2. Administración de Sistemas Linux

2.1. Introducción y objetivos

En este tema vamos a estudiar los conceptos asociados al sistema operativo Linux. Si bien hay muchas distribuciones y versiones se va a ver un repaso general del sistema para adquirir los conocimientos base de este sistema operativo que vamos a usar a lo largo del resto de temas de esta asignatura.

Empezaremos con un repaso teórico sobre Linux y su nacimiento e iremos viendo todas las áreas importantes para un administrador de sistemas o un operado de sistemas como son gestión de usuarios, seguridad, redes, almacenamiento, Shell, comandos principales, procesos, daemons y sistema.

Con el estudio de este tema pretendemos alcanzar los siguientes objetivos:

* Obtener la habilidad suficiente para realizar las tareas principales de administración de sistemas en Linux.
* Desarrollar las habilidades necesarias para un administrador de sistemas Linux.

En los vídeos de este tema podremos aprender los siguientes conceptos:

Vídeo 1. Instalación en un entorno virtualizado de una distribución de Linux Red Hat. Veremos todas las opciones de configuración y realizaremos el proceso completo para tener un sistema operativo Linux empresarial instalado.

Vídeo 2. Comandos Linux. Aprenderemos más de 30 comandos importantes para administradores de sistemas.

2.2. Primeros pasos en Linux

¿Por qué debería aprender sobre Linux?

Según RedHat (2022), es muy importante que los profesionales de TI entiendan la tecnología de Linux.

El uso de Linux está generalizado; si usa Internet, probablemente ya esté interactuando con sistemas Linux en su vida diaria. Quizás la forma más obvia en que interactúa con sistemas Linux es al navegar por la World Wide Web o al usar sitios de comercio electrónico para comprar y vender productos.

Sin embargo, Linux se usa para mucho más que eso. Linux administra los sistemas de punto de venta y los mercados de valores del mundo y, también, se usa en los televisores inteligentes y en los sistemas de entretenimiento durante el vuelo. Se usa en la mayoría de las quinientas supercomputadoras más importantes del mundo.

Linux proporciona las tecnologías fundamentales que impulsan la revolución de la nube y las herramientas utilizadas para desarrollar la próxima generación de aplicaciones de microservicios basadas en contenedores, tecnologías de almacenamiento basado en software y soluciones de grandes volúmenes de datos.

En el centro de datos modernos, Linux y Microsoft Windows son los principales actores y, además, Linux es un segmento en crecimiento en ese espacio. Algunas de las muchas razones por las que debe aprender a usar Linux incluyen:

* Un usuario de Windows necesita interoperar con Linux.
* En el desarrollo de aplicaciones, Linux aloja la aplicación o su tiempo de ejecución.
* En la computación en nube, las instancias de la nube en el entorno de nube privada o pública usan Linux como sistema operativo.
* Con las aplicaciones móviles o Internet de las cosas (IoT), hay muchas posibilidades de que el sistema operativo de su dispositivo use Linux.
* Si está buscando nuevas oportunidades en TI, hay mucha demanda de habilidades de Linux.

¿Qué hace que Linux sea excelente?

Hay muchas respuestas diferentes a la pregunta «¿Qué hace que Linux sea excelente?». Tres de ellas son:

* Linux es software de código abierto. Al ser de código abierto, no solo se puede ver cómo funciona el sistema, sino también se pueden probar cambios y compartirlos libremente para que otros los utilicen. El modelo de código abierto facilita el desarrollo de mejoras, lo que acelera la innovación.
* Linux proporciona fácil acceso a una interfaz de línea de comandos (CLI) potente y programable. Linux se construyó alrededor de la filosofía de diseño básica de que los usuarios pueden realizar todas las tareas de administración desde la CLI. Esto facilita la automatización, la implementación y el aprovisionamiento, así como simplifica la administración de sistemas locales y remotos. A diferencia de otros sistemas operativos, estas capacidades se han incorporado desde el principio y la idea ha sido siempre permitir estas importantes capacidades.
* Linux es un sistema operativo modular que hace posible reemplazar o eliminar componentes fácilmente. Los componentes del sistema se pueden actualizar y mejorar según sea necesario. Un sistema Linux puede ser una estación de trabajo de desarrollo de uso general o un dispositivo de software en su mínima expresión.

¿Qué es el software de código abierto?

El software de código abierto es software con código fuente que cualquiera puede usar, estudiar, modificar y compartir.

El código fuente es el conjunto de instrucciones en lenguaje humano que se usan para hacer un programa.

Puede interpretarse como un script o compilarse en un archivo ejecutable binario que la computadora ejecuta directamente. Al crear el código fuente, se le aplican los derechos de autor y su titular controla los términos bajo los cuales se puede copiar, adaptar y distribuir el software. Los usuarios pueden usar este software bajo una licencia de software.

Otros tipos de software tienen un código fuente que solo la persona, el equipo o la organización que lo crearon pueden verlo, cambiarlo o distribuirlo. Este software a veces se denomina software «patentado» o de «código cerrado». Por lo general, la licencia solo permite que el usuario final ejecute el programa y no proporciona acceso al código fuente o brinda acceso muy limitado a este.

El software de código abierto es diferente. Cuando el titular de los derechos de autor proporciona software con una licencia de código abierto, otorga a los usuarios el derecho de ejecutar el programa y también de ver, modificar, compilar y redistribuir el código fuente a otros sin pagar regalías.

El código abierto promueve la colaboración, el intercambio, la transparencia y la innovación rápida, ya que alienta a otras personas, más allá de los desarrolladores originales, a realizar modificaciones y mejoras en el software, así como a compartirlo con los demás.

El hecho de que el software sea de código abierto no significa que de alguna manera no se pueda usar o proporcionar para fines comerciales. El código abierto es una parte crítica de las operaciones comercialesde muchas organizaciones. Algunas licencias de código abierto permiten que el código se reutilice en productos de código cerrado.

Se puede vender el código fuente abierto, pero los términos de las verdaderas licencias de código abierto, generalmente, permiten que el cliente redistribuya el código fuente. Más comúnmente, los proveedores como Red Hat proporcionan ayuda comercial con la implementación, el soporte y la ampliación de soluciones basadas en productos de código abierto.

El código abierto tiene muchos beneficios para el usuario:

* Control: vea lo que hace el código y cámbielo para mejorarlo.
* Formación: aprenda del código real y desarrolle aplicaciones más útiles.
* Seguridad: inspeccione el código confidencial y corríjalo ya sea con la ayuda de los desarrolladores originales o sin ella.
* Estabilidad: el código puede sobrevivir a la pérdida del desarrollador o distribuidor original.

La conclusión es que el código abierto permite la creación de mejor software con un mayor retorno de la inversión mediante la colaboración.

Tipos de licencias de código abierto

Hay más de una forma de proporcionar software de código abierto. Los términos de la licencia de software controlan cómo puede reutilizarse el código fuente o cómo puede combinarse con otro código y existen cientos de licencias de código abierto diferentes. Sin embargo, para ser de código abierto, las licencias deben permitir que los usuarios usen, vean, cambien, compilen y distribuyan libremente el código.

Hay dos clases amplias de licencias de código abierto que son particularmente importantes:

* Las licencias copyleft, que están diseñadas para fomentar que el código se mantenga abierto.
* Las licencias permisivas, que están diseñadas para maximizar la reutilización del código.

A black background with a black square

Description automatically generated with medium confidence

Figura 1. Símbolo licencias con copyleft. Fuente: Wikipedia, 2022.

Las licencias copyleft o de «uso compartido para todos» requieren que cualquier persona que distribuya el código fuente, con o sin cambios, también deba extender esta libertad para que los demás también puedan copiar, cambiar y distribuir el código. La ventaja básica de estas licencias es que ayudan a que el código existente y las mejoras a ese código sigan siendo de código abierto, para aumentar la cantidad de código fuente abierto disponible. Las licencias copyleft más comunes son la Licencia Pública General de GNU (GNU General Public License, GPL) y la Licencia Pública General Reducida de GNU (GNU Lesser General Public License, LGPL).

Las licencias permisivas están diseñadas para maximizar la reutilización del código fuente. Los usuarios pueden usar el código fuente para cualquier propósito siempre que se conserven las declaraciones de derechos de autor y de licencia, incluso al reutilizar ese código bajo licencias más restrictivas o incluso de propiedad exclusiva. Esto simplifica mucho la reutilización de este código, pero con el riesgo de incentivar mejoras de propiedad exclusiva. Algunas licencias permisivas de código abierto de uso común son la licencia MIT/X11, la licencia BSD simplificada y la licencia de software Apache 2.0.

¿Quién desarrolla el software de código abierto?

