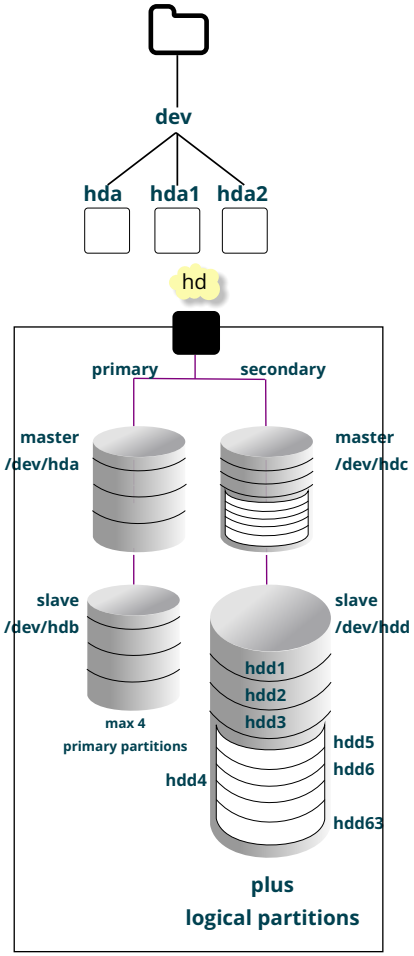
Preparing for installation

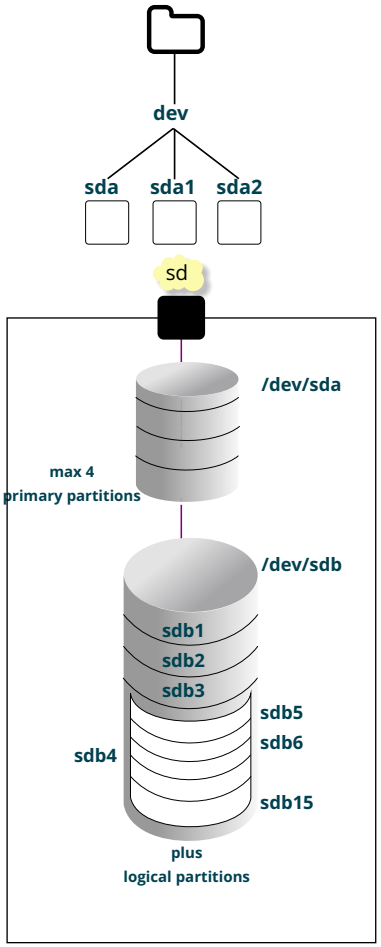
P-ATA

(major=3, minor=0-63)



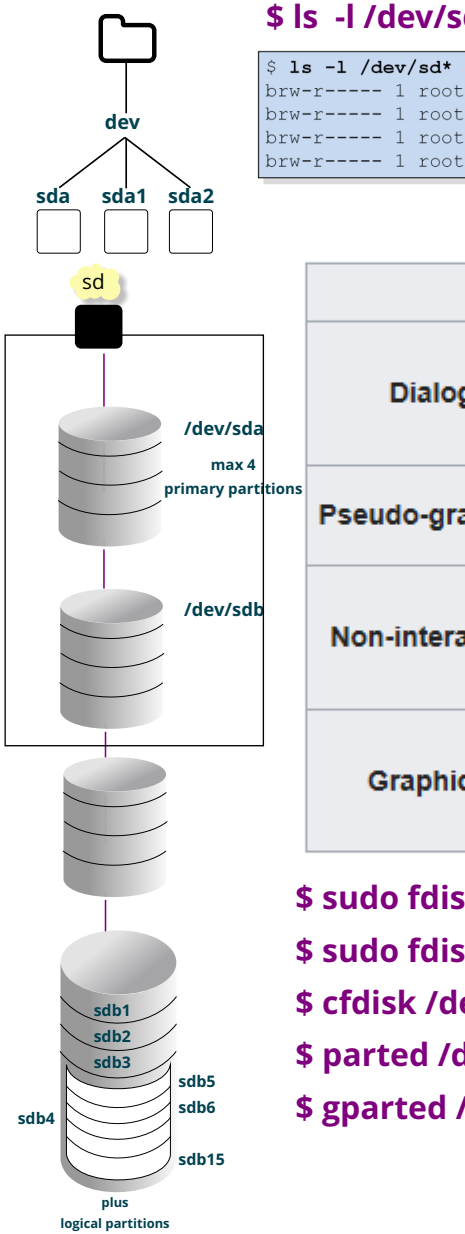
S-ATA

(major=8, minor=0-15)



SCSI

(major=8, minor=0-15)



$ ls -l /dev/sd*

```
$ ls -l /dev/sd*
brw-r----- 1 root disk 8,  0 May 15 04:11 /dev/sda
brw-r----- 1 root disk 8,  1 May 15 04:11 /dev/sda1
brw-r----- 1 root disk 8, 16 May 15 04:13 /dev/sdb
brw-r----- 1 root disk 8, 17 May 15 04:13 /dev/sdb1
```

Partitioning tools

	MBR	GPT
Dialog	fdisk parted	fdisk **gptfdisk** gdisk parted
Pseudo-graphics	cfdisk	cfdisk cgdisk
Non-interactive	sfdisk parted	sfdisk sgdisk parted
Graphical	GParted gnome-disk-utility partitionmanager	GParted gnome-disk-utility partitionmanager

$ sudo fdisk -l /dev/sdb

$ sudo fdisk /dev/sdb

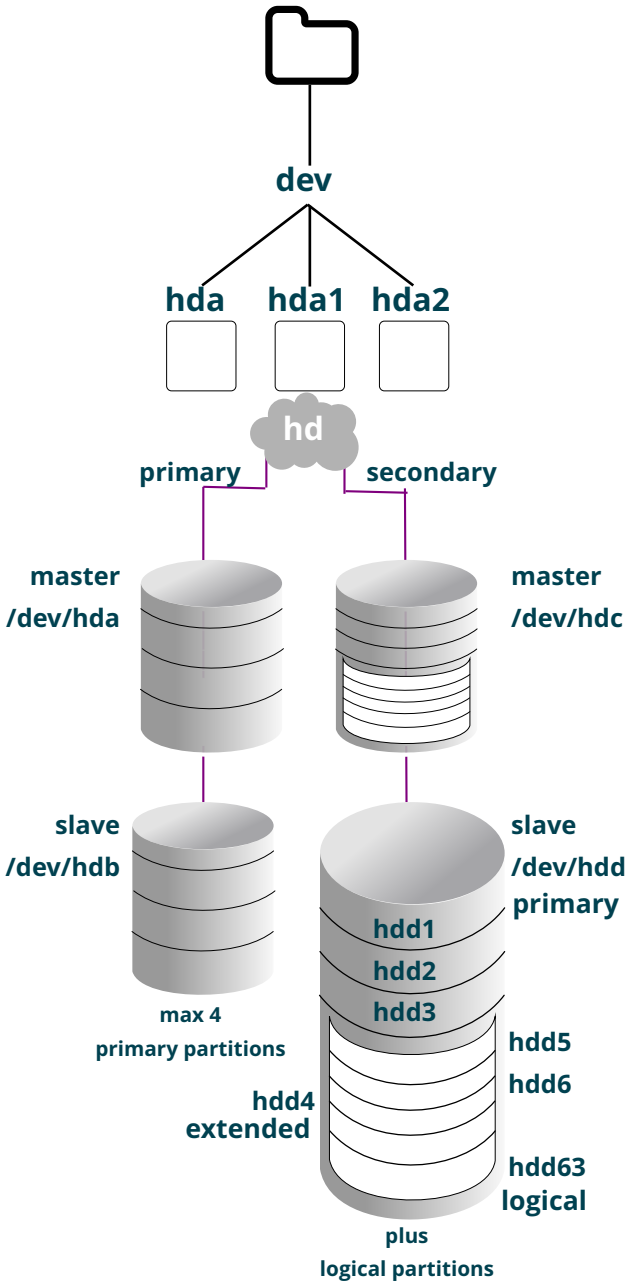
$ cfdisk /dev/sdb

$ parted /dev/sdb

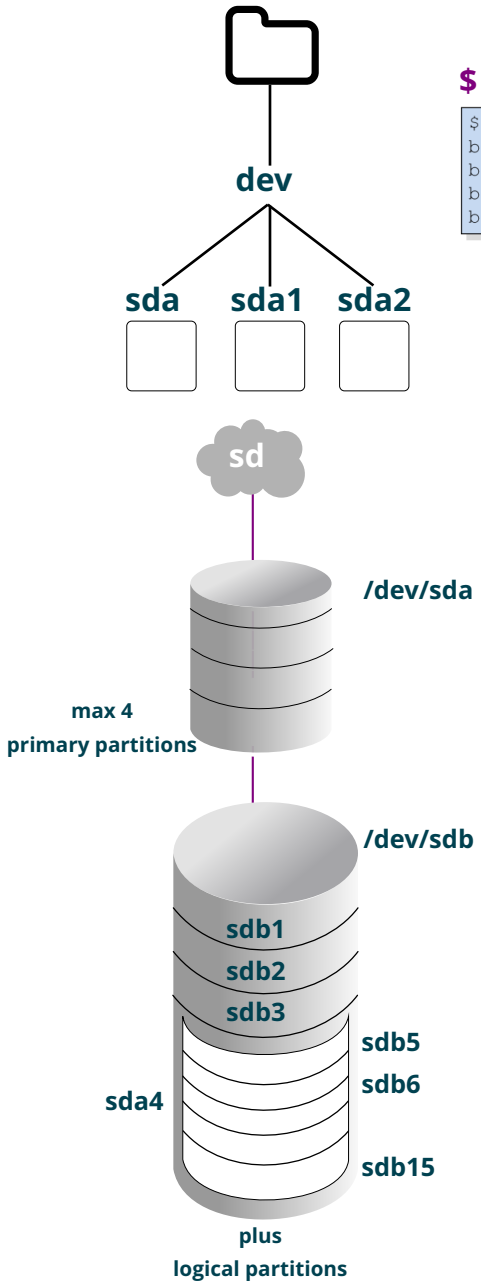
$ gparted /dev/sdb

Preparing for installation

P-ATA (major=3)



S-ATA/SCSI (major=8)



$ ls -l /dev/sd*

$ ls -l /dev/hd*

$ ls -l /dev/vd*

```
$ ls -l /dev/sd*
brw-r----- 1 root disk 8,  0 May 15 04:11 /dev/sda
brw-r----- 1 root disk 8,  1 May 15 04:11 /dev/sda1
brw-r----- 1 root disk 8, 16 May 15 04:13 /dev/sdb
brw-r----- 1 root disk 8, 17 May 15 04:13 /dev/sdb1
```

Partitioning tools

	MBR	GPT
Dialog	fdisk parted	fdisk **gptfdisk** gdisk parted
Pseudo-graphics	cfdisk	cfdisk cgdisk
Non-interactive	sfdisk parted	sfdisk sgdisk parted
Graphical	GParted gnome-disk-utility partitionmanager	GParted gnome-disk-utility partitionmanager

$ sudo fdisk -l /dev/sdb

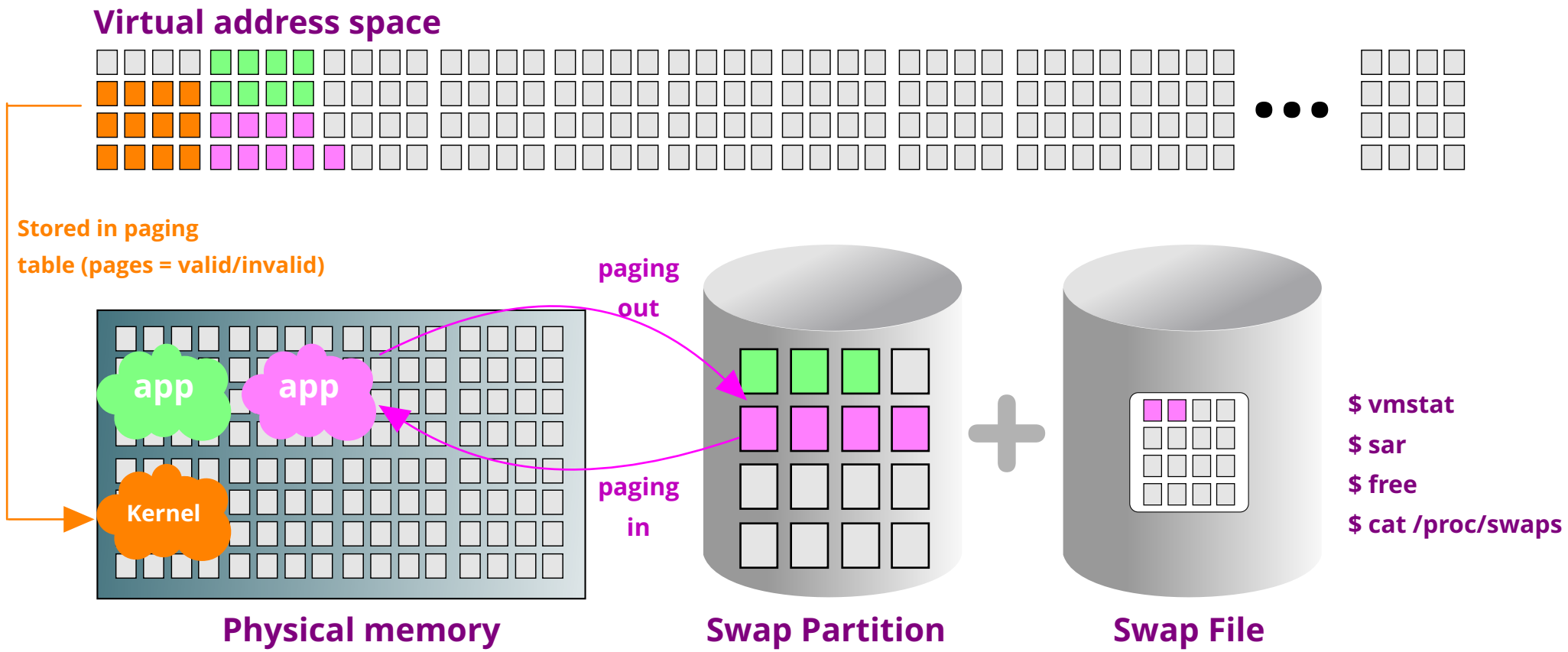
$ sudo fdisk /dev/sdb

$ cfdisk /dev/sdb

$ parted /dev/sdb

$ gparted /dev/sdb

Preparing for installation

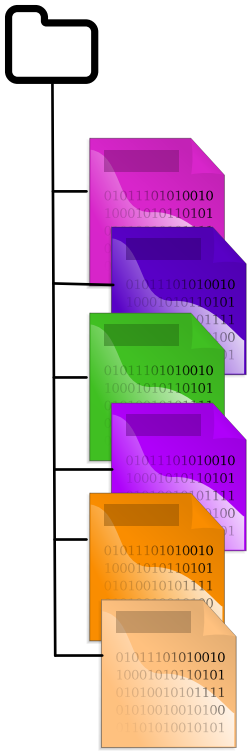


Adding Swap space

```
$ sudo dd if=/dev/zero of=/spare/swapfile  count=50000
$ sudo chmod 600 /spare/swapfile          # Protect permission of paging data
$ sudo mkswap /spare/swapfile             # Make the file as a swap device
$ sudo swapon  /spare/swapfile            # Enable swapfile for paging
$ free                                    # Display free and used memory
$ swapon -s                               # Display swap summary info
```

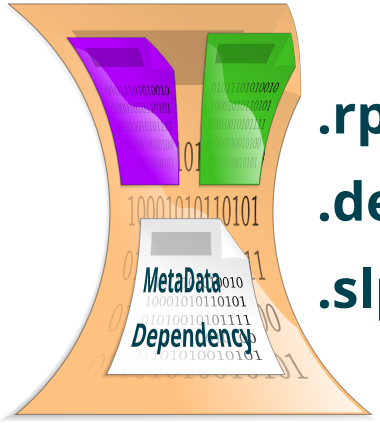
Software Package

Collection of all
files & dirs required
for software to execute



- Executable/binary
- Libraries
- Install/backout scripts
- Log files
- Device files
- Data files
- Config files
- Temp files
- READMEs
- Man Pages
- Docs
- Meta Data

