Linux speedrun: systemd

Getting (almost) everything out of systemd

Michael Boelen


```
michael@nluug$ systemd-analyze calendar "2025-11-20"
  Original form: 2025-11-20
Normalized form: 2025-11-20 00:00:00
    Next elapse: never

michael@nluug$ systemd-analyze timestamp "now"
  Original form: 2025-11-20 11:00
Normalized form: Thu 2025-11-20 11:00:00 CET
       (in UTC): Thu 2025-11-20 10:00:00 UTC
   UNIX seconds: @1763632800
       From now: 38us ago

michael@nluug$ systemd-analyze timespan "45m"
Original: 45m
      µs: 2700000000
   Human: 45min
```

$ whoami

Michael Boelen

- Blogger
  - Linux Audit
  - linux.vooreenbeginner.nl
  - Linux Lorem (videos)
  - meereco.nl
- Interests
  - Linux and Unix
  - Information Security
  - Chess ♟
  - Metal detecting
  - MeshCore
  - Video editing
- Open source developer
  - ~~rkhunter~~ / Lynis
- Volunteer
  - Communication Commission @ NLLGG
  - Webmaster @ NLUUG
  - Participation council (primary school)
  - Neighborhood watch program
- More at michaelboelen.com
- Mastodon: **@mboelen** (mastodon.nl and mastodon.social)

Let's learn something

Spelling

- **Correct:** systemd
- **Incorrect:** *all others*

Spelling

Yes, it is written **systemd**, not **system D** or **System D**, or even **SystemD**. And it isn't **system d** either. Why? Because it's a system daemon, and under Unix/Linux those are in lower case, and get suffixed with a lower case **d**. And since systemd manages the system, it's called systemd. It's that simple. But then again, if all that appears too simple to you, call it (but never spell it!) **System Five Hundred** since **D** is the roman numeral for 500 (this also clarifies the relation to System V, right?). The only situation where we find it OK to use an uppercase letter in the name (but don't like it either) is if you start a sentence with **systemd**. On high holidays you may also spell it *sÿstëmd*. But then again, ***Système D*** is not an acceptable spelling and something completely different (though kinda fitting).

Source: https://brand.systemd.io/ (highlighted text is for clarity)

Let's learn something

Releases

- First: **1** (2010)
- Latest: **258**

Question: Where are releases 45-182?
Answer:

```
systemd System and Service Manager

CHANGES WITH 183:
        * Note that we skipped 139 releases here in order to set the
          new version to something that is greater than both udev's
          and systemd's most recent version number.
```

FAQ

Question? At the end (or after the talk)

Got an insight? Do share!

- Mastodon: include **@nluug** (and me **@mboelen**)
- Use the hashtag: **#nluug**

Slides? Yes, will be published (speedrun series)

- Links to useful resources
- Cheat sheets
- Bonus levels!

Before we begin

Goals today

- Maximize coverage of systemd (in <45 minutes)
- Learn what's **possible**
- Cherry pick the things **interesting to you**

Ready? Let's play!

Level 1: It's-A Me, systemd!

What is systemd?

It's a:

- System manager
- Service manager

_____ +

= Two managers

What is systemd?

System manager

- PID 1
- Started by the kernel (init)
- Controls system state
  - Start
  - Shutdown
  - Hibernate
  - Reboot
- Provides additional system building blocks
  - journald (event log)
  - logind (login session manager)
  - udevd (hardware and kernel events)
  - and more...

What is systemd?

Service manager

- Same thing: PID 1
- Deals with services
- Provides more structure
- Increased speed
  - Parallelization
  - Smart(er) dependencies
- A new best friend: **systemctl**

systemd version?

systemctl --version

- **systemd 249** (249.11-0ubuntu3.16)

+PAM +AUDIT +SELINUX +APPARMOR +IMA +SMACK +SECCOMP +GCRYPT +GNUTLS +OPENSSL +ACL +BLKID +CURL +ELFUTILS +FIDO2 +IDN2 -IDN +IPTC +KMOD +LIBCRYPTSETUP +LIBFDISK +PCRE2 -PWQUALITY -P11KIT **-QRENCODE** +BZIP2 +LZ4 +XZ +ZLIB +ZSTD -XKBCOMMON +UTMP +SYSVINIT default-hierarchy=unified

- **systemd 252** (252.39-1~deb12u1)

+PAM +AUDIT +SELINUX +APPARMOR +IMA +SMACK +SECCOMP +GCRYPT -GNUTLS +OPENSSL +ACL +BLKID +CURL +ELFUTILS +FIDO2 +IDN2 -IDN +IPTC +KMOD +LIBCRYPTSETUP +LIBFDISK +PCRE2 -PWQUALITY +P11KIT **+QRENCODE** **+TPM2** +BZIP2 +LZ4 +XZ +ZLIB +ZSTD **-BPF_FRAMEWORK** -XKBCOMMON +UTMP +SYSVINIT default-hierarchy=unified

- **systemd 258** (258.1-1-arch)

+PAM +AUDIT -SELINUX +APPARMOR -IMA +**IPE** +SMACK +SECCOMP +GCRYPT +GNUTLS +OPENSSL +ACL +BLKID +CURL +ELFUTILS +FIDO2 +IDN2 -IDN +IPTC +KMOD +LIBCRYPTSETUP +LIBCRYPTSETUP_PLUGINS +LIBFDISK +PCRE2 +PWQUALITY +P11KIT +QRENCODE **+TPM2** +BZIP2 +LZ4 +XZ +ZLIB +ZSTD +**BPF_FRAMEWORK** +**BTF** +**XKBCOMMON** +UTMP **-SYSVINIT** +**LIBARCHIVE**

Is systemd a monolithic beast?

Not in file size

```
$ file /bin/init
/bin/init: symbolic link to ../lib/systemd/systemd
```

```
$ file /lib/systemd/systemd
/lib/systemd/systemd: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked,
interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=dad1165775a2eff463ca99f0e763d05572a97904, for
GNU/Linux 4.4.0, stripped
```

```
$ stat /lib/systemd/systemd
  File: /lib/systemd/systemd
  Size: 137400      Blocks: 272        IO Block: 4096   regular file
Device: 0,29   Inode: 450985      Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2025-11-14 18:48:44.000000000 +0100
Modify: 2025-11-10 10:11:12.000000000 +0100
Change: 2025-11-14 18:48:44.675129521 +0100
 Birth: 2025-11-14 18:48:44.675129521 +0100
```

Is systemd a monolithic beast?

In binaries then?

