ARCH LINUX

System Administration & KDE Plasma

Services · initcpio · X11/Wayland · Display Managers · Users · KDE Plasma · fish

A hands-on workshop for Arch Linux power users

01 Service Management	02 initcpio Hooks	03 X11 vs Wayland	04 Display Managers	05 Desktop Environments
06 Users & Permissions	07 KDE Plasma Setup	08 Plasma Personalization	09 Dolphin & fish Shell	10 KWin Scripts

SECTION 01

01

Service Management

systemd – the heart of Arch Linux

systemctl – basic commands

`systemctl start <svc>`

Start a service immediately

`systemctl stop <svc>`

Stop a running service

`systemctl restart <svc>`

Stop then start (new config)

`systemctl reload <svc>`

Reload config – no kill

`systemctl enable <svc>`

Enable at next boot

`systemctl disable <svc>`

Remove from autostart

`systemctl status <svc>`

Status + last log lines

`systemctl list-units`

List all active units

✦ HANDS-ON

```
$ systemctl status sshd                  # is SSH running?
$ systemctl enable --now NetworkManager  # enable AND start in one step
$ systemctl list-units --failed          # find broken services
```

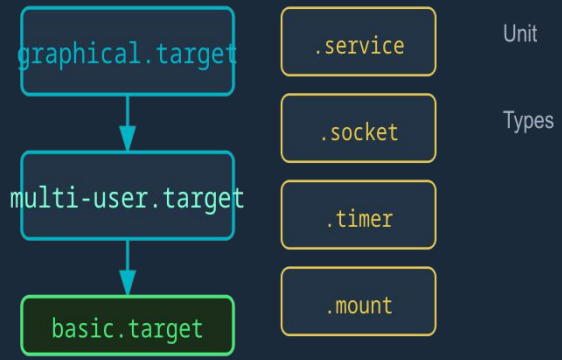
systemd units and targets

UNIT TYPES

.service – processes and daemons
.socket – socket-based activation
.timer – cron replacement
.mount – filesystem mount points
.target – groups of units
.path – watch files/dirs

IMPORTANT TARGETS

multi-user.target – CLI multi-user
graphical.target – GUI (default)
rescue.target – minimal + root shell
emergency.target – absolute minimum
default.target – symlink to current




```
$ systemctl get-default             # show current default
$ systemctl set-default multi-user.target   # change (persistent)
$ systemctl isolate multi-user.target       # switch NOW (live)
$ systemctl list-units --type=target --state=active
```

✦ HANDS-ON

```
$ systemctl get-default
$ systemctl list-units --type=target --state=active
```

Writing a custom systemd service

UNIT FILE SECTIONS

[Unit]
- Description= human label
- After= ordering dep
- Wants= soft dep
- Requires= hard dep

[Service]
- ExecStart= main command
- Restart= on-failure/always
- User= run as this user

[Install]
- WantedBy= pulled by target

```
# /etc/systemd/system/mybackup.service
[Unit]
Description=Nightly backup script
After=network-online.target
Wants=network-online.target

[Service]
Type=oneshot
ExecStart=/usr/local/bin/backup.sh
User=backupuser
StandardOutput=journal

[Install]
WantedBy=multi-user.target
```

✦ HANDS-ON

1. `sudo nano /etc/systemd/system/hello.service`
   `[Unit] Description=Hello [Service] ExecStart=/usr/bin/echo hello [Install] WantedBy=multi-user.target`
2. `sudo systemctl daemon-reload && sudo systemctl start hello.service`
3. `journalctl -u hello.service -n 10 # check output`

02

SECTION 02

initcpio Hooks

Custom early-boot scripts in the initramfs

initcpio hook structure

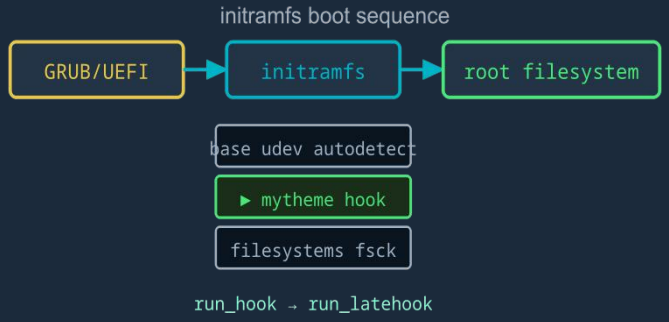
install/ FILE (build time)

build() – runs during mkinitcpio
add_runscript → include hooks/ file
add_binary cmd → copy binary + libs
add_file src dst → copy any file
add_module name → add kernel module
help() – shown by mkinitcpio -H

hooks/ FILE (run time)

run_hook() – before root mount
run_latehook() – after root mount ✓
run_cleanuphook() – before pivot_root