S/w distributed as a
Package = Compressed Archive



.rpm = RHEL/SUSE/Fedora/CentOS

.deb = Debian/Ubuntu/Kali/Mint

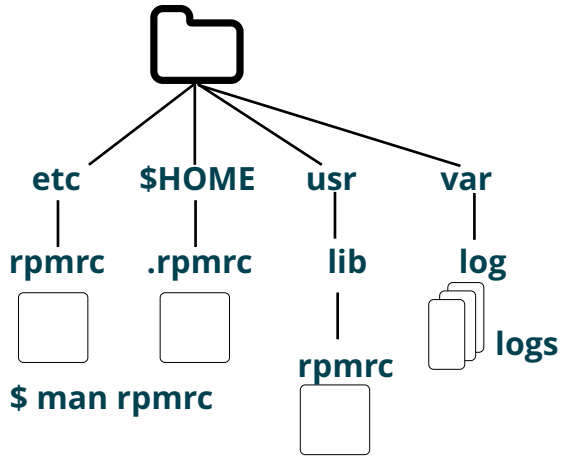
.slp

CLI/TUI/GUI tools

RPM - Redhat Package Manager

Naming Convention = name-version_release.lang.architecture.rpm

```
$ rpm --help
$ rpm -qa | more          # Query all
$ rpm -qa | grep 'bash'   # Query pkg
$ rpm -q bash             # Query
$ rpm -qi bash            # Query info
$ rpm -qs httpd           # Query status
$ rpm -qc httpd           # Query configs
$ rpm -ql httpd           # Query list files
$ rpm -qd awk             # Query docs
```




```
$ rpm -q --whatprovides /etc/passwd   # Query package that requires file
$ rpm -q --whatrequires bash          # Query packages use package
```

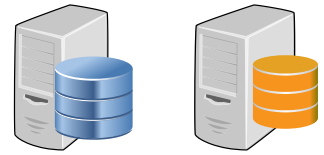
```
$ sudo rpm -ivh --test httpd-2.4.5-87.el7.centos.x86_64.rpm   # Simulate install, hash progress, verbose
$ sudo rpm -ivh httpd-2.4.5-87.el7.centos.x86_64.rpm          # Install, hash progress, verbose
$ sudo rpm -Uvh httpd-2.4.6-93.el7.centos.x86_64.rpm          # Update, hash progress, verbose
$ sudo rpm -evh --test httpd                                  # Simulate erase, hash progress, verbose
$ sudo rpm -evh httpd                                         # Erase, hash progress, verbose
```

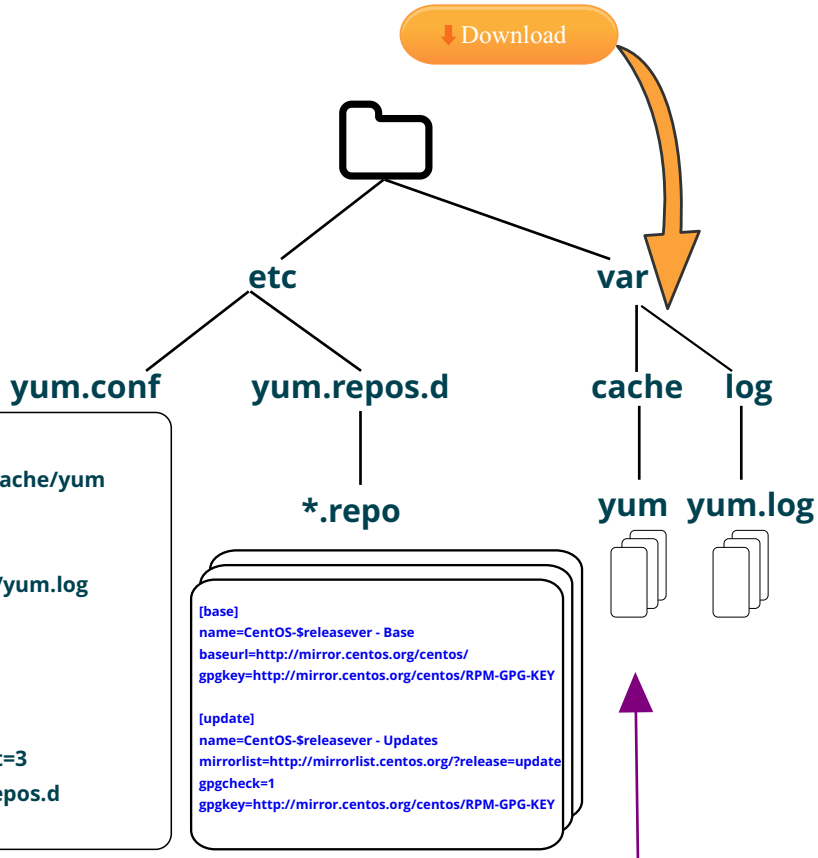
But no dependency management

YUM - Yellowdog Updater, Modifed

Better Dependency management and logging.

```
$ yum --help
$ yum repolist
$ yum list all              # List all installed and available
$ yum list available        # List available packages
$ yum list installed        # List installed packages
$ yum list updates          # List available updates
$ yum list recent           # List recent updates
$ yum history {list|info|redo|rollback|undo}   # Manage transactions
$ yum info bind             # Display package info
$ yum provides /etc/passwd # What package provides file
$ yum requires sed          # What package requires this

$ yum search httpd                  # Search for related packages
$ sudo yum install {-y|-n} httpd    # Install package + dependencies
$ sudo yum update {-y|-n} httpd     # Update package
$ sudo yum update                   # Update all - be CAREFUL!
$ sudo yum update kernel            # Update kernel on RHEL
$ sudo yum install --nogpgcheck -y httpd # No Sig checking
$ sudo yum remove {-y|-n} httpd     # Remove installed package.
$ yum group list
$ yum group info "Desktop Tools"
$ sudo yum group install "Desktop Tools"
$ sudo yum group remove "Desktop Tools"
$ sudo yum shell
```






```
$ sudo yum clean all
$ sudo yum clean packages
```

```
$ sudo yumdownloader htop httpd sed
$ sudo yum install --downloadonly --downloaddir='.' htop httpd sed
$ sudo yum localinstall cowsay.rpm
$ sudo yum install -y epel-release
```

Now try replacing yum with dnf = Dandified yum (RHEL 8) - better and replaces yum!

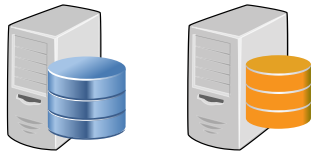
DNF - Dandified Yum

Introduced in Fedora 18 (2013), default in 20 (2015) and RHEL 8 (2020)

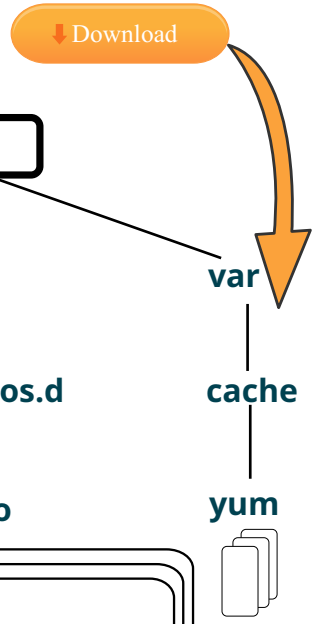
Improved dependency management (hawkey/libsolv).

Optimised memory efficiency

Support in Python 3



Redhat Centos




```
$ dnf  --help
$ dnf repolist; dnf info bash
$ dnf list all              # List installed all installed and available
$ dnf list available        # List available packages
$ dnf list installed        # List installed packages
$ dnf list updates          # List available updates
$ dnf list recent           # List recent updates
$ dnf history {list|info|redo|rollback|undo}     # Manage transactions
$ dnf info bind             # Display package info
$ dnf provides /etc/passwd  # What package provides file

$ dnf search httpd              # Search for related packages
$ sudo dnf install {-y|-n} httpd    # Install package + dependencies
$ sudo dnf update {-y|-n} httpd     # Update package
$ sudo dnf update               # Update all - be careful!
$ sudo dnf update kernel            # Update kernel on RHEL
$ dnf group list
$ dnf group list
$ dnf group info “Desktop Tools”
$ sudo dnf group install “Desktop Tools”
$ sudo dnfgroup remove “Desktop Tools”
$ sudo dnf shell
```

```
[main]
arch=x86_64
gpgcheck=False
best=False
installonlypkgs=kernel
install_only_limit=3
keepcache=False
proxy=http://myproxy.com/
throttle=0
clean_requirements_on_remove=True
```

```
[base]
name=CentOS-$releasever - Base
baseurl=http://mirror.centos.org/centos/
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY

[update]
name=CentOS-$releasever - Updates
mirrorlist=http://mirrorlist.centos.org/?release=update
gpgcheck=1
gpgkey=http://mirror.centos.org/centos/RPM-GPG-KEY
```

```
$ sudo dnf clean all
$ sudo dnf clean packages
```

```
$ dnf --downloadonly htop httpd sed
$ sudo dnf localinstall cowsay.rpm
```

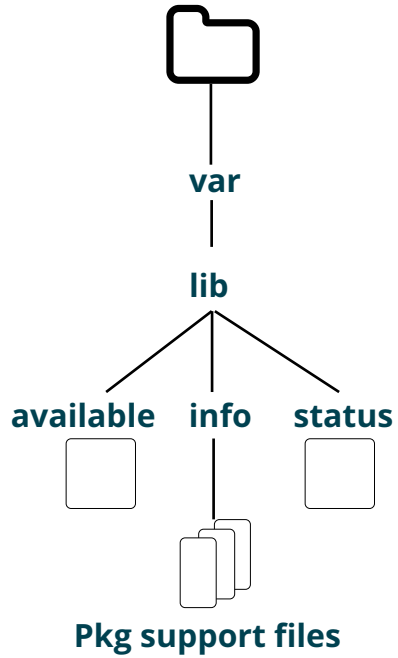
```
$ sudo dnf install -y epel-release
```

DPKG - Debian Package Manager

Naming Convention = name-version_release.architecture.deb

```
$ dpkg --help
$ dpkg -l | less          # List installed packages
$ dpkg -I httpd           # Package info
$ dpkg -s httpd           # Package status
$ dpkg -c httpd           # Package contents
$ dpkg -S bash            # Search for package file
$ dpkg -L bash            # List package libraries
```

```
$ sudo dpkg -i bash_5.0-4_i386.deb              # Install package
$ sudo dpkg -i --force-depends pkg.deb          # Install and force dependencies into warnings
$ sudo dpkg -i --force-overwrite bash_5.0-4_i386.deb  # Install and overwrite older files
$ sudo dpkg -i -R $HOME/packages                # Install package recursively
$ sudo dpkg --unpack bash_5.0-4_i386.deb        # Unpack package
$ sudo dpkg --configure bash                    # Configure package
$ sudo dpkg -r bash                             # Remove package
$ sudo dpkg -P bash                             # Purge package and configs
```

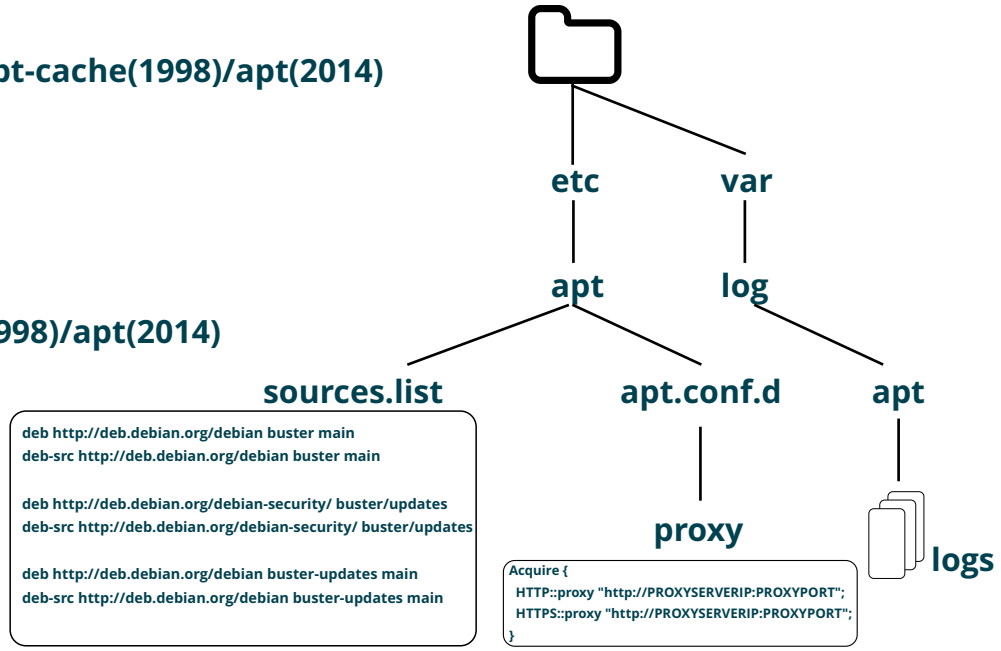


But no dependency management

APT - Advanced Package Tool

Better Dependency management and logging.

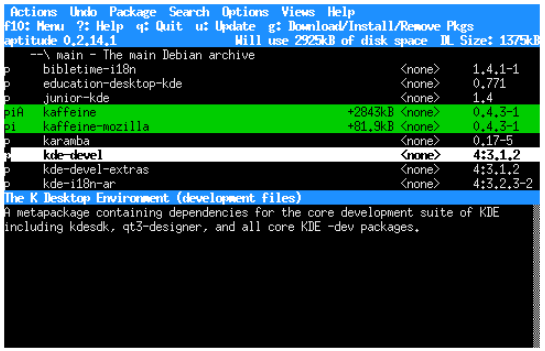
```
$ apt-cache search bash      # Search for related packages apt-cache(1998)/apt(2014)
$ apt show httpd             # Show package info
$ apt version httpd          # Show version info
$ apt depends bash           # Show package dependencies

$ sudo apt-get update        # Update all packages apt-get(1998)/apt(2014)
$ sudo apt upgrade           # Upgrade all packages
$ sudo apt install gawk      # Install package
$ sudo apt remove gawk       # Remove package
$ sudo apt purge gawk        # Remove package + Config files
$ sudo apt dist-upgrade      # Update to latest distribution
$ sudo apt clean             # Clean retrieved pkg cache
$ sudo apt autoclean         # Clean useless packages
```



Or use aptitude = front end to APT

Install software by system role/task.

```
$ sudo aptitude
```



Convert package between formats.

```
$ sudo alien --to-deb bash_5.0-4_i386.rpm
$ sudo alien --to-rpm bash_5.0-4_i386.deb
```

Installing from source code

Example: gitHub/SourceForge/FreshMeat/vendor

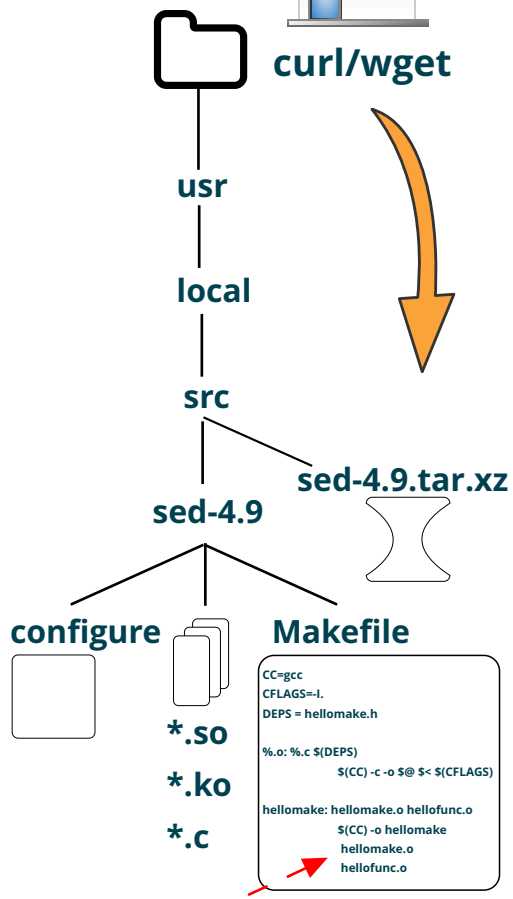
Sometimes the only way - but good luck with dependencies and pirates!

