

systemd

ytshih (2025, CC BY-SA), credits to csjhuang

Outline

- Intro to systemd
- Intro to control groups
- systemd unit
 - how to write / edit a systemd unit
- systemd service
- systemd target
- systemd timer
- systemd journal

Systemd

- A system and service manager for Linux
- The default init system for Debian since Debian 8 ("jessie")
- Transactional dependency-based service control logic
- Supports SysV and LSB init scripts and works as a replacement for sysvinit
- Keeps track of processes using Linux control groups

Systemd

- Includes a **logging daemon** and utilities to control
 - hostname
 - time / date
 - locale
 - user sessions
 - system accounts
 - runtime directories
 - network configuration
 - name resolution
 - network time synchronornization

Control Groups v2 (cgroups)

- A mechanism to organize processes hierarchically
- Distribute system resources along the hierarchy
 - i.e. CPU, Memory, I/O, PID

```
• linux1
  State: running
  Units: 389 loaded (incl. loaded aliases)
  Jobs: 0 queued
  Failed: 0 units
  Since: Tue 2025-10-07 00:59:21 CST; 1 month 10 days ago
  systemd: 252.39-1~deb12u1
  CGroup: /
          └─init.scope
              └─┬─ /sbin/init
                  └─system.slice
                      ├─autofs.service
                      │   └─┬─1170 /usr/sbin/automount --pid-file /var/run/autofs.pid
                      │       └─cron.service
                      │           └─┬─1153 /usr/sbin/cron -f
                      │               └─dbus.service
                      │                   └─┬─881 /usr/bin/dbus-daemon --system --address=systemd: --nofork ->n
                      │                       └─open-vm-tools.service
                      │                           └─┬─870 /usr/bin/vmtoolsd
                      │                               └─rsyslog.service
                      │                                   └─┬─885 /usr/sbin/rsyslogd -n -iNONE
                      │                                       └─system-getty.slice
                      │                                           └─┬─getty@tty1.service
                      │                                               └─┬─1156 /sbin/agetty -o "-p -- \u" --noclear - linux
                      │                                                   └─systemd-journald.service
                      │                                                       └─┬─448 /lib/systemd/systemd-journald
                      │                                                           └─systemd-logind.service
                      │                                                               └─┬─1154 /lib/systemd/systemd-logind
                      │                                                                   └─systemd-timesyncd.service
                      │                                                                       └─┬─872 /lib/systemd/systemd-timesyncd
```

Systemctl

- `systemctl status [unit]`
- `systemctl cat unit`
- `systemctl start / stop`
- `systemctl enable / disable`
- `systemctl kill`
- `systemctl mask / unmask`
- `systemctl list-unit-files`
- `systemctl dependencies`

Systemd unit

- [Unit]
 - Generic information about the unit
- [Service]
 - Service specific configuration
 - Including options in systemd.exec
- [Install]
 - Installation information for the unit
 - Used by enable / disable

```
# /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
```

Systemd drop-in snippet

- systemctl edit [unit]
- Edit or replace a drop-in snippet
 - Or the main unit file if --full is specified
- Using default editor

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

### Edits below this comment will be discarded

### /usr/lib/systemd/system/ssh.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
```


Systemd unit search path

- System
 - /etc/systemd/system/*
 - /usr/local/lib/systemd/system/*
 - /usr/lib/systemd/system/*
- User
 - \$XDG_CONFIG_DIRS/systemd/user/*
 - /etc/systemd/user/*
 - \$XDG_DATA_HOME/systemd/user/*
 - \$XDG_DATA_DIRS/systemd/user/*
 - /usr/local/lib/systemd/user/*
 - /usr/lib/systemd/user/*

Systemd unit dependencies

- Requires=
 - List the dependencies for this unit
 - **All the dependencies will be executed in parallel**
 - If one of the dependencies failed, this unit file will be labeled as failed
- Wants=
 - Similar to Requires but less restrictive
 - If one of the dependencies failed, this unit file **will not** be labeled as failed
 - Recommend in most cases
- BindsTo=
 - Similar to Requires but more restrictive
 - If one of the dependencies crashed, this unit file **will stop**
- Uphold=
 - Similar to BindsTo but more aggressive
 - If one of the dependencies crashed, **the dependencies will be restarted**