Shell: busybox sh (POSIX only!)
Path: /usr/lib/initcpio/{hooks,install}/




```
# /etc/mkinitcpio.conf – order matters!
HOOKS=(base udev autodetect modconf block mytheme filesystems fsck)

mkinitcpio -P          # rebuild ALL presets
```

✦ HANDS-ON

```
$ mkinitcpio -H mytheme     # show hook help
$ mkinitcpio -M             # dry-run autodetect
```

Example: terminal color palette hook (mytheme)

/etc/initcpio/hooks/mytheme

```
#!/bin/sh
run_latehook() {
 local palette=\
 "002b36 dc322f 859900 b58900 \
 268bd2 d33682 2aa198 eee8d5 \
 073642 cb4b16 586e75 657b83 \
 839496 6c71c4 93a1a1 fdf6e3"
 local i=0 seqs=''
 for col in $palette; do
   seqs="${seqs}\033]P$(printf \
         '%X' $i)${col}"
   i=$((i+1))
 done
 for tty in /dev/tty[0-6]; do
   [ -w "$tty" ] || continue
   printf "$seqs" > "$tty"
   printf "\033[H\033[J" > "$tty"
   [ -f /font.psf.gz ] &&
     setfont -C "$tty" /font.psf.gz
 done
}
```

/etc/initcpio/install/mytheme

```
#!/bin/bash
build() {
  add_runscript
  add_binary setfont
  add_binary printf
  local fp=$(find \
   /usr/share/kbd/consolefonts/\
   -name 'ter-v28b*' | head -1)
  if [ -n "$fp" ]; then
    add_file "$fp" /font.psf.gz
  else
    echo 'WARNING: no font!'
  fi
}
help() { cat <<EOF
Sets Solarized Dark palette
on all VTs at boot.
EOF
}
```

✦ HANDS-ON

Test live: `printf "\033]P0002b36\033]P1dc322f" > /dev/tty1   # change colors without reboot`

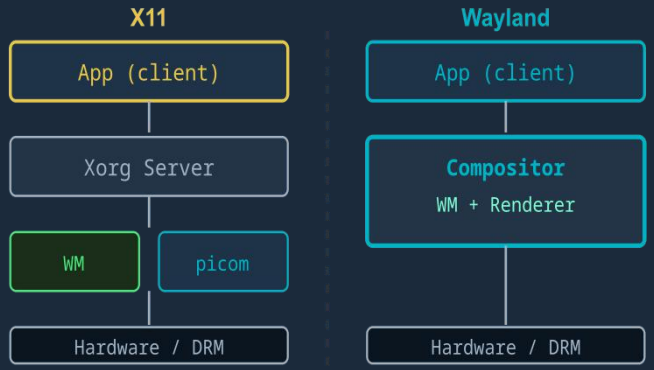
SECTION 03

03

X11 vs Wayland

Display protocols, compositors and WM management

Architecture comparison



X11 / Xorg

- Client-server architecture
- Exists since 1984
- Any WM plugs into Xorg
- Network transparent: DISPLAY=host:0
- Launch: startx or xinit
- Live WM swap: --replace flag
- Config: ~/.xinitrc /etc/X11/
- Picom as separate compositor
- Mature NVIDIA driver support

Wayland

- Protocol – Compositor = WM
- Designed 2008, stable ~2018
- Better security / isolation
- No network transparency
- Launch: compositor from TTY
- No live WM swap (restart)
- Config per compositor
- Native HiDPI, touch, stylus
- XWayland for legacy X11 apps

Launching X11 and live WM switching

```
# Start X11 from TTY:
startx                    # reads ~/.xinitrc
xinit /usr/bin/i3 -- :1   # explicit WM

# Minimal ~/.xinitrc content:
exec i3
```

```
# Start Wayland session:
sway
dbus-launch --exit-with-session sway

# Required env vars:
export XDG_SESSION_TYPE=wayland
```

