сеть, куча компов, которые соединены кабелями. Используется динамическая маршрутизация. Мы у себя на локалке настраивали статическую маршрутизацию.

TCP/UDP

Маршрутизация происходит с помощью портов. TCP идет с подтверждением доставки сообщения и соединения. У UDP такого нет, просто шлем данные, не знаем дошли или не дошли данные.

UDP используется в видеосвязи. В игровых серверах, где много игроков и много трафика также используется UDP для экономии трафика, т.к отправляется только 1 пакет.

В TCP идёт тройное рукопожатие (чтобы установить соединение идут уже 3 пакета + обратно приходит пакет).

Чтобы смотреть трафик – утилита tcpdump

sudo tcpdump -vvnni enp0s3 host vk.com and tcp port 443 – слушаем трафик на vk.com 443 порт. Ключ –vv детальный вывод, -nn вместо доменов ip адреса, -i интерфейс.

telnet vk.com 443 – установить tcp соединение

sudo tcpdump –vvnni enp0s3 udp port 53 and host 8.8.8.8 слушаем трафик через 8.8.8.8

Динамическая маршрутизация — вид маршрутизации, при котором таблица маршрутизации редактируется программно. В случае UNIX-систем *демонами маршрутизации*.

Демоны маршрутизации обмениваются между собой информацией, которая позволяет им заполнить таблицу маршрутизации наиболее оптимальными маршрутами. Протоколы, с помощью которых производится обмен информацией между демонами, называется *протоколами динамической маршрутизации*.

AS(Автономная система в интернете) — это система IP-сетей и маршрутизаторов, управляемых одним или несколькими операторами, имеющими единую политику маршрутизации с Интернетом.

traceroute – показывает маршрут

По умолчанию traceroute работает с udp. Он высылает сразу 3 udp пакета. TTL – время жизни пакета, каждый маршрутизатор, через который будет проходить пакет уменьшает его на единицу. При TTL=0 пакет уничтожается, а отправителю отсылается сообщение Time Exceeded.. traceroute отправляет пакет с TTL=1 и смотрит адрес ответившего узла, дальше TTL=2, TTL=3 и так пока не достигнет цели. Каждый раз отправляется по три пакета и для каждого из них измеряется время прохождения. Пакет

отправляется на случайный порт, который, скорее всего, не занят. Когда утилита traceroute получает сообщение от целевого узла о том, что порт недоступен трассировка считается завершенной.

traceroute –I – ICMP пакеты

traceroute –T – TCP пакеты

whois ip – информация о ip адресе

`sudo systemctl status bird – информация о статусе демона bird (динам. маршрутизация)`

`sudo tcpdump -vvnni enp0s3 udp port 520 – слушаем 520 порт интерфейс enp0s3`

`iptables – таблица фаервола.`

Все TCP-соединения начинаются с тройного рукопожатия. До того как клиент и сервер могут обменяться любыми данными приложения, они должны «договориться» о начальном числе последовательности пакетов, а также о ряде других переменных, связанных с этим соединением. Числа последовательностей выбираются случайно на обоих сторонах ради безопасности.

SYN

Клиент выбирает случайное число X и отправляет SYN-пакет, который может также содержать дополнительные флаги TCP и значения опций.

SYN ACK

Сервер выбирает свое собственное случайное число Y, прибавляет 1 к значению X, добавляет свои флаги и опции и отправляет ответ.

ACK

Клиент прибавляет 1 к значениям X и Y и завершает хэндшейк, отправляя ACK-пакет.