```
# /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
```

Systemd default dependencies

- Some dependencies are added unless `DefaultDependencies=no` is set
 - `Requires=sysinit.target`
 - `After=sysinit.target basic.target`
 - `Conflicts=shutdown.target`
 - `Before=shutdown.target`

Systemd unit orders

- Before=
 - This unit should run **before** the units listed in Before=
- After=
 - This unit should run **after** the units listed in After=
- These **is not** dependencies
 - Only startup order is specified, use Requires=, Wants=, BindsTo= for dependencies

```
# /usr/lib/systemd/system/ssh.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
```

Systemd unit installation

- Use `systemctl enable / disable` to install and uninstall systemd unit
- `systemctl` will install a symlink in `[target-name].wants/` directory

```
# /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
```

```
> sudo systemctl enable sshd.service
Created symlink '/etc/systemd/system/multi-user.target.wants/sshd.service' →
'/usr/lib/systemd/system/sshd.service'.
```

Systemd Target

- Systemd tasks are organized as **units** (e.g. `ssh.service`)
- **Targets** are groups of units
- Targets call units to put the system together
 - `graphical.target` calls all units that are necessary to start a workstation with graphical user interface
- Targets can build on top of another or depend on other targets
- At boot time, systemd activates the target **default.target**
 - Alias for another target, e.g. `graphical.target`
 - Try `systemctl get-default`

Common systemd targets

SysV Runlevel	systemd Target	Notes
0	poweroff.target	Halt the system
1	rescue.target	Single user mode
2, 3, 4	multi-user.target	Multi-user, non-graphical, users can usually login via multiple consoles or via the network
5	graphical.target	Multi-user, graphical
6	reboot.target	Reboot
-	emergency.target	Emergency shell
-	network-online.target	Network is online

```

Welcome to Fedora 20 (Heisenbug)!

[ OK ] Reached target Remote File Systems.
[ OK ] Listening on Delayed Shutdown Socket.
[ OK ] Listening on /dev/initctl Compatibility Named Pipe.
[ OK ] Reached target Paths.
[ OK ] Reached target Encrypted Volumes.
[ OK ] Listening on Journal Socket.
      Mounting Huge Pages File System...
      Mounting POSIX Message Queue File System...
      Mounting Debug File System...
      Starting Journal Service...
[ OK ] Started Journal Service.
      Mounting Configuration File System...
      Mounting FUSE Control File System...
[ OK ] Created slice Root Slice.
[ OK ] Created slice User and Session Slice.
[ OK ] Created slice System Slice.
[ OK ] Reached target Slices.
[ OK ] Reached target Swap.
      Mounting Temporary Directory...
[ OK ] Reached target Local File Systems (Pre).
      Starting Load Random Seed...
      Starting Load/Save Random Seed...
[ OK ] Mounted Huge Pages File System.
[ OK ] Mounted POSIX Message Queue File System.
[ OK ] Mounted Debug File System.
[ OK ] Mounted Configuration File System.
[ OK ] Mounted FUSE Control File System.
[ OK ] Mounted Temporary Directory.
[ OK ] Started Load Random Seed.
[ OK ] Started Load/Save Random Seed.
[ OK ] Reached target Local File Systems.
      Starting Recreate Volatile Files and Directories...
      Starting Trigger Flushing of Journal to Persistent Storage...
[ OK ] Started Recreate Volatile Files and Directories.
      Starting Record System Reboot/Shutdown in UTMPTMP...
[ OK ] Started Trigger Flushing of Journal to Persistent Storage.
[ OK ] Started Record System Reboot/Shutdown in UTMPTMP.
[ OK ] Reached target System Initialization.
[ OK ] Reached target Timers.
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Reached target Sockets.
[ OK ] Reached target Basic System.
      Starting Permit User Sessions...
      Starting D-Bus System Message Bus...
[ OK ] Started D-Bus System Message Bus.
      Starting Login Service...
      Starting Cleanup of Temporary Directories...
[ OK ] Started Permit User Sessions.
[ OK ] Started Cleanup of Temporary Directories.
      Starting Console Getty...
[ OK ] Started Console Getty.
[ OK ] Reached target Login Prompts.
[ OK ] Started Login Service.
[ OK ] Reached target Multi-User System.

Fedora release 20 (Heisenbug)
Kernel 3.9.2-200.fc18.x86_64 on an x86_64 (console)

fedora login:

```

Systemd target hierarchy

- graphical.target
 - sddm.service (desktop manager)
 - multi-user.target
 - basic.target
 - sockets.target
 - slices.target
 - sysinit.target
 - timers.target
 - getty.target
 - remote-fs.target
 - ...