Requires development tools installed (and possibly a few others).

```
$ sudo apt -y install build-essential
$ sudo dnf group install “Development Tools”
```

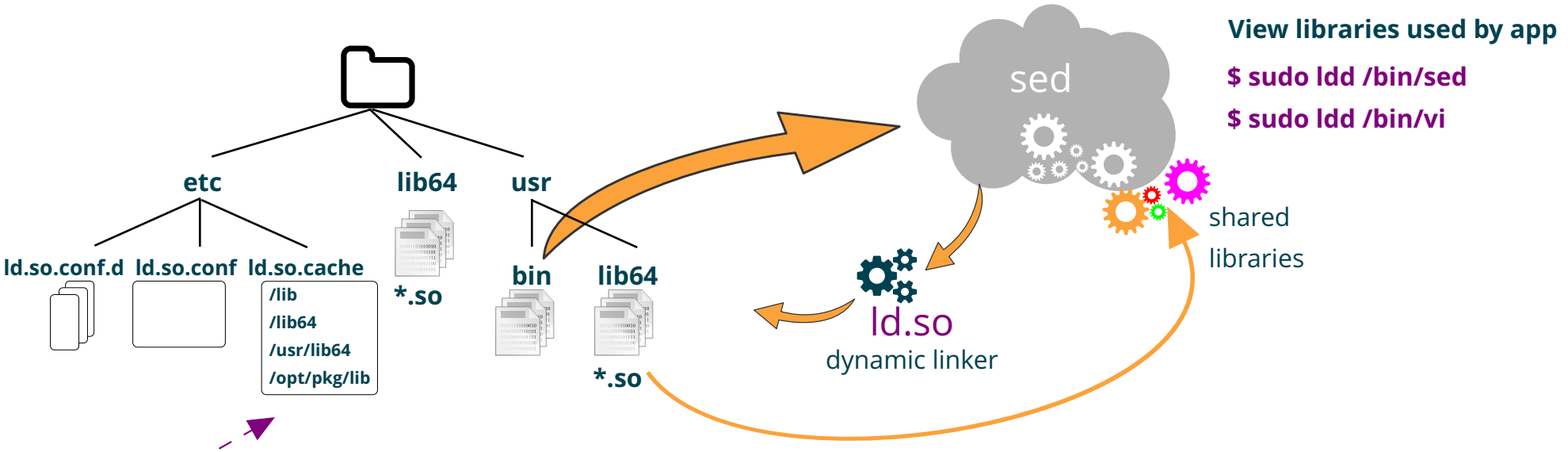
```
$ sudo -i; cd /usr/local/src
$ curl https://ftp.gnu.org/gnu/sed/sed-4.9.tar.xz -o sed-4.9.tar.xz    # Download src package
$ wget https://ftp.gnu.org/gnu/sed/sed-4.9.tar.xz -o logfile
```

```
$ sudo tar xJvf sed-4.9.tar.xz      # Extract source code files from tarball
$ cd sed-4.9                        # Change into package directory with Makefile
$ sudo ./configure                  # Run script to check dependencies and libraries
$ sudo make                         # Compile programs and libraries using Makefile
$ sudo make install                 # Copy binaries and libraries into file system
```



Read the README and Licence files

Shared Libraries



View libraries used by app

$ sudo ldd /bin/sed

$ sudo ldd /bin/vi

$ sudo ldconfig -v **# Print version**

$ sudo ldconfig -p **# Print list of cached libraries**

After installing software you may have to update cache

$ sudo ldconfig

Or

$ sudo vim /etc/ld.so.conf

$ sudo ldconfig

Or **${HOME}/.bash_profile**

```
export LD_LIBRARY_PATH="${LD_LIBRARY_PATH}:${HOME}/lib"
```

8 Defined Run Levels (SUS)

ssh

GNU Unix
terminal emulator

secure remote login

putty

$ who -r # Display Run level

$ runlevel # Display Run level

Systemd
Targets

5 — Default=Desktop/VM — graphical mode/target → graphical.target

4 — gdm — user-definable → CTRL-ALT-F1/7

3 — Default=Server/VM/Cloud instances — multi-user mode (net services)

net services: sshd, httpd, ftpd, dhcpd

2 — multi-user mode (client services) → multi-user.target → network.target

6 Linux Virtual Consoles

local services: getty, getty, getty

>login: CTRL-ALT-F2

>login: CTRL-ALT-F3

>login: CTRL-ALT-F6

1,s — single-user mode (root) = Admin → rescue.target

sys services: KERNEL

6=reboot

emergency.target
basic.target

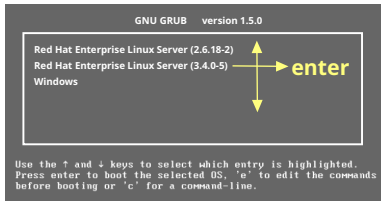
0 → poweroff.target

Bootloader=grub

POST

ON

Boot Management using BIOS and MBR what happens after



Initialise kernel

Kernel — Load kernel

PID=1 systemd

or

PID=1 initd

GRUB legacy stage 2

/boot/grub/menu.lst

/boot/grub/grub.conf

/boot/grub/devices

446 bytes

GRUBv1 stage1 | AA55

64 bytes

2 bytes

MBR

(hd0,0)

(hd0,1)

(hd0,2)

max 2TB

max 4 partitions

C/H/S Addressing

POST

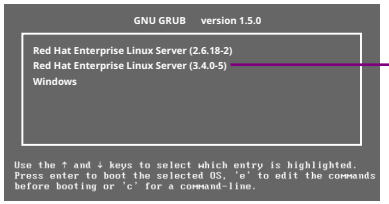
BIOS + cmos

set T.O.D

set boot device

```
default=0
fallback=1
timeout=10
color red/black green/blue
splashimage=(hd0,0)/grub/splash.xpm.gz
#hiddenmenu
                    location of /boot if separate partition
title Red Hat Enterprise Linux Server (2.6.18-2)
 root     (hd0,1)
 kernel   /boot/vmlinuz-2.6.18-2.el ro root=LABEL=/ rhgb quiet
 initrd   /boot/initrd-2.6.18-2.el.img

title Red Hat Enterprise Linux Server (3.4.0-5)
 root     (hd0,1)
 kernel   /boot/vmlinuz-3.4.0-5.el ro root=LABEL=/ rhgb quiet
 initrd   /boot/initrd-3.4.0-5.el.img

# section to load Windows
title Windows
 rootnoverify   (hd1,0)
 chainloader +1
```

make permanent change



Press 'e' to edit

temp change

Press Key to interrupt

Arrow Keys to select

$ man bootparams

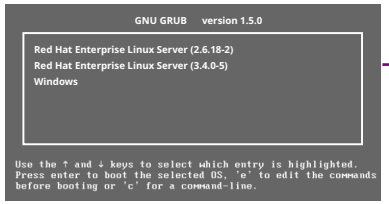
Boot Management using UEFI and GPT what happens after

Kernel

Initialise kernel

PID=1

systemd



Press 'e' to edit

ctrl-x to save

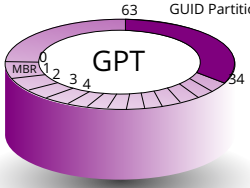
$ man bootparams

GRUB2

/boot can now be xfs/btrfs/ext4/lvm

/etc/grub2.cfg → /boot/efi/EFI/rocky/grub.cfg
/boot/grub2/grub.cfg

63 GUID Partitioning Table



(hd0,0)
(hd0,1)

LBA 64 bit addressing
= 2^64 * 512 = 2 Zbytes
max 128 partitions

POST → Menu

UEFI + cmos

Support for larger disks
/encryption/compression/LVM/diagnostics

0

```
# DO NOT EDIT THIS FILE. IT IS AUTO GENERATED!
menuentry 'CentOS Linux (3.10.0-514.el7.x86_64) 7 (Core)' --class centos --class
gnu-linux --class gnu --class os --unrestricted $menuentrygrubbid_option 'gnulinux-
3.10.0-514.el7.x86_64-advanced-10bac86a-a9be-478c-b63f-46d3ca061e03' {
        load_video
        insmod gzio
        insmod part_msdos
        insmod xfs
        set root='hd0,msdos1'
        if [ x$feature_platform_search_hint = xy ]; then
          search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos1 --hint-
efi=hd0,msdos1 --hint-baremetal=ahci0,msdos1 --hint='hd0,msdos1'  a52207e4-
01fd-4f12-98b7-681d56f21dc5
        else
          search --no-floppy --fs-uuid --set=root a52207e4-01fd-4f12-98b7-
681d56f21dc5
        fi
        linux16  /vmlinuz-3.10.0-514.el7.x86_64 root=/dev/mapper/cl-root ro
crashkernel=auto rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet LANG=en_US.UTF-8
        initrd16  /initramfs-3.10.0-514.el7.x86_64.img
}
```

1

```
menuentry 'CentOS Linux (0-rescue-9cdb9ab3246a4b3f9c0849ecd597f25e) 7
(Core)' --class centos --class gnu-linux --class gnu --class os --unrestricted
```

$ sudo grep '^menuentry' /boot/grub2/grub.cfg
$ sudo grubby --info ALL # Centos/RHEL 8


```
GRUB_TIMEOUT=5
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="rd.lvm.lv=fedora_fedora25vm/root
edora_fedora25vm/swap
dora_fedora25vm/usr rhgb quiet"
E_RECOVERY="true"
E_BLSCFG=true
```

NOW run grubby to add to
/boot/loader/entries

00_header
10_linux
20_os_prober
40_custom

Edit this for a custom entry!

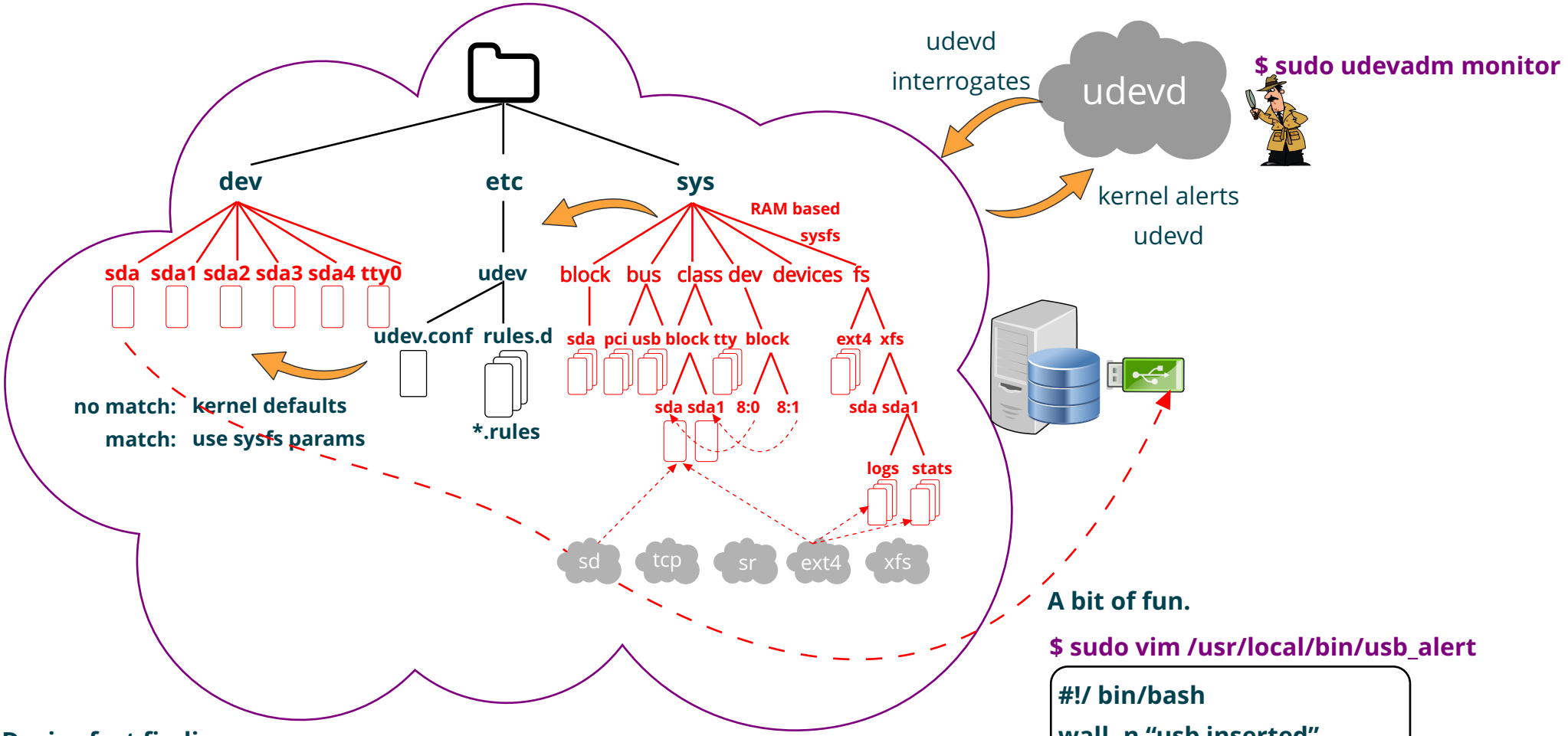
copy & paste

Run one of these to update Menu Entries
$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
$ sudo grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg
$ sudo update-grub # Debian/Ubuntu

Run this to set Default Boot instead of editing file.
$ sudo grub2-set-default 1 → /boot/grub2/grub.env
$ sudo grub2-editenv list

$ grubby --default-kernel
$ grubby --default-index
$ sudo grubby --set-default /boot/vmlinuz-VERS.x86_64
Run this command instead of creating Custom File!
$ sudo grubby --add-kernel=/boot/vmlinuz-VERS.x86_64
--initrd=/boot/initramfs-VERS.x86.img
--title="entry_title" --copy-default

Handling hardware - Linux PnP



Device fact finding

```
$ lsblk /dev/sda1                # List block information
$ lsblk -o name,size,ro,type,uuid,mountpoint
$ sudo blkid /dev/sda1           # Display block UUID

$ udevadm info -a -p /sys/block/sda/sda1 | grep 'serial'    # Display device properties
$ sudo systool
$ cat /proc/modules  /proc/partitions
$ lspci -vv
```

A bit of fun.

```
$ sudo vim /usr/local/bin/usb_alert
```

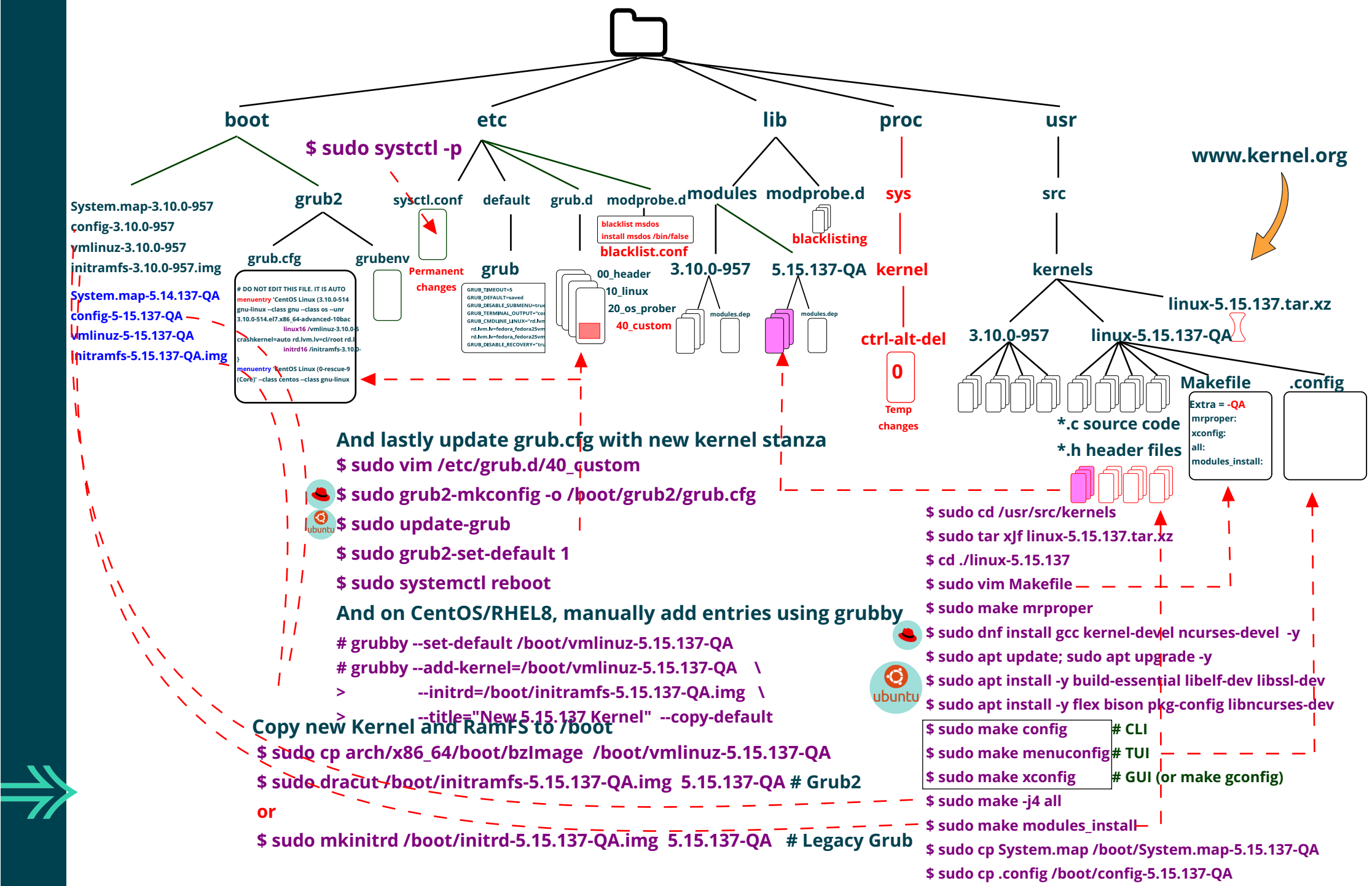
```
#!/ bin/bash
wall -n "usb inserted"
```

```
$ sudo chmod +x /usr/local/bin/usb_alert
$ sudo udevadm monitor
$ udevadm info -a -n /dev/sdb | more
$ sudo vim /etc/udev/rules.d/50-local.rules
```

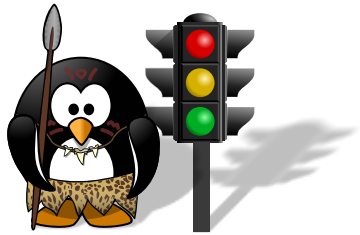
```
SUBSYSTEM=="block", ACTION=="add",
  RUN+="/usr/local/bin/usb-alert"
```

```
$ sudo udevadm control --reload
```