Es un error común pensar que el código abierto lo desarrolla únicamente un «ejército de voluntarios». En la actualidad, el desarrollo de código abierto es una actividad mayoritariamente profesional. Muchas organizaciones pagan a los desarrolladores para que trabajen en proyectos de código abierto para desarrollar y contribuir con las mejoras que ellos y sus clientes necesitan.

Los voluntarios y la comunidad académica desempeñan un papel importante y pueden hacer contribuciones vitales, en especial en áreas de tecnologías nuevas. La combinación de desarrollo formal e informal proporciona un entorno altamente dinámico y productivo.

¿Qué es una distribución de Linux?

Una distribución de Linux es un sistema operativo instalable construido a partir de un kernel de Linux, que admite librerías y programas de usuario. Un sistema operativo Linux completo no está desarrollado por una sola organización, sino por un conjunto de comunidades independientes de desarrollo de código abierto que trabajan con componentes individuales de software. Una distribución proporciona una manera fácil para que los usuarios instalen y administren un sistema Linux que funcione.

En 1991, un joven estudiante de informática llamado Linus Torvalds desarrolló un kernel tipo Unix y lo denominó Linux, con licencia de software de código abierto bajo la GPL. El kernel es el componente central del sistema operativo, que administra el hardware, la memoria y la programación de los programas en ejecución.

A continuación, este kernel de Linux podría complementarse con otro software de código abierto, como utilidades y programas del Proyecto GNU, la interfaz gráfica de X Window System de MIT y muchos otros componentes de código abierto, como el servidor de correo Sendmail o el servidor web Apache HTTP, para desarrollar un sistema operativo completo de código abierto tipo Unix.

Sin embargo, uno de los desafíos para los usuarios de Linux era reunir todas estas piezas de muchas fuentes diferentes. Muy temprano en su historia, los desarrolladores de Linux comenzaron a trabajar para proporcionar una distribución de herramientas precompiladas y probadas que los usuarios pudieran descargar y usar para configurar los sistemas Linux rápidamente.

Existen muchas distribuciones de Linux diferentes, con diferentes objetivos y criterios tanto para seleccionar como para apoyar el software proporcionado por su distribución. Sin embargo, las distribuciones generalmente tienen muchas características comunes:

* Las distribuciones constan de un kernel de Linux y programas de espacio de usuario compatibles.
* Las distribuciones pueden ser pequeñas y de propósito único o incluir miles de programas de código abierto.
* Las distribuciones deben proporcionar un medio para instalar y actualizar la distribución y sus componentes.
* El proveedor de la distribución debe admitir ese software, e idealmente, debe participar directamente en la comunidad que lo desarrolla.

A collection of logos on a white background

Description automatically generated

Figura 2. Existen múltiples distribuciones Linux. Fuente: Redes zone, 2015.

2.3. Introducción a línea de comandos

Introducción a la Shell Bash

RedHat (2022) afirma que una línea de comandos es una interfaz basada en texto que puede usarse para introducir instrucciones en un sistema de cómputo. Un programa denominado shell proporciona la línea de comandos de Linux. Se han desarrollado diversas opciones para el programa de shell a lo largo de los años y pueden configurarse diferentes usuarios para que usen diferentes shells. Sin embargo, la mayoría de los usuarios usan la shell predeterminada actual.

Uno de los Shell de Linux es GNU Bourne-Again Shell (bash). Bash es una versión mejorada de una de las shells más exitosas que se usa en los sistemas tipo UNIX: la shell Bourne (sh).

Si una shell se usa de manera interactiva, muestra una cadena cuando espera un comando del usuario. Esto recibe el nombre de prompt de shell. Cuando un usuario normal inicia una shell, el prompt predeterminado finaliza con un carácter $, como se muestra a continuación.

[user@host ~]$

El carácter $ se reemplaza con el carácter # si la shell se está ejecutando como el superusuario root. Con esto, resulta más evidente que se trata de una shell de superusuario, lo que permite evitar accidentes y errores que pueden afectar a todo el sistema. El prompt de shell de superusuario se muestra a continuación.

[root@host ~]#

El uso de bash para ejecutar comandos puede ser eficaz. La shell bash proporciona un lenguaje de secuencia de comandos capaz de admitir la automatización de tareas. La shell tiene capacidades adicionales que pueden simplificar operaciones o posibilitar aquellas que son difíciles de realizar con herramientas gráficas.

Conceptos básicos de la shell

Los comandos ingresados en el prompt de shell están compuestos por tres partes básicas:

* Comando que se ejecuta.
* Opciones que permiten ajustar el comportamiento del comando.
* Argumentos que generalmente son destinos del comando.

El comando es el nombre del programa que se ejecuta. Puede estar seguido de una o más opciones que ajustan el comportamiento del comando o lo que hará. Las opciones generalmente comienzan con uno o dos guiones (-a o --all, por ejemplo) para que se distingan de los argumentos. Los comandos pueden estar seguidos por uno o más argumentos, que generalmente indican un destino sobre el cual el comando debe operar.

Por ejemplo, el comando usermod -L user01 tiene un comando (usermod), una opción (-L) y un argumento (user01). El efecto de este comando es bloquear la contraseña de la cuenta del usuario user01.

Muchos administradores de sistemas eligen no ejecutar un entorno gráfico en sus servidores. Esto permite que los servicios del servidor usen los recursos que de otra forma usaría el entorno gráfico.

Si inicia sesión con la pantalla de inicio de sesión gráfica, su entorno gráfico se iniciará en la primera consola virtual que no se encuentre en uso en una sesión de inicio de sesión. Normalmente, la sesión gráfica reemplazará al prompt de inicio de sesión en la segunda consola virtual (tty2). Sin embargo, si esa consola está en uso en una sesión de inicio de sesión de texto activa (no solo en un prompt de inicio de sesión), en su lugar, se usa la siguiente consola virtual libre.

La pantalla de inicio de sesión gráfico se sigue ejecutando en la primera consola virtual (tty1). Si ya ha iniciado sesión en una sesión gráfica e inicia sesión con otro usuario en la pantalla de inicio de sesión gráfica, o si usa el ítem de menú Switch User (Cambiar usuario) para cambiar de usuario en el entorno gráfico sin cerrar sesión, se iniciará otro entorno gráfico para ese usuario en la siguiente consola virtual libre.

Cuando cierra sesión en un entorno gráfico, la sesión se cerrará y la consola física volverá automáticamente a la pantalla de inicio de sesión gráfica de la primera consola virtual.

Inicio de sesión a través de la red

Los usuarios y administradores de Linux a menudo necesitan obtener acceso de shell a un sistema remoto conectándose a él a través de la red. En un entorno de cómputo moderno, muchos servidores sin periféricos son en realidad máquinas virtuales o se ejecutan como instancias de nube pública o privada. Estos sistemas no son físicos y no tienen consolas de hardware reales. Es posible que ni siquiera puedan proporcionar acceso a su consola física o consola serial (simulada).

En Linux, la forma más común de obtener un prompt de shell en un sistema remoto es usar Secure Shell (SSH). La mayoría de los sistemas Linux y macOS proporcionan el programa de línea de comandos OpenSSH ssh para este propósito.

En este ejemplo, un usuario con un prompt de shell en la máquina host usa ssh para iniciar sesión en el sistema Linux remoto remotehost como el usuario remoteuser:

[user@host ~]$ ssh remoteuser@remotehost

remoteuser@remotehost's password: password

[remoteuser@remotehost ~]$

El comando ssh cifra la conexión para proteger la comunicación contra el espionaje o el robo de contraseñas y contenido.

Algunos sistemas (como las nuevas instancias en la nube) no permiten que los usuarios usen una contraseña para iniciar sesión con ssh para mejorar la seguridad. Una forma alternativa de autenticarse en una máquina remota sin ingresar una contraseña es a través de la autenticación de clave pública.

Con este método de autenticación, los usuarios tienen un archivo de identidad especial que contiene una clave privada, equivalente a una contraseña, que mantienen en secreto. Su cuenta en el servidor está configurada con una clave pública, que no tiene que ser secreta. Al iniciar sesión, los usuarios pueden configurar ssh para proporcionar la clave privada y, si su clave pública coincidente está instalada en esa cuenta en ese servidor remoto, iniciarán sesión sin necesitar una contraseña.

En el siguiente ejemplo, un usuario con un prompt de shell en la máquina host inicia sesión en remotehost como remoteuser con ssh, mediante autenticación de clave pública. La opción -i se usa para especificar el archivo de clave privada del usuario, que es mylab.pem. La clave pública coincidente ya está configurada como una clave autorizada en la cuenta remoteuser.

[user@host ~]$ ssh -i mylab.pem remoteuser@remotehost

[remoteuser@remotehost ~]$

Para que esto funcione, el archivo de clave privada debe ser legible solo para el propietario del archivo. En el ejemplo anterior, donde la clave privada está en el archivo mylab.pem, el comando chmod 600 mylab.pem podría usarse para garantizar esto. La forma de configurar los permisos de archivo se describe en más detalle en un capítulo posterior.