Nope

```
❯ pacman -Ql systemd | grep /bin/ | awk '{print $2}' | grep -v '/$' | column
/usr/bin/bootctl                        /usr/bin/systemd-ask-password           /usr/bin/systemd-nspawn
/usr/bin/busctl                         /usr/bin/systemd-cat                    /usr/bin/systemd-path
/usr/bin/coredumpctl                    /usr/bin/systemd-cgls                   /usr/bin/systemd-pty-forward
/usr/bin/homectl                        /usr/bin/systemd-cgtop                  /usr/bin/systemd-repart
/usr/bin/hostnamectl                    /usr/bin/systemd-confext                /usr/bin/systemd-resolve
/usr/bin/importctl                      /usr/bin/systemd-creds                  /usr/bin/systemd-run
/usr/bin/journalctl                     /usr/bin/systemd-cryptenroll            /usr/bin/systemd-socket-activate
/usr/bin/kernel-install                 /usr/bin/systemd-cryptsetup             /usr/bin/systemd-stdio-bridge
/usr/bin/localectl                      /usr/bin/systemd-delta                  /usr/bin/systemd-sysext
/usr/bin/loginctl                       /usr/bin/systemd-detect-virt            /usr/bin/systemd-sysusers
/usr/bin/machinectl                     /usr/bin/systemd-dissect                /usr/bin/systemd-tmpfiles
/usr/bin/mount.ddi                      /usr/bin/systemd-escape                 /usr/bin/systemd-tty-ask-password-agent
/usr/bin/networkctl                     /usr/bin/systemd-firstboot              /usr/bin/systemd-umount
/usr/bin/oomctl                         /usr/bin/systemd-home-fallback-shell    /usr/bin/systemd-vmspawn
/usr/bin/portablectl                    /usr/bin/systemd-hwdb                   /usr/bin/systemd-vpick
/usr/bin/resolvectl                     /usr/bin/systemd-id128                  /usr/bin/timedatectl
/usr/bin/run0                           /usr/bin/systemd-inhibit                /usr/bin/udevadm
/usr/bin/systemctl                      /usr/bin/systemd-machine-id-setup       /usr/bin/userdbctl
/usr/bin/systemd-ac-power               /usr/bin/systemd-mount                  /usr/bin/varlinkctl
/usr/bin/systemd-analyze                /usr/bin/systemd-notify
```

Is systemd a monolithic beast?

In functionality?

Maybe

But is it really a problem? Probably not

Up to the next level!

Level 2: Units

systemd units

Everything is a unit!

- Goals
- Service definitions
- File systems and paths
- Tasks
- Resource groups

systemd units

What units does my system have?

- systemctl list-units # show if service is active and description
- systemctl list-units "*net*" # restrict to units like **net**work
- systemctl list-units "*timer" # all timers or other unit types
- systemctl list-unit-files # show enabled state and preset

```
❯ systemctl list-units "*path"
  UNIT                                LOAD   ACTIVE SUB     DESCRIPTION
  grub-btrfs-snapper.path             loaded active waiting Monitors for new snapshots
  systemd-ask-password-plymouth.path loaded active waiting Forward Password Requests to Plymouth Directory Watch
  systemd-ask-password-wall.path      loaded active waiting Forward Password Requests to Wall Directory Watch

Legend: LOAD   → Reflects whether the unit definition was properly loaded.
        ACTIVE → The high-level unit activation state, i.e. generalization of SUB.
        SUB    → The low-level unit activation state, values depend on unit type.

3 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

```
❯ systemctl list-unit-files "*ssh*"
UNIT FILE                   STATE      PRESET
sshd-unix-local@.service    alias      -
sshd-vsock@.service         alias      -
sshd.service                enabled    disabled
sshd@.service               indirect   disabled
sshdgenkeys.service         disabled   disabled
gpg-agent-ssh@.socket       static     -
sshd-unix-local.socket      generated  -
sshd-vsock.socket           generated  -
ssh-access.target           static     -

9 unit files listed.
```

systemd units and location

Where are they stored?

- systemctl show -p UnitPath --value
- systemctl --user show -p UnitPath --value

There is an easier way...

systemd units and location

systemd-analyze unit-paths

Tip: usually you will be using /etc/systemd/systemd

```
❯ systemd-analyze unit-paths
/etc/systemd/system.control
/run/systemd/system.control
/run/systemd/transient
/run/systemd/generator.early
/etc/systemd/system
/etc/systemd/system.attached
/run/systemd/system
/run/systemd/system.attached
/run/systemd/generator
/usr/local/lib/systemd/system
/usr/lib/systemd/system
/run/systemd/generator.late
```

systemd-analyze unit-files

Tip: supports regular expressions

```
❯ systemd-analyze unit-files "ssh*"
ids: sshd-vsock.socket → /run/systemd/generator/sshd-vsock.socket
ids: sshd-unix-local.socket → /run/systemd/generator/sshd-unix-local.socket
ids: sshd-unix-local@.service → sshd@.service
ids: sshd@.service → /usr/lib/systemd/system/sshd@.service
ids: sshdgenkeys.service → /usr/lib/systemd/system/sshdgenkeys.service
ids: sshd-vsock@.service → sshd@.service
ids: ssh-access.target → /usr/lib/systemd/system/ssh-access.target
ids: sshd.service → /usr/lib/systemd/system/sshd.service
aliases: sshd-vsock.socket ← sshd-vsock.socket
aliases: sshd.service ← sshd.service
aliases: sshdgenkeys.service ← sshdgenkeys.service
aliases: ssh-access.target ← ssh-access.target
aliases: sshd-unix-local.socket ← sshd-unix-local.socket
aliases: sshd@.service ← sshd-unix-local@.service, sshd@.service, sshd-vsock@.service
```

systemd unit types

It's all about units!

- service
- timer
- mount
- automount
- device
- path
- scope
- socket
- swap
- target
- slice

systemd services

- **Goal**
  - Processes
  - Single execution or longer period
  - Can be triggered by other unit types
- **Which ones?**
  - systemctl --type=service
  - systemctl list-units "*service"
- **Interact**
  - systemctl (start|stop|restart) NAME.service
  - systemctl status NAME.service
  - systemctl cat NAME.service

systemd targets

- **Goal**
  - A state to achieve
- **Which ones?**
  - systemctl --type=target
  - systemctl get-default
    - graphical
  - systemctl set-default multi-user
  - systemctl isolate multi-user (← run on test system first)

systemd timers

- **Goal**
  - Scheduled task
  - Fine-grained cron
  - Monotonic
    - Run after time span (boot, service up)
- **Show available timers**
  - systemctl list-timers
  - systemctl list-timers --all

systemd mount and automount

- **Mount**
  - Mounts a file system
- **Automount**
  - Mount a path upon usage
  - Requires a mount unit with same name

systemd paths

- **Goal**
  - Monitors a file path
  - Can trigger a service, useful for event-driven actions
- **Example**

1) [**Unit**]
2) Description=Monitor nginx virtual host configuration and certificates
3) [**Path**]
4) PathModified=/path/to/nginx
5) PathModified=/path/to/nginx/virtual-hosts
6) PathModified=/path/to/nginx/certificates
7) Unit=monitor-nginx-configuration-changes-and-reload.service
8) [**Install**]
9) WantedBy=multi-user.target

systemd slices

- **What are slices?**
  - Group of processes
  - Allows settings resource limits, like:
    - CPU
    - Memory
    - Tasks
    - Block I/O
  - Uses Linux Control Group (cgroup)
    - cgroup v2
    - Support cgroup v1 removed in 258

Level 3: systemctl

systemctl

- **Basic management**
  - systemctl **cmd** MYAPP.service
    - start | stop | restart | reload | reload-or-restart | try-reload-or-restart
    - enable | disable | mask | unmask
- **Information**
  - systemctl **cmd** MYAPP.service
    - list-dependencies
    - status
    - show
- **Unit configuration**
  - systemctl **cmd** MYAPP.service
    - cat | edit

systemctl service information

systemctl status sshd.service