Compiling a Kernel - and creating alternate boot environment



Boot Management - System Service Control using SYS V initd



PID=1

initd → /etc/inittab

```
id:3:initdefault:
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6
```

Kernel

GRUB legacy stage 2

446 bytes

GRUBv1 stage1 | AA55 | 64 bytes | 2 bytes

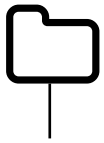
MBR

(hd0,0)
(hd0,1)
(hd0,2)
max 2TB
max 4 partitions
C/H/S Addressing

POST

BIOS + cmos

set T.O.D
set boot device



etc

rc.d

rc.sysinit

```
hostname
mounts /proc, /sys, /dev
enable SELINUX
load device drivers
start udevd
initialise kernel params
starts mapper for LVM
remounts / r/w
enable disk quotas
mounts fs in /etc/fstab
dumps dmesg into log file
```

3

rc

```
for script in rc3.d
do
    ${script}/S*  start
done
```

```
for script in rc3.d
do
    ${script}/K*  stop
done
```

rc0.d rc1.d rc2.d rc3.d rc5.d

K* stop S* start K* stop S* start K* stop S* start K* stop S* start

init.d

at crond sshd junk

```
#! /bin/bash
case $1 in
    start)   /bin/junk
    stop)   kill -9 $PID
    restart)kill -9 $PID
            /bin/junk
esac
exit 0
```

Naming Convention:

S[0-99]cron

K[0-99]cron

Stop/Start services manually

```
# cd /etc/rc.d/init.d
# ./crond stop
# ./crond start|status|restart
# service crond stop|start|restart
```

```
# init 6
# shutdown -g0 -i6 “Shutting down” -y
# shutdown now
# shutdown -r -t 17:00
# reboot
# poweroff
```

```
# who -r
# runlevel
# ps -ef | less
# ps -ef | grep ‘crond’
```

Enable/Disable on next reboot or

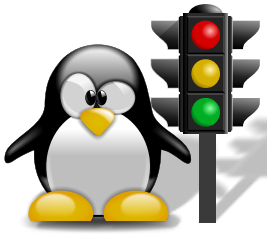
```
# ln -s junk ../rc0.d/K20junk
# ln -s junk ../rc1.d/K20junk
# ln -s junk ../rc2.d/S80junk
# ln -s junk ../rc3.d/S80junk
# ln -s junk ../rc5.d/S80junk
```

```
# chkconfig  --list
# chkconfig  --list crond
```

```
# chkconfig crond on
# chkconfig crond off
```

```
# chkconfig --level 2345 httpd on
# chkconfig --level 016 httpd off
```

Boot Management - System Service Control using systemd



journalctl

journald

PID=1

systemd

unit default.target → unit graphical.target

requires multi-user.target (unit)

wants: display-manager.service (unit), various GUI services (unit)

requires after: basic.target (unit)

requires: timers.target (unit), paths.target (unit), sockets.target (unit), rescue.target (unit)

rescue.target → rescue.service

sysinit.target (unit)

requires: local-fs.target (unit), cryptsetup.target (unit), udevd.service / tmpfs.service / sysctl.service (unit), configfs.mount / debugfs.mount / etc.mount (unit)

/etc/systemd/system /usr/lib/systemd/system

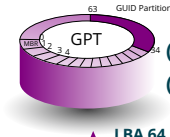
```
# ls -l  /etc/systemd/system/default.target
# systemctl get-default
# systemctl set-default multi-user.target
# systemctl isolate emergency.target
# systemctl
# systemctl poweroff
# systemctl emergency
# systemctl default
# systemctl reboot
```

Kernel

e to edit

GRUB2

/boot can now be xfs/btrfs/ext4/lvm



LBA 64 bit addressing
= 2^64 * 512 = 2 Zbytes
max 128 partitions

POST → Menu

UEFI + cmos

Display/Edit unit files

```
# systemctl cat sshd.service
# systemctl show sshd.service
# systemctl edit sshd.service
# systemctl edit --full sshd.service
```

Bit of fun and good for troubleshooting

```
# systemd-analyze critical-chain
# systemd-analyze blame
# systemd-analyze plot > bootinfo.svg
# eog bootinfo.svg
```

linux16 systemd.unit=rescue.target
systemd.unit=debug-shell
systemd.unit=emergency.target

Fact Finding

```
# systemctl -t help
# systemctl list-units
# systemctl list-units --all
# systemctl list-units --type=target
# systemctl list-units --type=service
# systemctl list-units --type=mount
# systemctl list-unit-files
# systemctl [--all] | grep 'sshd'
# systemctl list-sockets
# systemctl list-jobs
# systemctl list-dependencies [--all]
```

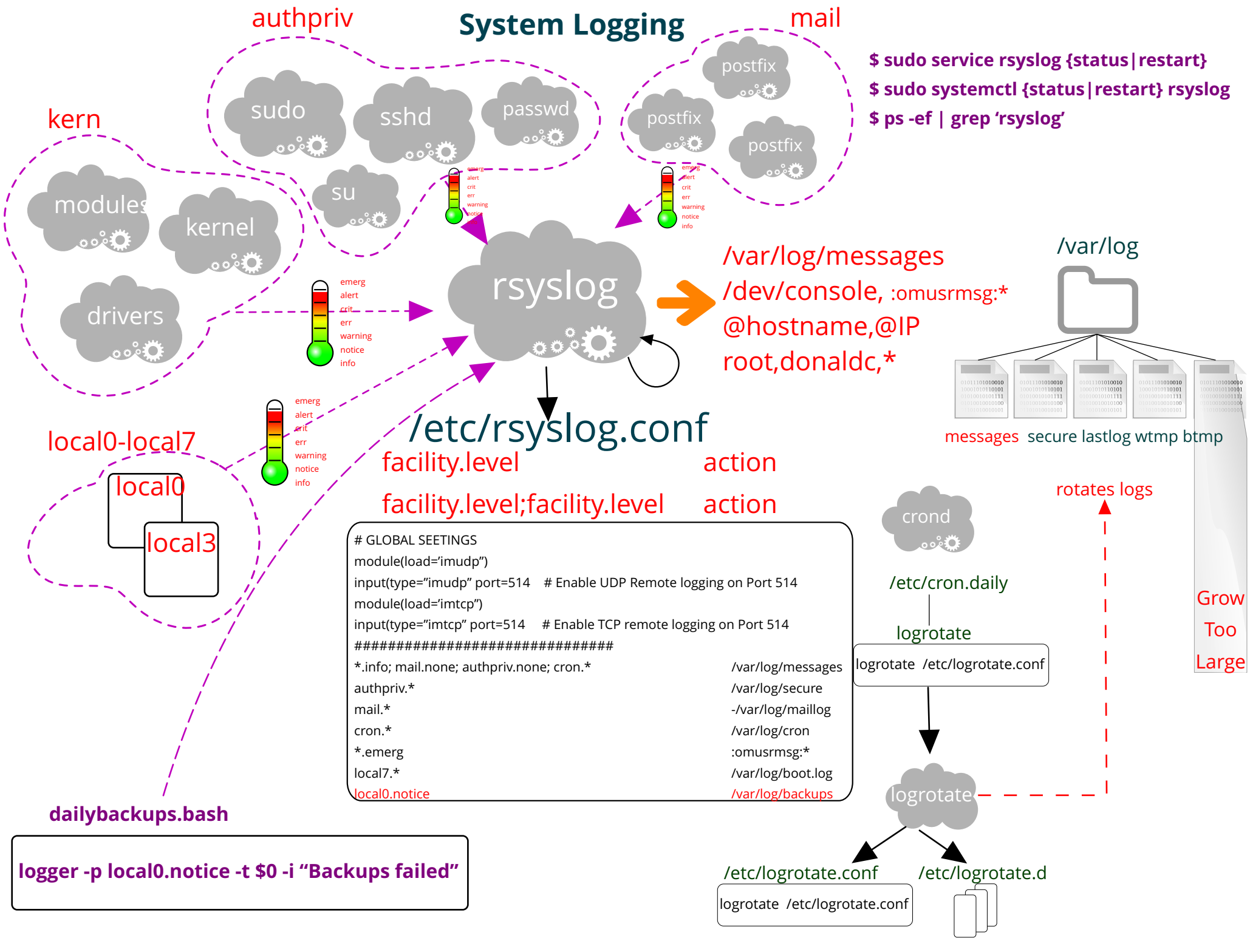
Controlling services manually

```
# systemctl status sshd.service
# systemctl stop|start|status|restart sshd
```

Enabling/Disabling services on reboot

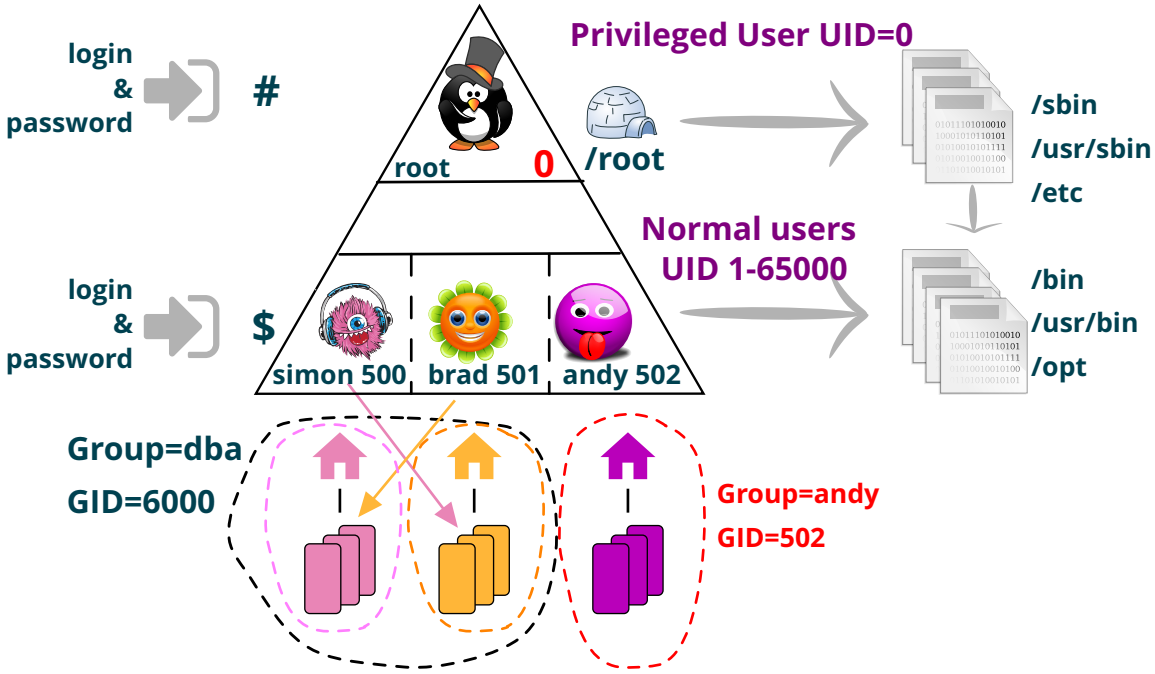
```
# systemctl disable sshd
# systemctl enable sshd
# systemctl mask sshd
# systemctl unmask sshd
```

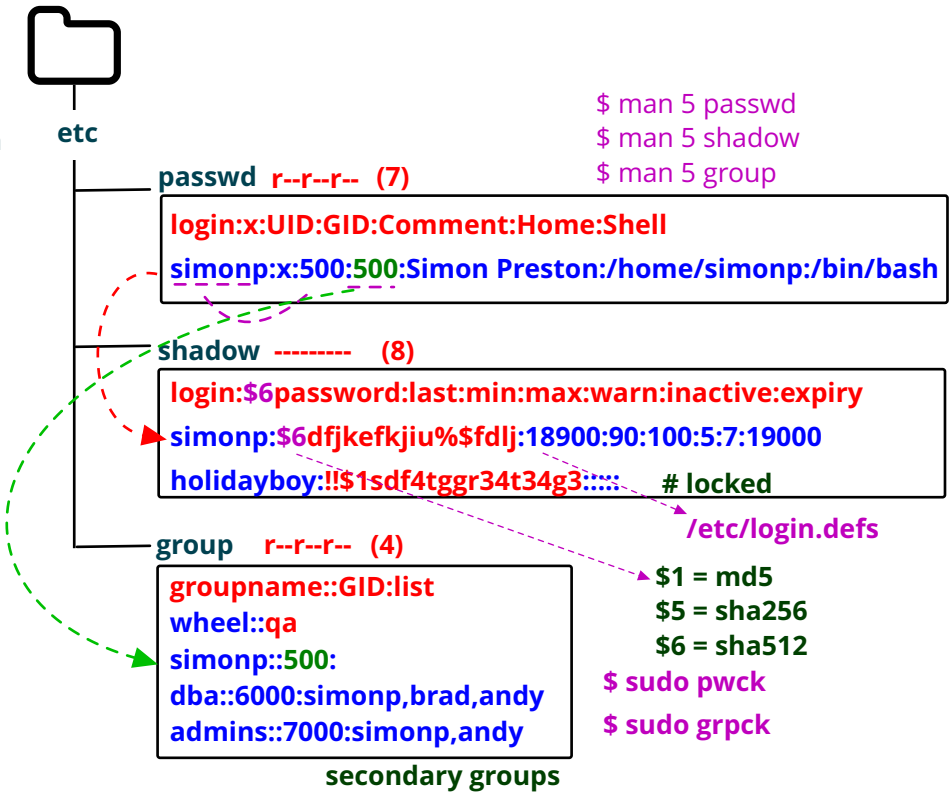
System Logging




```
$ sudo service rsyslog {status|restart}
$ sudo systemctl {status|restart} rsyslog
$ ps -ef | grep 'rsyslog'
```

Linux User Accounts





Login:	donaldc, dcameron, student1
Password:	> 7,8 characters +
UID:	[0 - 499], 500 - 65000
GID:	[0 - 499], 500 - 65000
Home directory:	/home/simon, /root
Default Shell:	/bin/bash, /bin/ksh, /bin/sh

1x primary GID
128x optional secondary GIDs

Creating user accounts

```
$ sudo vim /etc/group
$ sudo vim /etc/passwd
$ sudo vipw
$ sudo mkdir /home/simonp
$ sudo chown simonp:simonp  /home/simonp
$ sudo pwconv                  # sync passwd and shadow
$ sudo passwd simonp           # activate account with password
$ sudo passwd -l simonp        # lock account
$ sudo passwd -u simonp        # unlock account
$ sudo passwd -n90 -x100 -w7 simonp
$ sudo chage -m90  -M100 -W5 -E 2024-04-01 -I 7 simonp
```

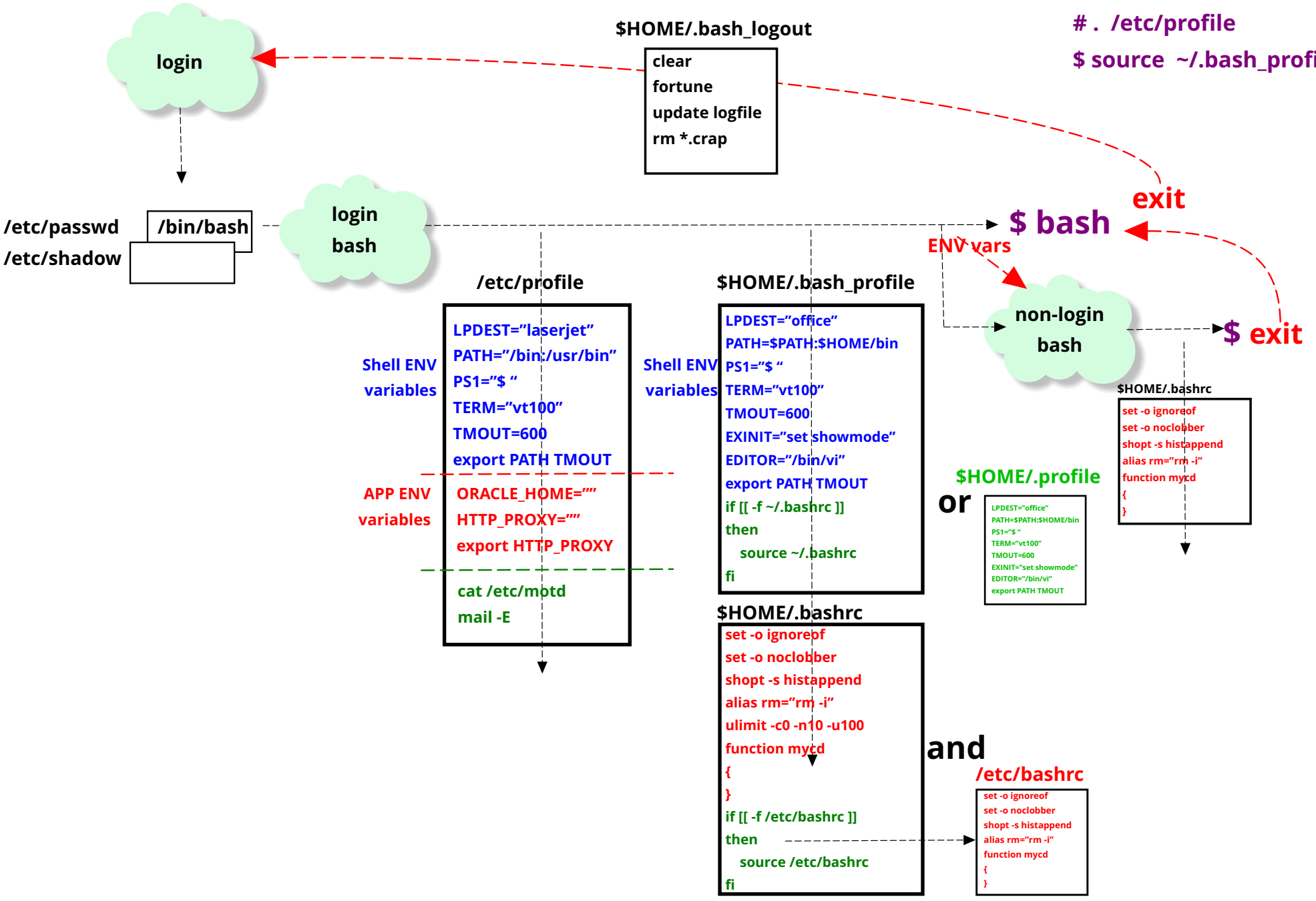
```
$ id simonp
$ groups simonp
$ grep 'simonp' /etc/passwd
```