Los usuarios también pueden tener configuradas claves privadas que se prueban automáticamente, pero ese asunto está fuera del alcance de esta sección. En las referencias que aparecen al final de esta sección, hay más información acerca de este tema.

Introducción al entorno de escritorio de GNOME

El entorno de escritorio es la interfaz gráfica de usuario en un sistema Linux. Proporciona un escritorio integrado para usuarios y una plataforma de desarrollo unificada sobre un marco (framework) gráfico proporcionado por Wayland (opción predeterminada) o la herramienta heredada X Window System.

GNOME Shell proporciona las funciones principales de la interfaz de usuario para el entorno de escritorio GNOME. La aplicación GNOME Shell es muy personalizable. Cualquier tema se puede seleccionar persistentemente en el inicio de sesión al hacer clic en el ícono de engranaje junto al botón Sign In (iniciar sesión) que está disponible después de seleccionar su cuenta, pero antes de ingresar la contraseña.

Figura 3. Partes Shell de Gnome. Fuente: Red Hat, 2021.

La primera vez que inicia sesión como usuario nuevo se ejecuta un programa de configuración inicial para ayudar a configurar parámetros de cuenta básicos. Una vez completa, se inicia la aplicación GNOME Help en la pantalla Getting Started with GNOME (primeros pasos con GNOME). Esta pantalla incluye vídeos y documentación para ayudar a orientar a los nuevos usuarios hacia el entorno de GNOME3. Para iniciar rápidamente GNOME Help, haga clic en el botón Activities (Actividades) a la izquierda de la barra superior y en el guion que aparece a la izquierda de la pantalla; haga clic en el icono de la boya salvavidas para iniciarla.

GNU Bourne-Again Shell (bash) es un programa que interpreta comandos escritos por el usuario. Cada secuencia escrita en la shell puede tener un máximo de tres partes: el comando, las opciones (que suelen comenzar con - o --) y los argumentos. Cada palabra escrita en la shell está separada de las otras por espacios. Los comandos son nombres de programas que están instalados en el sistema. Cada comando tiene sus propias opciones y argumentos.

Cuando esté listo para ejecutar un comando, presione la tecla Intro. Escriba cada comando en una línea separada. La salida del comando se muestra antes de que aparezca el siguiente prompt de shell.

[user@host]$ whoami

user

[user@host]$

Si quiere escribir más de un comando en una sola línea, use un punto y coma (;) como separador de comando. El punto y coma pertenece a la clase de caracteres denominada metacaracteres que tienen un significado especial para bash. En este caso, la salida de ambos comandos se mostrará antes de que aparezca el siguiente prompt de shell.

En el siguiente ejemplo, se muestra cómo combinar dos comandos (command1 y command2) en la línea de comandos.

[user@host]$ command1;command2

Ejemplos de comandos simples

El comando date muestra la fecha y la hora actuales. Además, puede usarlo el superusuario para configurar el reloj del sistema. Un argumento que comienza con el signo más (+) especifica una secuencia de formato para el comando de fecha.

[user@host ~]$ date

Sat Jan 26 08:13:50 IST 2019

[user@host ~]$ date +%R

08:13

[user@host ~]$ date +%x

01/26/2019

El comando passwd cambia la contraseña propia del usuario. La contraseña original de la cuenta debe especificarse antes de que se permita un cambio.

De manera predeterminada, passwd se configura para solicitar una contraseña más sólida que esté compuesta por letras minúsculas, letras mayúsculas, números y símbolos y que no se base en una palabra del diccionario. El superusuario puede usar el comando passwd para cambiar las contraseñas de otros usuarios.

[user@host ~]$ passwd

Changing password for user.

Current password: <strong>old\_password

New password: <strong>new\_password

Retype new password: <strong>new\_password

passwd: all authentication tokens updated successfully.

Linux no requiere de extensiones de nombre de archivo para clasificar los archivos por tipo. El comando file detecta el comienzo del contenido de un archivo y muestra qué tipo de archivo es. Los archivos que se clasificarán pasan como argumentos para el comando.

[user@host ~]$ file /etc/passwd

/etc/passwd: ASCII text

[user@host ~]$ file /bin/passwd

/bin/passwd: setuid ELF 64-bit LSB shared object, x86-64, version 1

(SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2,

for GNU/Linux 3.2.0, BuildID[sha1]=a3637110e27e9a48dced9f38b4ae43388d32d0e4, stripped

[user@host ~]$ file /home

/home: directory

Visualización del contenido de los archivos

Uno de los comandos más simples y de uso frecuente en Linux es cat. El comando cat permite crear archivos individuales o múltiples, ver el contenido de los archivos, concatenar el contenido de varios archivos y redirigirlo a un terminal o a archivos.

En el ejemplo, se muestra cómo ver el contenido del archivo /etc/passwd.

[user@host ~]$ cat /etc/passwd

root:x:0:0:root:/root:/bin/bash

bin:x:1:1:bin:/bin:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin

adm:x:3:4:adm:/var/adm:/sbin/nologin

...output omitted...

Use el siguiente comando para mostrar el contenido de varios archivos.

[user@host ~]$ cat file1 file2

Hello World!!

Introduction to Linux commands.

Algunos archivos son muy largos y pueden ocupar más espacio para mostrar que el proporcionado por el terminal. El comando cat no muestra el contenido de un archivo como páginas. El comando less muestra una página del archivo a la vez y le permite desplazarse a su gusto.

El comando less le permite avanzar y retroceder por las páginas de los archivos que son más largos de lo que puede mostrar una ventana de terminal. Use las teclas UpArrow (Flecha arriba) y DownArrow (Flecha abajo) para desplazarse hacia arriba y hacia abajo. Presione q para salir del comando.

Los comandos head y tail muestran el comienzo y el final de un archivo, respectivamente. De manera predeterminada, estos comandos muestran diez líneas del archivo, pero ambos tienen la opción -n que permite la especificación de una cantidad diferente de líneas. El archivo que se mostrará pasa como un argumento para estos comandos.

[user@host ~]$ head /etc/passwd

root:x:0:0:root:/root:/bin/bash

bin:x:1:1:bin:/bin:/sbin/nologin

daemon:x:2:2:daemon:/sbin:/sbin/nologin

adm:x:3:4:adm:/var/adm:/sbin/nologin

lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin

sync:x:5:0:sync:/sbin:/bin/sync

shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown

<span style="font-size: 12.8px;">halt:x:7:0:halt:/sbin:/sbin/halt

</span>mail:x:8:12:mail:/var/spool/mail:/sbin/nologin

operator:x:11:0:operator:/root:/sbin/nologin

[user@host ~]$ tail -n 3 /etc/passwd

gdm:x:42:42::/var/lib/gdm:/sbin/nologin

gnome-initial-setup:x:977:977::/run/gnome-initial-setup/:/sbin/nologin

avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin

El comando wc cuenta líneas, palabras y caracteres en un archivo. Puede usar la opción -l, -w o -c para mostrar solo la cantidad de líneas, palabras o caracteres, respectivamente.

[user@host ~]$ wc /etc/passwd

45 102 2480 /etc/passwd

[user@host ~]$ wc -l /etc/passwd ; wc -l /etc/group

45 /etc/passwd

70 /etc/group

[user@host ~]$ wc -c /etc/group /etc/hosts

966 /etc/group

516 /etc/hosts

1482 total

Continuación de un comando largo en otra línea

Los comandos con muchas opciones y argumentos pueden volverse largos rápidamente y la ventana de comandos los delimita automáticamente cuando el cursor llega al margen derecho. En cambio, para facilitar la lectura del comando, puede escribir un comando largo utilizando más de una línea.

Para ello, utilizará un carácter de barra invertida (\), al que se conoce como carácter de escape, para ignorar el significado del carácter que se encuentra inmediatamente después de la barra invertida. Aprendió que, al ingresar un carácter de nueva línea, presionando la tecla Enter (Intro) se le indica a la shell que la entrada del comando está completa y que ejecute el comando. Con el escape del carácter de nueva línea, se le indica a la shell que se traslade a una nueva línea de comando sin la ejecución del comando. La shell reconoce la solicitud mostrando un prompt de continuación, denominado prompt secundario, mediante el carácter mayor que (>) de forma predeterminada, en una nueva línea vacía. Los comandos pueden continuarse a lo largo de muchas líneas.

[user@host]$ head -n 3 \

> /usr/share/dict/words \

> /usr/share/dict/linux.words

==> /usr/share/dict/words <==

1080

10-point

10th

==> /usr/share/dict/linux.words <==

1080

10-point

10th

[user@host ~]$

Historial de comandos

El comando history muestra una lista de los comandos ejecutados anteriormente con un número de comando como prefijo.

