Gestión del Sistema con Systemd

Sumario

Introducción

Systemd es un sistema de gestión del entorno de usuario de los sistemas GNU/Linux modernos. Aunque fue desarrollado originalmente por RedHat se ha ido imponiendo poco a poco, y no sin cierta controversia, entre las distribuciones GNU/Linux más conocidas.

Tiene múltiples ventajas respecto a sysvinit ya que permite gestionar de un modo mucho más simple y eficiente todos los aspectos relacionados con el arranque. También tiene sus detractores y ha recibido numerosas críticas por parte de miembros de la comunidad que consideran que systemd es una ruptura con la filosofía fundamental de Linux: "Haz algo concreto y hazlo bien".

Algunos administradores de sistemas han denominado a systemd como el "segundo kernel". Conocemos el kernel de Linux, es decir, el sistema operativo propiamente dicho, que ejecuta en modo privilegiado, con acceso total al hardware. El resto de procesos ejecutados en el sistema pertenecen al denominado "espacio de usuario", es decir el modo de ejecución no privilegiado, que requiere de la intermediación del sistema operativo para la asignación de recursos a esos procesos. Systemd toma el control de la mayoría de aspectos de gestión relacionados con los procesos de espacio de usuario. Por este motivo se le atribuye el nombre de "segundo kernel", en el sentido de abarca, algunos dicen que demasiados, los aspectos de gestión de los recursos de los procesos en espacio de usuario.

La filosofía clásica de Unix y Linux es "haz algo simple y hazlo bien". Esta filosofía parece claramente contrapuesta a los objetivos de diseño y funcionamiento de systemd, de ahí la controversia generada.

El concepto fundamental en systemd es el de **Unit**. Una Unit representa a un recurso administrable mediante systemd. La definición de Units se realiza mediante los **Unit Files**.

Systemd units

Con systemd se pueden gestionar muchos aspectos de configuración de un sistema, servicios, puntos de montaje, logs, estados del sistema, etc. La gestión del sistema que realiza systemd está basada en el concepto de unit (unidad). Una unit es un objeto que representa un recurso del sistema manejado por systemd. Los metadatos, o configuración, de las units se realiza mediante unit files, los cuales permiten la definición y administración de los aspectos de cada unit. Cada tipo de unit viene identificada por el sufijo de su unit file. Por ejemplo para la gestión de servicios se utiliza el sufijo .service

Uno de los aspectos fundamentales gestionados mediante units son los services (servicios) del sistema

Comandos de gestión básicos

Gestión de inicio y parada de la máquina

- Apagado de la máquina

```
systemctl poweroff
```

- Reinicio

```
systemctl reboot
```

- Reinicio en modo rescate

```
systemctl rescue
```

Gestión básica de units

El comando **systemctl** se usa para administrar las units. Permite gestionar muchos aspectos, entre los cuales podemos incluir los servicios. En los siguientes ejemplos se hará explícito el sufijo .service, que indica que se está trabajando con units de tipo servicio. Sin embargo systemd es lo suficientemente inteligente para que no sea necesario hacer explícto ese sufijo.

Veamos algunos comandos de gestión básica de units

- Conocer el estado de una unit

```
systemctl status nginx.service
```

Muestra el estado de la unit (de tipo service). En este caso el estado del servidor web nginx

- Indicar si el servicio correspondiente está activo

```
systemctl is-active nginx.service
```

- Indica si el servicio inicia durante el arranque

```
systemctl is-enabled nginx.service
```

- Comprueba si el servicio ha tenido algún problema al iniciar

```
systemctl is-failed nginx.service
```

- Listar las units del sistema

```
systemctl list-units
```

Mostraría todas las units del sistema activas. Con el modificador --all mostraría también aquellas no activas

```
systemctl list-units-files
```

Mostraría toda la información de units instaladas en el sistema, a partir de sus unit files, incluso aquellas que no están cargadas en memoria

- Mostrar todos los detalles de una unit

```
systemctl show nginx.service
```

- Ver los detalles de unit file

```
systemctl cat nginx.service
```

- Mostrar las dependencias de units

```
systemctl list-dependencies –all nginx.service
```

Mostraría todas las dependencias de la unit de tipo service nginx.service. Con el modificador --all se usa de modo recursivo, es decir muestra a su vez las dependencias de las dependencias

Gestión de Unit Files

Las units como ya vimos son el elemento fundamental de gestión de recursos en systemd. Cualquier aspecto administrado con systemd tendrá una unit asociada. Existen varios tipos de units:

- **service**: gestión de servicios
- **socket**: gestión de los sockets utilizados por otras units (como servicios)
- **device**: gestión de dispositivos
- **mount y automount**: gestión de puntos de montaje
- **swap**: gestión de espacios de intercambio
- **target**: gestión de targets
- **path**: utilizada para activación de units .service basada en eventos del sistema de archivos referenciado por el path
- **timer**: gestión de eventos, al estilo cron o at
- **snapshot**: gestión de instantáneas del sistema
- **slice**: gestión de recursos asociados a procesos usando cgroup
- **scope**: gestión de procesos organizados y creados por systemd

Como puede verse pueden gestionarse muchos aspectos del sistema con systemd. Para definir el comportamiento y definición de cada uno de los recursos gestionados utilizamos los **unit files**, los cuales contienen directivas asociadas al tipo de unit. Los unit file tienen un nombre del tipo nombre_unit.tipo_unit, por ejemplo apache.service o multi-user.target.

La sintaxis de definición de los unit files está basada en **secciones**, cada sección indica un aspecto, en el que se define, a través de **directivas**, todos los parámetros asociados a ese aspecto. Algunas secciones de unit files

- [**Unit**]: Define directivas específicas de la definición de la unit
- [**Install**]: Define directivas de gestión de units estableciendo relaciones en aspectos como, por ejemplo, la relación de la unit con targets asociados
- [**Service**]: Define directivas específicas de un service
- [**Socket**]: Define directivas de configuración de sockets asociados a services
- [**Mount**] y [**Automount**]: Define directivas para puntos de montaje
- [**Swap**]: Directivas que definen y habilitan espacios de intercambio para páginas de memoria virtual anónimas
- [**Timer**]: Definición y gestión de eventos temporales
- [**Path**]: Monitorización del sistema de archivos
- [**Slice**]: Gestión de asignación de recursos a los procesos

Ejemplo de Unit file

```
[Unit]
Description=MyApp
After=docker.service
Requires=docker.service

[Service]
TimeoutStartSec=0
ExecStartPre=-/usr/bin/docker kill busybox1
ExecStartPre=-/usr/bin/docker rm busybox1
ExecStartPre=/usr/bin/docker pull busybox
ExecStart=/usr/bin/docker run --name busybox1 busybox /bin/sh -c "while true; do echo Hello World; sleep 1; done"

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

El unit file anterior define una unit de tipo service.

Podemos observar como algunas secciones son **genéricas**, como [**Unit**], que identifica aspectos generales de la unit, o [**Install**], que identifica, entre otras cosas, en que target se activa la unit. Otras secciones son específicas del tipo de unit concreto, como en el caso anterior [**Service**], en la que se definen los parámetros que afectan al servicio.

Edición de unit files

Para editar un unit file, es decir los metadatos de definición de una unit, podemos utilizar el comando systemctl edit.

Veamos un ejemplo:

```
systemctl edit nginx.service
```

Permite añadir y sobreescribir configuraciones del servicio nginx. Con la opción --full se permite modificar el contenido completo del unit file.

Tras modificar el unit file es necesario recargar el servicio

```
systemctl daemon-reload
```

Cuando editamos del modo anterior un unit file se está creando un directorio en **/etc/systemd/system** con nombre **nombre_unit.d**, en el caso anterior nginx.service.d. Dentro de ese directorio se ubicará un snippet con nombre **override.conf** que aplicará cambios en la definición de la unit file original, de modo que las modificaciones del snippet tengan preferencia.

Para editar la unit file completa podemos usar:

```
systemctl edit --full nginx.service
```

De este modo se creará un archivo en **/etc/systemd/sytem** con el nombre de la unit, copia modificada del unit original en **/lib/systemd/system** (ubicación original de los units cuando se instala el software) y que tomará preferencia sobre éste.

Para eliminar los cambios introducidos en override.conf:

```
rm -r /etc/systemd/system/nginx.service.d
```

Para eliminar la copia exacta del unit file creado con la opción --full de edit:

```
rm /etc/systemd/system/nginx.service
```

Tras borrar es necesario recargar el proceso systemd para que tome las copias de los unit files correctas

```
systemctl daemon-reload
```

Procedimientos de administración con systemd

Gestión de targets (runlevels)

Gestión de Servicios con systemd

Gestión de Puntos de Montaje

Gestión de timers

Gestión de logs

Monitorización del sistema de archivos con systemd

Gestión de recursos asignados a procesos

Referencias

Documentación man systemd

https://www.freedesktop.org/software/systemd/man/index.html

Volver

JavierFP 17:44 11 dec 2017 (CET)