ANÁLISIS Y ALMACENAMIENTO DE REGISTROS

RH124

Capitulo 10

Visión general:	
Meta	Ubicar e interpretar correctamente archivos de registro del sistema relevantes para la solución de problemas.
Objetivos	• Describir la arquitectura básica syslog en Red Hat Enterprise Linux 7. • Interpretar entradas en archivos syslog relevantes para la solución de problemas o revisar el estado del sistema. • Buscar e interpretar entradas en el journal de systemd para solucionar problemas o revisar el estado del sistema. • Configurar systemd-journald para almacenar el diario en disco en lugar de almacenarlo en memoria. • Mantener una sincronización de tiempos y configuración de zona horaria precisas para garantizar sellos de tiempo correctos en los registros del sistema.
Secciones	• Arquitectura de registro de sistema (y práctica) • Revisión de archivos Syslog (y práctica) • Revisión de entradas del Journal de systemd (y práctica) • Conservación del Journal de systemd (y práctica) • Mantenimiento de tiempo exacto (y práctica)
Trabajo de laboratorio	• Análisis y almacenamiento de registros

Arquitectura de registro del sistema

RH124

Objetivos

Tras finalizar esta sección, los estudiantes deberían poder describir la arquitectura básica de syslog en Red Hat Enterprise Linux 7.

Inicio de sesión del sistema

Los procesos y el kernel del sistema operativo deben poder llevar un registro de los eventos que suceden. Estos registros pueden ser útiles para realizar una auditoría del sistema y solucionar problemas. Por convención, se almacenan de forma persistente en el directorio **`/var/log`**.

Red Hat Enterprise Linux incluye un sistema de registro estándar que se basa en el protocolo Syslog. Muchos programas utilizan este sistema para registrar eventos y organizarlos en archivos de registro. En Red Hat Enterprise Linux 7, hay dos servicios que se encargan de los mensajes de syslog: **`systemd-journald`** y **`rsyslog`**.

El demonio **`systemd-journald`** proporciona un servicio de administración de registros mejorado que recopila mensajes del kernel, las primeras etapas del proceso de arranque, la salida estándar y los errores de demonios a medida que se inician y ejecutan, y syslog. Escribe estos mensajes en un diario estructurado de eventos que, de manera predeterminada, no se conserva entre un reinicio y otro. Esto permite recopilar en una base de datos central los mensajes de syslog y los eventos que syslog omite. Los mensajes de syslog son reenviados de **`systemd-journald`** a **`rsyslog`** para su posterior procesamiento.

El servicio **`rsyslog`** luego ordena los mensajes de syslog por tipo (o utilidad) y prioridad, y los escribe en archivos persistentes en el directorio **`/var/log`**.

Generalidades de los archivos de registro del sistema

Archivo de registro	Propósito
`/var/log/messages`	La mayoría de los mensajes de syslog se registran aquí. Las excepciones son mensajes relacionados con tareas de autenticación y procesamiento de correos electrónicos, que realizan periódicamente trabajos, y aquellos relacionados exclusivamente con tareas de depuración.
`/var/log/secure`	El archivo de registro para errores y mensajes relacionados con seguridad y autenticación.
`/var/log/maillog`	El archivo de registro con mensajes relacionados con el servidor de correo.
`/var/log/cron`	El archivo de registro relacionado con tareas ejecutadas en forma periódica.
`/var/log/boot.log`	Los mensajes relacionados con el arranque del sistema se registran aquí.

Referencias

Páginas del manual: **systemd-journald.service**(8), **rsyslogd**(8), **rsyslog.conf**(5)

Es posible encontrar información adicional 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: Componentes de registro de sistema

Revisión de archivos Syslog

RH124

Objetivos

Tras finalizar esta sección, los estudiantes deberían poder interpretar las entradas en los archivos syslog correspondientes para solucionar problemas o revisar el estado del sistema.

Archivos syslog

Muchos programas usan el protocolo *syslog* para registrar eventos en el sistema. Cada mensaje se clasifica por instalación (tipo de mensaje) y prioridad (gravedad del mensaje). Las instalaciones disponibles se documentan en la página del manual **`rsyslog.conf`**(5).