El carácter de signo de admiración, (!), es un metacarácter que se usa para expandir los comandos anteriores sin tener que volver a escribirlos. El comando !number se expande hasta el comando que coincide con el número especificado. El comando !string se expande hasta el comando más reciente que comienza con la secuencia especificada.

<strong>[user@host ~]$ history

</strong>...output omitted...

23 clear

24 who

25 pwd

26 ls /etc

27 uptime

<span style="font-size: 12.8px;">28 ls -l

</span>29 date

<span style="font-size: 12.8px;">30 history

</span><strong>[user@host ~]$ !ls

</strong><span style="font-size: 12.8px;">ls -l

</span>total 0

drwxr-xr-x. 2 user 6 Mar 29 21:16 Desktop

...output omitted...

<strong>[user@host ~]$ !26

</strong>ls /etc

abrt hosts pulse

adjtime hosts.allow purple

aliases hosts.deny qemu-ga

...output omitted...

2.4. Administración de archivos desde línea de comandos

RedHat (2022) afirma que todos los archivos de un sistema Linux se guardan en sistemas de archivos que están organizados en un árbol de directorios invertido individual conocido como jerarquía de sistema de archivos. Este árbol está invertido porque se dice que la root del árbol está en la parte superior de la jerarquía y las ramas de los directorios y subdirectorios se extienden debajo de root.

A diagram of folders with text

Description automatically generated

Figura 4. Directorios importantes del sistema de archivos en Red Hat. Fuente: Red Hat, 2021.

El directorio / es el directorio raíz (root) que está en la parte superior de la jerarquía del sistema de archivos. El carácter / también se usa también como un separador de directorio en los nombres de archivo. Por ejemplo, si etc es un subdirectorio del directorio /, podemos llamar a ese directorio /etc. De la misma manera, si el directorio /etc contiene un archivo con el nombre issue, podemos referirnos a ese archivo como /etc/issue.

Los subdirectorios de / se usan con fines estandarizados para organizar archivos por tipo y objetivo. Esto facilita la posibilidad de encontrar archivos. Por ejemplo, en el directorio raíz, el subdirectorio /boot se usa para guardar archivos que se necesitan para arrancar el sistema.

Los siguientes términos ayudan a describir el contenido del directorio del sistema de archivos:

* El contenido estático no se modifica hasta que se edita o se reconfigura en forma explícita.
* El contenido dinámico o variable se puede modificar o adjuntar mediante procesos activos.
* El contenido persistente se mantiene después de un arranque nuevo, como los parámetros de configuración.
* El contenido de tiempo de ejecución es contenido específico del proceso o del sistema que se elimina en el reinicio.

A close-up of a list

Description automatically generated

Tabla 1. Directorios importantes en la distribución de Red Hat Enterprise Linux. Fuente: elaboración propia.

Rutas absolutas y rutas relativas

A screenshot of a computer

Description automatically generated

Figura 5. La vista del explorador de archivos habitual (izquierda) equivale a la vista descendente (derecha). Fuente: Red Hat, 2021.

La ruta de un archivo o directorio especifica su ubicación única en el sistema de archivos. Si se sigue la ruta de un archivo, pasa por uno o más subdirectorios con nombre que están delimitados por una barra (/) hasta que se llega al destino. Los directorios, también llamados carpetas, contienen otros archivos y otros subdirectorios. Se puede hacer referencia a estos de la misma manera que con los archivos.

Rutas absolutas

Una ruta absoluta es un nombre totalmente calificado que especifica la ubicación exacta de los archivos en la jerarquía del sistema de archivos. Comienza en el directorio raíz (/) y especifica cada subdirectorio que se debe atravesar para llegar al archivo específico. Cada archivo del sistema de archivos tiene un único nombre de ruta absoluta, reconocido con una regla simple: un nombre de ruta con una barra (/) como primer carácter es el nombre de la ruta absoluta. Por ejemplo, el nombre de ruta absoluta para el archivo de registro de mensajes del sistema es /var/log/messages. Los nombres de rutas absolutas pueden ser extensos; en consecuencia, los archivos también pueden ubicarse en forma relativa al directorio de trabajo actual para su prompt de shell.

El directorio de trabajo actual y las rutas relativas

Cuando un usuario inicia sesión y abre una ventana de comandos, la ubicación inicial es, por lo general, el directorio de inicio del usuario. Los procesos del sistema también tienen un directorio inicial. Los usuarios y los procesos navegan hacia otros directorios, según sea necesario; los términos directorios de trabajo o directorio de trabajo actual se refieren a su ubicación actual.

Al igual que una ruta absoluta, una ruta relativa identifica un archivo único y especifica solo la ruta necesaria para llegar al archivo desde el directorio de trabajo. Para reconocer nombres de ruta relativas, se sigue una regla simple: un nombre de ruta que no tenga otro carácter más que una barra como primer carácter es un nombre de ruta relativa. Un usuario en el directorio /var podría referirse en forma relativa al archivo de registro del mensaje como log/messages.

Los sistemas de archivos Linux, que incluyen, entre otros, ext4, XFS, GFS2 y GlusterFS, distinguen entre letras minúsculas y mayúsculas. Si se crea un archivo FileCase.txt y un archivo filecase.txt en el mismo directorio, se obtienen dos archivos exclusivos.

Los sistemas de archivos que no pertenecen a Linux podrían funcionar de manera diferente. Por ejemplo, VFAT, NTFS de Microsoft y HFS+ de Apple tienen un comportamiento de conservación de tipo de letra. A pesar de que estos sistemas de archivos no distinguen entre minúsculas y mayúsculas, muestran los nombres de archivo con el uso de mayúsculas original usado cuando se creó el archivo. Por lo tanto, si intentó crear los archivos en el ejemplo anterior en un sistema de archivos VFAT, ambos nombres se tratarían como nombres que apuntan al mismo archivo y no a dos archivos diferentes.

Administración de archivos de la línea de comandos

Para administrar archivos, debe poder crearlos, eliminarlos, copiarlos y moverlos. También debe organizarlos de forma lógica en directorios, los cuales también debe poder crear, eliminar, copiar y mover.

La siguiente tabla resume algunos de los comandos de administración de archivos más comunes. El resto de esta sección analizará con más detalle las formas de usar estos comandos.

A blue and white list with white text

Description automatically generated

Tabla 2. Comandos comunes de administración de archivos. Fuente: elaboración propia.

Editar archivos con Vim

Un principio clave de diseño de Linux es que la configuración y la información habitualmente se almacenan en archivos basados en texto. Estos archivos se pueden estructurar de varias maneras, como listas de configuraciones, en formatos tipo INI, como XML o YAML estructurados, etc. Sin embargo, la ventaja de los archivos de texto es que se pueden ver y editar con cualquier editor de texto simple.

Vim es una versión mejorada del editor vi que se distribuye con los sistemas Linux y UNIX. Vim es altamente configurable y eficaz para usuarios avanzados; incluye funciones como edición en pantalla partida, formateo de color y resaltado para la edición de texto.

Debe saber cómo usar al menos un editor de texto que se puede utilizar desde un prompt de shell de solo texto. Así, puede editar archivos de configuración basados en texto desde una ventana de terminal o desde inicios de sesión remotos a través de ssh o de la consola web. Entonces, no necesitará acceso a un escritorio gráfico para editar archivos en un servidor y, de hecho, es posible que ese servidor no necesite ejecutar un entorno de escritorio gráfico.

Pero entonces, ¿por qué aprender Vim en lugar de otras opciones posibles?

La razón clave es que Vim casi siempre está instalado en un servidor, si hay un editor de texto. Esto se debe a que vi se especificó a través del estándar POSIX que Linux y muchos otros sistemas operativos tipo UNIX cumplen en gran parte.

Además, Vim se utiliza a menudo como la implementación de vi en otros sistemas operativos o distribuciones comunes. Por ejemplo, macOS actualmente incluye una instalación ligera de Vim de forma predeterminada. Por lo tanto, las habilidades de Vim que aprenda para Linux también podrían ayudarlo a resolver cosas en otro lugar.

Inicio de Vim

Vim se puede instalar en Linux de dos maneras diferentes. Esto puede afectar las funciones y los comandos Vim que tiene disponibles.

Es posible que su servidor solo tenga instalado el paquete vim-minimal. Esta es una instalación muy ligera que incluye solo el conjunto de características principales y el comando vi básico. En este caso, puede abrir un archivo para editarlo con vi filename y tendrá disponibles todas las funciones básicas que se analizan en esta sección.

Alternativamente, su servidor puede tener el instalado el paquete vim-enhanced. Esto proporciona un conjunto de características mucho más completo, un sistema de ayuda en línea y un programa tutorial. Para iniciar Vim en este modo mejorado, utiliza el comando vim.