```
❯ systemctl status sshd.service
● sshd.service - OpenSSH Daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: disabled)
     Active: active (running) since Fri 2025-11-14 13:25:34 CET; 4 days ago
 Invocation: 4c63e3f94ee943b4a8b552add17a6e78
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 2644 (sshd)
      Tasks: 1 (limit: 4659)
     Memory: 7.3M (peak: 24.9M, swap: 932K, swap peak: 2.8M)
        CPU: 147ms
     CGroup: /system.slice/sshd.service
             └─2644 "sshd: /usr/bin/sshd -D [listener] 0 of 10-100 startups"

nov 14 13:25:34 cachyos systemd[1]: Starting OpenSSH Daemon...
nov 14 13:25:34 cachyos sshd[2644]: Server listening on 0.0.0.0 port 22.
nov 14 13:25:34 cachyos sshd[2644]: Server listening on :: port 22.
nov 14 13:25:34 cachyos systemd[1]: Started OpenSSH Daemon.
```

systemctl service information

systemctl cat sshd.service

```
❯ systemctl cat sshd.service
# /usr/lib/systemd/system/sshd.service
[Unit]
After=network.target sshdgenkeys.service
Before=ssh-access.target
Description=OpenSSH Daemon
Documentation=man:sshd(8) man:sshd_config(5)
Wants=sshdgenkeys.service ssh-access.target

[Service]
Type=notify-reload
ExecStart=/usr/bin/sshd -D
KillMode=process
Restart=always

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

systemctl service information

systemctl list-dependencies sshd.service

```
❯ systemctl list-dependencies sshd.service
sshd.service
○ ├─sshdgenkeys.service
● ├─system.slice
● ├─ssh-access.target
● └─sysinit.target
●   ├─dev-hugepages.mount
●   ├─dev-mqueue.mount
●   ├─kmod-static-nodes.service
○   ├─ldconfig.service
●   ├─lvm2-lvmpolld.socket
●   ├─lvm2-monitor.service
●   ├─plymouth-read-write.service
●   ├─plymouth-start.service
●   ├─proc-sys-fs-binfmt_misc.automount
●   ├─sys-fs-fuse-connections.mount
●   ├─sys-kernel-config.mount
●   ├─sys-kernel-debug.mount
●   ├─sys-kernel-tracing.mount
○   ├─systemd-ask-password-console.path
○   ├─systemd-binfmt.service
○   ├─systemd-boot-random-seed.service
○   ├─systemd-firstboot.service
○   ├─systemd-hibernate-clear.service
○   ├─systemd-hwdb-update.service
○   ├─systemd-journal-catalog-update.service
●   ├─systemd-journal-flush.service
●   ├─systemd-journald.service
```

systemctl service information

systemctl show sshd.service

```
❯ systemctl show sshd.service
Id=sshd.service
Names=sshd.service
Requires=system.slice sysinit.target
Wants=ssh-access.target sshdgenkeys.service
WantedBy=multi-user.target
Conflicts=shutdown.target
Before=shutdown.target multi-user.target ssh-access.target
After=system.slice sshdgenkeys.service systemd-journald.socket basic.target sysinit.target network.target
Documentation="man:sshd(8)" "man:sshd_config(5)"
Description=OpenSSH Daemon
LoadState=loaded
ActiveState=active
FreezerState=running
SubState=running
FragmentPath=/usr/lib/systemd/system/sshd.service
UnitFileState=enabled
UnitFilePreset=disabled
StateChangeTimestamp=Fri 2025-11-14 13:25:34 CET
StateChangeTimestampMonotonic=3158397243
InactiveExitTimestamp=Fri 2025-11-14 13:25:34 CET
InactiveExitTimestampMonotonic=3158382408
ActiveEnterTimestamp=Fri 2025-11-14 13:25:34 CET
ActiveEnterTimestampMonotonic=3158397243
ActiveExitTimestampMonotonic=0
InactiveEnterTimestampMonotonic=0
CanStart=yes
CanStop=yes
CanReload=yes
```

Customizing systemd services

- **Override**
  - Complements initial configuration
  - Default name: override.conf
- **How?**
  - systemctl edit --full APP.service
  - **systemctl edit APP.service**
  - File: /usr/lib/systemd/system/APP.service.d/*.conf
  - File: /etc/systemd/system/APP.service.d/*.conf

Customizing systemd services

```
### Editing /etc/systemd/system/sshd.service.d/override.conf
### Anything between here and the comment below will become the contents of the drop-in file

TYPE SMART THINGS HERE!!!11!

### Edits below this comment will be discarded

STUPID THINGS CAN GO OVER HERE (^ read the line above)

### /usr/lib/systemd/system/sshd.service
# [Unit]
# After=network.target sshdgenkeys.service
# Before=ssh-access.target
# Description=OpenSSH Daemon
# Documentation=man:sshd(8) man:sshd_config(5)
# Wants=sshdgenkeys.service ssh-access.target
#
# [Service]
# Type=notify-reload
# ExecStart=/usr/bin/sshd -D
# KillMode=process
# Restart=always
#
# [Install]
# WantedBy=multi-user.target
```

Customizing systemd services

```
### Editing /etc/systemd/system/sshd.service.d/override.conf
### Anything between here and the comment below will become the contents of the drop-in file

[Service]
IPAccounting=yes

### Edits below this comment will be discarded


### /usr/lib/systemd/system/sshd.service
# [Unit]
# After=network.target sshdgenkeys.service
# Before=ssh-access.target
# Description=OpenSSH Daemon
# Documentation=man:sshd(8) man:sshd_config(5)
# Wants=sshdgenkeys.service ssh-access.target
#
# [Service]
# Type=notify-reload
# ExecStart=/usr/bin/sshd -D
# KillMode=process
# Restart=always
#
# [Install]
# WantedBy=multi-user.target
```

Customizing systemd services

Then what?

- Reload the configuration
  - Use: **systemctl daemon-reload**
  - **Do NOT use:** systemctl reload
- Restart your service
  - systemctl restart UNIT.service

Level 4: Logging

Journals

Migration from syslog to journals

- Text to binary format
  - We gain meta data
  - More speed (when used correctly)
- Your new friend: **journalctl**

journalctl

Retrieve information

- journalctl --disk-usage
- journalctl --verify

```
❯ journalctl --disk-usage
Archived and active journals take up 16.5M in the file system.
```

```
❯ journalctl --verify
PASS: /var/log/journal/f4d314a409514ba689f9d9bc37ecbc09/user-1000.journal
PASS: /var/log/journal/f4d314a409514ba689f9d9bc37ecbc09/system@2525f4cef0ee4e1a92298c2ce58662e2-0000000000000001-000642c4d1bf5496.journal
PASS: /var/log/journal/f4d314a409514ba689f9d9bc37ecbc09/system.journal
PASS: /var/log/journal/f4d314a409514ba689f9d9bc37ecbc09/system@2525f4cef0ee4e1a92298c2ce58662e2-0000000000000efc-000642c94c429ed4.journal
```

journalctl

Useful flags and filters

- -e = go to the end (of the pager)
- -x = show more details or help when available
- -f = follow, like tail -f
- --no-pager

journalctl

Limit output

- -n 10 = last 10 lines
- -u UNIT = show output of unit
- --since "today"
- --since="2025-11-01" --until="2025-11-08"

More useful flags and filters, like searching?