Live WM switching (X11 only — Wayland requires session restart)

```
openbox --replace &       # replace current WM with Openbox
i3 --replace &            # replace with i3
pkill -x openbox && bspwm &   # bspwm has no --replace

wmctrl -m                 # identify running WM  (pacman -S wmctrl)
echo $XDG_SESSION_TYPE    # wayland or x11?
```

✦ HANDS-ON

```
$ echo $XDG_SESSION_TYPE      # wayland or x11?
$ echo $WAYLAND_DISPLAY       # set only on Wayland sessions
```

SECTION 04

04

Display Managers

Graphical login screens and session management

Popular Display Managers on Arch

SDDM

```
pacman -S sddm
```

Qt6-based. Default for KDE/Plasma. Excellent Wayland support, theme engine, HiDPI.

GDM

```
pacman -S gdm
```

GNOME Display Manager. Tight systemd-logind integration. Default for GNOME.

LightDM

```
pacman -S lightdm
```

Lightweight, DE-agnostic. Multiple greeter frontends (GTK, Qt, WebKit).

ly

```
pacman -S ly
```

TUI login in the terminal. No graphics dependencies, extremely minimal.

greetd

```
pacman -S greetd
```

Modern IPC-based daemon. Pairs with tuigreet or agreety frontend.

LXDM

```
pacman -S lxdm
```

Part of LXDE project. Simple GTK3 interface, low resource usage.

✦ HANDS-ON

```
$ systemctl status display-manager.service   # which DM is active?
$ pacman -Qi sddm | grep -E 'Name|Version'   # inspect installed DM
```

Enabling, switching and configuring a DM

```
# Install and enable SDDM:
pacman -S sddm
systemctl enable sddm.service

# Switch DM (ALWAYS disable old first!):
systemctl disable lightdm.service
systemctl enable sddm.service && reboot
```

```
# Session files (DM reads these):
ls /usr/share/xsessions/         # X11
ls /usr/share/wayland-sessions/  # Wayland

# Force session in ~/.dmrc:
[Desktop]
Session=plasmawayland
```

```
# /etc/sddm.conf.d/kde.conf:
[Autologin]
User=myuser
Session=plasmawayland

[Theme]
Current=breeze
EnableHiDPI=true
```

✦ HANDS-ON

```
$ ls /usr/share/xsessions/ /usr/share/wayland-sessions/  # available sessions
$ cat /etc/sddm.conf.d/*.conf 2>/dev/null || echo 'no sddm config'
$ systemctl is-enabled sddm                              # enabled?
```

SECTION 05

05

Desktop Environments

GNOME · KDE Plasma · XFCE · and more

Popular desktop environments on Arch

GNOME

gtk4, Wayland-first

```
pacman -S gnome
systemctl enable gdm
```

KDE Plasma

Qt6, X11 + Wayland

```
pacman -S plasma
systemctl enable sddm
```

XFCE

GTK3, lightweight

```
pacman -S xfce4 xfce4-goodies
systemctl enable lightdm
```

Cinnamon

GTK3, GNOME 2 fork

```
pacman -S cinnamon
systemctl enable lightdm
```

MATE

GTK3, classic layout

```
pacman -S mate mate-extra
systemctl enable lightdm
```

LXQt

Qt5, very lightweight

```
pacman -S lxqt
systemctl enable sddm
```

SECTION 06

06

Users & Permissions

useradd · chmod · chown · chattr

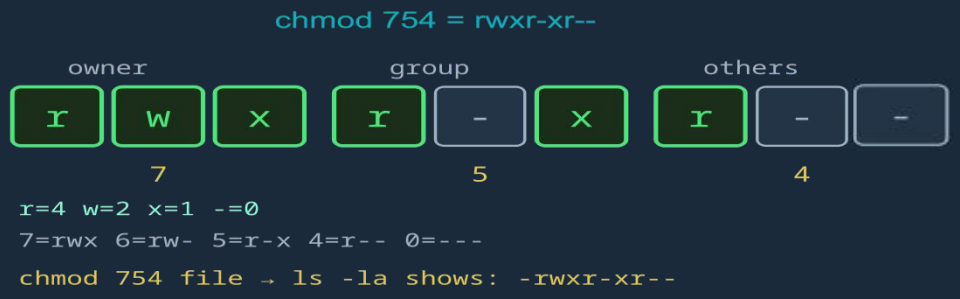
Managing users and groups

```
# Create a user with groups:
useradd -m -G wheel,audio,video \
  -s /bin/bash alice
passwd alice

# Modify user:
usermod -aG docker alice  # add to group
usermod -s /usr/bin/fish alice
userdel -r alice          # delete + home
```

```
# Group operations:
groupadd devs
gpasswd -a alice devs   # add user
gpasswd -d alice devs   # remove user
groupdel devs

# Inspect:
id alice        # UID, GID, all groups
groups alice    # group names only
```

