CONTROL DE SERVICIOS Y DEMONIOS

RH124

Capitulo 8

Descripción general	
Meta	Controlar y monitorear servicios de red y demonios del sistema con systemd.
Objetivos	• Enumerar los demonios del sistema y los servicios de red iniciados por el servicio systemd y las unidades socket. • Controlar los demonios del sistema y los servicios de red con `systemctl`.
Secciones	• Identificación de procesos del sistema comenzados en forma automática (y práctica) • Control de servicios del sistema (y práctica)
Trabajo de laboratorio	• Control de servicios y demonios

Identificación de procesos del sistema comenzados en forma automática

RH124

Objetivos

Tras finalizar esta sección, los estudiantes deberían poder enumerar los demonios del sistema y los servicios de red iniciados por el servicio **`systemd`** y las unidades socket.

Introducción a `systemd`

El arranque del sistema y los procesos del servidor son administrados por *el sistema systemd y el administrador del servicio*. Este programa proporciona un método para activar los recursos del sistema, los demonios del servidor y otros procesos, tanto en el momento del arranque como en un sistema que está en funcionamiento.

Los demonios son procesos que esperan o se ejecutan en segundo plano y realizan varias tareas. Generalmente, los demonios se inician automáticamente en el momento del arranque y continúan ejecutándose hasta que se apaga el sistema o son detenidos manualmente. Por convención, los nombres de muchos programas demonios finalizan con la letra "d".

Para estar atento a las conexiones, un demonio usa un *socket*. Este es el canal de comunicación principal con los clientes locales o remotos. Los sockets pueden ser creados por los demonios o pueden ser separados del demonio y ser creados por otro proceso, como systemd. El socket pasa al demonio cuando el cliente establece una conexión.

Un poco de historia

Durante muchos años, la ID 1 de proceso de los sistemas Linux y UNIX ha sido el proceso **`init`**. Este proceso era responsable de activar otros servicios en el sistema y es el origen del término "init system". Los demonios usados con más frecuencia se iniciaban en los sistemas en el momento del arranque con las secuencias de comandos init *System V* y *LSB* . Estas son secuencias de comandos de la shell y pueden variar de una distribución a otra. Los demonios usados con menos frecuencia se iniciaban a pedido por otro servicio, como **`initd`** o **`xinetd`**, que escucha las conexiones del cliente. Estos sistemas tienen muchas limitaciones, que son resueltas con systemd.

En Red Hat Enterprise Linux 7, la ID 1 de proceso es **`systemd`**, que es el sistema init nuevo. Algunas de las funciones nuevas que proporciona systemd son:

- Capacidades de paralelización, que aumentan la velocidad de arranque de un sistema.
- Inicio a pedido de los demonios sin necesidad de otro servicio.
- Administración de dependencia del servicio automática, que puede prevenir los tiempos de inactividad prolongados, como evitar que se inicie un servicio de red cuando la red no está disponible.
- Método para realizar el seguimiento de los procesos relacionados en forma conjunta con el uso de los grupos de control de Linux.

nota

Con systemd, se usan las secuencias de comandos del servicio basado en la shell solo para algunos servicios heredados. Por lo tanto, se reemplazan los archivos de configuración con las variables de la shell, como aquellos que se encuentran en **`/etc/sysconfig`**. Aquellos que todavía están en uso están incluidos como archivos del entorno systemd y se leen como pares NOMBRE=VALOR. Ya no se proporcionan como una secuencia de comandos de la shell.

Unidades `systemctl` y `systemd`

El comando **`systemctl`** se usa para administrar diferentes tipos de objetos de systemd denominados *unidades*. Con **`systemctl -t help`** puede mostrarse una lista de los tipos de unidades disponibles.

Importante

El **`systemctl`** puede abreviar u "omitir" los nombres de unidad, las entradas de árbol de proceso y las descripciones de unidad, a menos que se ejecute con la opción **`-l`**.