[user@host ~]$ vim filename

Modos de operación de Vim

Una característica inusual de Vim es que tiene varios modos de operación, incluido el modo de comandos, el modo de comandos ampliado, el modo de edición y el modo visual. Según el modo, es posible que emita comandos, edite texto o trabaje con bloques de texto en el modo visual. Como usuario nuevo de Vim, siempre debe estar al tanto del modo actual, ya que las pulsaciones de teclas tienen diferentes efectos en los diferentes modos.

A diagram of a computer program

Description automatically generated

Figura 6. Cambio entre modos de Vim. Fuente: Red Hat Academy, 2021.

Cuando se abre Vim por primera vez, arranca en el modo de comandos, usado para navegar, cortar y pegar y otro tipo de manipulación de texto. Ingrese en cada uno de los otros modos con pulsaciones de tecla de caracteres únicos para acceder a funciones de edición específicas:

* Al pulsar la tecla i se ingresa al modo de inserción, en el cual todo el texto ingresado se convierte en contenido de archivo. Al presionar Esc se vuelve al modo de comandos.
* Al pulsar la tecla v se ingresa al modo visual, en el cual se pueden seleccionar varios caracteres para la manipulación de texto. Utilice Shift+V (Mayús V) para líneas múltiples y Ctrl+V para seleccionar en bloque. La misma pulsación de tecla utilizada para ingresar al modo visual (v, Shift+V (Mayús V) o Ctrl+V) se utiliza para salir.
* Al pulsar la tecla: se inicia el modo de comandos ampliado para tareas como la escritura del archivo (para guardarlo) y la salida del editor Vim.

En el siguiente vídeo, *Instalación de sistemas operativos Linux,* vamos a realizar una instalación de un sistema operativo Linux Red Hat Enterprise. Veremos la configuración de todos los parámetros necesarios para empezar a trabajar profesionalmente con esta distribución de Linux.

2.5. Usuarios, grupos y permisos

RedHat (2022) afirma que una cuenta de usuario se utiliza para proporcionar límites de seguridad entre diferentes personas y programas que pueden ejecutar comandos.

Los usuarios tienen nombres de usuario para identificarlos como usuarios humanos y facilitar el trabajo con ellos. Internamente, el sistema distingue las cuentas de usuario por el número de identificación único que se les asigna, el ID de usuario o UID. Si las personas utilizan una cuenta de usuario, generalmente se le asignará una contraseña secreta que el usuario utilizará para demostrar que es el usuario autorizado real al iniciar sesión.

Las cuentas de usuario son fundamentales para la seguridad del sistema. Cada proceso (programa en ejecución) en el sistema se ejecuta como un usuario particular. Cada archivo tiene un usuario particular como su propietario. La propiedad del archivo ayuda al sistema a aplicar el control de acceso para los usuarios de los archivos. El usuario asociado con un proceso de ejecución determina los archivos y directorios accesibles para ese proceso.

Hay tres tipos principales de cuenta de usuario: el superusuario, el usuario del sistema y el usuario normal.

* La cuenta de superusuario es para la administración del sistema. El nombre del superusuario es root y la cuenta tiene UID 0. El superusuario tiene acceso completo al sistema.
* El sistema tiene cuentas de usuario del sistema que utilizan los procesos que proporcionan servicios de soporte. Estos procesos o daemons, por lo general, no necesitan ejecutarse como superusuario. Son cuentas asignadas sin privilegios que les permiten proteger sus archivos y otros recursos entre sí y de los usuarios habituales del sistema. Los usuarios no inician sesión de forma interactiva mediante una cuenta de usuario del sistema.
* La mayoría de los usuarios tienen cuentas de usuario normal que utilizan para su trabajo diario. Al igual que los usuarios del sistema, los usuarios normales tienen acceso limitado al sistema.

Puede utilizar el comando id para mostrar información acerca del usuario con sesión iniciada actualmente.

[user01@host ~]$ id

uid=1000(user01) gid=1000(user01) groups=1000(user01) context=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023

Para ver el propietario de un archivo, use el comando ls -l. Para ver el propietario de un directorio, use el comando ls -ld. En la siguiente salida, la tercera columna muestra el nombre de usuario.

[user01@host ~]$ ls -l file1

-rw-rw-r--. 1 user01 0 Feb 5 11:10 file1

[user01@host]$ ls -ld dir1

drwxrwxr-x. 2 user01 6 Feb 5 11:10 dir1

Para ver la información del proceso, use el comando ps. La opción predeterminada es mostrar solo los procesos que están en la shell actual. Agregue la opción a para ver todos los procesos con un terminal. Para ver el usuario relacionado con un proceso, incluya la opción u. En la siguiente salida, la primera columna muestra el nombre de usuario.

[user01@host]$ ps -au

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

root 777 0.0 0.0 225752 1496 tty1 Ss+ 11:03 0:00 /sbin/agetty -o -p -- \u --noclear tty1 linux

root 780 0.0 0.1 225392 2064 ttyS0 Ss+ 11:03 0:00 /sbin/agetty -o -p -- \u --keep-baud 115200,38400,9600

user01 1207 0.0 0.2 234044 5104 pts/0 Ss 11:09 0:00 -bash

user01 1319 0.0 0.2 266904 3876 pts/0 R+ 11:33 0:00 ps au

En la salida de los comandos anteriores, se muestran los usuarios por nombre, pero, internamente, el sistema operativo usa los UID para realizar un seguimiento de los usuarios. La asignación de nombres de usuario a UID se define en las bases de datos de la información de la cuenta. De forma predeterminada, los sistemas usan el archivo /etc/passwd para almacenar información sobre los usuarios locales.

Cada línea del archivo /etc/passwd contiene información sobre un usuario. Se divide en siete campos separados por dos puntos. Este es un ejemplo de una línea de /etc/passwd:

user01: x: 1000: 1000: User One: /home/user01: /bin/bash

1. Nombre de usuario para este usuario (user01).
2. La contraseña del usuario se almacenaba aquí en formato cifrado. Se trasladó al archivo /etc/shadow, que se analizará más adelante. Este campo siempre debe ser X.
3. El número de UID para esta cuenta de usuario (1000).
4. El número de GID para el grupo principal de esta cuenta de usuario (1000). Los grupos se analizarán más adelante en esta sección.
5. El nombre real para este usuario. (User One).
6. El directorio de inicio de este usuario. (/home/user01). Este es el directorio de trabajo inicial cuando se inicia la shell y contiene los datos del usuario y los parámetros de configuración.
7. El programa de shell predeterminado para este usuario, que se ejecuta al iniciar sesión (/bin/bash). Para un usuario normal, por lo general este es el programa que proporciona el prompt de línea de comando del usuario. Un usuario del sistema podría usar /sbin/nologin si no se permiten inicios de sesión interactivos para ese usuario.

¿Qué es un grupo?

Un grupo es una colección de usuarios que necesitan compartir el acceso a archivos y otros recursos del sistema.

Los grupos se pueden utilizar para otorgar acceso a los archivos a un conjunto de usuarios en lugar de a un solo usuario.

Como los usuarios, los grupos tienen nombres de grupo para facilitar el trabajo con ellos. Internamente, el sistema distingue los grupos por el número de identificación único que se les asigna, el ID de grupo o GID.

La asignación de nombres de grupo a GID se define en las bases de datos de la información de la cuenta de grupo. De forma predeterminada, los sistemas usan el archivo /etc/group para almacenar información sobre los grupos locales.

Cada línea del archivo /etc/group contiene información sobre un grupo. Cada entrada de grupo se divide en cuatro campos separados con dos puntos. Este es un ejemplo de una línea de /etc/group:

group01: x: 10000: user01,user02,user03

1. Nombre de grupo para este grupo (group01).
2. Campo de contraseña de grupo obsoleto. Este campo siempre debe ser X.
3. El número de GID para este grupo (10 000).
4. Una lista de usuarios que son miembros de este grupo como grupo adicional (user01, user02 y user03).

Grupos principales y grupos adicionales

Cada usuario tiene exactamente un grupo principal. Para usuarios locales, este es el grupo enumerado por número de GID en el archivo /etc/passwd. De forma predeterminada, este es el grupo que será propietario de los nuevos archivos creados por el usuario.

Normalmente, cuando crea un nuevo usuario normal, se crea un nuevo grupo con el mismo nombre que ese usuario. Ese grupo se usa como el grupo principal del nuevo usuario y ese usuario es el único miembro de este grupo privado de usuarios. Se deduce que esto ayuda a simplificar la administración de los permisos de archivos, que se analizará más adelante en este curso.