Las ocho prioridades también se estandarizan y clasifican de la siguiente manera:

Descripción general de las prioridades de syslog

Código	Prioridad	Gravedad
0	emerg	El sistema no se puede usar.
1	alert	Se debe implementar una acción de inmediato.
2	crit	Condición crítica.
3	err	Condición de error no crítica.
4	warning	Condición de advertencia.
5	notice	Evento normal pero importante.
6	info	Evento informativo.
7	debug	Mensaje de nivel de depuración.

El servicio rsyslogd usa la instalación y la prioridad de los mensajes de registro para determinar cómo resolverlos. Esto se configura mediante el archivo **`/etc/rsyslog.conf`** y los archivos **`*.conf`** en **`/etc/rsyslog.d`**. Los programas y los administradores pueden cambiar la configuración de **`rsyslogd`**, de tal manera que no pueda sobrescribirse con las actualizaciones de **`rsyslog`** mediante la inclusión de archivos personalizados que tienen el sufijo .conf en el directorio **`/etc/rsyslog.d`**.

En la sección **`#### RULES ####`** de **`/etc/rsyslog.conf`**, se incluyen directivas que definen dónde se almacenan los mensajes de registro. En el lado izquierdo de cada línea, se indican la instalación y la gravedad del mensaje de registro que se corresponde con la directiva.
El archivo rsyslog.conf puede contener el carácter * como comodín en los campos de instalación y gravedad, donde es válido para todas las instalaciones o todas las gravedades.
En el lado derecho de cada línea, se indica en qué archivo se debe guardar el mensaje de registro. Generalmente, los mensajes de registro se guardan en archivos ubicados en el directorio **`/var/log`**.

nota

Los archivos de registro se conservan mediante el servicio **`rsyslog`**, y el directorio **`/var/log`** contiene una variedad de archivos de registro específicos para determinados servicios. Por ejemplo, el servidor web Apache o Samba generan sus propios archivos de registro en el subdirectorio correspondiente del directorio **`/var/log`**.

Un mensaje manejado por **`rsyslog`** puede aparecer en varios archivos de registro diferentes. Para evitar eso, el campo de gravedad puede configurarse como **`none`**, lo que significa que ninguno de los mensajes dirigidos hacia esta instalación se agregan al archivo de registro especificado.

En lugar de registrar mensajes de syslog en un archivo, pueden imprimirse en las terminales de todos los usuarios que hayan iniciado sesión. En el archivo **`rsyslog.conf`** predeterminado, esto se hace para todos los mensajes que tienen la prioridad "emerg".

Sección de reglas de muestra de rsyslog.conf

```
#### RULES ####

# Log all kernel messages to the console.
# Logging much else clutters up the screen.
#kern.*                                                 /dev/console

# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none                /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/secure

# Log all the mail messages in one place.
mail.*                                                  -/var/log/maillog


# Log cron stuff
cron.*                                                  /var/log/cron

# Everybody gets emergency messages
*.emerg                                                 :omusrmsg:*

# Save news errors of level crit and higher in a special file.
uucp,news.crit                                          /var/log/spooler

# Save boot messages also to boot.log
local7.*                                                /var/log/boot.log
```

nota

El archivo **`rsyslog.conf`** está documentado en la página del manual **`rsyslog.conf`**(5) y en la amplia documentación HTML de **`/usr/share/doc/rsyslog-*/manual.html`** que está en el *rsyslog-doc*, que está disponible en el canal de software de Red Hat Enterprise Linux 7, pero no está incluido en el medio de instalación.

Rotación del archivo de registro

Los registros se "rotan" mediante la utilidad **`logrotate`** para evitar que llenen el sistema de archivos que contiene **`/var/log/`**. Cuando se rota un archivo de registro, se le cambia el nombre con una extensión que indica la fecha en que se rotó: el archivo **`/var/log/messages`** anterior puede pasar a ser **`/var/log/messages-20141030`** si se rota el 30 de octubre de 2014. Una vez que se rotó el archivo de registro anterior, se crea un nuevo archivo de registro y se notifica al servicio que escribe en este.