→ journalctl cheat sheet

Level 5: Control Tools

bootctl

Show boot related information

- TPM2 support
- EFI details
- Secure boot support
- Boot loader

busctl

Write or retrieve D-Bus information

- Services available on the bus (D-Bus)
- Status
  - busctl status org.freedesktop.resolve1
- Not for typical usage or system administration

coredumpctl

Useful for troubleshooting

- Subcommands
  - **debug**: start gdb on last coredump
  - **info**: show details for a specified entry
  - **list**: show entries for a provided EXE path

hostnamectl

Basic system details

- Hostname
- Hardware
- Virtualization
- Kernel
- Linux distribution
- Firmware

```
❯ hostnamectl
 Static hostname: cachyos
       Icon name: computer-vm
         Chassis: vm
      Machine ID: f4d314a409514ba689f9d9bc37ecbc09
         Boot ID: d7983952dbac4f9c96f4c6284cd27cd7
    AF_VSOCK CID: 1
  Virtualization: kvm
Operating System: CachyOS
          Kernel: Linux 6.17.6-2-cachyos
    Architecture: x86-64
 Hardware Vendor: QEMU
  Hardware Model: Standard PC _Q35 + ICH9, 2009_
Hardware Version: pc-q35-6.2
Firmware Version: 1.15.0-1
   Firmware Date: Tue 2014-04-01
    Firmware Age: 11y 7month 2w 3d
```

loginctl

Useful for system management

- Looks like *who*
- Can show, kill, lock sessions
- Enable lingering!
  - Run user services without active session

```
❯ loginctl
SESSION  UID USER    SEAT  LEADER CLASS   TTY   IDLE SINCE
      2 1000 michael seat0 719    user    tty2  no   -
      3 1000 michael -     725    manager -     no   -
      7 1000 michael -     42372  user    pts/1 no   -

3 sessions listed.
```

resolvectl

Useful for system management

- DNS configuration
- Active resolver

```
❯ resolvectl
Global
          Protocols: +LLMNR +mDNS -DNSOverTLS DNSSEC=no/unsupported
    resolv.conf mode: foreign
Fallback DNS Servers: 9.9.9.9#dns.quad9.net 2620:fe::9#dns.quad9.net 1.1.1.1#cloudflare-dns.com 2606:4700:4700::1111#cloudflare-dns.com
                      8.8.8.8#dns.google 2001:4860:4860::8888#dns.google

Link 2 (enp1s0)
    Current Scopes: DNS LLMNR/IPv4 LLMNR/IPv6 mDNS/IPv4 mDNS/IPv6
         Protocols: +DefaultRoute +LLMNR +mDNS -DNSOverTLS DNSSEC=no/unsupported
Current DNS Server: 192.168.122.1
       DNS Servers: 192.168.122.1
     Default Route: yes
```

timedatectl

Time synchronization

```
❯ timedatectl
               Local time: ma 2025-11-17 19:50:18 CET
           Universal time: ma 2025-11-17 18:50:18 UTC
                 RTC time: ma 2025-11-17 18:50:18
                Time zone: Europe/Amsterdam (CET, +0100)
System clock synchronized: yes
              NTP service: active
          RTC in local TZ: no
```

userdbctl

Useful for system management

- /etc/passwd++?

```
❯ userdbctl
  NAME                          DISPOSITION        UID   GID REALNAME                       HOME              SHELL
  root                          intrinsic            0     0 -                              /root             /usr/bin/bash
┌ ↓ begin system users ↓        system               1     - First system user              -                 -
  bin                           system               1     1 -                              /                 /usr/bin/nologin
  daemon                        system               2     2 -                              /                 /usr/bin/nologin
  mail                          system               8    12 -                              /var/spool/mail   /usr/bin/nologin
  ftp                           system              14    11 -                              /srv/ftp          /usr/bin/nologin
  rpc                           system              32    32 Rpcbind Daemon                 /var/lib/rpcbind  /usr/bin/nologin
  http                          system              33    33 -                              /srv/http         /usr/bin/nologin
  rpcuser                       system              34    34 RPC Service User               /var/lib/nfs      /usr/bin/nologin
  named                         system              40    40 BIND DNS Server                /                 /usr/bin/nologin
  dbus                          system              81    81 System Message Bus             /                 /usr/bin/nologin
  ntp                           system              87    87 Network Time Protocol          /var/lib/ntp      /bin/false
  polkitd                       system             102   102 User for polkitd               /                 /usr/bin/nologin
  rtkit                         system             133   133 RealtimeKit                    /proc             /usr/bin/nologin
  sddm                          system             960   960 SDDM Greeter Account           /var/lib/sddm     /usr/bin/nologin
  pcscd                         system             961   961 PC/SC Smart Card Daemon        /                 /usr/bin/nologin
  passim                        system             962   962 Local Caching Server           /usr/share/empty  /usr/bin/nologin
  openvpn                       system             963   963 OpenVPN                        /                 /usr/bin/nologin
  nm-openvpn                    system             964   964 NetworkManager OpenVPN         /                 /usr/bin/nologin
  git                           system             965   965 git daemon user                /                 /usr/bin/git-shell
  fwupd                         system             966   966 Firmware update daemon         /var/lib/fwupd    /usr/bin/nologin
  dnsmasq                       system             967   967 dnsmasq daemon                 /                 /usr/bin/nologin
  _talkd                        system             969   969 User for legacy talkd server   /                 /usr/bin/nologin
  avahi                         system             970   970 Avahi mDNS/DNS-SD daemon       /                 /usr/bin/nologin
  alpm                          system             971   971 Arch Linux Package Management  /                 /usr/bin/nologin
  uuidd                         system             972   972 UUID generator helper daemon   /var/lib/libuuid  /usr/bin/nologin
  tss                           system             973   973 tss user for tpm2              /                 /usr/bin/nologin
  systemd-timesync              system             974   974 systemd Time Synchronization   /                 /usr/bin/nologin
  systemd-resolve               system             975   975 systemd Resolver               /                 /usr/bin/nologin
  systemd-journal-remote        system             976   976 systemd Journal Remote         /                 /usr/bin/nologin
  systemd-oom                   system             977   977 systemd Userspace OOM Killer   /                 /usr/bin/nologin
  systemd-network               system             978   978 systemd Network Management     /                 /usr/bin/nologin
  systemd-coredump              system             979   979 systemd Core Dumper            /                 /usr/bin/nologin
└ ↑ end system users ↑          system             999     - Last system user               -                 -
  michael                       regular           1000  1000 Michael                        /home/michael     /bin/fish
┌ ↓ begin systemd-homed users ↓ regular          60001     - First systemd-homed user       -                 -
└ ↑ end systemd-homed users ↑   regular          60513     - Last systemd-homed user        -                 -
┌ ↓ begin dynamic greeter users ↓ dynamic        60578     - First dynamic greeter user     -                 -
└ ↑ end dynamic greeter users ↑ dynamic          60705     - Last dynamic greeter user      -                 -
┌ ↓ begin dynamic system users ↓ dynamic         61184     - First dynamic system user      -                 -
└ ↑ end dynamic system users ↑  dynamic          65519     - Last dynamic system user       -                 -
  nobody                        intrinsic         65534 65534 Kernel Overflow User         /                 /usr/bin/nologin
┌ ↓ begin container users ↓     container        524288     - First container user        -                 -
└ ↑ end container users ↑       container    1879048191     - Last container user         -                 -
┌ ↓ begin foreign users ↓       foreign      2147352576     - First foreign user          -                 -
└ ↑ end foreign users ↑         foreign      2147418111     - Last foreign user           -                 -

34 users listed.
```