chmod – file permissions

```
chmod 755 script.sh     # rwxr-xr-x
chmod 644 config.txt    # rw-r--r--
chmod u+x,g-w file.sh   # symbolic
chmod -R 750 /srv/app/  # recursive
ls -la file   stat file # view perms
```



✦ HANDS-ON

```
$ touch t && ls -la t   # default perms?   $ chmod 600 t && ls -la t
```

chown, chattr and special permission bits

```
# chown – change ownership:
chown alice file.txt
chown alice:devs file.txt
chown -R www-data:www-data /var/www
chgrp audio /dev/snd/*
```

```
# chattr – filesystem attributes:
chattr +i file.txt    # immutable
chattr -i file.txt    # remove immutable
chattr +a log.txt     # append-only
lsattr file.txt       # display attrs
```

Special permission bits (SUID · SGID · Sticky)

Bit	Octal	Set with	Effect	Real example
SUID	4xxx	`chmod u+s prog`	Runs as file owner	`/usr/bin/passwd`
SGID	2xxx	`chmod g+s dir`	Files inherit parent group	`/srv/shared/`
Sticky	1xxx	`chmod +t /tmp`	Only owner/root can delete	`/tmp` `/var/tmp`

✦ HANDS-ON

```
$ ls -la /usr/bin/passwd    # -rwsr-xr-x ← SUID     $ ls -la /tmp    # drwxrwxrwt ← sticky
```

07

SECTION 07

KDE Plasma Setup

Installation, packages and SDDM

Required packages – Arch main repos only

PLASMA & CORE

plasma-meta
kde-applications-meta
konsole dolphin
kwrite kate
kdeconnect
discover packagekit-qt6

FISH & CLI

fish starship
bat fd
ripgrep fzf
eza zoxide
htop neofetch
git base-devel

KWIN & DISPLAY

kwin
xorg-xwayland
qt6-wayland
kscreen libkscreen
plasma-pa plasma-nm
bluedevil powerdevil

FONTS & ICONS

ttf-jetbrains-mono-nerd
noto-fonts
noto-fonts-emoji
papirus-icon-theme
ffmpegthumbs
dolphin-plugins

✦ HANDS-ON

```
sudo pacman -S plasma-meta kde-applications-meta fish starship bat fd ripgrep fzf eza
sudo systemctl enable --now sddm
```

SECTION 08

08

Plasma Personalization

Themes · Colors · Icons · Fonts · SDDM

Global theme, color scheme and icons

GLOBAL THEME

System Settings → Appearance
→ Global Theme
Applies: decorations, colors,
cursor, icons, splash
Path: ~/.local/share/plasma/
look-and-feel/

COLOR SCHEME

System Settings → Appearance
→ Colors
Edit or import .colors file
Custom: ~/.local/share/
color-schemes/
Format: KDE Color Scheme

ICON THEME

Settings → Appearance → Icons
Recommended: Papirus-Dark
pacman -S papirus-icon-theme

Custom: ~/.local/share/icons/
System: /usr/share/icons/

```
plasma-apply-colorscheme BreezeDark
plasma-apply-colorscheme --list-schemes          # list all available

plasma-apply-lookandfeel -a org.kde.breezedark.desktop
plasma-apply-lookandfeel --list                  # browse installed

kwriteconfig6 --file kdeglobals --group Icons --key Theme Papirus-Dark
```

✦ HANDS-ON

```
$ plasma-apply-colorscheme --list-schemes | head -10
$ plasma-apply-lookandfeel --list | head -10
```

Fonts, cursors, splash screen and SDDM

FONTS

System Settings → Appearance → Fonts
Categories: General, Fixed-width,
Small, Toolbar, Menu, Window Title