Después de una determinada cantidad de rotaciones, habitualmente después de cuatro semanas, el archivo de registro anterior se descarta para liberar espacio en disco. Una tarea de cron ejecuta el programa de rotación de archivos de registros a diario para verificar si es necesario rotar algún registro. La mayoría de los archivos de registro se rotan semanalmente, pero el programa de rotación de archivos de registros rota algunos más rápido o más lento, o cuando alcanzan un tamaño determinado.

La configuración de logrotate no se aborda en este curso. Si desea obtener más información, consulte la página del manual **`logrotate`**(8).

Análisis de una entrada de syslog

Los registros del sistema escritos por **rsyslog** comienzan con el mensaje más antiguo en la parte superior y el mensaje más nuevo al final del archivo de registro. Todas las entradas en los archivos de registro administrados por **rsyslog** se graban en formato estándar. El siguiente ejemplo explicará la anatomía de un mensaje de archivo de registro en el archivo de registro **/var/log/secure**:

```
1Feb 11 20:11:48 2localhost 3sshd[1433]: 4Failed password for student from
 172.25.0.10 port 59344 ssh2
```

1. La marca de tiempo cuando se grabó la entrada de registro.
2. El host desde donde se envió el mensaje de registro.
3. El programa o el proceso que envió el mensaje de registro.
4. El mensaje real enviado.

Monitoreo de un archivo de registro con `tail`

Para reproducir problemas e inconvenientes, puede ser especialmente útil controlar uno o más archivos de registro para eventos. El comando **`tail -f /path/to/file`** proporciona las últimas 10 líneas del archivo especificado y continúa ofreciendo líneas nuevas a medida que se escriben en el archivo monitoreado.

Para monitorear los intentos de inicio de sesión fallidos en un terminal, ejecute **ssh** como usuario root mientras otro usuario intenta iniciar sesión en la máquina serverX:

```
[root@serverX ~]$ tail -f /var/log/secure
...
Feb 10 09:01:13 localhost sshd[2712]: Accepted password for root from 172.25.254.254
 port 56801 ssh2
Feb 10 09:01:13 localhost sshd[2712]: pam_unix(sshd:session): session opened for user
 root by (uid=0)
```

Envío de un mensaje de syslog con `logger`

El comando **`logger`** puede enviar mensajes al servicio **`rsyslog`**. De manera predeterminada, envía el mensaje al usuario de la instalación con el aviso de gravedad (**`user.notice`**), a menos que se especifique lo contrario con la opción **`-p`**. Es especialmente útil, probar los cambios en la configuración de **`rsyslog`**.

Para enviar un mensaje a **`rsyslogd`** que se graba en el archivo de registro **`/var/log/boot.log`**, ejecute lo siguiente:

```
[root;@serverX ~]$ logger -p local7.notice "Log entry created on serverX"
```

Referencias

Páginas del manual: **logger**(1), **tail**(1), **rsyslog.conf**(5) y **logrotate**(8)

rsyslog Manual

- **/usr/share/doc/rsyslog-*/manual.html** provisto por el paquete *rsyslog-doc*

Es posible encontrar información adicional 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: Encontrar entradas de registro

Revisión de las entradas del journal de systemd

RH124

Objetivos

Tras finalizar esta sección, los estudiantes deberían poder encontrar e interpretar las entradas de registro en el journal de systemd para solucionar problemas o revisar el estado del sistema.

Cómo encontrar eventos con `journalctl`

El journal de systemd almacena datos de registro en un archivo binario estructurado e indicado. Estos datos incluyen información adicional sobre el evento de registro. En el caso de los eventos de syslog, esto puede incluir, por ejemplo, el recurso y la prioridad del mensaje original.

Importante

En Red Hat Enterprise Linux 7, el diario de systemd se almacena en **`/run/log`** de manera predeterminada, y sus contenidos se borran después del reinicio. Esta configuración puede ser modificada por el administrador del sistema y se analiza en otra parte de este curso.

El comando **journalctl** muestra el journal del sistema completo que comienza con la entrada de registro más antigua, cuando se ejecuta como usuario root:

```
[root@serverX ~]# journalctl
Feb 13 10:01:01 server1 run-parts(/etc/cron.hourly)[8678]: starting 0yum-hourly.cron
Feb 13 10:01:01 server1 run-parts(/etc/cron.hourly)[8682]: finished 0yum-hourly.cron
Feb 13 10:10:01 server1 systemd[1]: Starting Session 725 of user root.
Feb 13 10:10:01 server1 systemd[1]: Started Session 725 of user root.
Feb 13 10:10:01 server1 CROND[8687]: (root) CMD (/usr/lib64/sa/sa1 1 1)
```

El comando **journalctl** resalta en negrita los mensajes de texto con aviso o advertencia de prioridad, y los mensajes con error de prioridad y superiores se resaltan en rojo.