More commands

Other commands available

- Depends on systemd version
- See overview: systemd commands

Overview of systemd commands

Command	Purpose	Available since
bootctl	Manage EFI firmware boot settings for the boot loader	198
busctl	Connect to *D-Bus* to query information	199
coredumpctl	List and process stored *core dumps*	195
homectl	Create, change, remove, inspect home directories	245
hostnamectl	Shows hostname, system type, hardware architecture, OS and kernel version, and allows hostname and location change	195
importctl	Download, import, and export disk images via systemd-importd	256
journalctl	Show systemd *journal* entries	183
kernel-install	Manage kernel and initrd images on the boot partition	198
localectl	Manage system locale and keyboard layout	195
loginctl	Controls the systemd login manager and can also be used to query information about users	183
machinectl	Introspect and control virtual machine and container registry registration manager	205
networkctl	Query or modify network links known to systemd-networkd	216

Level 6: Security and Service Hardening

systemd uses seccomp

- **seccomp**
  - Secure Computing
  - Defines which **system calls** are allowed
  - Policy: **continue** or **instant kill**
  - Great for **sandboxing**
  - Also used by Flatpak, snap, and others
- systemd + seccomp =
  - systemd-nspawn (containers)
  - service units

systemd-analyze security

Built-in security audit

```
❯ systemd-analyze security
UNIT                                        EXPOSURE PREDICATE HAPPY
NetworkManager.service                           7.8 EXPOSED   🙁
accounts-daemon.service                          5.5 MEDIUM    😐
alsa-state.service                               9.6 UNSAFE    😨
ananicy-cpp.service                              4.8 OK        🙂
archlinux-keyring-wkd-sync.service               2.0 OK        😀
auditd.service                                   9.4 UNSAFE    😨
avahi-daemon.service                             9.6 UNSAFE    😨
dbus-broker.service                              8.7 EXPOSED   🙁
dirmngr@etc-pacman.d-gnupg.service               9.6 UNSAFE    😨
dm-event.service                                 9.5 UNSAFE    😨
emergency.service                                9.5 UNSAFE    😨
getty@tty1.service                               9.6 UNSAFE    😨
gpg-agent@etc-pacman.d-gnupg.service             9.6 UNSAFE    😨
keyboxd@etc-pacman.d-gnupg.service               9.6 UNSAFE    😨
lvm2-lvmpolld.service                            9.5 UNSAFE    😨
plymouth-start.service                           9.5 UNSAFE    😨
polkit.service                                   1.2 OK        😀
power-profiles-daemon.service                    1.0 OK        😀
qemu-guest-agent.service                         9.6 UNSAFE    😨
rescue.service                                   9.5 UNSAFE    😨
rtkit-daemon.service                             7.2 MEDIUM    😐
scx_loader.service                               9.6 UNSAFE    😨
sddm.service                                     9.6 UNSAFE    😨
shadow.service                                   1.2 OK        😀
snapper-cleanup.service                          6.7 MEDIUM    😐
spice-vdagentd.service                           9.2 UNSAFE    😨
sshd.service                                     9.6 UNSAFE    😨
systemd-ask-password-console.service             9.4 UNSAFE    😨
systemd-ask-password-plymouth.service            9.5 UNSAFE    😨
systemd-ask-password-wall.service                9.4 UNSAFE    😨
systemd-bsod.service                             9.5 UNSAFE    😨
systemd-hostnamed.service                        1.7 OK        😀
systemd-importd.service                          5.0 MEDIUM    😐
systemd-journald.service                         4.9 OK        🙂
systemd-logind.service                           2.8 OK        😀
systemd-machined.service                         6.2 MEDIUM    😐
systemd-oomd.service                             1.8 OK        😀
systemd-resolved.service                         2.2 OK        🙂
systemd-rfkill.service                           9.4 UNSAFE    😨
systemd-timesyncd.service                        2.1 OK        😀
systemd-udevd.service                            7.0 MEDIUM    😐
systemd-userdbd.service                          2.3 OK        😀
udisks2.service                                  9.6 UNSAFE    😨
upower.service                                   2.4 OK        🙂
user@1000.service                                9.4 UNSAFE    😨
vboxservice.service                              9.6 UNSAFE    😨
vmtoolsd.service                                 9.6 UNSAFE    😨
vmware-vmblock-fuse.service                      9.6 UNSAFE    😨
```

```
❯ systemd-analyze security sshd.service
  NAME                                                    DESCRIPTION
✗ RootDirectory=/RootImage=                               Service runs within the host's root directory
  SupplementaryGroups=                                    Service runs as root, option does not matter
  RemoveIPC=                                              Service runs as root, option does not apply
✗ User=/DynamicUser=                                      Service runs as root user
✗ CapabilityBoundingSet=~CAP_SYS_TIME                     Service processes may change the system clock
✗ NoNewPrivileges=                                        Service processes may acquire new privileges
✓ AmbientCapabilities=                                    Service process does not receive ambient capabilities
✗ PrivateDevices=                                         Service potentially has access to hardware devices
✗ ProtectClock=                                           Service may write to the hardware clock or system clock
✗ CapabilityBoundingSet=~CAP_SYS_PACCT                    Service may use acct()
✗ CapabilityBoundingSet=~CAP_KILL                         Service may send UNIX signals to arbitrary processes
✗ ProtectKernelLogs=                                      Service may read from or write to the kernel log ring buffer
✗ CapabilityBoundingSet=~CAP_WAKE_ALARM                   Service may program timers that wake up the system
✗ CapabilityBoundingSet=~CAP_(DAC_*|FOWNER|IPC_OWNER)     Service may override UNIX file/IPC permission checks
✗ ProtectControlGroups=                                   Service may modify the control group file system
✗ CapabilityBoundingSet=~CAP_LINUX_IMMUTABLE              Service may mark files immutable
✗ CapabilityBoundingSet=~CAP_IPC_LOCK                     Service may lock memory into RAM
✗ ProtectKernelModules=                                   Service may load or read kernel modules
✗ CapabilityBoundingSet=~CAP_SYS_MODULE                   Service may load kernel modules
✗ CapabilityBoundingSet=~CAP_BPF                          Service may load BPF programs
✗ CapabilityBoundingSet=~CAP_SYS_TTY_CONFIG               Service may issue vhangup()
✗ CapabilityBoundingSet=~CAP_SYS_BOOT                     Service may issue reboot()
✗ CapabilityBoundingSet=~CAP_SYS_CHROOT                   Service may issue chroot()
✗ PrivateMounts=                                          Service may install system mounts
✗ SystemCallArchitectures=                                Service may execute system calls with all ABIs
✗ CapabilityBoundingSet=~CAP_BLOCK_SUSPEND                Service may establish wake locks
✗ MemoryDenyWriteExecute=                                 Service may create writable executable memory mappings
✗ RestrictNamespaces=~user                                Service may create user namespaces
✗ RestrictNamespaces=~pid                                 Service may create process namespaces
✗ RestrictNamespaces=~net                                 Service may create network namespaces
✗ RestrictNamespaces=~uts                                 Service may create hostname namespaces
✗ RestrictNamespaces=~mnt                                 Service may create file system namespaces
✗ CapabilityBoundingSet=~CAP_LEASE                        Service may create file leases
✗ CapabilityBoundingSet=~CAP_MKNOD                        Service may create device nodes
✗ RestrictNamespaces=~cgroup                              Service may create cgroup namespaces
✗ RestrictSUIDSGID=                                       Service may create SUID/SGID files
✗ RestrictNamespaces=~ipc                                 Service may create IPC namespaces
✗ ProtectHostname=                                        Service may change system host/domainname
```

