Engineer

systemd

det som starter alt annet på Linux

Hva det er

systemd er programmet med prosess-ID 1 på nesten alle store Linux-distribusjoner: system- og tjenestestyreren som tar maskinen opp ved oppstart, starter og passer på tjenestene mens den går, og stenger alt ned i ordnet rekkefølge når den slås av. Navnet skrives med liten forbokstav, aldri SystemD eller System D, og d-en til slutt er det gamle Unix-tegnet for en daemon. systemd hører til familien av init-systemer. Der finner du også SysV-init-skriptene og Upstart, som det har avløst i de fleste distribusjoner, OpenRC og runit, som enkelte mindre distribusjoner fortsatt velger, og launchd, som gjør den samme jobben på macOS.

Alt systemd styrer, er en enhet (på engelsk *unit*), beskrevet i en kort tekstfil med seksjoner og par av nøkkel og verdi. Manualen regner opp 11 typer: service, socket, target, device, mount, automount, timer, swap, path, slice og scope. En service kjører et program. En timer starter en annen enhet etter en tidsplan. En socket holder en nettverksport åpen og starter tjenesten sin først når den første tilkoblingen kommer inn. En target samler enheter til en tilstand maskinen kan nå, f.eks. et flerbrukersystem med nettverk. Krav og rekkefølge er to ulike innstillinger, så «denne trenger den» og «denne starter etter den» står på hver sin linje. systemd gjør dem om til én transaksjon av start- og stoppjobber, og den transaksjonen kan sjekkes før den kjøres.

To designvalg skiller systemd fra de skriptbaserte init-systemene som kom før. Hver tjeneste kjører i sin egen kontrollgruppe i Linux, så styreren holder oversikt over prosessene i gruppen og kan stoppe dem samlet, sette tak på minnet deres og føre regnskap med prosessortiden. Dessuten er prosjektet en hel samling verktøy og ikke bare en styrer: på sitt eget nettsted lister det opp en daemon for logging, journalen, og komponenter for innlogging, nettverksoppsett,

tidssynkronisering og navneoppslag. Hver distribusjon velger selv hvilke av tilleggene den slår på. Styreren på PID 1 er den delen som er så godt som universell.

Bruk begrepet i en beslutning

Bruk «Hva er systemd?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Daglig ettersyn av servere, e-post, kontoer og oppdateringer, slik at det alle er avhengige av, forblir kjedelig.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Drev 10 PostgreSQL- og MS SQL-servere på Ubuntu — 10 databaseservere produksjonssatt og holdt i god stand

Drev 10 PostgreSQL- og MS SQL-servere på Ubuntu Ta kontakt →

Hvorfor det er viktig

På en Linux-server er spørsmålet sjelden om du skal kjøre systemd, for det valget har distribusjonen allerede tatt. Det du faktisk avgjør, er om du bruker systemd som tjenestestyrer og planlegger, eller om du legger en annen prosessbehandler, omstartsregelen til en container-daemon eller cron oppå.

Bruker du systemd direkte, samles oppstartsrekkefølge, omstartsregel, logger, ressursgrenser og sandkasse i én fil per tjeneste. Filen leses av styreren som allerede kjører, uten en ekstra prosess for tjenestestyring. En enhet kan sette et tak på minnet med `MemoryMax=`, gjøre operativsystemets egne kataloger skrivebeskyttet for tjenesten med `ProtectSystem=` og kjøre den som en engangsbruker med `DynamicUser=`. Kommandoen `systemd-analyze security` gir så hver tjeneste en score for hvor utsatt den er, på en skala fra 0,0 til 10,0.

En jobb som en timer starter, skiller seg fra en linje i crontab. Den logger til journalen, får den samme sandkassen som enhver tjeneste, og `systemctl list-timers` viser når den gikk sist og når den går neste gang. Kunnskapen er dessuten flyttbar: en ingeniør som kan lese en enhetsfil, kan drifte de fleste Linux-servere, uansett hvilken distribusjon de kjører.

Prisen er også reell. Enhetene finnes bare på Linux, så et team som også leverer til macOS, skriver property lists for launchd, og et team som leverer til BSD-systemene, skriver rc-skript. Inne i de fleste containere finnes systemd ikke: der er selve applikasjonen PID 1, og omstartsregelen tilhører det som kjører containeren. Der containere og vanlige tjenester deler en vert, gjør Quadlet

fra Podman avstanden mindre. Quadlet gjør en containerbeskrivelse om til en vanlig tjenesteenhet, slik at begge typer arbeidslast står under én og samme tjenestestyrer.

Slik ser det ut i praksis

systemd bærer tre deler av selskapets egen plattform.

