Engineer

systemd

Gestor que inicia el sistema y los servicios de Linux

Qué es

En casi todas las distribuciones Linux de uso mayoritario, el proceso con ID 1 es systemd: el gestor del sistema y de servicios que pone la máquina en marcha durante el arranque, inicia y supervisa sus servicios mientras funciona y detiene todo ordenadamente al apagarla. El proyecto escribe su nombre en minúsculas, nunca SystemD ni System D, y la d final es la antigua marca Unix de un daemon. Pertenece a la familia de sistemas init, junto a los scripts de SysV init y Upstart que sustituyó en la mayoría de las distribuciones, OpenRC y runit, que aún eligen algunas distribuciones pequeñas, y launchd, que hace el mismo trabajo en macOS.

Todo lo que gestiona systemd es una unidad, descrita en un breve archivo de texto con secciones y pares clave-valor. Su manual enumera 11 tipos: service, socket, target, device, mount, automount, timer, swap, path, slice y scope. Un servicio ejecuta un programa. Un temporizador inicia otra unidad según una programación. Un socket mantiene un puerto a la escucha e inicia su servicio cuando llega la primera conexión. Un target agrupa unidades en un estado que la máquina puede alcanzar, como un sistema multiusuario con red. Los requisitos y el orden son ajustes separados, por lo que «esto necesita aquello» y «esto se inicia después de aquello» son dos líneas distintas; systemd las convierte en una transacción de trabajos de inicio y parada que puede comprobar antes de ejecutarla.

Dos decisiones de diseño lo diferencian de los sistemas init anteriores basados en scripts de shell. Cada servicio se ejecuta en su propio grupo de control de Linux, de modo que el gestor sigue los procesos del grupo y puede detenerlos juntos, limitar su memoria y contabilizar su uso de CPU. Además, el proyecto es un conjunto de herramientas, no solo un gestor: su propia web enumera un daemon de registros, el journal, junto a componentes para los inicios de sesión, la

configuración de red, la sincronización horaria y la resolución de nombres. Las distribuciones eligen cuáles de esos componentes adicionales activan; la parte casi universal es el gestor con PID 1.

Utilice el concepto en una decisión

Aplique «¿Qué es systemd?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. El cuidado diario de servidores, correo, cuentas y actualizaciones para que los servicios de los que depende todo el mundo sigan funcionando sin sorpresas.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. 10 servidores PostgreSQL y MS SQL en Ubuntu — 10 servidores de bases de datos desplegados y mantenidos

10 servidores PostgreSQL y MS SQL en Ubuntu Póngase en contacto →

Por qué importa

En un servidor Linux rara vez se decide si ejecutar systemd, porque la distribución ya lo eligió. La decisión real es si utilizarlo como supervisor y planificador o superponerle otro gestor de procesos, la política de reinicio de un daemon de contenedores o cron.

Utilizarlo directamente reúne el orden de inicio, la política de reinicio, los registros, los límites de recursos y el aislamiento en un archivo por servicio, que lee el gestor que ya está en ejecución, sin un segundo proceso supervisor. Una unidad puede limitar la memoria con `MemoryMax=`, hacer que los directorios del propio sistema operativo sean de solo lectura para su servicio con `ProtectSystem=` y ejecutarse con un usuario temporal mediante `DynamicUser=`; `systemd-analyze security` puntúa la exposición de cada servicio en una escala de 0,0 a 10,0. Un trabajo iniciado por un temporizador, a diferencia de una línea de crontab, registra su actividad en el journal, utiliza el mismo aislamiento que cualquier servicio y muestra su última y próxima ejecución en `systemctl list-timers`. El conocimiento también se transfiere: un profesional de ingeniería que sabe leer un archivo de unidad puede operar la mayoría de los servidores Linux, sea cual sea su distribución.

Los costes también son reales. Las unidades solo funcionan en Linux, así que un equipo que también publique para macOS escribe listas de propiedades de launchd, y uno que publique para los BSD escribe scripts rc. Dentro de la mayoría de los contenedores no hay systemd: la propia aplicación es PID 1 y la política de reinicio corresponde a lo que ejecute el contenedor. Cuando contenedores y

servicios normales comparten servidor, Quadlet de Podman reduce esa distancia al convertir la descripción de un contenedor en una unidad de servicio corriente, de modo que ambos tipos de carga quedan bajo un mismo supervisor.

Cómo se aplica en la práctica

La plataforma propia de la empresa se apoya en systemd en tres puntos.