```
systemctl list-dependencies multi-user.target
multi-user.target
● |dbus-broker.service
○ |nftables.service
● |nix-daemon.service
● |sshd.service
● |systemd-ask-password-wall.path
● |systemd-logind.service
● |systemd-networkd.service
● |systemd-user-sessions.service
● |basic.target
● |  |--.mount
● |  |--tmp.mount
● |  |--paths.target
● |  |--slices.target
● |  |  |--.slice
● |  |    |--system.slice
● |  |--sockets.target
```


Systemd service type

- oneshot
 - After the main process **exits**
- simple
 - Change state to **started immediately** after the main service process has been **forked**
 - After fork()
- exec
 - Similar to simple but **immediately after** the binary has been **executed**
 - After fork() returned
- notify, notify-reload
 - Similar to simple but expect the **service sends a "READY=1"** via sd_notify(3)
 - Reload supported by sending signal **SIGHUP** to the process

```
# /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
```

Systemd service restart policy

- Restart=
 - Configure whether the service shall be restarted

	no	always	on-success	on-failure	on-abnormal	on-abort
Clean exit code	✗	✓	✓	✗	✗	✗
Unclean exit code	✗	✓	✗	✓	✗	✗
Unclean signal	✗	✓	✗	✓	✓	✓
Timeout	✗	✓	✗	✓	✓	✗
OOM	✗	✓	✗	✓	✓	✗

Systemd exec

- ExecStart=
 - Commands that are executed when this service is started
- ExecStartPre=, ExecStartPost=
 - Additional commands that are executed **before or after** the command in ExecStart=
 - Can be prefixed with a "-", allowing this command to fail and not affect the rest
- ExecReload=
 - Commands to execute to trigger a configuration reload
- ExecStop=
 - Commands to execute to stop the service started via ExecStart=
- TimeoutSec=, TimeoutStartSec=, TimeoutStopSec=
 - The time to wait for startup and stop

Systemd exec - paths

- WorkingDirectory=
 - Working directory, relative to the service's root directory
- RootDirectory=
 - Service root directory, relative to the host's root directory

Systemd exec - users

- User=, Group=
 - Set the UNIX user and group that the processes are executed as
 - Default is **root**
- DynamicUser=
 - If set, a UNIX user and group pair is allocated dynamically when the unit is started
 - Allocated from the UID / GID range 61184-65519

Systemd exec - resources

- LimitCPU=, LimitFSIZE=, ...
 - Set soft and hard limits on various resources for executed processes
 - e.g. LimitCPU= equals to `ulimit -t`
- Nice=
 - Set the default nice level (scheduling priority) for executed processes
- CPUSchedulingPolicy=
 - Set the CPU scheduling policy.
 - Takes one of **other**, **batch**, **idle**, **fifo**, **rr**
- CPUSchedulingPriority=
 - Set the CPU scheduling priority processes
- CPUAffinity=
 - Controls the CPU affinity of the executed processes

Systemd exec - unit directories

- RuntimeDirectory=
 - Below path /run/ for system units
 - Use RuntimeDirectoryPreserve= to preserve runtime directory
- StateDirectory=
 - Below path /var/lib/ for system units
- CacheDirectory=
 - Below path /var/cache/ for system units
- LogsDirectory=
 - Below path /var/log/ for system units
- ConfigurationDirectory=
 - Below path /etc/ for system units

Systemd environment

- Environment=
 - Set environment variables for executed processes
 - e.g. Environment="VAR1=word1 word2" "VAR2=word3"
- EnvironmentFile=
 - Read environment variables from a text file
 - Optionally prefixed with "-", allowing the file to be absent

Systemd standard I/O and logging

- **StandardInput=**
 - Controls where file descriptor 0 (stdin) of the executing processes is connected to
 - Takes one of *null*, *tty*, *file:path*, *socket*, *fd:name*, ...
- **StandardOutput=, StandardError=**
 - Controls where file descriptor 1 (stdout) and 2 (stderr)
 - Takes one of *inherit*, *null*, ***journal***, *kmsg*, *file:path*, *append:path*, *truncate:path*, *socket*, *fd:name*

systemd-journald

- A system service collects logs
 - Stores logs into a structured, indexed journals
 - From kmsg, syslog, service unit, ...
- Stores log data in either
 - Persistent in /var/log/journal
 - Volatile in /run/log/journal, lost at reboot

systemd.timer (5)

- A “systemd” way to control periodic process
- A timer named **X.timer** will trigger **X.service**
- Usually installed as dependency of *timers.target*
 - *multi-user.target* =(depends)=> *basic.target* =(depends)=> *timers.target*
- systemd unit (including service) will be introduced in the next course

systemd.time (7)

- OnCalendar is similar to cron time format
 - *weekday year-month-date hour:minute:second*
- RandomizedDelaySec can prevent timers on multiple machines from executing at the same time
- Use systemd-analyze to check syntax