A continuación, se enumeran algunos de los tipos de unidades más usados:

- Las unidades de servicio tienen la extensión .service y representan servicios del sistema. Este tipo de unidad se usa para iniciar los demonios usados con más frecuencia, como un servidor web.

- Las unidades socket tienen la extensión .socket y representan sockets de comunicación entre procesos (IPC). El control del socket pasará a un demonio o servicio iniciado recientemente cuando se realice una conexión de cliente. Las unidades de socket se usan para demorar el inicio de un servicio en el momento del arranque y para iniciar servicios usados con menos frecuencia a pedido. En principio, son similares a los servicios que usan el superservidor **`xinetd`** para iniciar a pedido.

- Las unidades de ruta tienen la extensión .path y se usan para demorar la activación de un servicio hasta que ocurra un cambio en el sistema de archivos específico. Esto se usa con más frecuencia en servicios que utilizan directorios de cola, como los sistemas de impresión.

Estados de servicio

El estado de un servicio puede visualizarse con **`systemctl status` *`name.type`***. Si no se proporciona el tipo de unidad, **`systemctl`** mostrará el estado de la unidad de servicio, en caso de que exista una.

```
[root@serverX ~]# systemctl status sshd.service
sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled)
   Active: active (running) since Thu 2014-02-27 11:51:39 EST; 7h ago
 Main PID: 1073 (sshd)
   CGroup: /system.slice/sshd.service
           └─1073 /usr/sbin/sshd -D

Feb 27 11:51:39 server0.example.com systemd[1]: Started OpenSSH server daemon.
Feb 27 11:51:39 server0.example.com sshd[1073]: Could not load host key: /et...y
Feb 27 11:51:39 server0.example.com sshd[1073]: Server listening on 0.0.0.0 ....
Feb 27 11:51:39 server0.example.com sshd[1073]: Server listening on :: port 22.
Feb 27 11:53:21 server0.example.com sshd[1270]: error: Could not load host k...y
Feb 27 11:53:22 server0.example.com sshd[1270]: Accepted password for root f...2
Hint: Some lines were ellipsized, use -l to show in full.
```

En el resultado del estado, se pueden encontrar varias palabras clave que indican el estado del servicio:

Palabra clave:	Descripción:
loaded (cargado)	Se procesó el archivo de configuración de la unidad.
active (activo); en ejecución	En ejecución con uno o más procesos en curso.
active (activo); cerrado	Se completó correctamente la configuración de una sola vez.
active (activo); en espera	En ejecución, pero a la espera de un evento.
inactive (inactivo)	Detenido.
habilitado	Se iniciará en el momento del arranque.
deshabilitado	No se iniciará en el momento del arranque.
estático	No puede habilitarse, pero puede iniciarse por una unidad habilitada en forma automática.

Enumeración de los archivos de unidad con `systemctl`

En este ejemplo, continúe con los próximos pasos mientras el instructor realiza una demostración sobre cómo obtener la información de estado de los servicios.

nota

Observe que el comando **`systemctl`** paginará automáticamente el resultado con **`less`**.

1. Consulte el estado de todas las unidades para verificar el arranque del sistema.

```
[root@serverX ~]# systemctl
```

2. Consulte el estado solo de las unidades de servicio.

```
[root@serverX ~]# systemctl --type=service
```

3. Investigue alguna unidad que tenga el estado de falla o mantenimiento. Otra alternativa es agregar la opción **-l** para mostrar el resultado completo.

```
[root@serverX ~]# systemctl status rngd.service -l
```

4. El argumento **status** también puede usarse para determinar si una unidad en particular está activa y mostrar si la unidad está habilitada para iniciarse en el momento del arranque. Los comandos alternativos también pueden mostrar con facilidad los estados activo y habilitado:

```
[root@serverX ~]# systemctl is-active sshd
[root@serverX ~]# systemctl is-enabled sshd
```