La clave para usar en forma correcta el journal para la solución de problemas y auditorías es limitar las búsquedas en el journal para mostrar solo el resultado relevante. En los siguientes párrafos, se presentarán varias estrategias diferentes para restringir el resultado de consultas del journal.

De manera predeterminada, **journalctl -n** muestra las 10 últimas entradas de registro. Se necesita un parámetro opcional para la cantidad de las últimas entradas de registro que se deben mostrar. Para mostrar las últimas 5 entradas de registro, ejecute:

```
[root@serverX ~]# journalctl -n 5
```

Al solucionar problemas, puede ser práctico filtrar el resultado del journal por prioridad de las entradas del diario. El comando **journalctl -p** usa el nombre o el número de los niveles de prioridad conocidos y muestra los niveles indicados y todas las entradas de nivel

más alto. Los niveles de prioridad conocidos para **journalctl** son depuración, información, aviso, advertencia, error, gravedad, alerta y emergencia.

Para filtrar el resultado del comando **journalctl** a fin de que solo enumere cualquier entrada de registro de error de prioridad o superior, ejecute:

```
[root@serverX ~]# journalctl -p err
```

Al igual que el comando **tail -f**, **journalctl -f** ofrece las últimas 10 líneas del journal y continúa proporcionando las entradas del journal nuevas a medida que se escriben en el journal.

```
[root@serverX ~]# journalctl -f
```

Cuando se buscan eventos específicos, puede ser útil limitar el resultado a un lapso de tiempo específico. El comando **journalctl** tiene dos opciones para limitar el resultado a un intervalo de tiempo determinado, las opciones **--since** y **--until**. Ambas opciones toman un parámetro de tiempo con el formato **YYYY-MM-DD hh:mm:ss**. Si se omite la fecha, el comando asume que la fecha es hoy y si no se indica la parte de la hora, se asume que el día completo comienza a las 00:00:00. Ambas opciones consideran **yesterday**, **today** y **tomorrow** como parámetros válidos, además del campo de fecha y hora.

Proporciona todas las entradas del journal que se registraron hoy:

```
[root@serverX ~]# journalctl --since today
```

Además del contenido visible del journal, existen campos adjuntos a las entradas del registro que solo pueden verse cuando se activa el resultado de explicación extensa. Para filtrar el resultado de una consulta del journal, pueden usarse todos los campos adicionales que se muestran. Esto es útil para restringir el resultado de búsquedas complejas para determinados eventos del journal.

```
[root@serverX ~]# journalctl -o verbose
Thu 2014-02-13 02:06:00.409345 EST [s=0b47abbf995149c191a8e539e18c3f9c;
i=d28;b=1ea26e84667848af9a4a2904a76ff9a5;m=4d6878ff5a;t=4f244525daa67;
x=880bc65783036719]
    PRIORITY=6
    _UID=0
    _GID=0
    _BOOT_ID=1ea26e84667848af9a4a2904a76ff9a5
    _MACHINE_ID=4513ad59a3b442ffa4b7ea88343fa55f
    _CAP_EFFECTIVE=0000001fffffffff
    _TRANSPORT=syslog
    SYSLOG_FACILITY=10
    SYSLOG_IDENTIFIER=sshd
    _COMM=sshd
    _EXE=/usr/sbin/sshd
    _SYSTEMD_CGROUP=/system.slice/sshd.service
    _SYSTEMD_UNIT=sshd.service
    _SELINUX_CONTEXT=system_u:system_r:sshd_t:s0-s0:c0.c1023
    _HOSTNAME=serverX
    _CMDLINE=sshd: root [priv]
    SYSLOG_PID=6833
    _PID=6833
    MESSAGE=Failed password for root from 172.25.X.10 port 59371 ssh2
    _SOURCE_REALTIME_TIMESTAMP=1392275160409345
```