```
> systemd-analyze calendar '*--* 00:00:00'  
Normalized form: *--* 00:00:00  
Next elapse: Mon 2025-10-27 00:00:00 CST  
  (in UTC): Sun 2025-10-26 16:00:00 UTC  
From now: 8h left
```

```
# /lib/systemd/system/apt-daily-upgrade.timer  
[Unit]  
Description=Daily apt upgrade and clean activities  
After=apt-daily.timer  
  
[Timer]  
OnCalendar=*--* 6:00  
RandomizedDelaySec=60m  
Persistent=true  
  
[Install]  
WantedBy=timers.target
```

systemd unit template

- Unit name can be parameterized by a single argument
 - The argument is called “instance name”
- Must have a single “@” at the end of the unit name prefix
 - e.g. btrfs.scrub@.service, wg-quick@.service
- Use %i or %I specifier in unit file to get the instance name

```
[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/usr/bin/wg-quick up %i
ExecStop=/usr/bin/wg-quick down %i
ExecReload=/bin/bash -c 'exec /usr/bin/wg synccnf %i <(exec /usr/bin/wg-quick strip %i)'
Environment=WG_ENDPOINT_RESOLUTION_RETRIES=infinity
```

Example - 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
```

Example - haproxy.service

```
# /usr/lib/systemd/system/haproxy.service
[Unit]
Description=HAProxy Load Balancer
After=network-online.target
Wants=network-online.target

[Service]
EnvironmentFile=-/etc/default/haproxy
EnvironmentFile=-/etc/sysconfig/haproxy
Environment="CONFIG=/etc/haproxy/haproxy.cfg" "PIDFILE=/run/haproxy.pid" "EXTRA_OPTS=-S
/run/haproxy-master.sock"
ExecStart=/usr/bin/haproxy -Ws -f $CONFIG -p $PIDFILE $EXTRA_OPTS
ExecReload=/usr/bin/haproxy -Ws -f $CONFIG -c $EXTRA_OPTS
ExecReload=/bin/kill -USR2 $MAINPID
KillMode=mixed
Restart=always
SuccessExitStatus=143
Type=notify

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

Example - apt-daily@.timer / .service

```
# /lib/systemd/system/apt-daily.service
[Unit]
Description=Daily apt download activities
Documentation=man:apt(8)
ConditionACPower=true
After=network.target network-online.target
systemd-networkd.service NetworkManager.service
connman.service

[Service]
Type=oneshot
ExecStartPre=/usr/lib/apt/apt-helper wait-online
ExecStart=/usr/lib/apt/apt.systemd.daily update
```

```
# /lib/systemd/system/apt-daily.timer
[Unit]
Description=Daily apt download activities

[Timer]
OnCalendar=*-*-* 6,18:00
RandomizedDelaySec=12h
Persistent=true

[Install]
WantedBy=timers.target
```


Other utilities

- `systemd-hostnamed` (8) / `hostnamectl` (1)
- `systemd-timedated` / `timedatectl` (1)
- `systemd-localed` (8) / `localectl` (1)
- `systemd-logind` (8) / `loginctl` (1)
- `systemd-sysusers` (8)
- `systemd-networkd` (8) / `networkctl` (1)
- `systemd-resolved` (8) / `resolvectl` (1)
- `systemd-timesyncd` (8) / `timedatectl` (1)
- `systemd-boot` (7) / `bootctl` (1)
- `systemd-nspawn` (1) / `machinectl` (1)

Reference

- systemd.unit (5)
- systemd.service (5)
- systemd.timer (5)
- systemd.exec (5)
- systemctl (1)
- Debian Wiki - systemd <https://wiki.debian.org/systemd>
- Control Group v2 - The Linux Kernel documentation
<https://docs.kernel.org/admin-guide/cgroup-v2.html#>