Los usuarios también pueden tener grupos adicionales. La pertenencia a grupos adicionales se determina a través del archivo /etc/group. A los usuarios se les otorga acceso a los archivos en función de si alguno de sus grupos tiene acceso. No importa si el grupo o los grupos que tienen acceso son principales o adicionales para el usuario.

Por ejemplo, si el usuario user01 tiene un grupo principal usero01 y grupos adicionales wheel y webadmin, entonces ese usuario puede leer archivos legibles para cualquiera de esos tres grupos.

El comando id también se puede utilizar para averiguar sobre la membresía de grupo de un usuario.

[user03@host ~]$ id<strong>

</strong>uid=1003(user03) gid=1003(user03) groups=1003(user03),10(wheel),10000(group01) context=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023

En el ejemplo anterior, user03 tiene el grupo user03 como su grupo principal (gid). El elemento groups enumera todos los grupos de este usuario y, además del grupo principal user03, el usuario tiene los grupos wheel y group01 como grupos adicionales.

El superusuario

La mayoría de los sistemas operativos tienen una especie de superusuario; un usuario que tiene todo el poder sobre el sistema.

Es el usuario root. Este usuario tiene el poder de anular los privilegios normales del sistema de archivos y se usa para manejar y administrar el sistema. Para realizar tareas como la instalación o eliminación de software y para administrar los directorios y los archivos del sistema, los usuarios deben aumentar sus privilegios al usuario root.

El usuario root solo entre los usuarios normales puede controlar la mayoría de los dispositivos, pero hay algunas excepciones. Por ejemplo, los usuarios normales pueden controlar dispositivos desmontables, como dispositivos USB. Por lo tanto, los usuarios normales pueden agregar y eliminar archivos y administrar de otro modo un dispositivo desmontable, pero solo el usuario root puede administrar los discos duros «fijos» de manera predeterminada.

Sin embargo, este privilegio ilimitado viene acompañado de una responsabilidad. El usuario root tiene poder ilimitado para dañar el sistema: eliminar archivos y directorios, eliminar cuentas de usuarios, agregar puertas traseras, etc. Si la cuenta del usuario root está comprometida, alguien más tendrá control administrativo del sistema. A lo largo de este curso, se les indicará a los administradores que inicien sesión como usuario normal y que escalen los privilegios a root solo cuando sea necesario.

La cuenta root en Linux es casi equivalente a la cuenta de administrador local en Microsoft Windows. En Linux, la mayoría de los administradores del sistema inician sesión en el sistema como un usuario sin privilegios y usan distintas herramientas para ganar privilegios de usuario root temporalmente.

Los comandos useradd, userdel y userdel nos permiten gestionar a los usuarios. Los correspondientes comandos para grupos son groupadd, groupmod y groupdel.

Permisos del sistema de archivos de Linux

Los permisos de archivos controlan el acceso a los archivos. Los permisos de archivos de Linux son simples pero flexibles, lo que hace que sean fáciles de comprender y de aplicar y, al mismo tiempo, permiten gestionar fácilmente los casos más normales de permisos.

Los archivos tienen tres categorías de usuario a las que se les aplican permisos. El propietario del archivo es un usuario que normalmente es quien lo creó. También es posible que el propietario del archivo sea un solo grupo, generalmente el grupo primario del usuario que creó el archivo, pero esto se puede cambiar. Pueden establecerse diferentes permisos para el usuario o grupo propietario y para otros usuarios del sistema que no sean el usuario o un miembro del grupo propietario.

Los permisos más específicos tienen prioridad. Los permisos de usuario anulan los permisos de grupo, lo cual anula los permisos de otros.

En la Figura 7 joshua es un miembro de los grupos joshua y web y allison es miembro de los grupos allison, wheel y web. Cuando joshua y allison necesitan colaborar, los archivos deben asociarse con el grupo web y los permisos del grupo deben permitir el acceso deseado.

A blue and purple circles with white text

Description automatically generated

Figura 7. Ejemplo de pertenencia a un grupo para facilitar la colaboración. Fuente: Red Hat, 2021.

Se utilizan tres categorías de permisos: lectura, escritura y ejecución. En la siguiente tabla, se explica cómo estos permisos afectan el acceso a archivos y directorios.

A blue and white text box

Description automatically generated

Tabla 3. Efectos de los permisos en archivos y directorios. Fuente: elaboración propia.

Los usuarios normalmente poseen permisos tanto de lectura como de ejecución en los directorios de solo lectura, por lo que pueden enumerar el directorio y tener acceso completo de solo lectura a su contenido.

Si un usuario solo posee acceso de lectura en un directorio, los nombres de los archivos dentro de este pueden enumerarse, pero no estará disponible otra información, incluidos permisos o marcas de tiempo y tampoco se podrá acceder a ellos. Si un usuario solo tiene acceso de ejecución en un directorio, no puede enumerar nombres de archivo en el directorio. Si el usuario conoce el nombre de un archivo para el cual tiene permiso de lectura, puede acceder al contenido de ese archivo desde fuera del directorio al especificar explícitamente el nombre de archivo relativo.

Todo usuario que sea propietario o cuente con permisos de escritura para el directorio donde se encuentra el archivo puede quitar un archivo, sin importar la propiedad ni los permisos del archivo en sí. Esto puede anularse con un permiso especial, el sticky bit, que trataremos más adelante en este capítulo.

Nota

Los permisos de archivos Linux funcionan de manera diferente al sistema de permisos del sistema de archivos NTFS para Microsoft Windows.

En Linux, los permisos se aplican solo al archivo o directorio en el que están configurados. Es decir, los permisos en un directorio no se heredan de forma automática por los subdirectorios o los archivos que se encuentran en él. Sin embargo, los permisos de un directorio pueden bloquear el acceso al contenido del directorio según qué tan restrictivos sean.

El permiso de lectura en un directorio de Linux es casi equivalente a Mostrar el contenido de la carpetaen Windows.

El permiso de escritura en un directorio de Linux es equivalente a Modify (Modificación) en Windows. Esto implica la posibilidad de eliminar archivos y subdirectorios. En Linux, si escritura y sticky bit están establecidos en un directorio, solo el propietario del archivo o subdirectorio puede eliminarlos, lo que se asemeja al comportamiento del permiso de escritura de Windows.

El usuario root en Linux posee los permisos equivalentes a Full Control (Control total) de Windows en todos los archivos. Sin embargo, el usuario root puede tener acceso restringido por la política de SELinux del sistema que utiliza contextos de seguridad del proceso y de los archivos. SELinux se analizará en un curso posterior.

Visualización de permisos y propiedades de archivos y directorios

La opción -l del comando ls muestra información detallada sobre permisos y propiedad:

[user@host~]$ ls -l test<strong>

</strong>-rw-rw-r--. 1 student 0 Feb 8 17:36 test

Utilice la opción -d para mostrar información detallada sobre un directorio y no sobre su contenido.

[user@host ~]$ ls -ld /home

drwxr-xr-x. 5 root 4096 Jan 31 22:00 /home

2.6. Procesos, servicios y daemons. Monitorización

Definición de un proceso

RedHat (2022) afirma que un proceso es una instancia de un programa ejecutable que se inició y se encuentra en funcionamiento. Un proceso consta de lo siguiente:

* Un espacio de direcciones que incluye la memoria asignada
* Características de seguridad que incluyen credenciales y privilegios de propiedad.
* Uno o más subprocesos de ejecución de código de programa.
* Estado del proceso.

El entorno de un proceso incluye lo siguiente:

* Variables locales y globales.
* Un contexto de programación actual.
* Recursos asignados del sistema, como descriptores de archivos y puertos de red.

Un proceso (principal) existente duplica su propio espacio de direcciones (fork) para crear una nueva estructura de proceso (secundaria). Se asigna una identificación de proceso (PID) única a cada proceso nuevo para su rastreo y por motivos de seguridad. La PID y la identificación del proceso principal (PPID) son elementos del entorno del proceso nuevo. Cualquier proceso puede crear un proceso secundario.

A través del proceso de bifurcación, un proceso secundario hereda identidades de seguridad, descriptores de archivos actuales y anteriores, privilegios de recursos y puertos, variables de entorno y código de programa. Luego, un proceso secundario puede ejecutar su propio código de programa.

Normalmente, un proceso principal se encuentra inactivo mientras se ejecuta el proceso secundario, lo que establece que una solicitud (espera) se señale cuando finalice el proceso secundario. Tras su finalización, el proceso secundario ya ha cerrado o descartado sus recursos y su entorno.

El único recurso restante, llamado zombi, es una entrada en la tabla de procesos. El proceso principal, que se señala como activo una vez que finaliza el proceso secundario, limpia la tabla de procesos de la entrada del proceso secundario y libera el último recurso del proceso secundario. El proceso principal continúa con la ejecución de su propio código de programa.