5. Enumere el estado activo de todas las unidades cargadas. Otra opción es limitar el tipo de unidad. La opción **--all** agregará unidades inactivas.

```
[root@serverX ~]# systemctl list-units --type=service
[root@serverX ~]# systemctl list-units --type=service --all
```

6. Visualice los parámetros de configuración de habilitado e inhabilitado para todas las unidades. Otra opción es limitar el tipo de unidad.

```
[root@serverX ~]# systemctl list-unit-files --type=service
```

7. Visualice solo los servicios con fallas.

```
[root@serverX ~]# systemctl --failed --type=service
```

Referencias

Páginas del manual **systemd**(1), **systemd.unit**(5), **systemd.service**(5), **systemd.socket**(5) y **systemctl**(1)

Es posible encontrar información adicional en el capítulo sobre la administración de servicios con **systemd** en la *Guía del administrador del sistema Red Hat Enterprise Linux* para Red Hat Enterprise Linux 7, que se puede encontrar en

> https://access.redhat.com/documentation/

Práctica: Identificar el estado de unidades systemd

Control de servicios del sistema

RH124

Objetivos

Tras finalizar esta sección, los estudiantes deberían poder controlar los demonios del sistema y los servicios de red con **`systemctl`**.

Iniciar y detener demonios del sistema en un sistema en funcionamiento.

Los cambios realizados en un archivo de configuración u otros tipos de actualizaciones de servicio posiblemente requieran el reinicio del servicio. Un servicio que ya no se utiliza puede detenerse antes de quitar el software. Un servicio que no se utilice frecuentemente puede ser iniciado manualmente por un administrador solo cuando sea necesario.

En este ejemplo, realice los siguientes pasos mientras el instructor realiza una demostración de cómo administrar servicios en un sistema en funcionamiento.

1. Vea el estado de un servicio.

```
[root@serverX ~]# systemctl status sshd.service
```

2. Verifique que el proceso esté en funcionamiento.

```
[root@serverX ~]# ps -up PID
```

3. Detenga el servicio y verifique el estado.

```
[root@serverX ~]# systemctl stop sshd.service
[root@serverX ~]# systemctl status sshd.service
```

4. Inicie el servicio y vea el estado. La ID del proceso ha cambiado.

```
[root@serverX ~]# systemctl start sshd.service
[root@serverX ~]# systemctl status sshd.service
```

5. Detenga y, luego, inicie el servicio con un solo comando.

```
[root@serverX ~]# systemctl restart sshd.service
[root@serverX ~]# systemctl status sshd.service
```

6. Emita instrucciones para que un servicio lea y vuelva a cargar su archivo de configuración sin que se detenga completamente y se inicie. La ID del proceso no cambiará.

```
[root@serverX ~]# systemctl reload sshd.service
[root@serverX ~]# systemctl status sshd.service
```

Dependencias de unidades

Los servicios pueden iniciarse como dependencias de otros servicios. Si una unidad de socket está habilitada y la unidad de servicio con el mismo nombre no lo está, el servicio se iniciará automáticamente cuando se realice una solicitud en el socket de red. Los servicios también pueden ser activados por unidades de ruta cuando se cumple una condición del sistema de archivos. Por ejemplo, un archivo colocado en el directorio de colas de impresión hará que el servicio de impresión **cups** se inicie si no está funcionando.

```
[root@serverX ~]# systemctl stop cups.service
Warning: Stopping cups, but it can still be activated by:
  cups.path
  cups.socket
```

Para detener completamente los servicios de impresión en un sistema, detenga las tres unidades. Al deshabilitar el servicio, se deshabilitarán las dependencias.

El comando **`systemctl list-dependencies`** ***`UNIT`*** puede utilizarse para imprimir un árbol de las otras unidades que deben iniciarse si se inicia la unidad especificada. Según la dependencia exacta, la otra unidad posiblemente deba estar funcionando antes o después de que se inicia la unidad especificada. La opción **`--reverse`** de este comando mostrará las unidades que deben tener la unidad especificada iniciada para ejecutarse.