Entre las opciones más prácticas para buscar líneas que sean relevantes para un proceso o evento especial están:

- _COMM, el nombre del comando
- _EXE, la ruta hacia el ejecutable para el proceso
- _PID, la PID del proceso
- _UID, la UID del usuario que ejecuta el proceso
- _SYSTEMD_UNIT, la unidad systemd que inició el proceso

Puede combinarse más de una de estas opciones. Por ejemplo, la siguiente consulta muestra las entradas del journal relacionadas con los procesos que fueron iniciados por el archivo de unidad de systemd, sshd.service, que también tiene el PID 1182:

```
[root@serverX ~]# journalctl _SYSTEMD_UNIT=sshd.service _PID=1182
```

nota

Para obtener una lista de los campos más usados del journal, consulte la página del manual systemd.journal-fields(7).

Referencias

Páginas del manual: (1) y systemd.journal-fields (7)**`journalctl`**

Es posible encontrar información adicional 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: búsqueda de eventos con journalctl

Preservando el journal de systemd

RH124

Objetivos

Tras finalizar esta sección, los estudiantes deberían poder configurar **`systemd-journald`** para que almacene el diario en el disco y no en la memoria.

Almacenar el journal del sistema de manera permanente.

De manera predeterminada, el journal de systemd se conserva en **`/run/log/journal`**, lo que significa que se borra cuando se reinicia el sistema. El journal es un mecanismo nuevo en Red Hat Enterprise Linux 7, y para la mayoría de las instalaciones, basta con un journal detallado que comienza con el último inicio.

Si el directorio **`/var/log/journal`** existe, el journal se registrará, en cambio, en ese directorio. La ventaja es que los datos históricos estarán disponibles de inmediato en el inicio. Sin embargo, incluso cuando el journal sea persistente, no todos los datos se conservarán para siempre. El journal tiene un mecanismo de rotación de registro incorporado que se activará mensualmente. Además, de manera predeterminada, el journal no podrá tener más del 10 % del sistema de archivos en el que está ubicado ni dejar menos del 15 % del sistema de archivos libre. Estos valores pueden ajustarse en **`/etc/systemd/journald.conf`**, y los límites actuales del tamaño del journal se registran cuando comienza el proceso **`systemd-journald`**, como puede verse con el siguiente comando, que muestra las dos primeras líneas de la salida de **`journalctl`**:

```
[root@serverX ~]# journalctl | head -2
-- Logs begin at Wed 2014-03-05 15:13:37 CST, end at Thu 2014-03-06 21:57:54 CST. --
Mar 05 15:13:37 serverX.example.com systemd-journal[94]: Runtime journal is using 8.0M
 (max 277.8M, leaving 416.7M of free 2.7G, current limit 277.8M).
```

El journal de systemd puede hacerse persistente si se crea el directorio **/var/log/journal** como usuario raíz:

```
[root@serverX ~]# mkdir /var/log/journal
```

Asegúrese de que el directorio **/var/log/journal** sea propiedad del usuario raíz y del grupo systemd-journal, y que tenga los permisos 2755.

```
[root@serverX ~]# chown root:systemd-journal /var/log/journal
[root@serverX ~]# chmod 2755 /var/log/journal
```

Es necesario que se reinicie el sistema o que se envíe la señal especial **USR1** como usuario root al proceso **systemd-journald**.

```
[root@serverX ~]# killall -USR1 systemd-journald
```

Puesto que el journal de systemd ahora es persistente en todos los reinicios, **journalctl -b** puede reducir la salida si solo muestra los mensajes de registros desde el último inicio del sistema.

```
[root@serverX ~]# journalctl -b
```

nota

Cuando se depura el bloqueo de un sistema con un journal constante, generalmente es necesario limitar la cola del journal al reinicio anterior al bloqueo. La opción **`-b`** puede estar acompañada por un número negativo que indica la cantidad de arranques anteriores del sistema a la que debe limitarse la salida. Por ejemplo, **`journalctl -b -1`** limita la salida al inicio anterior.