Creating user accounts

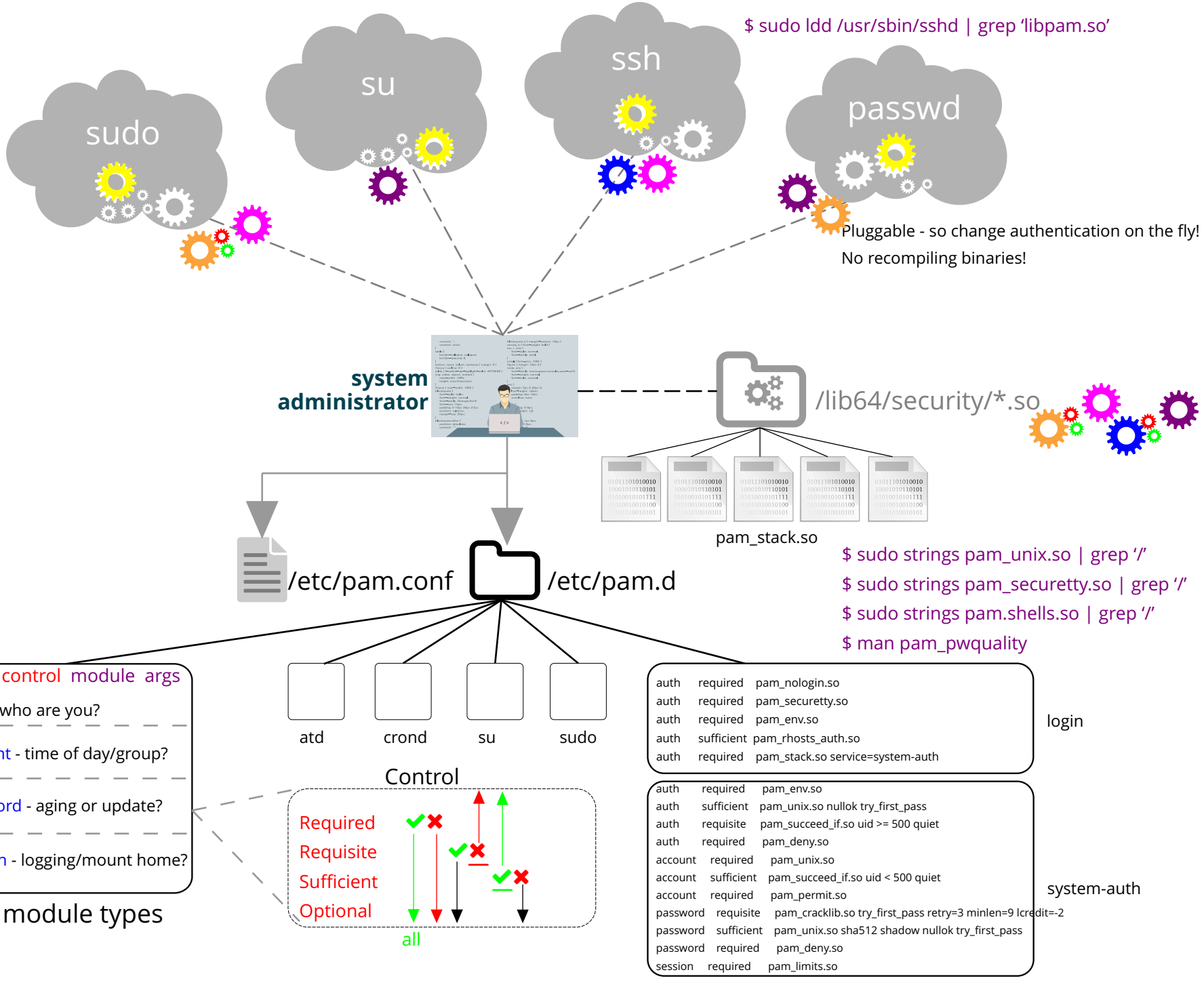
```
$ sudo groupadd -g6000 dba
$ sudo groupmod -n admins dba
$ sudo groupdel dba
$ sudo useradd donaldc
$ sudo useradd -u 500 -g 500 -G dba,admins -m -d /home/simonp -s /bin/bash \
        -c "Simon Preston" -k /etc/skel -f 7 -e 2021-04-01 simonp
$ sudo usermod -l spreston -s /bin/ksh simonp
$ sudo usermod -aGwheel simonp          # Append wheel group

$ sudo passwd simonp          # activate account with password
$ sudo passwd -n90 -x100 -w7 donaldc
$ sudo chage -m90  -M100 -W5 -E 2021-04-01 -I 7 donaldc
$ sudo chage -l donaldc
$ sudo chage -d0 simonp       # User forced to set password on first login
$ chsh -s /bin/bash           # Normal users can change their shells

$ find / -user simonp|500 -exec mv {} /var/tmp/simon_stuff 2> /dev/null
$ sudo userdel -r simonp
```

/etc/login.defs
/etc/default/useradd

Bash Initialisation Files



PAM - Pluggable Authentication Modules



Linux Supported File Systems

original (minix fs) 1991

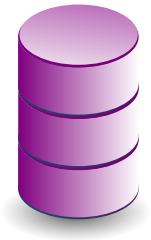
64MB
filename=14

ext 1992

2GB
filename=255

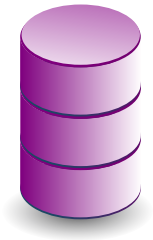
ext2 1993

4TB/Limit 2TB
filename=255
ACLs
32 bit FS



ext3 2001

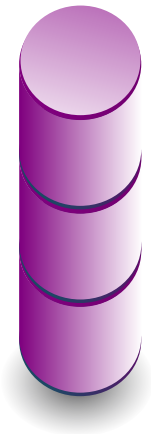
4TB-16TB
filename=255
ACLs/Journaling
FileMax = 2Gb
32 bit FS



De-facto
Standard

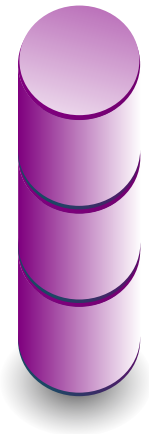
ext4 2008

Max 1ExaByte
filename=255
ACLs/Journaling
FileMax = 16Tb
64 bit FS
Extents



xfs 1994

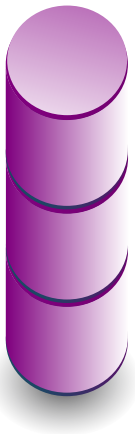
Max 8ExaByte
Limited 16TiB
filename=255
ACLs/Journaling
FileMax = 16Tib
64 bit FS
B+ Trees
Extents



De-facto
Standard
RHEL 7/8

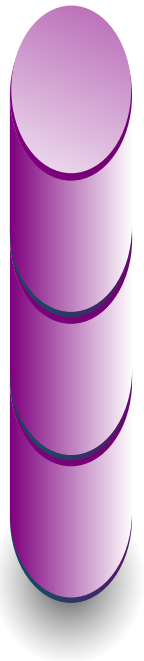
btrfs 2011

Max 16ExaByte
filename=255
ACLs/Journaling
FileMax = 16Eib
64 bit FS
B+ Trees
Extents

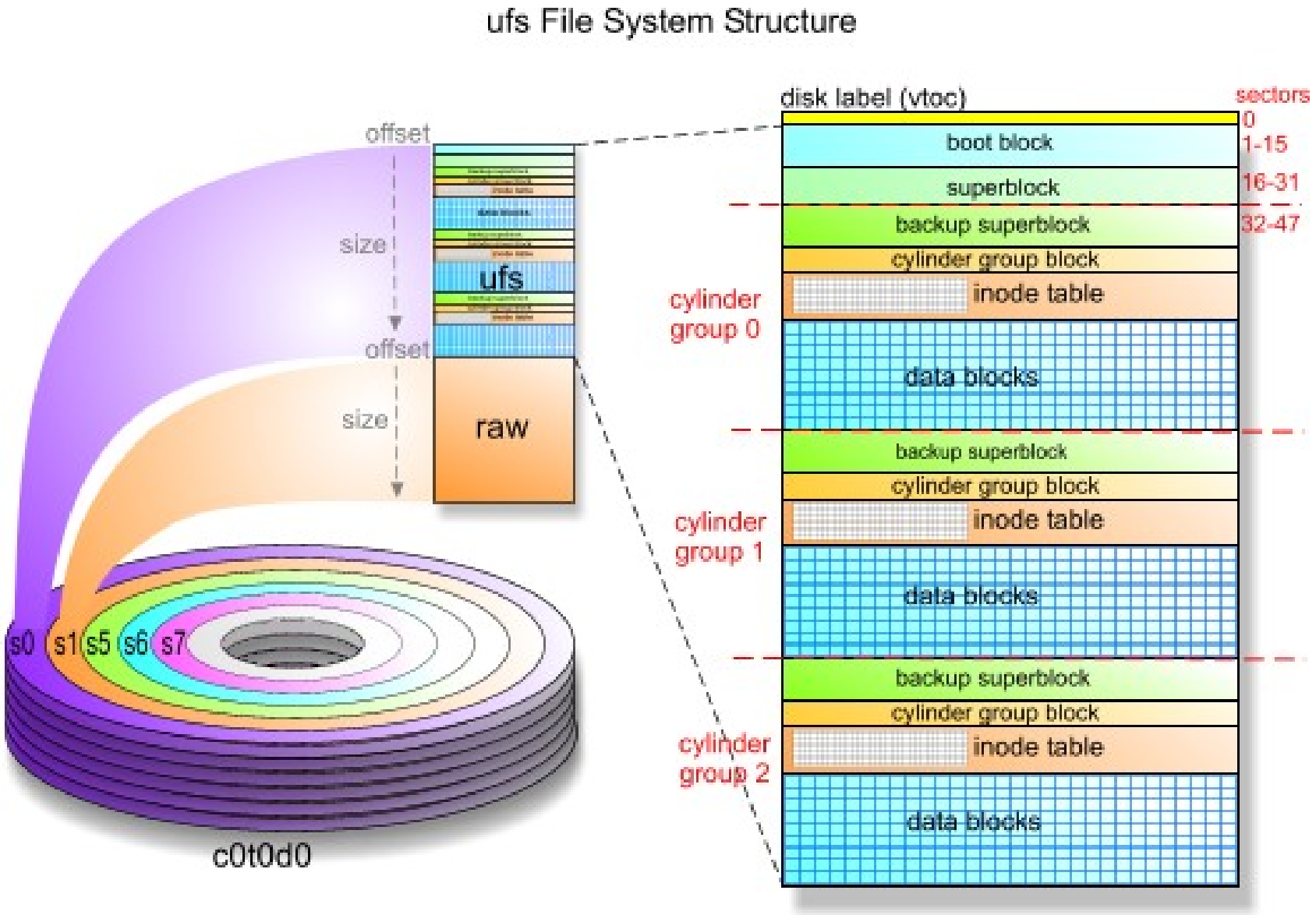


zfs 2004

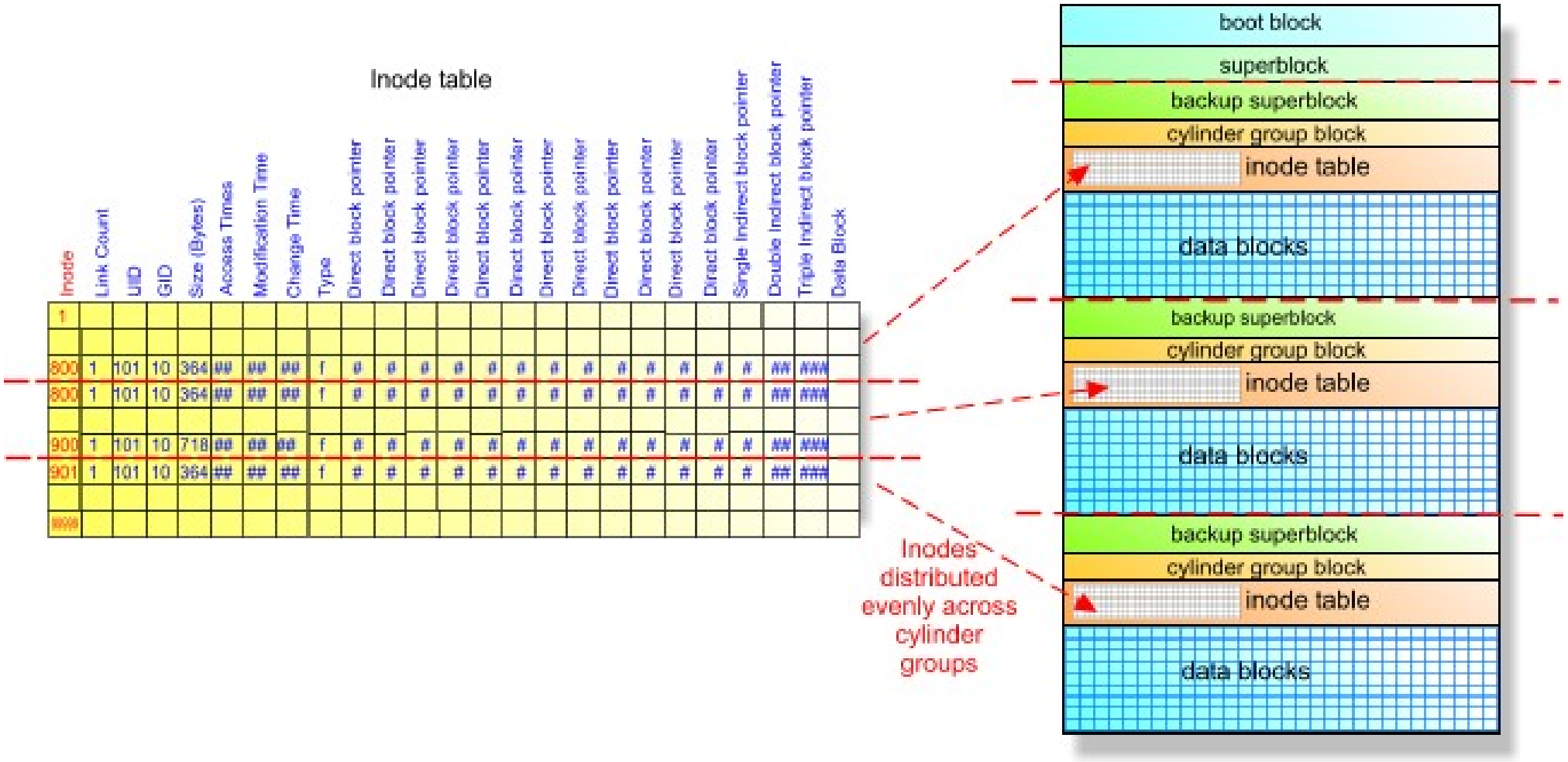
Max 256 Zetabytes
filename=255
ACLs/Journaling
Self Healing
Copy on Write
128 bit FS
B+ Trees
Extents