Enmascaramiento de servicios

En ocasiones, es posible que en un sistema haya servicios en conflicto instalados. Por ejemplo, hay múltiples métodos para administrar redes (red y NetworkManager) y firewalls (iptables y firewalld). A fin de evitar que un administrador inicie un servicio por error, existe la opción de *enmascarar*ese servicio. El enmascaramiento creará un enlace en los directorios de configuración de modo que nada ocurra en caso de que se inicie el servicio.

```
[root@serverX ~]# systemctl mask network
ln -s '/dev/null' '/etc/systemd/system/network.service'
[root@serverX ~]# systemctl unmask network
rm '/etc/systemd/system/network.service'
```

Importante

Un servicio deshabilitado no se iniciará automáticamente en el arranque ni a través de otros archivos de unidad, pero puede iniciarse manualmente. Un servicio enmarcarado no puede iniciarse de manera manual ni automática.

Habilitación de demonios del sistema para que se inicien o detengan durante el arranque

El inicio de un servicio en un sistema en funcionamiento no garantiza el inicio del servicio cuando se vuelva a arrancar el sistema. De manera similar, el detenimiento de un servicio en un sistema en funcionamiento no evitará que se reinicie cuando se vuelva a arrancar el sistema. Los servicios se inician durante el proceso de arranque cuando se crean enlaces en los directorios de configuración **`systemd`** correspondientes. Dichos vínculos se crean y quitan con comandos **`systemctl`**.

En este ejemplo, realice los siguientes pasos mientras el instructor realiza una demostración sobre cómo habilitar y deshabilitar los servicios.

1. Vea el estado de un servicio.

```
[root@serverX ~]# systemctl status sshd.service
```

2. Deshabilite el servicio y verifique el estado. Tenga en cuenta que la deshabilitación de un servicio no detiene el servicio.

```
[root@serverX ~]# systemctl disable sshd.service
[root@serverX ~]# systemctl status sshd.service
```

3. Habilite el servicio y verifique el estado.

```
[root@serverX ~]# systemctl enable sshd.service
[root@serverX ~]# systemctl is-enabled sshd.service
```

Resumen de los comandos `systemctl`

Los servicios pueden iniciarse y detenerse en un sistema en funcionamiento, y habilitarse o deshabilitarse para que se inicien automáticamente durante el proceso de arranque.

Tarea:	Comando:
Ver información detallada sobre el estado de una unidad.	`systemctl status UNIT`
Detener un servicio en un sistema en funcionamiento.	`systemctl stop UNIT`
Iniciar un servicio en un sistema en funcionamiento.	`systemctl start UNIT`
Reiniciar un servicio en un sistema en funcionamiento.	`systemctl restart UNIT`
Volver a cargar el archivo de configuración de un servicio en ejecución.	`systemctl reload UNIT`
Deshabilitar completamente el inicio (tanto manual como durante el proceso de arranque) de un servicio.	`systemctl mask UNIT`
Poner un servicio enmascarado a disposición.	`systemctl unmask UNIT`
Configurar un servicio para que se inicie durante el proceso de arranque.	`systemctl enable UNIT`
Deshabilitar el inicio de un servicio durante el proceso de arranque.	`systemctl disable UNIT`
Enumerar unidades necesarias y deseadas por la unidad especificada.	`systemctl list-dependencies UNIT`

Referencias

Páginas del manual **systemd**(1), **systemd.unit**(5), **systemd.service**(5), **systemd.socket**(5) y **systemctl**(1)

Es posible encontrar información adicional en el capítulo sobre la administración de servicios con **systemd** en la *Guía del administrador del sistema Red Hat Enterprise Linux* para Red Hat Enterprise Linux 7, que se puede encontrar en

> https://access.redhat.com/documentation/

Práctica: Uso de systemctl para administrar servicios