Referencias

Páginas del manual: **`mkdir`**(1), **`systemd-journald`**(1), **`killall`**(1)

Es posible encontrar información adicional 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: Configuración del journal de systemd constante

Mantenimiento de la hora correcta

RH124

Objetivos

Tras finalizar esta sección, los estudiantes deberían poder conservar la sincronización precisa de la hora y la configuración de la zona horaria para garantizar que las marcas de tiempo sean correctas en los registros del sistema.

Configure los relojes y la zona horaria local.

La hora correcta del sistema sincronizado es muy importante para el análisis del archivo de registro en varios sistemas. El *Protocolo de tiempo en red* (*NTP*) es una manera estándar para que las máquinas proporcionen y obtengan la información de la hora correcta de Internet. Una máquina puede obtener información de la hora correcta de los servicios NTP públicos en Internet, como el NTP Pool Project. Otra opción es un reloj de hardware de alta calidad para proporcionar la hora precisa a los clientes locales.

El comando **`timedatectl`** muestra una descripción general de los parámetros de configuración relacionados con la hora, que incluyen la hora actual, la zona horaria y los parámetros de configuración de sincronización de NTP del sistema.

```
[student@serverX ~]$ timedatectl
      Local time: Thu 2014-02-13 02:16:15 EST
  Universal time: Thu 2014-02-13 07:16:15 UTC
        RTC time: Thu 2014-02-13 07:16:15
        Timezone: America/New_York (EST, -0500)
     NTP enabled: yes
NTP synchronized: no
 RTC in local TZ: no
      DST active: no
 Last DST change: DST ended at
                  Sun 2013-11-03 01:59:59 EDT
                  Sun 2013-11-03 01:00:00 EST
 Next DST change: DST begins (the clock jumps one hour forward) at
                  Sun 2014-03-09 01:59:59 EST
                  Sun 2014-03-09 03:00:00 EDT
```

Está disponible una base de datos con las zonas horarias conocidas y puede enumerarse con:

```
[student@serverX ~]$ timedatectl list-timezones
Africa/Abidjan
Africa/Accra
Africa/Addis_Ababa
Africa/Algiers
Africa/Asmara
Africa/Bamako
...
```

Los nombres de las zonas horarias se basan en la base de datos de zonas horarias "tz" (o "zoneinfo") públicas que están a cargo de la Autoridad para la Asignación de Números de Internet (IANA). Las zonas horarias se nombran según el continente u océano; luego, por lo general, pero no siempre, la ciudad más grande dentro de la región de la zona horaria. Por ejemplo, la mayoría de la zona horaria de montaña de los EE. UU. se denomina "América/Denver".

La elección del nombre correcto puede ser no intuitiva en casos donde las localidades dentro de una zona horaria tienen normas horarias de aprovechamiento de la luz solar. Por ejemplo, en los EE. UU., gran parte del estado de Arizona (hora de la zona montañosa de los EE. UU.) no modifica la hora para aprovechar la luz solar y su huso horario es el de "América/Phoenix".

El comando **tzselect** es práctico para identificar los nombres de la zona horaria zoneinfo correcta. De manera interactiva, se le formulan preguntas al usuario sobre la ubicación del sistema y se proporciona el nombre de la zona horaria correcta. No implementa cambios en la configuración de la zona horaria del sistema.

La configuración del sistema para la zona horaria actual puede modificarse como usuario root:

```
[root@serverX ~]# timedatectl set-timezone America/Phoenix
[root@serverX ~]# timedatectl
      Local time: Thu 2014-02-13 00:23:54 MST
  Universal time: Thu 2014-02-13 07:23:54 UTC
        RTC time: Thu 2014-02-13 07:23:53
        Timezone: America/Phoenix (MST, -0700)
     NTP enabled: yes
NTP synchronized: no
 RTC in local TZ: no
      DST active: n/a
```

Para cambiar los parámetros de configuración de fecha y hora actuales con el comando **timedatectl**, está disponible la opción **set-time**. La hora se especifica con el formato "DD-MM-AAA hh:mm:ss", donde se puede omitir la fecha o la hora. Para cambiar la hora a 09:00:00, ejecute:

```
[root@serverX ~]$ timedatectl set-time 9:00:00
[root@serverX ~]$ timedatectl
      Local time: Thu 2014-02-13 09:00:27 MST
  Universal time: Thu 2014-02-13 16:00:27 UTC
        RTC time: Thu 2014-02-13 16:00:28
        Timezone: America/Phoenix (MST, -0700)
     NTP enabled: yes
NTP synchronized: no
 RTC in local TZ: no
      DST active: n/a
```