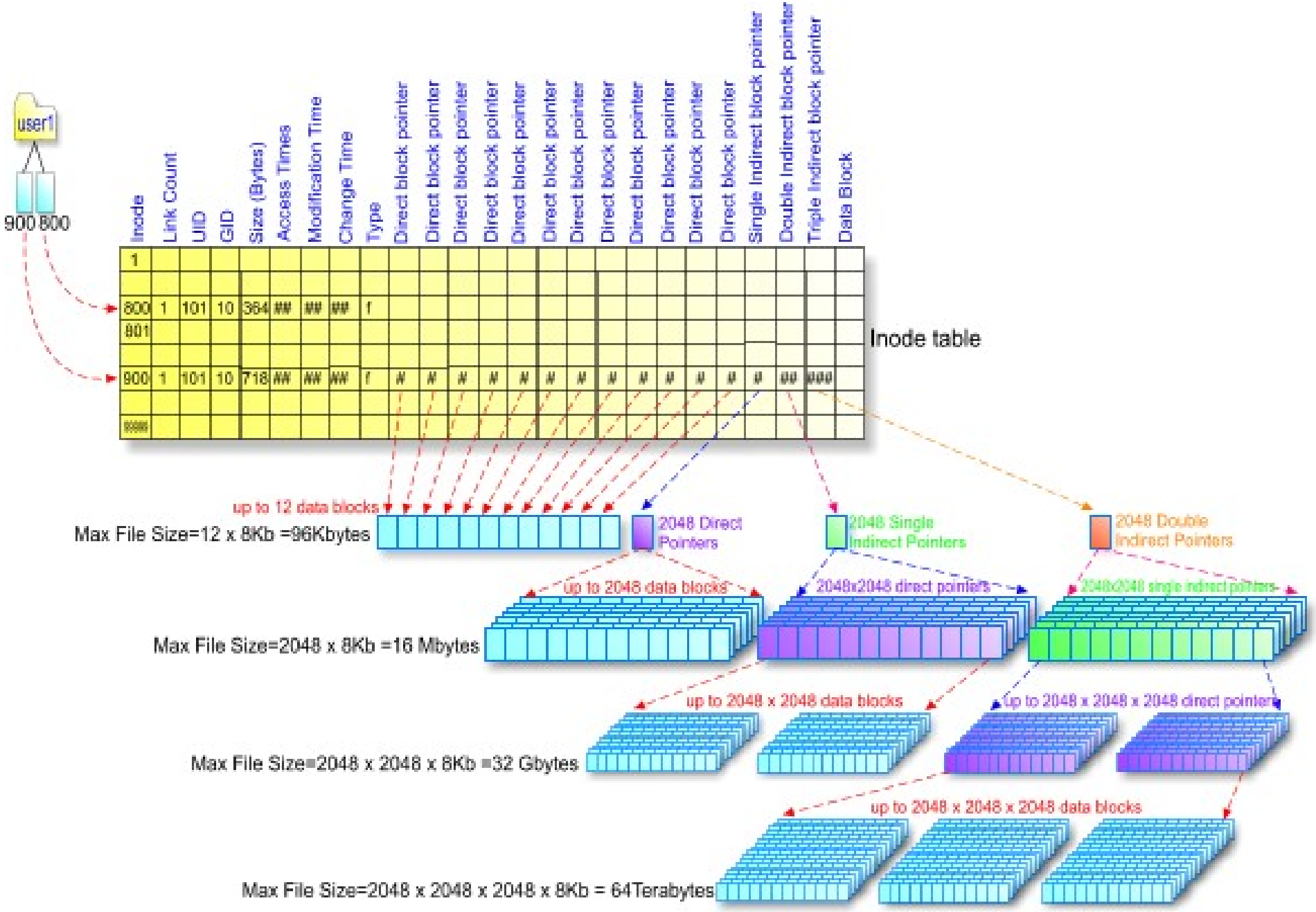
Linux Supported File Systems



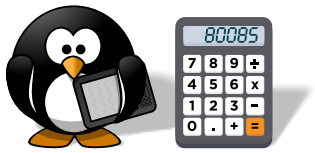
Linux Supported File Systems

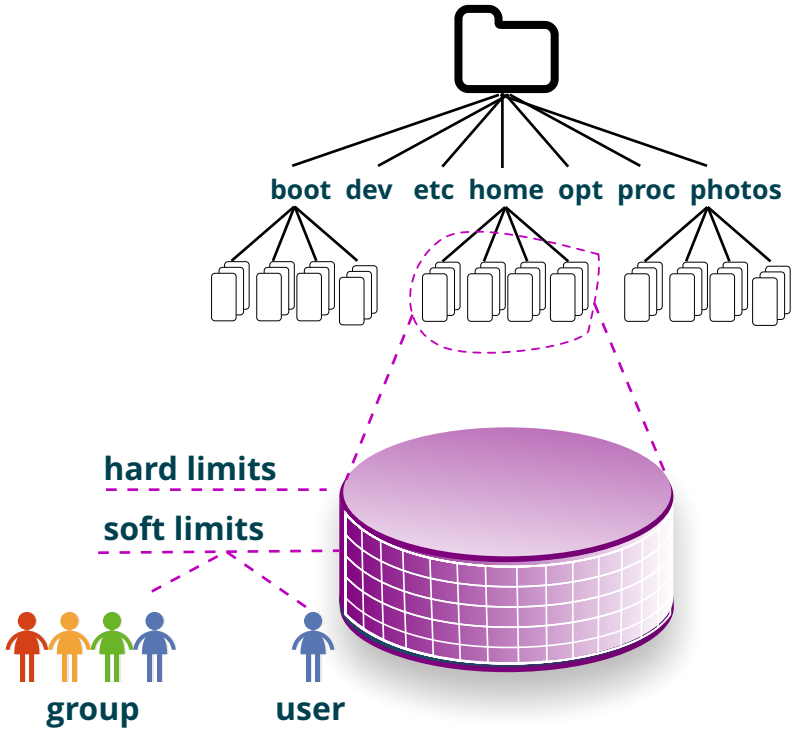


Linux Supported File Systems



Disk Quotas - ext and xfs





Enabling quotas

```
$ sudo vim /etc/fstab
```

```
UUID="34kh2129sdfsdsd2"   /home   ext3   defaults,usrquota,grpquota  1  2
```

```
$ sudo mount -t ext3 -o remount,usrquota,grpquota /home
$ sudo quotacheck -vug /home     # Check current quotas
$ sudo quotaon -vug /home        # Enable quotas
$ sudo quotaon -p /home          # Query quota
```

Editing user quotas

```
$ sudo edquota -u -f /home paul
$ sudo edquota -u -f /home paul -T
$ sudo edquota -g -f /home admins
```

Fact Finding

```
$ sudo repquota -vugs /home
$ quota
$ quota -vugs
```

Enabling quotas on XFS File Systems

```
$ sudo vim /etc/fstab
```

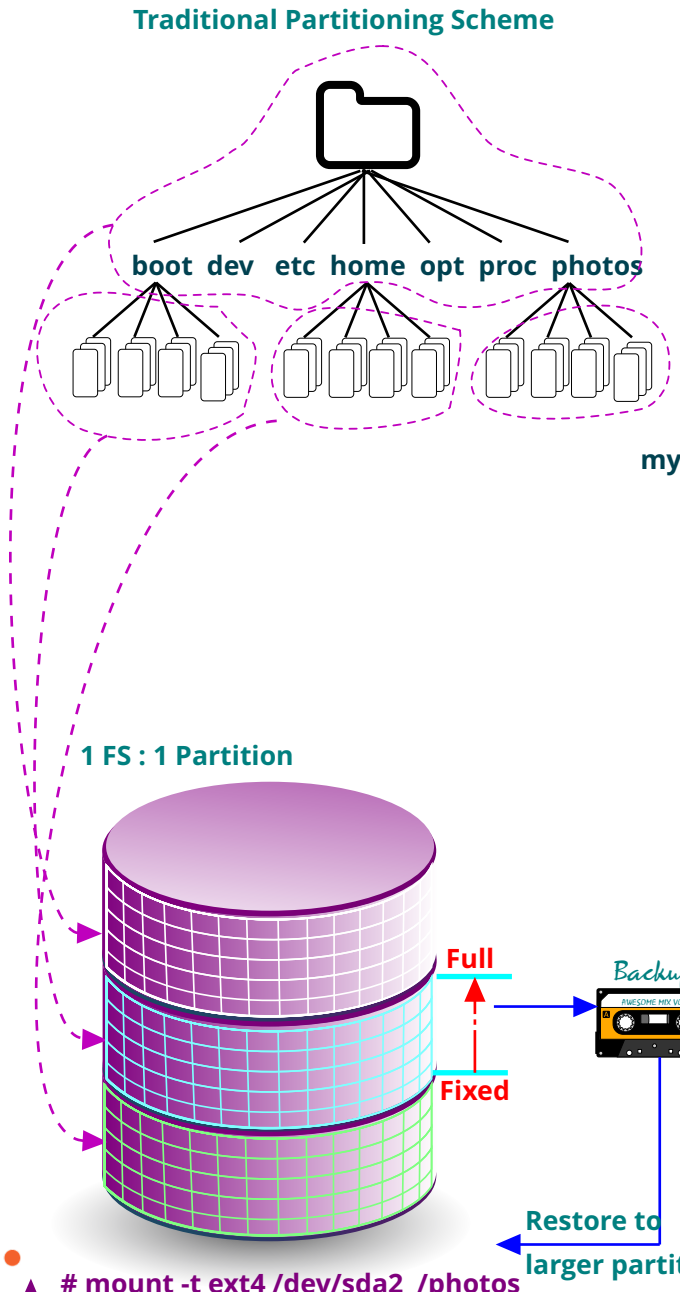
```
UUID="34kh2129sdfsdsd2"   /home   xfs  defaults,quota,gquota  1  2
```

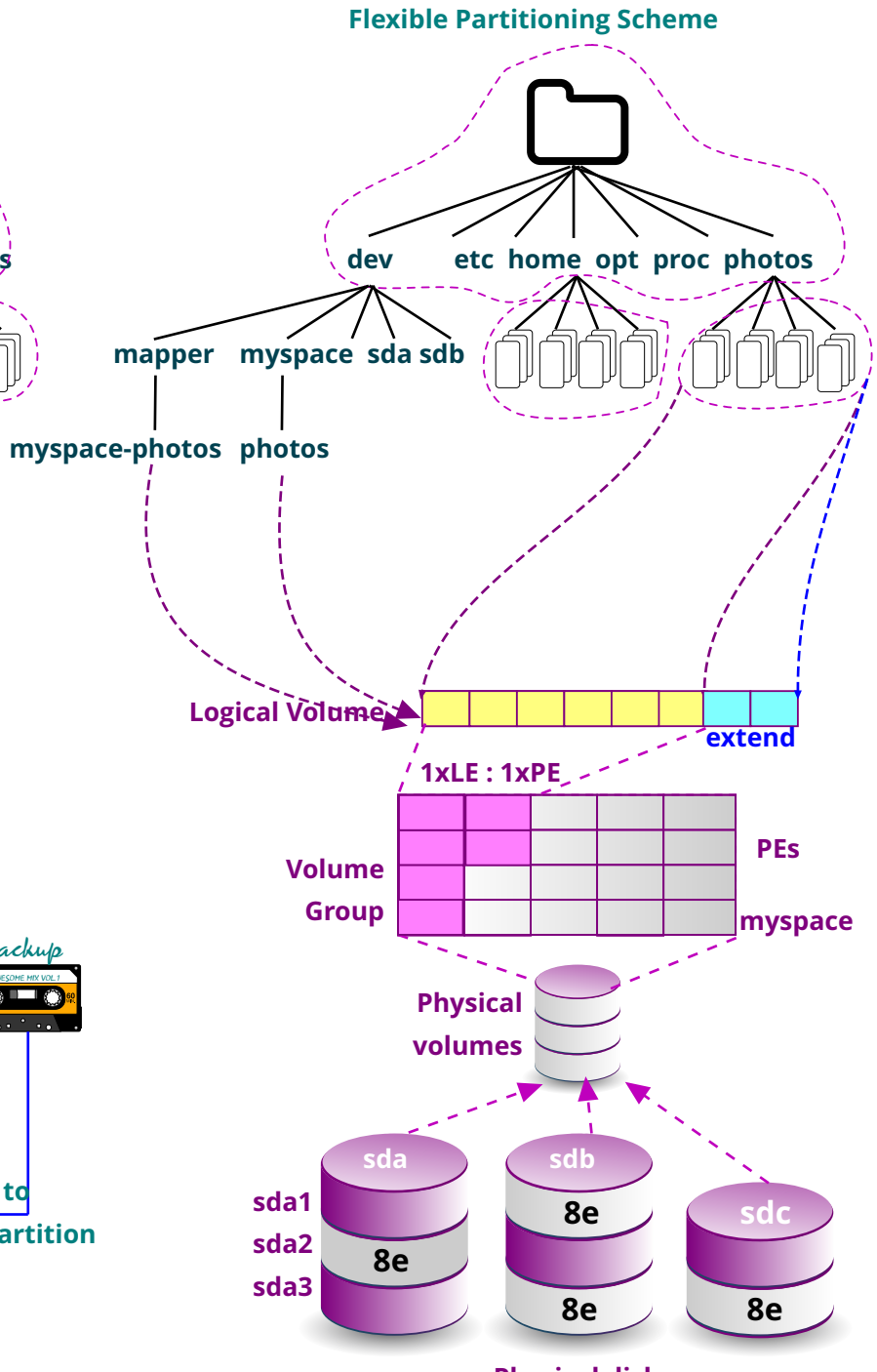
```
$ sudo mount -t xfs -o remount,quota,gquota /home
$ sudo xfs_quota -x -c state
```

Editing user quotas

```
$ sudo xfs_quota -x -c 'limit -u bsoft=5m bhard=6m paulb'  /home
$ sudo xfs_quota -x -c 'limit -g isoft=20 ihard=30 admins'  /home

$ sudo xfs_quota -x -c print
$ sudo xfs_quota -x -c 'free -hb'          # Display free blocks
$ sudo xfs_quota -x -c 'free -hi'          # Display free inodes
$ sudo xfs_quota -x -c 'report' /home      # Display quota report
```

LVM - your flexible friend






```
# mount -t ext4 /dev/sda2  /photos
# mkdir /photos
# mkfs -t ext4 /dev/sda2
# fdisk /dev/sda; blkid; lsblk
```

Reduce a LV and REDUCE ext (only) FS

```
# mount /dev/myspace/photos
# lvreduce -L50M -r /dev/myspace/photos
# resize2fs -p /dev/myspace/photos 50M
# umount /photos
```

Extend a LV and resize filesystem

```
# mount /dev/myspace/photos
# resize2fs -p /dev/myspace/photos
# umount /dev/myspace/photos
# lvextend  -l+4  -r /dev/myspace/photos
# lvextend -L60M -r /dev/myspace/photos
```

```
# mount /dev/myspace/photos /photos
# vim /etc/fstab
# mkdir /photos
# mkfs -t ext4 /dev/myspace/photos
```

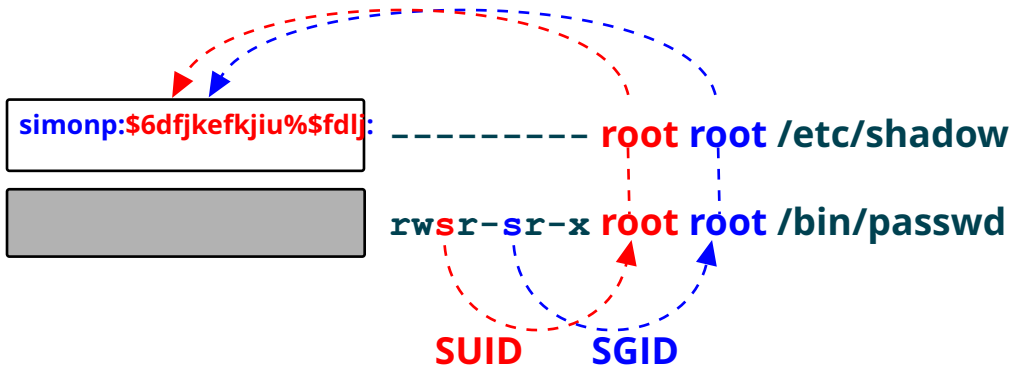
```
# lvdisplay
# lvscan
# lvcreate -n photos -l20  myspace
# lvcreate -n photos -L50M myspace
```

```
# vgdisplay
# vgscan
# vgextend myspace /dev/sdc2
# vgcreate myspace /dev/sda2  /dev/sdb1
```

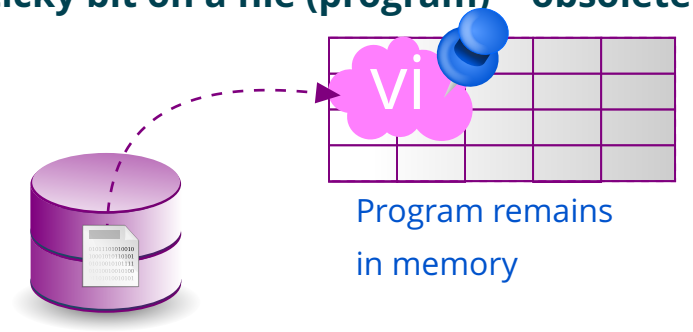
```
# pvdisplay
# pvscan
# pvcreate /dev/sda2 /dev/sdb1 /dev/sdc2
# fdisk /dev/sda
```