Implementeringsbeskrivelsen fra september 2026 om å drive selskapets offentlig tilgjengelige tjenester på én vert med 512 MB beskriver produksjonsverten som en maskin med én kjerne og omtrent 464 MB brukbart minne. På den kjørte en webserver som håndterer TLS for sju domener, git-forgen, onion-tjenesten, én server for hver av Gemini og Gopher og utestenging av inntrengere. Automatisering av sikkerhetskopiering er implementert separat; det fastslår ikke at krypterte sikkerhetskopier er aktive. Med så lite minne måtte hver fast kjørende prosess begrunne minnebruken sin. Derfor var systemd tjenestestyreren, uten en annen tjenestestyrerprosess lagt oppå, og containerne kjørte som Quadlet-enheter under den samme styreren i stedet for under en egen container-daemon. Da spørsmålet var om verten også kunne romme en onion-tjeneste, viste et dokumentert døgn med stikkprøvemålinger at tilgjengelig minne aldri falt under omtrent 310 MB av 464.

Implementeringen av sikkerhetskopiering og gjenopprettingstester bruker systemd-enheter for en databasedump der dette er konfigurert, et kryptert, deduplisert restic-øyeblikksbilde over SFTP til lagring hos en annen leverandør, opprydding etter oppbevaringsregelen og en integritetssjekk. En separat månedlig arbeidsflyt gjenoppretter en kjent fil og sammenligner bytene, og sikkerhetskopiering og gjenoppretting har separate signaler ved vellykket kjøring. Timerne avhenger av et konfigurert sikkerhetskopirepo; implementeringsdokumentasjonen datert 2026-10-27 sier at de var deaktivert uten et slikt repo, og dokumenterer ingen fullført gjenoppretting i drift. Arbeidet møtte også en av de skarpere kantene i systemd, beskrevet nedenfor: passfrasen ligger i en beskyttet fil som enhetene leser, fordi escape-sekvenser i en verdi sendt gjennom miljøet til en enhet blir behandlet, og en omvendt skråstrek kunne endre verdien uten varsel.

Den tredje er et prinsipp. Teknologisiden trekker den generelle grensen: vanlige systemd-enheter som standard, og Docker og Kubernetes for de arbeidslastene som gjør seg fortjent til dem.

Hvor det går galt

En endret enhet kjører videre med de gamle innstillingene. systemd leser enhetsfilene inn i minnet, så en endring på disk gjør ingenting før `systemctl daemon-reload` bygger styrerens bilde opp på nytt. Imens kan `systemctl cat` vise en fil som avviker fra den styreren bruker. Gjør endringer med `systemctl edit` , som laster inn på nytt for deg, eller last inn på nytt og start deretter tjenesten igjen.

En hemmelighet kommer fram til tjenesten med et tegn for lite. Verdier i `Environment=` går gjennom siteringsreglene i systemd, der escape-sekvenser i C-stil tolkes og en bokstavelig omvendt skråstrek må dobles. Dokumentasjonen til systemd advarer også om at miljøvariabler er synlige for klienter uten privilegier over D-Bus. Legg hemmeligheter i en fil og gi dem videre med `LoadCredential=` , som gir tjenesten en skrivebeskyttet kopi som bare tjenestens egen bruker og root kan lese.

En pakkeoppgradering tar stille bort en lokal rettelse. Endringer i leverandørens fil under `/usr/lib/systemd/system/` blir overskrevet neste gang pakkene oppdateres. Legg endringene i en drop-in under `/etc/systemd/system/<unit>.d/` , som flettes inn oppå og overlever oppgraderinger. Kommandolister legges sammen i stedet for å byttes ut, så en drop-in som endrer startkommandoen, må først tømme den med en tom tilordning:

```
# /etc/systemd/system/app.service.d/override.conf
[Service]
ExecStart=
ExecStart=/usr/local/bin/app --config /etc/app/app.toml
Restart=on-failure
```

En nattlig jobb hopper over natten serveren sto avslått. En kalendertimer som skulle ha gått mens maskinen var nede, venter rett og slett på neste tidspunkt, fordi `Persistent=` står på false som standard. Sett `Persistent=true` på sikkerhetskopiering og andre jobber som må kjøre, så går den tapte kjøringen når timeren er aktiv igjen, med en tilfeldig forsinkelse dersom en slik er konfigurert.

En tjeneste som krasjer, blir liggende nede. `Restart=` står på `no` som standard, så en tjeneste som dør, forblir død til noen merker det. Sett `Restart=on-failure` på tjenester som skal gå lenge. Vit også at når en enhet når startgrensen

sin, gir styreren den opp, og en manuell start må vente til intervallet er over eller til noen har kjørt `systemctl reset-failed` .

Les mer

- systemd(1) — selve styreren: enhetstypene, oppstartsforløpet og signalene den svarer på.
- systemd.unit(5) — enhetsfiler, avhengigheter, startgrenser og drop-in-katalogene som overstyrer dem.
- systemd.service(5) — tjenestetyper, omstartsregler og hvordan kommandolister nullstilles.
- systemd.exec(5) — kjøremiljøet, valgene for sandkasse og credentials, filene som bringer hemmeligheter inn i tjenesten.
- systemd.timer(5) — kalendertimere og monotone timere, med `Persistent=` .
- systemctl(1) — kommandolinjen for å starte, redigere, laste inn på nytt og undersøke enheter.

Ordliste

Velg ut fra deres faktiske rammer

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er systemd?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/systemd/