Install Nerd Fonts (terminal icons):
pacman -S ttf-jetbrains-mono-nerd
pacman -S noto-fonts noto-fonts-emoji

Rebuild font cache: fc-cache -fv

CURSOR + SPLASH + SDDM

Cursor: Settings → Appearance → Cursors
pacman -S xcursor-themes
Custom: ~/.icons/

Splash: Settings → Workspace Behavior
→ Splash Screen → None (faster login)

SDDM: /etc/sddm.conf.d/kde.conf
[Theme] Current=breeze EnableHiDPI=true

```
kwriteconfig6 --file kcminputrc --group Mouse --key cursorTheme breeze_cursors
kwriteconfig6 --file kdeglobals --group KScreen --key ScaleFactor 1.5
sudo bash -c 'echo -e "[General]\nEnableHiDPI=true" > /etc/sddm.conf.d/hidpi.conf'
```

✦ HANDS-ON

```
$ fc-list | grep -i jetbrains       # confirm font installed
$ cat /etc/sddm.conf.d/*.conf 2>/dev/null || echo 'no sddm config'
```

SECTION 09

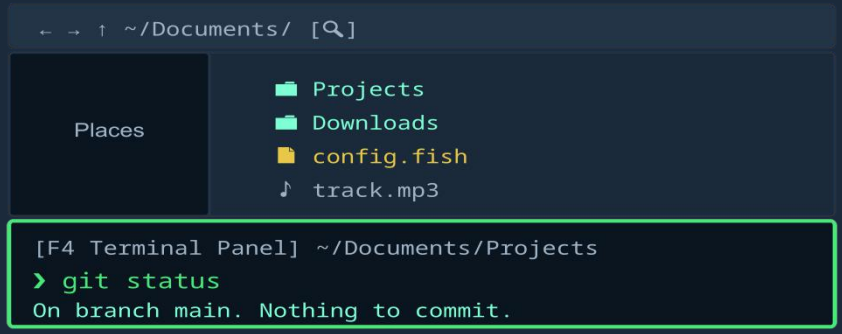
09

Dolphin & fish Shell

File manager power features · fish · Starship · Konsole

Dolphin – advanced configuration

Dolphin panels (F3 F4 F11)



DOLPHIN KEY SHORTCUTS

F4 – toggle terminal panel
F3 – split view (dual pane)
F11 – information panel
Ctrl+H – show hidden files
Ctrl+L – location bar (type path)
Alt+← / → – back / forward

```
# Essential plugins:
pacman -S dolphin-plugins    # git/VCS icons
pacman -S ffmpegthumbs       # video previews
pacman -S kdegraphics-thumbnailers ark
kbuildsycoca6                # rebuild cache
```

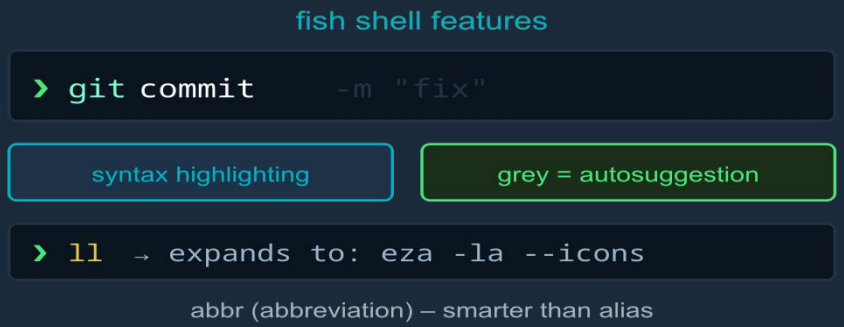
```
# ~/.config/dolphinrc:
[General]
ShowFullPath=true
SortRole=name

# Set fish as default shell:
chsh -s /usr/bin/fish
```

✦ HANDS-ON

1. Open Dolphin → press F4 (terminal panel)
2. Press F3 (split pane) → Ctrl+H (hidden files) → F11 (info panel)

```
$ sudo pacman -S dolphin-plugins ffmpegthumbs ark && kbuildsycoca6
```

fish shell setup and Starship prompt




```
# ~/.config/fish/config.fish:
starship init fish | source
zoxide init fish | source
fzf --fish | source

abbr --add gs  'git status'
abbr --add ll  'eza -la --icons'
abbr --add lt  'eza --tree --level=2'
```