systemd service properties

- Security options for units
  - 40+ properties (and counting)
  - Most are for sandboxing
  - Available in official documentation, but...
    - Technical
    - Difficult
    - Lack of examples

systemd service properties

Overview:

systemd unit settings

Setting	Description	Available since
CapabilityBoundingSet	Define what capabilities are allowed within the service unit	21
DeviceAllow	Allow access to a device	208
DevicePolicy	Define level of access to devices in /dev	208
ExecPaths	Define the paths from which programs can be executed	231
InaccessiblePaths	Define paths that should not be accessible	231
IPAccounting	Define if accounting on network packets and bytes should be used	235
KeyringMode	Controls kernel session keyring and define what is available to the service	235
LockPersonality	Prevent processes switching their personality, a kernel execution domain	235
MemoryDenyWriteExecute	Block creation or alteration of memory segments to become writable and executable as well	231
NoExecPaths	Exclude paths from which programs can be executed	231
NoNewPrivileges	Prevent processes from gaining new privileges	187
PrivateDevices	Only allow access to a subset of devices in /dev	209
PrivateMounts	Provides a separated mount namespace to the service	239
PrivateNetwork	Defines if access to the network interfaces of the host is possible	33
PrivatePIDs	Define a new PID namespace for the process and its children	257
PrivateTmp	Define new namespace for /tmp and /var/tmp directory	1
PrivateUsers	Define a new user namespace for the process and its children	232
ProcSubset	Define the subset of access by unit to /proc	247
ProtectClock	Limit access to clock information	245
ProtectControlGroups	Limit write access to control groups directory structure under /sys/fs/cgroup	232
ProtectHome	Define what level of access is possible to home directories	214
ProtectHostname	Defines if hostname or NIS domain name can be changed	242

systemd security hardening

```
###############################################################################
# Source: https://linux-audit.com/systemd/hardening-profiles/nginx/
# Profile version: 0.4 [2025-01-06]
###############################################################################
# Customizations:
# - Insert here the changes you made to the profile
###############################################################################

[Service]
# ==============================
# Paths
# ==============================
# Deny access to /dev/shm directory, suggested when using MemoryDenyWriteExecute=yes
# Details: https://linux-audit.com/systemd/settings/units/inaccessiblepaths/
InaccessiblePaths=/dev/shm
# Do not allow execution of files, except nginx itself
# Details: https://linux-audit.com/systemd/settings/units/noexecpaths/
NoExecPaths=/
# Details: https://linux-audit.com/systemd/settings/units/execpaths/
ExecPaths=/usr/sbin/nginx /usr/lib
# Allow creation of PID file and writing to log files (access|error).log
# Details: https://linux-audit.com/systemd/settings/units/readwritepaths/
ReadWritePaths=/run /var/log/nginx

# ==============================
# Capabilities and system calls
# ==============================
# Only allow: bind to ports < 1024, change file permissions/ownership so nginx workers can use them
# Details: https://linux-audit.com/systemd/settings/units/capabilityboundingset/
CapabilityBoundingSet=CAP_NET_BIND_SERVICE CAP_CHOWN CAP_DAC_OVERRIDE CAP_SETGID CAP_SETPCAP CAP_SETUID
```

systemd security hardening

Be aware after enabling new settings: Errors

```
# systemctl restart nginx.service
Job for nginx.service failed because the control process exited with error code.
See "systemctl status nginx.service" and "journalctl -xeu nginx.service" for details.
# systemctl status nginx.service
× nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
    Drop-In: /etc/systemd/system/nginx.service.d
             └─override.conf
     Active: failed (Result: exit-code) since Tue 2025-03-11 14:22:15 CET; 2min 8s ago
   Duration: 2d 4h 36.240s
       Docs: man:nginx(8)
    Process: 3235 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=203/EXEC)
        CPU: 26ms

Mar 11 14:22:14 debian-test systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Mar 11 14:22:15 debian-test (nginx)[3235]: nginx.service: Failed to execute /usr/sbin/nginx: Permission denied
Mar 11 14:22:15 debian-test (nginx)[3235]: nginx.service: Failed at step EXEC spawning /usr/sbin/nginx: Permission denied
Mar 11 14:22:15 debian-test systemd[1]: nginx.service: Control process exited, code=exited, status=203/EXEC
Mar 11 14:22:15 debian-test systemd[1]: nginx.service: Failed with result 'exit-code'.
Mar 11 14:22:15 debian-test systemd[1]: Failed to start nginx.service - A high performance web server and a reverse proxy server.
```

systemd security hardening

Troubleshooting security hardening issues

- Generic tips
  - Apply in small steps
  - systemctl status myapplication.service
  - journalctl -u myapplication.service
  - systemd-analyze unit-shell UNIT.service ls
- Capabilities and system calls
  - Use **ldd** and **strace**
  - Use property **SystemCallLog**
    - SystemCallLog=~@system-service chroot

Level 7: Health Monitoring, Recovery, Statistics

systemd-analyze

Timing options

- Default option: **time**
- More detailed:
  - **blame**
  - **critical-chain**

```
❯ systemd-analyze time
Startup finished in 5.678s (kernel) + 6.332s (userspace) = 12.011s
graphical.target reached after 6.327s in userspace.
```

```
❯ systemd-analyze blame
8.708s plocate-updatedb.service
4.310s plymouth-quit-wait.service
4.309s plymouth-quit.service
 598ms grub-btrfs-snapper.service
 530ms man-db.service
 408ms dev-mapper-luks\x2daca9da5e\x2de81b\x2d4a7a\x2dbb5a\x2d5b93a9b8bbbd.device
 194ms systemd-udev-trigger.service
 190ms ufw.service
 172ms NetworkManager.service
 105ms systemd-tmpfiles-setup-dev-early.service
  95ms systemd-udevd.service
  92ms user@1000.service
  91ms systemd-timesyncd.service
  83ms udisks2.service
  78ms systemd-resolved.service
  70ms systemd-zram-setup@zram0.service
  65ms systemd-remount-fs.service
  65ms dev-zram0.swap
  59ms systemd-tmpfiles-setup-dev.service
  58ms systemd-logind.service
  57ms systemd-tmpfiles-clean.service
  55ms systemd-journald.service
  54ms upower.service
```

systemd failed units

- **Monitoring**
  - systemctl list-units --state=failed
  - systemctl is-failed UNIT
    - response: failed
  - systemctl reset-failed [UNIT]

systemd failed units

Example:

1) × monitor-nginx-configuration-changes-and-reload.path - Monitor virtual host configuration of Nginx
2) Loaded: loaded (/etc/systemd/system/monitor-nginx-configuration-changes-and-reload.path; enabled; preset: enabled)
3) Active: failed (Result: **unit-start-limit-hit**) since Tue 2023-03-11 10:30:48 UTC; 8 months 1 day ago
4) Duration: 10month 5d 7h 36min 18.847s
5) Triggers: ● monitor-nginx-configuration-changes-and-reload.service
6)
7) May 05 17:54:29 hostname systemd[1]: Started monitor-nginx-configuration-changes-and-reload.path - Monitor virtual host configuration of Nginx.
8) Mar 11 10:30:48 hostname systemd[1]: monitor-nginx-configuration-changes-and-reload.path: Failed with result 'unit-start-limit-hit'.

systemd failed units

Self-healing services

- Configuration options
  - **Restart**
    - always | on-success | on-failure | on-abnormal | on-abort | on-watchdog
  - **RestartSec**
  - **StartLimit*** configuration
- **Example**

1) Restart=on-failure # only restart on failure
2) RestartSec=5 # wait five seconds before doing a restart
3) StartLimitBurst=3 # maximum of three restarts during 60s period (below)
4) StartLimitIntervalSec=60 # set interval time for recovery

Monitoring

Use accounting of services

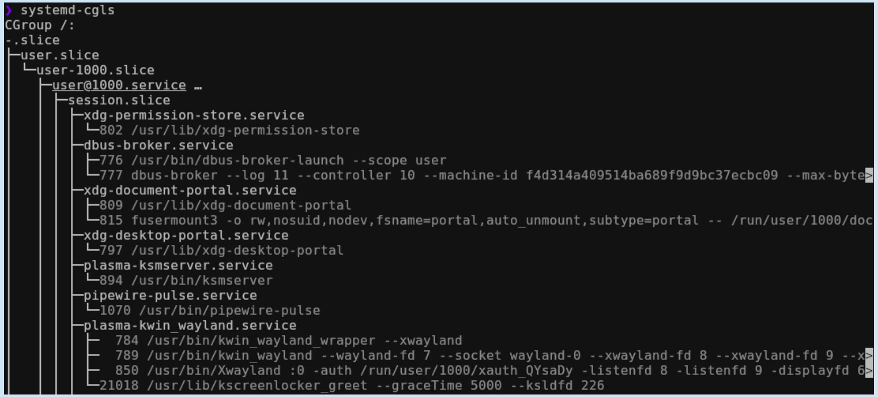
- IOAccounting
- IPAccounting
- MemoryAccounting
- TasksAccounting

```
❯ systemctl show sshd.service | grep -iE "^(CPU|IO|Mem)" | sort
CPUAffinityFromNUMA=no
CPUQuotaPeriodUSec=infinity
CPUQuotaPerSecUSec=infinity
CPUSchedulingPolicy=0
CPUSchedulingPriority=0
CPUSchedulingResetOnFork=no
CPUUsageNSec=96290000
CPUWeight=[not set]
IOAccounting=no
IOReadBytes=[not set]
IOReadOperations=[not set]
IOSchedulingClass=2
IOSchedulingPriority=4
IOWeight=[not set]
IOWriteBytes=[not set]
IOWriteOperations=[not set]
MemoryAccounting=yes
MemoryAvailable=2744881152
MemoryCurrent=7507968
MemoryDenyWriteExecute=no
MemoryHigh=infinity
MemoryKSM=no
MemoryLow=0
MemoryMax=infinity
MemoryMin=0
MemoryPeak=25907200
MemoryPressureThresholdUSec=200ms
MemoryPressureWatch=auto
MemorySwapCurrent=954368
MemorySwapMax=infinity
MemorySwapPeak=2953216
MemoryZSwapCurrent=0
MemoryZSwapMax=infinity
MemoryZSwapWriteback=yes
```

systemd cgroup / slices

Monitoring

- systemd-cgls
- systemd-cgtop



CGroup	Tasks	%CPU	Memory	Input/s	Output/s
/	470	-	1.2G	-	-
dev-hugepages.mount	-	-	48K	-	-
dev-mqueue.mount	-	-	108K	-	-
init.scope	1	-	18.4M	-	-
sys-fs-fuse-connections.mount	-	-	8K	-	-
sys-kernel-config.mount	-	-	4K	-	-
sys-kernel-debug.mount	-	-	20K	-	-
sys-kernel-tracing.mount	-	-	20K	-	-
system.slice	62	-	965.9M	-	-
system.slice/NetworkManager.service	4	-	14.3M	-	-
system.slice/accounts-daemon.service	4	-	1.4M	-	-
system.slice/ananicy-cpp.service	4	-	12.2M	-	-
system.slice/avahi-daemon.service	2	-	1M	-	-
system.slice/dbus-broker.service	2	-	3.1M	-	-
system.slice/dev-zram0.swap	-	-	188K	-	-
system.slice/home.mount	-	-	40K	-	-
system.slice/polkit.service	4	-	4.3M	-	-
system.slice/power-profiles-daemon.service	4	-	936K	-	-
system.slice/qemu-guest-agent.service	2	-	272K	-	-
system.slice/root.mount	-	-	24K	-	-
system.slice/rtkit-daemon.service	3	-	476K	-	-
system.slice/scx_loader.service	4	-	348K	-	-
system.slice/sddm.service	2	-	15.6M	-	-
system.slice/spice-vdagentd.service	3	-	1.6M	-	-
system.slice/srv.mount	-	-	20K	-	-
system.slice/sshd-vsock.socket	-	-	8K	-	-
system.slice/sshd.service	1	-	7M	-	-
system.slice/system-dbus\x2d:1.3\x2dorg.kde.powerdevil.discretegpuhelper.slice	-	-	4K	-	-
system.slice/system-dbus\x2d:1.3\x2dorg.kde.ufw.slice	-	-	64K	-	-

Summary

Management summary

It's a lot

Thanks for playing!

Slides available at michaelboelen.com

Want more in-depth videos about this subject?
Coming soon: @LinuxLorem (YouTube)

Bonus levels

run0

It's like *sudo*

- run0 CMD

Commands: systemd-*

Most you won't need daily, but some can be very handy

- systemd-ac-power (Report whether we are connected to an external power source)
- systemd-analyze (Analyze and debug system manager)
- systemd-ask-password (Query the user for a system password)
- systemd-cat (Connect a pipeline or program's output with the journal)
- systemd-cgls (Recursively show control group contents)
- systemd-cgtop (Show top control groups by their resource usage)
- systemd-confext (Activates System Extension Images)
- systemd-creds (Lists, shows, encrypts and decrypts service credentials)
- systemd-cryptenroll (Enroll PKCS#11, FIDO2, TPM2 token/devices to LUKS2 encrypted volumes)
- systemd-cryptsetup (Full disk decryption logic)
- systemd-delta (Find overridden configuration files)
- systemd-detect-virt (Detect execution in a virtualized environment)
- systemd-dissect (Dissect Discoverable Disk Images (DDIs))
- systemd-escape (Escape strings for usage in systemd unit names)
- systemd-firstboot (Initialize basic system settings on or before the first boot-up of a system)
- systemd-home-fallback-shell (command link)
- systemd-hwdb (Hardware database management tool)
- systemd-id128 (Generate and print sd-128 identifiers)
- systemd-inhibit (Execute a program with an inhibition lock taken)
- systemd-machine-id-setup (Initialize the machine ID in /etc/machine-id)
- systemd-mount (Establish and destroy transient mount or auto-mount points)

Commands: systemd-* (cont.)