Advanced Permissions and attributes

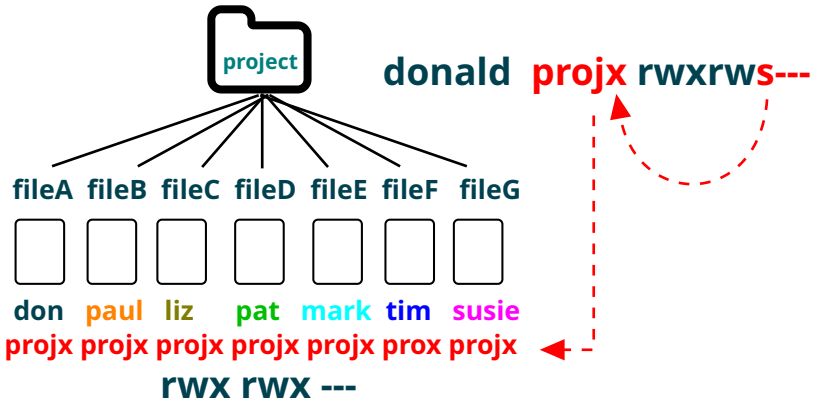
SUID and GID on a file (program)



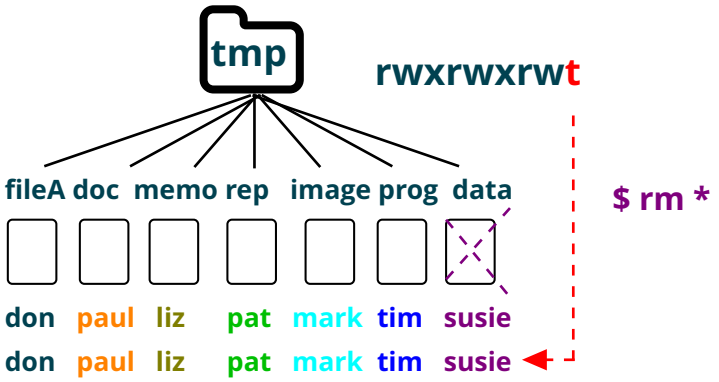
Sticky bit on a file (program) = obsolete



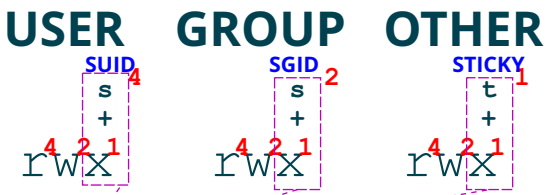
SGID on a directory



Sticky bit on a directory



		SUID	SGID	Sticky Bit
File	x	Run as owner s	Run as group owner s	t
	-	S	S	T
Dir	x	s	Inherit dir group name s	Only owner can delete t
	-	S	S	T




```
# chmod  4755  /bin/passwd  or chmod u+s
$ chmod  2770  /project     or chmod g+s
# chmod  1777  /tmp         or chmod o+t
$ find / -perm -4000  -ls 2> /dev/null
```

ACL - Access Control Lists

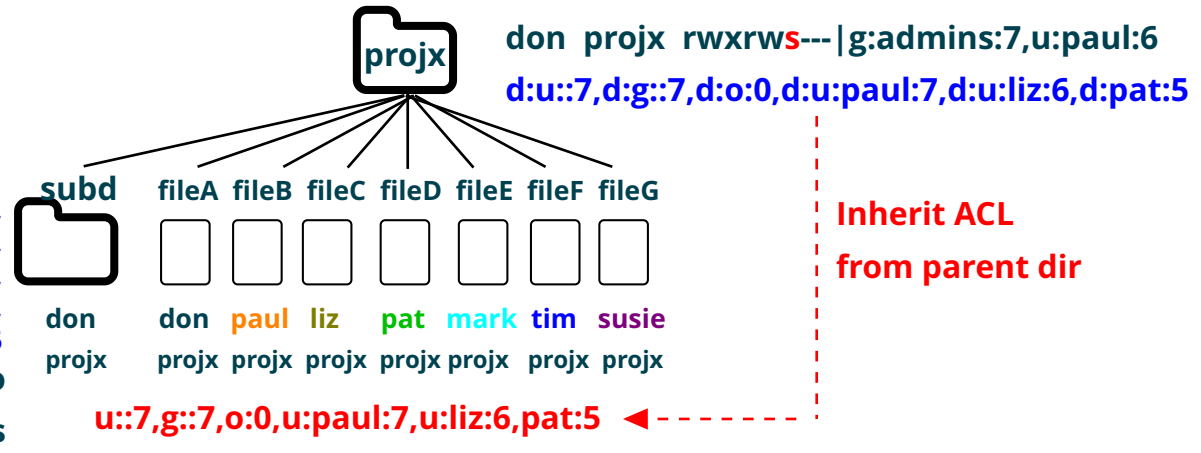
	USER	GROUP	OTHER
Basic File & Dir Permissions	7 rwx user::7 u::rwx	5 r-x group::5 g::r-x	0 --- other:0 o:---
ACL Permissions	user:alex:7 user:aaron:0 u:paul:6	group:dba:5 g:dba:r-x	mask:4 m:r--

```
$ chmod 750  datafile
$ chmod u=rwx,g=r-x,o=---  datafile

$ setfacl -m u::7,g::5,o:0  datafile
$ getfacl datafile

$ setfacl -m u::7,g::5,u:alex:7,g:dba:5,g:oracle:rw-,o:0  datafile
$ setfacl -m u::7,g::5,o:0,g:dba:5,g:oracle:rw-,m:4   datafile
$ getfacl datafile
$ setfacl -x u:alex,g:dba  datafile
```

Duplicate ACLs to another file

```
$ getfacl datafile > newfile
$ setfacl --set-file=datafile  newfile
$ getfacl datafile | setfacl --set-file=- newfile
$ getfacl datafile
```

Default permissions on directories



Network Configuration

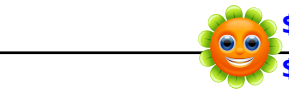
/etc/sysconfig/network
/etc/hostname
/etc/HOSTNAME
booboo

$ ssh localhost
$ ssh yogi

client=booboo

server=yogi

$ sudo systemctl start sshd.service
$ sudo systemctl status httpd.service
$ sudo systemctl enable httpd



sftp
ssh
wget

Interface
Application
+ Presentation
+ Session

NetworkManager
sshd
vsftpd
httpd

/etc/network/interfaces
/etc/netplan/01-netcfg.yaml
/etc/sysconfig/network-scripts/ifcfg-eno1

```
BOOTPROTO=none|dchp
ONBOOT=yes
TYPE=Ethernet
USERCTL=no
IPADDR=192.168.1.1
NETMASK=255.255.255.0
PREFIX=24
GATEWAY=192.168.1.254
HWADDR=01:18:ca:78:a4:c6
DNS1=192.168.1.255
DNS2=8.8.8.8
```

sshd

192.168.1.10/24
255.255.255.0

192.168.1.11/24
255.255.255.0

/etc/services

ftp	21/udp
ftp	21/tcp
ssh	22/tcp
ssh	22/udp

22 49124

21 22 23 25 80

socket: IP + Port + tcp/udp

/etc/hosts

127.0.0.1	localhost
192.168.1.10	booboo
192.168.1.11	yogi

/etc/nsswitch.conf

hosts: files dns ldap

Transport
QOS - ack/nack

tcp udp

Internet
Addressing/Routing

ip icmp arp rarp

MAC IP

Net Interface
Framing/Checkbits

e1000g

kernel

Netfilter=iptables

filter nat mangle
in out for

/etc/resolv.conf

```
192.168.1.1
8.8.8.8
8.8.4.4
```

DNS
Resolver

Hardware

eth0/eno1 NIC

NIC = MAC 48bit Address
01:18:ca:78:a4:c6

enp1s0 NIC = MAC 48bit Address

01:18:ca:89:10:a3

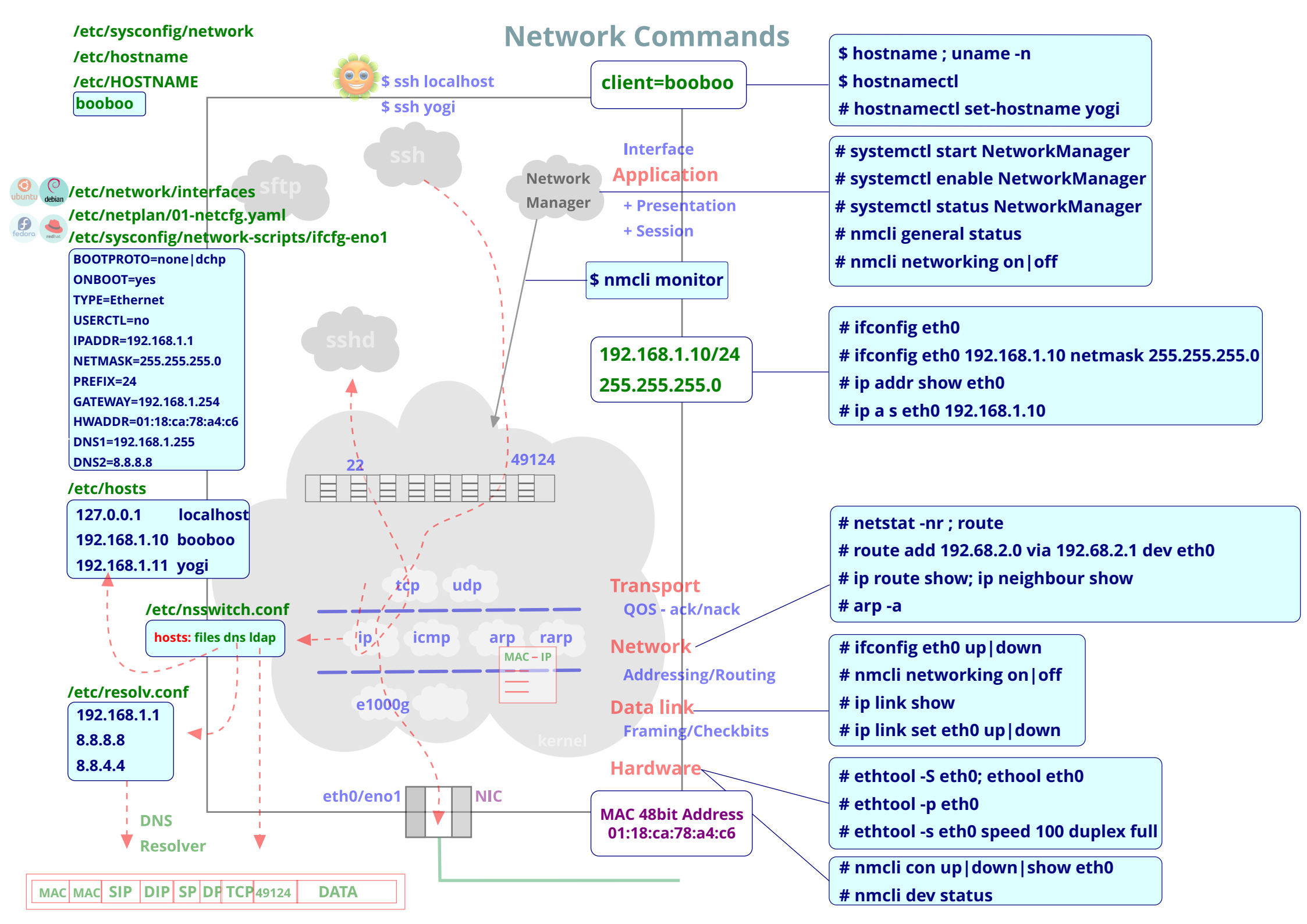
e to edit

linux16 biosdevname=0
or edit /etc/default/grub

AUTO GENERATED by NetworkManager

MAC	MAC	SIP	DIP	SP	DP	SEQ	TCP	49124	DATA

Network Commands

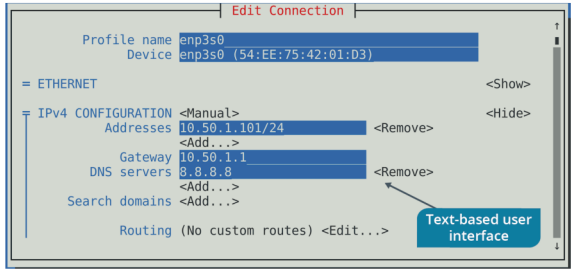


Network Manager and nmcli

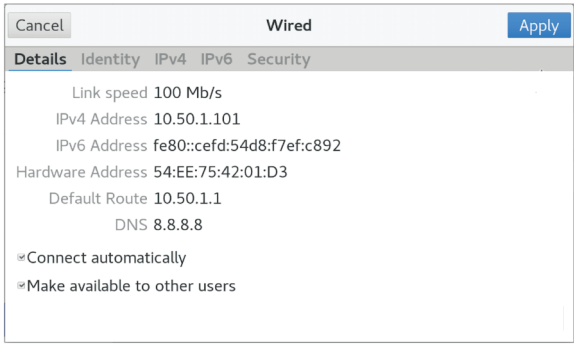
These commands are typically used with Centos/RHEL 7/8 onwards. Ubuntu uses the netplan command.

```
$ systemctl status NetworkManager
$ systemctl stop NetworkManager
$ systemctl start NetworkManager
$ sudo nmcli networking off                  # Turn NM on|of
$ sudo nmcli networking on                   # Turn networking on|off
$ nmcli -t -f RUNNING general                # Show Terse output
$ nmcli -p                                   # Show Pretty output
$ nmcli general                              # Show NM status
$ nmcli general hostname                     # Show hostname
$ nmcli general hostname booboo              # Change hostname
$ nmcli dev status                           # Show device status
$ nmcli con show                             # Show all connnections
$ nmcli con show ens32                       # Show connnection info
$ nmcli con up “Wired connection 1”          # Switch connection ID up
$ nmcli con down “Wired connection 1”# Switch connection ID down
$ nmcli con show ens32                       # Show connnection info
$ nmcli con edit ens32                       # Edit connnection info
$ nmcli con reload                           # Reload Config info
$ nmcli dev status                           # Show device status
$ nmcli monitor                              # Monitor realtime NM changes
$ nmcli con add ifname ens32 type ethernet ipv4.address 192.168.1.100 ipv4.gateway 192.168.1.1
```

nmtui TUI Tool



nmgui GUI Tool



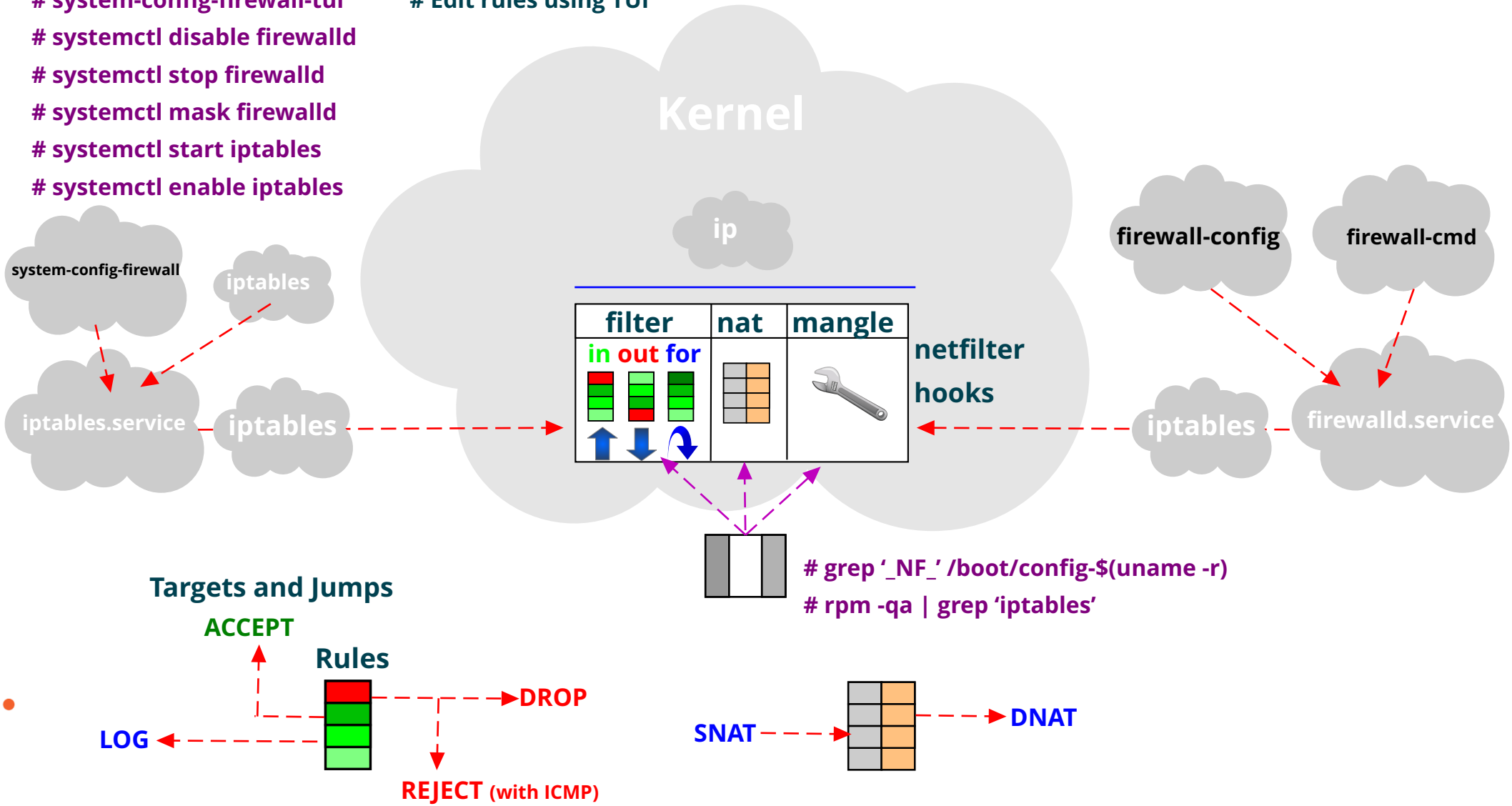
Firewall Protection: aka packet filtering

iptables

```
# iptables-save > /etc/sysconfig/iptables    # Save current rules
# dnf install -y iptables-services           # Install iptables
# vim /etc/sysconfig/iptables                # Edit iptables rules
# system-config-firewall                     # Edit rules using GUI
# system-config-firewall-tui                 # Edit rules using TUI
# systemctl disable firewalld
# systemctl stop firewalld
# systemctl mask firewalld
# systemctl start iptables
# systemctl enable iptables
```

firewalld (RHEL 7)

```
# systemctl disable iptables
# systemctl stop iptables
# systemctl mask iptables
# systemctl unmask iptables
```