Descripción de los estados del proceso

En un sistema operativo de funciones múltiples, cada CPU (o núcleo de CPU) puede trabajar en un proceso en un momento dado. Mientras se ejecuta un proceso, sus requisitos inmediatos en cuanto a asignación de recursos y tiempo de la CPU cambian. Los procesos reciben un estado que cambia en la medida en que las circunstancias así lo exijan.

A diagram of a running process

Description automatically generated

Figura 8. Estados de los procesos de Linux. Fuente: Red Hat Academy. Train the trainers, 2021. Manuscrito no publicado.

Los estados de los procesos de Linux se ilustran en el diagrama anterior y se describen en la siguiente tabla.

Por qué los estados de los procesos son importantes

Al solucionar problemas de un sistema, es importante comprender cómo se comunica el kernel con los procesos y cómo los procesos se comunican entre sí.

En la creación del proceso, el sistema asigna un estado al proceso. La columna S del comando top o la columna STAT de ps muestra el estado de cada proceso. En un sistema de una sola CPU, solo se puede ejecutar un proceso a la vez. Es posible ver varios procesos con un estado de R. Sin embargo, no todos se ejecutarán consecutivamente, algunos de ellos estarán en estado en espera.

[user@host ~]$ top

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

1 root 20 0 244344 13684 9024 S 0.0 0.7 0:02.46 systemd

2 root 20 0 0 0 0 S 0.0 0.0 0:00.00 kthreadd

...output omitted...

[user@host ~]$ ps aux

USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND

...output omitted...

root 2 0.0 0.0 0 0 ? <strong>S</strong> 11:57 0:00 [kthreadd]

student 3448 0.0 0.2 266904 3836 pts/0 <strong>R+ </strong>18:07 0:00 ps aux

...output omitted...

El proceso se puede suspender, detener, reanudar, finalizar e interrumpir mediante el uso de señales. Las señales se analizarán en más detalle, posteriormente, en este capítulo. Las señales pueden ser usadas por otros procesos, por el propio kernel o por usuarios conectados al sistema.

Lista de procesos

El comando ps se usa para elaborar una lista de los procesos actuales. Puede proporcionar información detallada de los procesos que incluye:

* La identificación del usuario (UID) que determina los privilegios del proceso.
* La identificación del proceso (PID) única.
* La CPU y el tiempo real empleado.
* La cantidad de memoria que el proceso ha asignado en diversas ubicaciones.
* La ubicación del proceso stdout, conocido como terminal de control.
* El estado del proceso actual.

Realización de trabajos en segundo plano

Cualquier comando o tubería puede iniciarse en segundo plano si se anexa el signo ampersand (&) al final de la línea de comandos. La shell Bash muestra un número de trabajo (exclusivo de la sesión) y el identificador de proceso del proceso secundario nuevo. La shell no espera que finalice el proceso secundario, sino que muestra el prompt de shell.

[user@host ~]$ sleep 10000 &

[1] 5947

[user@host ~]$

Monitoreo del proceso en tiempo real

El programa top es una vista dinámica de los procesos del sistema que muestra un encabezado del resumen seguido de un proceso o lista de subprocesos similares a la información de ps. A diferencia de la salida de ps estática, top continuamente se actualiza a un intervalo configurable y ofrece capacidades de reorganización, ordenado y resaltado de columnas. Las configuraciones del usuario se pueden guardar y hacer persistentes.

Las columnas de salidas predeterminadas se diferencian de otras herramientas de recursos en:

* La ID del proceso (PID).
* El nombre de usuario (USER) es el propietario del proceso.
* La memoria virtual (VIRT) es toda la memoria que está usando el proceso, incluido el conjunto residente, las bibliotecas compartidas y cualquier página de memoria asignada o intercambiada (con la etiqueta VSZ en el comando ps).
* La memoria residente (RES) es la memoria física que usa el proceso, incluido cualquier objeto residente compartido (con la etiqueta RSS en el comando ps).
* El estado del proceso S se muestra como:

D = Suspensión ininterrumpida.

R = En ejecución o ejecutable.

S = En espera.

T = Detenido o en seguimiento.

Z = Zombi.

* El tiempo de CPU (TIME) es el tiempo total de procesamiento desde que comenzó el proceso. Se puede alternar para incluir el tiempo acumulativo de todos los procesos secundarios.
* El nombre del comando de proceso (COMMAND).

Introducción a systemd

El daemon de systemd administra el inicio para Linux, incluido el inicio del servicio y la gestión de servicios en general. Activa los recursos del sistema, los daemons del servidor y otros procesos, tanto en el momento del arranque como en un sistema que está en funcionamiento.

Los daemons son procesos que esperan o se ejecutan en segundo plano y realizan diversas tareas. Generalmente, los daemons 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 daemon finalizan con la letra d.

A menudo, un servicio en el contexto de systemd hace referencia a uno o más daemons, pero iniciar o detener un servicio puede, en cambio, hacer una modificación por única vez en el estado del sistema, que no implica dejar un proceso daemon en ejecución después de esto (que se denomina oneshot).

Visualización de los estados de servicio

Visualice el estado de una unidad específica con systemctl status name.type. Si no se proporciona el tipo de unidad, systemctl mostrará el estado de una unidad de servicio, en caso de que exista una.

[root@host ~]# systemctl status sshd.service

● sshd.service - OpenSSH server daemon

Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; vendor preset: enabled)

Active: active (running) since Thu 2019-02-14 12:07:45 IST; 7h ago

Main PID: 1073 (sshd)

CGroup: /system.slice/sshd.service

─1073 /usr/sbin/sshd -D ...

Feb 14 11:51:39 host.example.com systemd[1]: Started OpenSSH server daemon.

Feb 14 11:51:39 host.example.com sshd[1073]: Could not load host key: /et...y

Feb 14 11:51:39 host.example.com sshd[1073]: Server listening on 0.0.0.0 ....

Feb 14 11:51:39 host.example.com sshd [1073]: Server listening on :: port 22.

Feb 14 11:53:21 host.example.com sshd[1270]: error: Could not load host k...y

Feb 14 11:53:22 host.example.com sshd[1270]: Accepted password for root f...2

...output omitted...

Verificación del estado de un servicio

El comando systemctl proporciona métodos para verificar los estados específicos de un servicio. Por ejemplo, use el siguiente comando para verificar que la unidad de servicio se encuentre activa (en ejecución) en ese momento:

[root@host ~]# systemctl is-active sshd.service

active

El comando devuelve el estado de la unidad de servicio, el cual generalmente es active (activo) o inactive(inactivo).

Ejecute el siguiente comando para verificar si una unidad de servicio está habilitada para iniciarse automáticamente durante el arranque del sistema:

[root@host ~]# systemctl is-enabled sshd.service

enabled

El comando informa si la unidad de servicio está habilitada para iniciarse en el momento del arranque, lo cual, generalmente, se informa como estado enabled (habilitado) o disabled (deshabilitado).

Para verificar si la unidad falló durante el arranque, ejecute el siguiente comando:

[root@host ~]# systemctl is-failed sshd.service

active

Reinicio y recarga de servicios

Durante el reinicio de un servicio en ejecución, el servicio se detiene y, luego, se inicia. Al reiniciar el servicio, la ID del proceso cambia y una nueva ID del proceso se asocia durante el arranque. Para reiniciar un servicio en ejecución, use el argumento restart con el comando systemctl. En el siguiente ejemplo, se muestra cómo reiniciar el servicio sshd.service:

[root@host ~]# systemctl restart sshd.service

Algunos servicios tienen la capacidad de volver a cargar sus archivos de configuración sin necesidad de reiniciar. Este proceso se llama recarga del servicio. La recarga de un servicio no cambia la ID del proceso asociada con diversos procesos del servicio. Para recargar un servicio en ejecución, use el argumento reload con el comando systemctl. En el siguiente ejemplo, se muestra cómo volver a cargar el servicio sshd.service después de modificar la configuración:

[root@host ~]# systemctl reload sshd.service

En caso de que no esté seguro de si el servicio cuenta con la funcionalidad para volver a cargar los cambios del archivo de configuración, use el argumento reload-or-restart con el comando systemctl. El comando vuelve a cargar los cambios de configuración si la funcionalidad de recarga está disponible. De lo contrario, el comando reinicia el servicio para implementar los nuevos cambios de configuración:

[root@host ~]# systemctl reload-or-restart sshd.service

¿Qué es OpenSSH?

OpenSSH implementa el protocolo Secure Shell o SSH en los sistemas Linux. El protocolo SSH permite que los sistemas se comuniquen de forma cifrada y segura a través de una red insegura.

Puede usar el comando ssh para crear una conexión segura a un sistema remoto, autenticarse como un usuario específico y obtener una sesión de shell interactiva en el sistema remoto como ese usuario. También puede usar el comando ssh para ejecutar un comando individual en el sistema remoto sin acceder a la shell interactiva.