- systemd-notify (Notify service manager about start-up completion and other daemon status changes)
- systemd-nspawn (Spawn a command or OS in a lightweight container)
- systemd-path (List and query system and user paths)
- systemd-pty-forward (Run a command with a custom terminal background color or title)
- systemd-repart (Automatically grow and add partitions, and generate disk images (DDIs))
- systemd-resolve (command link)
- systemd-run (Run programs in transient scope units, service units, or path-, socket-, or timer-tr...)
- systemd-socket-activate (Test socket activation of daemons)
- systemd-stdio-bridge (D-Bus proxy)
- systemd-sysext (Activates System Extension Images)
- systemd-sysusers (Allocate system users and groups)
- systemd-tmpfiles (Create, delete, and clean up files and directories)
- systemd-tty-ask-password-agent (List or process pending systemd password requests)
- systemd-umount (Establish and destroy transient mount or auto-mount points)
- systemd-vmspawn (Spawn an OS in a virtual machine)
- systemd-vpick (Resolve paths to ".v/" versioned directories)

systemd-analyze

A swiss-army toolkit for systemd

- architectures (List known CPU architectures)
- blame (Print list of running units ordered by time to init)
- calendar (Normalize repetitive calendar events and calculate when they elapse next)
- capability (List Linux capabilities along with their numeric IDs)
- cat-config (Show contents of a config file)
- compare-versions (Compare two version strings)
- condition (Evaluate Condition and Assert assignments)
- critical-chain (Print a tree of the time critical chain of units)
- dot (Output dependency graph in dot(1) format)
- dump (Output serialization of server state)
- exit-status (List exit statuses along with their class)
- fdstore (List contents of service unit's file descriptor store)
- filesystems (List filesystems)
- has-tpm2 (Report TPM2 support)
- image-policy (Analyze image policy string)
- inspect-elf (Parse and print ELF object packaging metadata)
- malloc (Output internal memory state of D-Bus service)
- pcrs (Show known TPM2 PCRs)
- plot (Output SVG graphic showing service initialization)
- security (Analyze security settings of specified service units)
- smbios11 (Show SMBIOS Type #11 strings passed to the system)
- srk (Read Storage Root Key from TPM2 device)
- syscall-filter (List system calls contained in the specified system call set)
- time (Print timing statistics)
- timespan (Parse time span and output the normalized form)
- timestamp (Parse timestamp and output the normalized form)
- unit-paths (List all directories from which unit files may be loaded)
- verify (Check unit files for correctness)

systemd-analyze

systemd-analyze calendar

```
1 [Unit]
2 Description=...
3 [Timer]
4 OnCalendar=*-*-* *:00:30
5 Persistent=...
6 Unit=...
7 [Install]
8 ...
```

```
michael@linux:~$ systemd-analyze calendar --iterations=6 "Sat,Sun 10:05:00"
  Original form: Sat,Sun 10:05:00
Normalized form: Sat,Sun *-*-* 10:05:00
    Next elapse: Sat 2025-11-15 10:05:00 CET
       (in UTC): Sat 2025-11-15 09:05:00 UTC
       From now: 2 days left
       Iter. #2: Sun 2025-11-16 10:05:00 CET
       (in UTC): Sun 2025-11-16 09:05:00 UTC
       From now: 3 days left
       Iter. #3: Sat 2025-11-22 10:05:00 CET
       (in UTC): Sat 2025-11-22 09:05:00 UTC
       From now: 1 week 2 days left
       Iter. #4: Sun 2025-11-23 10:05:00 CET
       (in UTC): Sun 2025-11-23 09:05:00 UTC
       From now: 1 week 3 days left
       Iter. #5: Sat 2025-11-29 10:05:00 CET
       (in UTC): Sat 2025-11-29 09:05:00 UTC
       From now: 2 weeks 2 days left
       Iter. #6: Sun 2025-11-30 10:05:00 CET
       (in UTC): Sun 2025-11-30 09:05:00 UTC
       From now: 2 weeks 3 days left
```

systemd-cat

Add entries to the journals

```
❯ echo "test entry" | systemd-cat --identifier=testapp

~ michael@cachyos
❯ journalctl -n1
nov 18 18:48:54 cachyos testapp[50410]: test entry
```

systemd-path

Find common (configuration) paths

```
❯ systemd-path
binfmt: /usr/lib/binfmt.d
catalog: /usr/lib/systemd/catalog
modules-load: /usr/lib/modules-load.d
search-binaries: /usr/local/sbin:/usr/local/bin:/usr/bin:/usr/bin/site_perl:/usr/bin/vendor_perl:/usr/bin/cor>
search-binaries-default: /usr/local/bin:/usr/bin
search-configuration: /home/michael/.config:/etc
search-configuration-factory: /usr/local/share/factory/etc:/usr/share/factory/etc
search-library-arch: /home/michael/.local/lib/x86_64-linux-gnu:/usr/lib
search-library-private: /home/michael/.local/lib:/usr/local/lib:/usr/lib
search-shared: /home/michael/.local/share:/usr/local/share:/usr/share
search-state-factory: /usr/local/share/factory/var:/usr/share/factory/var
sysctl: /usr/lib/sysctl.d
system-binaries: /usr/bin
system-configuration: /etc
system-configuration-factory: /usr/share/factory/etc
system-credential-store: /etc/credstore
system-credential-store-encrypted: /etc/credstore.encrypted
system-include: /usr/include
system-library-arch: /usr/lib
system-library-private: /usr/lib
system-runtime: /run
system-runtime-logs: /run/log
system-search-credential-store: /etc/credstore:/run/credstore:/usr/local/lib/credstore:/usr/lib/credstore
system-search-credential-store-encrypted: /etc/credstore.encrypted:/run/credstore.encrypted:/usr/local/lib/cr>
system-shared: /usr/share
system-state-cache: /var/cache
system-state-factory: /usr/share/factory/var
system-state-logs: /var/log
system-state-private: /var/lib
system-state-spool: /var/spool
systemd-initrd-preset: /usr/lib/systemd/initrd-preset
systemd-search-network: /etc/systemd/network:/run/systemd/network:/usr/local/lib/systemd/network:/usr/lib/sys>
systemd-search-system-environment-generator: /run/systemd/system-environment-generators:/etc/systemd/system-e>
```

systemd-run

Create a transient service

- Great for testing properties when hardening units

```
systemd-run --pty --property=PrivateNetwork=yes ip link
```

Useful links and resources

systemd

- Getting more out of systemd
  - **Units**
    - Linux Audit: Overview of systemd units
  - **Cheat sheets**
    - Linux Audit: journalctl
    - Linux Audit: systemctl

systemd security tips

- **Guides**
  - systemd service hardening
  - Resolving basic issues with failed systemd service
  - Steps to take when a service unit fails after hardening
- **Unit settings** (properties)
  - Overview
  - + Alternative description
  - + Examples
- **Ready-to-use[1] profiles**
  - Hardening profiles

[1] may need adjustments depending on your distribution and configuration

More slides?

See this and other speedruns:

- linuxlorem.com/speedruns/

Suggestions?

- **Contact**
  - Details: michaelboelen.com
  - @mboelen