Adding Rules to iptables

```
$ sudo iptables -L          # List rules to stdout
$ sudo iptables -F          # Flush the rules table
```

Set iptable rules

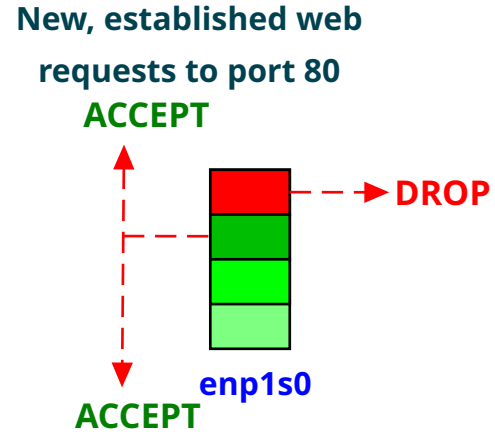
```
$ sudo iptables -A OUTPUT -d 195.26.2.0/24 -p icmp -j DROP     # Drop ping ICMP packets to network 195.26.2.0
$ sudo iptables -A INPUT -s 132.10.0.0/16 -j DROP              # Drop all packets from network 132.10.0.0
$ sudo iptables -A INPUT -s 132.10.24.12 -j ACCEPT             # Accept all packets from host 132.10.24.12
$ sudo iptables -A INPUT -s 102.168.42.1 -d 10.1.1.1 -p tcp --dport 22 -j ACCEPT  # Accept all ssh packets from source to destination.
$ sudo iptables -L --line-numbers                              # Display rules by line-numbers for each chain.
$ sudo iptables -I INPUT 2 -d 192.168.1.0/24 -p icmp -j DROP   # Insert INPUT rule before Rule 2.
$ sudo iptables -D INPUT 1                                     # Delete Rule 1 in INPUT chain.
```

Set chains policy to drop as LAST RULE (Good practice)

```
$ sudo iptables -A INPUT -j DROP
$ sudo iptables -A OUTPUT -j DROP
$ sudop iptables -A FORWARD -j DROP
```

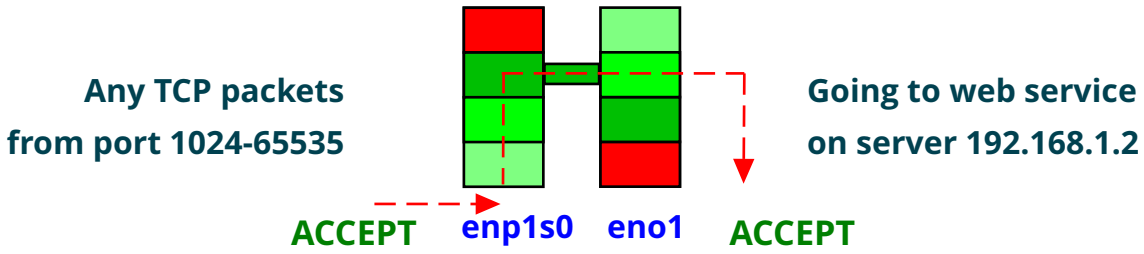
Example - opening web service ports

```
$ sudo iptables -A INPUT -i enp1s0 -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT
$ sudo iptables -A OUTPUT -o enp1s0 -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
$ sudo iptables -A OUTPUT -o enp1s0 -p tcp --sport 443 -m state --state ESTABLISHED -j ACCEPT
```



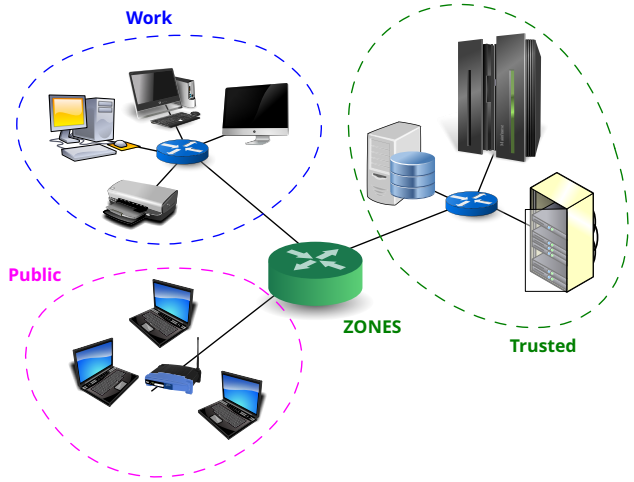
Example - matching connection state and forwarding

```
$ sudo iptables -A INPUT -i enp1s0 -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
$ sudo iptables -A FORWARD -s 0/0 -i enp1s0 -d 192.168.1.2 -o eno1 -p tcp --sport 1024:65535  --dport 80 -j ACCEPT
```



Adding Rules to firewalld (Default for RHEL/Centos 7 onwards)

ALLOW, REJECT or DROP INCOMING traffic to service



https://firewalld.org/documentation/man-pages/firewall-cmd.html

Manage ports (Run cmd TWICE with and without --permanent)

```
$ sudo firewall-cmd --permanent --add-port=22/tcp
$ sudo firewall-cmd --permanent --add-port=53/udp
$ sudo firewall-cmd --permanent --remove-port=22/tcp
```

Add/Remove a service (Run cmd TWICE!!)

```
$ sudo firewall-cmd --permanent --add-service=ssh
$ sudo firewall-cmd --permanent --remove-service=http
```

Disable and query INCOMING ICMP

```
$ sudo firewall-cmd --add-icmp-block={echo-reply,echo-request}
$ sudo firewall-cmd --query-icmp-block=echo-reply
```

TO manage OUTPUT rules, you need to use DIRECT rules

```
$ sudo firewall-cmd --permanent --direct --add-rule ipv4 \
> filter OUTPUT 0 -p icmp -j DROP
$ sudo firewall-cmd --permanent --direct --remove-rule ipv4 \
> filter OUTPUT 0 -p icmp -j DROP
$ sudo firewall-cmd --direct --get-all-rules
```

Starting the service

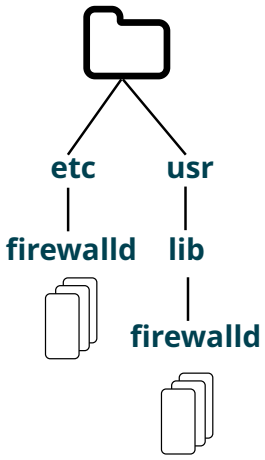
```
$ sudo systemctl start firewalld
$ sudo systemctl enable firewalld
```

Checking the status

```
$ sudo systemctl status firewalld
$ firewall-cmd --state
```

View default services

```
$ sudo firewall-cmd --get-services|get-ports
$ sudo firewall-cmd --list-services
$ sudo firewall-cmd --list-ports
```



View and set zones

```
$ sudo firewall-cmd --list-all-zones
$ sudo firewall-cmd --set-default-zone=public
$ sudo firewall-cmd --get-default-zone
$ sudo firewall-cmd --get-active-zones
$ sudo firewall-cmd --zone=public --change-interface=ens32
```

Allow IP Address

```
$ sudo firewall-cmd --permanent --add-source=192.168.1.254
$ sudo firewall-cmd --permanent --add-source=192.168.1.0/24
```

Block IP Address

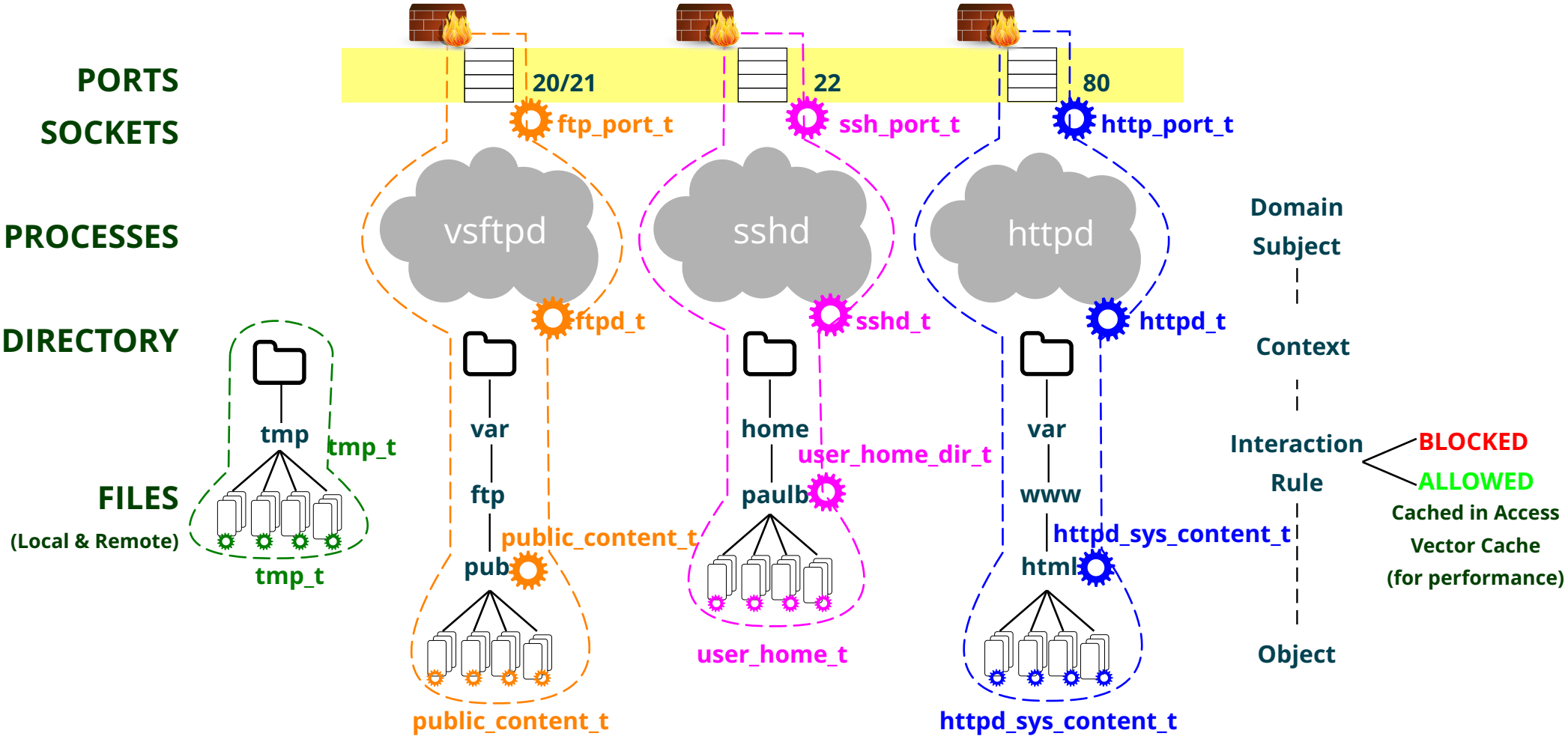
```
$ sudo firewall-cmd --permanent --add-rich-rule="rule family='ipv4'
                    source-address='192.168.1.66' reject"
$ sudo firewall-cmd --permanent --add-rich-rule="rule family='ipv4'
                    source-address='192.168.1.0/24 reject"
```

Saved rules

```
$ sudo firewall-cmd --runtime-to-permanent
$ sudo firewall-cmd --reload
```

Or, you could use a GUI: `$ sudo firewall-config`

SELinux - Security Enhanced Linux



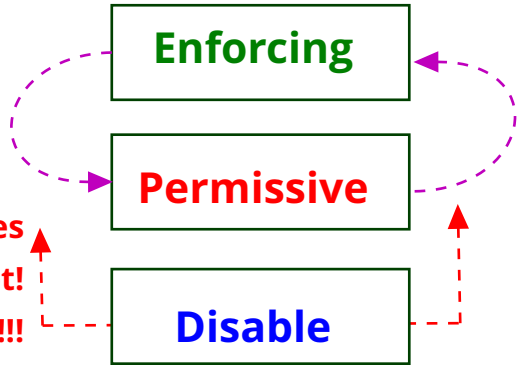
Every Object has a SELABEL (several contexts)

unconfined_u:object_r:httpd_sys_content_t:s0

SELinux user : SELinux role: SELinux type : sensitivity

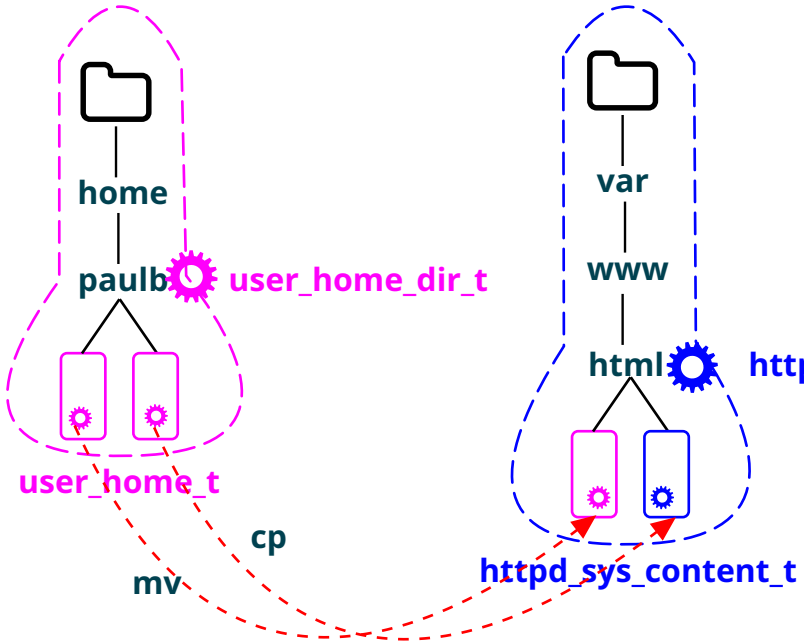
```
$ ps -efZ | more
$ ls -lZ /tmp
$ id -Z
$ semanage fcontext -l        # List all objects and file context types
```

3 Modes:



Logs = /var/log/audit/audit.log & /var/log/messages

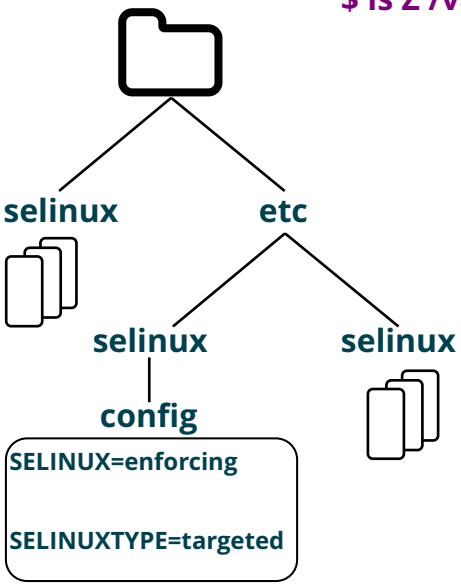
SELinux - Security Enhanced Linux



Fix labels

```
$ sudo restorecon -Rv /var/www/html
$ sudo chcon -R --type=httpd_sys_content_t  fileA
$ sudo chcon -R --reference=fileB  fileA
```

```
$ mv ~paulb/fileA  /var/www/html      # Preserve original label
$ cp ~paulb/fileB  /var/www/html      # Change to label of parent dir
$ cp -a ~paulb/fileB  /var/www/html   # Preserve original label
$ ls Z /var/ww/html
```



Change SELinux mode

```
$ getenforce                    # Display current SELinux mode
$ sudo setenforce 1             # Set SELinux mode to Enforcing
$ sudo setenforce permissive    # Set SELinux mode to Permissive
$ sestatus                      # Display current SELinux settings
```

e to edit - - - - ▶ linux16 selinux=0 # Set SELinux mode to Permissive

```
$ sudo touch /.autorelabel      # RELABEL entire filesystem with SELinux label on reboot!
```