En el registro de implementación de septiembre de 2026, el servidor de producción descrito en la ejecución de los servicios públicos de la empresa en un servidor de 512 MB era una máquina de un solo núcleo con unos 464 MB de memoria utilizable. En ella residían un servidor web que gestionaba TLS para siete dominios, la forja git, el servicio onion, un servidor para Gemini y otro para Gopher y un sistema de bloqueo de intrusiones. La automatización de las copias de seguridad se implementa por separado; eso no demuestra que haya copias cifradas activas. Con ese tamaño, cada proceso residente tenía que justificar su memoria, por lo que systemd era el supervisor, sin un segundo proceso supervisor superpuesto, y los contenedores se ejecutaban como unidades Quadlet bajo el mismo gestor en lugar de tener un daemon de contenedores propio. Cuando se planteó si el servidor también podía alojar un servicio onion, las muestras registradas durante un día de medición mostraron que la memoria disponible nunca bajó de unos 310 MB de 464.

La implementación de copias de seguridad y pruebas de restauración utiliza unidades systemd para un volcado de base de datos donde esté configurado, una instantánea restic cifrada y deduplicada en almacenamiento de otro proveedor mediante SFTP, la poda por retención y una comprobación de integridad. Un flujo mensual separado restaura un archivo conocido y compara sus bytes; las copias y la restauración tienen señales de éxito separadas. Los temporizadores dependen de un repositorio configurado; el registro de implementación fechado el 2026-10-27 dice que estaban desactivados por no tenerlo y no registra ninguna restauración real completada. El trabajo también tropezó con una de las aristas de systemd, descrita más abajo: la frase de contraseña reside en un archivo protegido que leen las unidades, porque se procesan las secuencias de escape de un valor pasado por el entorno de la unidad, y una barra inversa podía alterarlo silenciosamente.

La página de tecnologías marca la pauta general: unidades systemd normales por defecto, reservando Docker y Kubernetes para las cargas que justifiquen su uso.

Dónde surgen los problemas

Una unidad editada sigue funcionando con la configuración anterior. systemd lee los archivos de unidad y los guarda en memoria, por lo que un cambio en disco no hace nada hasta que `systemctl daemon-reload` actualiza la información que el gestor mantiene en memoria; `systemctl cat` puede mostrar un archivo distinto de lo que utiliza el gestor en ejecución. Haga los cambios con `systemctl edit` , que recarga por usted, o recargue y después reinicie el servicio.

Un secreto llega al servicio con un carácter menos. Los valores de `Environment=` pasan por las reglas de comillas de systemd, que procesan secuencias de escape de estilo C y exigen duplicar una barra inversa literal. La documentación de systemd también advierte de que las variables de entorno son visibles para clientes sin privilegios a través de D-Bus. Guarde los secretos en un archivo y páselo con `LoadCredential=` , que entrega al servicio una copia de solo lectura que solo pueden leer su usuario y root.

Una actualización de paquete deshace silenciosamente una corrección local. Los cambios en el archivo del proveedor bajo `/usr/lib/systemd/system/` se sobrescriben en la siguiente actualización del paquete. Colóquelos en un archivo complementario bajo `/etc/systemd/system/<unit>.d/` , que se combina con prioridad sobre la configuración de base y sobrevive a las actualizaciones. Las listas de comandos se acumulan en lugar de sustituirse, por lo que un archivo complementario que cambie el comando de inicio primero lo vacía con una asignación sin valor:

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

Un trabajo nocturno se salta una ejecución cuando el servidor está apagado. Un temporizador de calendario que venció con la máquina apagada simplemente espera a su siguiente turno, porque `Persistent=` es falso por defecto. Configure `Persistent=true` para las copias de seguridad y otros trabajos que deban ejecutarse; la ejecución omitida se iniciará cuando el temporizador vuelva a estar activo, sujeta a cualquier retraso aleatorio configurado.

Un servicio que falla se queda detenido. `Restart=` vale `no` por defecto, así que un servicio que termina inesperadamente permanece detenido hasta que alguien lo advierte. Configure `Restart=on-failure` para los servicios de larga duración y tenga presente que, una vez que una unidad alcanza su límite de inicios, el gestor deja de intentarlo; un inicio manual debe esperar a que transcurra el intervalo o a que se ejecute `systemctl reset-failed`.

Lecturas adicionales

- systemd(1) — el propio gestor: tipos de unidad, secuencia de arranque y señales a las que responde.
- systemd.unit(5) — archivos de unidad, dependencias, límites de inicio y directorios de archivos complementarios que prevalecen sobre su configuración.
- systemd.service(5) — tipos de servicio, política de reinicio y cómo se restablecen las listas de comandos.
- systemd.exec(5) — entorno de ejecución, opciones de aislamiento y credenciales.
- systemd.timer(5) — temporizadores de calendario y monotónicos, incluido `Persistent=`.
- systemctl(1) — la línea de comandos para iniciar, editar, recargar e inspeccionar unidades.

Glosario

Elija según sus restricciones reales

Describa la ruta de despliegue, los riesgos operativos y quiénes se encargan de las guardias. Podemos definir el alcance de un cambio con comprobaciones, una vía de recuperación y responsabilidades claras.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es systemd?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

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

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/systemd/