FISH KEY FEATURES

Autosuggestions – grey inline completions
Syntax highlighting – live as you type
Web config UI: fish_config (browser)
Universal vars: set -U VAR value
No .bashrc – all in ~/.config/fish/

```
# ~/.config/starship.toml:
[character]
success_symbol = '[❯](bold green)'
[git_branch]
symbol = ' '
[directory]
truncation_length = 3
```

✦ **HANDS-ON**

```
$ fish_config          # opens browser UI at localhost:8000
$ echo $fish_version && type eza && type bat
$ abbr --add ll 'eza -la --icons'  && ll   # test abbreviation
```

SECTION 10

10

KWin Scripts & Window Rules

Window management automation and shortcuts

KWin Scripts – install and write your own

INSTALL VIA GUI

System Settings → Window Management
→ KWin Scripts
Click 'Get New Scripts' for KDE Store
Scripts live in:
~/.local/share/kwin/scripts/

RECOMMENDED KWIN SCRIPTS

Krohnkite – i3-style tiling layout
Quarter Tiling – snap to quadrants
Window Rules – built-in per-app config
Parachute – expose / overview effect

```
# Install from file:
kpackagetool6 -t KWin/Script \
  -i myscript.kwinscript

# List / remove:
kpackagetool6 -t KWin/Script -l

# Reload without logout:
qdbus-qt6 org.kde.KWin /Scripting reload
```

```
// Minimal KWin script (main.js):
workspace.clientAdded.connect(
  function(client) {
    if (client.resourceClass
        === 'konsole') {
      client.frameGeometry = {
        x:0,y:0,
        width:1280,height:720
      };
    }
  }
);
```

✦ HANDS-ON

```
# metadata.json alongside main.js:
{ "KPlugin": { "Name": "My Script", "Description": "Auto-size Konsole", "Version": "1.0" } }
```

Window Rules and essential KWin shortcuts

WINDOW RULES (BUILT-IN)

System Settings → Window Management
→ Window Rules → Add New
Match by: class, title, type, role

Per-app you can force:
- Size and position
- Virtual desktop / activity
- Skip taskbar / always on top
- Initial opacity, maximize state

KWin Tiling (Meta + arrow keys)


```
# ~/.config/kwinrulesrc
[1]
Description=Firefox maximized
wmclass=firefox
wmclassmatch=1
maximizehoriz=true
maximizehorizrule=3
maximizevert=true
maximizevertrule=3
```

```
# Apply rules live (no restart):
qdbus-qt6 org.kde.KWin /KWin reconfigure

# Essential shortcuts:
Meta + ← / →   tile left / right
Meta + PgUp/Dn  switch virtual desktop
Meta + W        show all windows
Ctrl+F9         present windows
```

✦ HANDS-ON

Try: Meta+← tile a window left, Meta+→ tile another right → instant split-screen

SUMMARY

What we covered

01

Service Management

systemctl, unit files, custom services, targets

02

initcpio Hooks

Boot-time scripts, palettes, fonts in initramfs

03

X11 vs Wayland

Protocol differences, live WM switching

04

Display Managers

SDDM, GDM, LightDM, ly – enable and config

05

Desktop Environments

GNOME, KDE, XFCE – install and switch

06

Users & Permissions

useradd, chmod, chown, chattr, SUID/sticky

07

KDE Plasma Setup

Packages, meta-packages, SDDM activation

08

Plasma Personalization

Themes, colors, icons, fonts, HiDPI

09

Dolphin & fish Shell

Plugins, Starship, Konsole, eza, bat

10

KWin Scripts & Rules

Window automation, shortcuts, per-app rules

FURTHER READING

Resources & next steps

Arch Wiki

wiki.archlinux.org

The definitive resource. Search any package, config, or concept.

KDE Documentation

docs.kde.org

Official KDE user and developer docs, Plasma 6 guides.

KDE Store

store.kde.org

Themes, plasmoids, KWin scripts, SDDM themes, icons.

fish docs

fishshell.com/docs

Complete fish shell documentation and function reference.

Starship

starship.rs

All starship.toml configuration options with examples.

systemd docs

systemd.io

Unit file reference, journalctl, networkd, resolved.