Gestión de timers

Sumario

Introducción

Un aspecto fundamental a la hora de administrar un sistema GNU/Linux es la posibilidad de definir tareas programadas. Tradicionalmente para este cometido se ha usado la herramienta cron, la cual permite, de un modo simple y rápido, definir tareas periódicas.

Systemd incorpora la posibilidad de gestionar este aspecto mediante los Unit de tipo timer. Como es habitual definiremos su correspondiente Unit File para definir el comportamiento del mismo.

Trabajando con timers

A continuación mostraremos, con un ejemplo, la gestión básica de timers con systemd. El ejemplo consistirá en los siguientes pasos

- Definiremos un **unit file de tipo timer, ejemplo-timer.timer,** que indicará un aspecto temporal en el que se llevará a cabo una acción en el sistema. El timer definido será incluído en el target **basic.target**, predefinido en systemd. Además, es necesario vincular el timer con la acción que se ejecutará por cada expiración del mismo. Para ello necesitaremos un unit de tipo target
- Definiremos el **unit de tipo target, ejemplo-timer-target,** al que se hace referencia arriba. Este target será necesario para vincular el timer con la acción definida a través de un unit de tipo service.
- Definiremos el unit de tipo service, **ejemplo-timer-service**, que definirá la acción ejecutada cada vez que se verifique la condición temporal definida en el timer.

A continuación se muestra la relación entre los elementos que será necesario definir para habilitar el timer

Definiendo el timer

Para definir uno de estos objetos usaremos un unit file

Creamos el unit file **/etc/systemd/system/ejemplo-timer.timer**

```
[Unit]
Description=Timer de Prueba
[Timer]
OnBootSec=5min
OnCalendar=*:0/1
Unit=ejemplo-timer.target
[Install]
WantedBy=basic.target
```

Dentro de la sección **Timer** es donde se definen los parámetros específicos

- **OnBootSec**: tiempo hasta la activación después de iniciar el sistema
- **OnCalendar**: define el instante temporal de activación del timer
- **Unit**: Unit que se notifica por cada expiración del timer, en este caso una unit de tipo target que definiremos a continuación

La sección **Install** establece parámetros a la hora de instalar o desplegar el timer **WantedBy=basic.target** indica el target al que irá asociado el timer. Este target está predefinido y, como su nombre indica, actúa como un contenedor de units a procesar en un perfil de ejecución básico. Por tanto, cuando se procese el target correspondiente, basic.target, se activará el timer.

Definiendo un target

Definimos el unit file para el target **/etc/systemd/system/ejemplo-timer.target**

```
[Unit]
Description=Ejemplo Timer Target
StopWhenUnneeded=yes
```

Definiendo un service

Por último necesitamos una acción que se disparará asociada a los eventos de expiración del timer. Para ello crearemos un unit de tipo service a través del siguiente unit file **/etc/systemd/system/ejemplo-timer.service**

```
[Unit]
Description=Escribe la fecha/hora cada minuto
Wants=ejemplo-timer.timer
[Service]
ExecStart=/bin/date
[Install]
WantedBy=ejemplo-timer.target
```

Vemos como en la sección principal Unit se define la directiva

- **Wants=ejemplo-timer.timer**

que indica que esta unit dependerá del timer definido anteriormente, es decir, deberá ser procesado por systemd el unit correspondiente al timer para habilitar el service.

En la sección **Service** vemos la acción, en este caso invocación al comando date, que imprimirá la fecha y hora en los archivos de log del journal

Por último en la sección Install **WantedBy=ejemplo-timer.target** indica el target al que va asociado el service. De este modo conectamos el unit de tipo timer, definido al principio, con el service a través del unit de tipo target definido en el apartado anterior.

Lanzamiento del timer

A continuación ejecutamos los comando correspondientes para habilitar y activar el timer

```
systemctl enable ejemplo-timer.timer
systemctl start ejemplo-timer.timer
systemctl enable ejemplo-timer.service
```

Los 3 comandos anteriores:

- Habilita el timer para que se inicie en cada inicio del sistema
- Inicia el timer
- Habilita el service para que se inicie en cada inicio del sistema

Para ver el resultado de la activación de la lógica del timer consultamos los logs con el comando

```
journalctl -f -u ejemplo-timer.service
```

La salida mostrada en el log

```
nov 27 18:20:08 base date[1077]: lun nov 27 18:20:08 CET 2017
nov 27 18:21:08 base systemd[1]: Started Escribe la fecha/hora cada minuto.
nov 27 18:21:08 base date[1079]: lun nov 27 18:21:08 CET 2017
nov 27 18:22:08 base systemd[1]: Started Escribe la fecha/hora cada minuto.
nov 27 18:22:08 base date[1082]: lun nov 27 18:22:08 CET 2017
nov 27 18:23:08 base systemd[1]: Started Escribe la fecha/hora cada minuto.
```

NOTA

Si bien en el ejemplo anterior hacemos uso de 3 units: timer, target y service; el uso del target no sería estrictamente necesario. Con el target podemos

tomar control agrupando servicios bajo un mismo aspecto, sin embargo el ejemplo anterior funcionaría perfectamente definiendo timer y service del siguien modo:

- El unit file del timer **/etc/systemd/system/ejemplo-timer.timer**

```
[Unit]
Description=Timer de Prueba
[Timer]
OnBootSec=5min
OnCalendar=*:0/1
Unit=ejemplo-timer.service
[Install]
WantedBy=basic.target
```

- El unit file del service **/etc/systemd/system/ejemplo-timer.service**

```
[Unit]
Description=Escribe la fecha/hora cada minuto
Wants=ejemplo-timer.timer
[Service]
ExecStart=/bin/date
[Install]
WantedBy=basic.target
```

Conclusión

A priori no parece que se haya simplificado demasiado la gestión de este aspecto del sistema con systemd. El uso de la herramienta cron sigue siendo muy recomendable, dada su sencillez y su extendida utilización

Volver

JavierFP 17:44 11 dec 2017 (CET)