La opción **set-ntp** habilita o inhabilita la sincronización de NTP para el ajuste de hora automático. La opción requiere de un argumento **true** o **false** para activarla o desactivarla. Para activar la sincronización de NTP, ejecute:

```
[student@desktopX ~]$ timedatectl set-ntp true
```

Configuración y control de chronyd

El servicio **chronyd** se encarga de que el reloj de hardware local (RTC), que por lo general es impreciso, esté dentro de los parámetros establecidos mediante la sincronización con los servidores NTP configurados o, en caso de que no haya conectividad de red disponible, con la desviación del reloj de RTC calculada que se registra en el **driftfile** especificado en el archivo de configuración **/etc/chrony.conf**.

De manera predeterminada, **chronyd** usa servidores del NTP Pool Project para la sincronización del tiempo y no necesita otra configuración. Puede ser útil cambiar los servidores NTP cuando la máquina en cuestión esté en una red aislada.

La calidad de la fuente de la hora NTP está determinada por el valor del **estrato** informado por la fuente de la hora. El **estrato** determina la cantidad de saltos con que la máquina se aleja del reloj de referencia de alto rendimiento. El reloj de referencia es una fuente de hora de **estrato 0**. Un servidor NTP conectado en forma directa a dicho reloj es un **estrato 1**, mientras que una máquina que sincroniza la hora a partir de un servidor NTP es una fuente de hora **estrato 2**.

Existen dos categorías de fuentes de hora que pueden configurarse en el archivo de configuración **/etc/chrony.conf**, **server** y **peer**. El **server** se encuentra un estrato más arriba que el servidor NTP local y **peer** está en el mismo estrato. Puede especificarse más de un **server** y más de un **peer**, uno por línea.

Para volver a configurar el servidor **chronyd** para sincronizarlo con classroom.example.com, en lugar de hacerlo con los servidores predeterminados configurados en **/etc/chrony.conf**, elimine las otras entradas de servidor y reemplácelas con la siguiente entrada del archivo de configuración:

```
# Use public servers from the pool.ntp.org project.
server classroom.example.com iburst
```

Después de orientar **chronyd** hacia la fuente de hora local, classroom.example.com, es necesario reiniciar el servicio:

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

El comando **chronyc** actúa como cliente para el servicio **chronyd**. Después de configurar la sincronización NTP, puede ser práctico verificar si el servidor NTP se usó para sincronizar el reloj del sistema. Esto puede lograrse con el comando **chronyc sources** o, para un resultado más extenso con explicaciones adicionales sobre el resultado, con el comando **chronyc sources -v**:

```
[root@serverX ~]$ chronyc sources -v
210 Number of sources = 1

  .-- Source mode  '^' = server, '=' = peer, '#' = local clock.
 / .- Source state '*' = current synced, '+' = combined , '-' = not combined,
| /   '?' = unreachable, 'x' = time may be in error, '~' = time too variable.
||                                                 .- xxxx [ yyyy ] +/- zzzz
||                                                /   xxxx = adjusted offset,
||         Log2(Polling interval) -.             |    yyyy = measured offset,
||                                  \            |    zzzz = estimated error.
||                                   |           |
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* classroom.example.com         8   6    17    23   -497ns[-7000ns] +/-  956us
```

El carácter * en el campo **S** (estado Source) indica que el servidor classroom.example.com se usó como fuente de hora y el servidor NTP es la máquina que se toma actualmente como referencia para la sincronización.

nota

Red Hat Enterprise Linux 6 y las versiones anteriores usan **ntpd** y **ntpq** para administrar la configuración de NTP. Puede encontrar más información en la documentación de Red Hat Enterprise Linux 6.

Referencias

Páginas del manual **timedatectl**(1), **tzselect**(8), **chronyd**(8), **chrony.conf**(5) y **chronyc**(1)

Es posible encontrar información adicional 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/

> NTP Pool Project
> http://www.pool.ntp.org/

> Base de datos de zona horaria
> http://www.iana.org/time-zones

Práctica: Ajuste de la hora del sistema