Con el siguiente comando ssh, podría iniciar sesión en el servidor remoto remotehost utilizando el mismo nombre de usuario que el usuario local actual. En este ejemplo, el sistema remoto le solicita que se autentique con la contraseña de ese usuario.

[user01@host ~]$ ssh remotehost

user01@remotehost's password: redhat

...output omitted...

[user01@remotehost ~]$

2.7. Análisis y almacenamiento de registros

Registro del sistema

RedHat (2022) afirma que los procesos y el kernel del sistema operativo llevan un registro de los eventos que suceden. Estos registros se usan para realizar una auditoría del sistema y solucionar problemas.

Muchos sistemas registran registros de eventos en archivos de texto que se guardan en el directorio /var/log. Estos registros se pueden inspeccionar utilizando las utilidades de texto normales, como less y tail.

Varias distribuciones Linux incluye un sistema de registro estándar que se basa en el protocolo Syslog. Muchos programas usan este sistema para registrar eventos y organizarlos en archivos de registro.

El servicio systemd-journald está en el corazón de la arquitectura de registro de eventos del sistema operativo. Recolecta mensajes de eventos desde muchas fuentes, incluido el kernel, la salida de las primeras etapas del proceso de arranque, la salida y el error estándar de los daemons cuando se inician y se ejecutan y los eventos de syslog. Luego, los reestructura en un formato estándar y los escribe en un diario de sistema indexado y estructurado. De forma predeterminada, este diario se almacena en un sistema de archivos que no persiste en los reinicios.

Sin embargo, el servicio rsyslog lee los mensajes de syslog recibidos por systemd-journald desde el diario a medida que llegan. Luego procesa los eventos del registro del sistema, los registra en sus archivos de registro o los reenvía a otros servicios de acuerdo con su propia configuración.

El servicio rsyslog ordena y escribe mensajes de syslog en los archivos de registro que no persisten en los reinicios en /var/log. El servicio rsyslog clasifica los mensajes de registro en archivos de registro específicos según el tipo de programa que envió cada mensaje, o instalación, y la prioridad de cada mensaje de syslog.

Además de los archivos de mensajes de syslog, el directorio /var/log contiene archivos de registro de otros servicios en el sistema. En la siguiente tabla, se enumeran algunos archivos útiles del directorio /var/log.

A blue and white checklist with white text

Description automatically generated

Tabla 4. Archivos de registro del sistema seleccionados. Fuente: elaboración propia.

Registro de eventos en el sistema

Muchos programas usan el protocolo syslog para registrar eventos en el sistema. Cada mensaje se clasifica por facility (tipo de mensaje) y prioridad (gravedad del mensaje). Los tipos (facilities) disponibles se detallan en la página de manual rsyslog.conf.

En la siguiente tabla, se enumeran las ocho prioridades estándares de syslog, de mayor a menor.

A screenshot of a computer

Description automatically generated

Tabla 5. Descripción general de las prioridades de syslog. Fuente: elaboración propia.

El servicio rsyslog usa el tipo (facility) y la prioridad de los mensajes de registro para determinar cómo resolverlos. Esto está configurado por reglas en el archivo /etc/rsyslog.conf y cualquier archivo en el directorio /etc/rsyslog.d que tiene una extensión de nombre de archivo de .conf. Los paquetes de software pueden agregar reglas fácilmente instalando un archivo apropiado en el directorio /etc/rsyslog.d.

Cada regla que controla cómo ordenar los mensajes de syslog es una línea en uno de los archivos de configuración. En el lado izquierdo de cada línea, se indican el tipo (facility) y la gravedad del mensaje de syslog que se corresponde con la regla. En el lado derecho de cada línea, se indica en qué archivo se debe guardar el mensaje de registro (o a donde más enviar el mensaje). Un asterisco (\*) es un comodín que coincide con todos los valores.

Por ejemplo, la siguiente línea registraría los mensajes enviados al facility authpriv en cualquier prioridad para el archivo /var/log/secure:

authpriv.\* /var/log/secure

Los mensajes de registro a veces coinciden con más de una regla en rsyslog.conf. En tales casos, un mensaje se almacena en más de un archivo de registro. Para limitar los mensajes almacenados, la palabra clave none en el campo de prioridad señala que no se deben almacenar mensajes para el tipo (facility) indicado en el archivo dado.

En lugar de registrar mensajes de syslog en un archivo, también pueden imprimirse en las terminales de todos los usuarios que hayan iniciado sesión. El archivo rsyslog.conf tiene una configuración para imprimir todos los mensajes de syslog con la prioridad emerg en las terminales de todos los usuarios que hayan iniciado sesión.

Rotación del archivo de registro

La herramienta logrotate rota los archivos de registro para evitar que ocupen demasiado espacio en el sistema de archivos que contiene el directorio /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. Por ejemplo, el archivo /var/log/messages anterior puede pasar a ser /var/log/messages-20190130 si se rota el 30-1-2019. 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 más viejo se descarta para liberar espacio en disco. Un trabajo programado ejecuta el programa logrotate a diario para verificar si es necesario rotar algún registro. La mayoría de los archivos de registro se rotan semanalmente, pero logrotate rota un poco más rápidamente o más lento, o cuando alcanza 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 mensajes de registro comienzan con el mensaje más antiguo en la parte superior y el mensaje más nuevo al final del archivo de registro. El servicio rsyslog usa un formato estándar al registrar las entradas en los archivos de registro. En el siguiente ejemplo, se explica la anatomía de un mensaje de registro en el archivo de registro /var/log/secure.

Feb 11 20:11:48 localhost sshd[1433]: Failed password for student from 172.25.0.10 port 59344 ssh2

* La marca de tiempo cuando se grabó la entrada de registro.
* El host desde donde se envió el mensaje de registro.
* El nombre del programa o el proceso y el número de PID que envió el mensaje de registro.
* El mensaje real enviado.

Monitoreo de registros

Para reproducir problemas e inconvenientes, es útil monitorear uno o más archivos de registro para eventos. El comando tail -f /path/to/file proporciona las últimas diez líneas del archivo especificado y continúa ofreciendo líneas nuevas en el archivo a medida que se escriben.

Por ejemplo, para monitorear los intentos fallidos de inicio de sesión, ejecute el comando tail en un terminal y, luego, en otro terminal, ejecute el comando ssh como el usuario root mientras un usuario intenta iniciar sesión en el sistema.

En el primer terminal, ejecute el siguiente comando tail:

[root@host ~]# tail -f /var/log/secure

En el segundo terminal, ejecute el siguiente comando ssh:

[root@host ~]# ssh <a href="mailto:root@localhost">root@localhost

</a>root@localhost's password: redhat

...output omitted...

[root@host ~]#

Envío manual de mensajes Syslog

El comando logger puede enviar mensajes al servicio rsyslog. De manera predeterminada, envía el mensaje al tipo (facility) user con la notice prioridad (user.notice), a menos que se especifique lo contrario con la opción -p. Es útil probar los cambios en la configuración del servicio rsyslog.

Para enviar un mensaje al servicio rsyslog que se graba en el archivo de registro /var/log/boot.log, ejecute el siguiente comando logger:

[root host ~]# logger -p local7.notice "Log entry created on host"

Cómo encontrar eventos

El servicio systemd-journald almacena datos de registro en un archivo binario estructurado e indexado,que se denomina diario (journal). Estos datos incluyen información adicional sobre el evento de registro. Por ejemplo, en el caso de los eventos de syslog, esto incluye el tipo (facility) y la prioridad del mensaje original.

[root@host ~]# journalctl

...output omitted...

Feb 21 17:46:25 host.lab.example.com systemd[24263]: Stopped target Sockets.

Feb 21 17:46:25 host.lab.example.com systemd[24263]: Closed D-Bus User Message Bus Socket.

lines 1464-1487/1487 (END) q

El comando journalctl destaca los mensajes de registro importantes: los mensajes con prioridad notice o warning se muestran en negrita y los mensajes con prioridad error o una prioridad mayor se muestran en rojo.

La clave para usar en forma correcta el diario (journal) para la solución de problemas y auditorías es limitar las búsquedas en el diario (journal) para mostrar solo la salida relevante.

De manera predeterminada, journalctl -n muestra las diez últimas entradas de registro. Puede ajustar esto con un argumento opcional que especifique cuántas entradas de registro se mostrarán.

2.8. Administración de redes

Recopilación de la información de interfaz de red

Identificación de interfaces de red

RedHat (2022) afirma que el comando ip link listará todas las interfaces de red disponibles en su sistema:

[user@host ~]$ ip link show

1: lo: <LOOPBACK,UP,LOWER\_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000

    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00

2: ens3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000

    link/ether 52:54:00:00:00:0a brd ff:ff:ff:ff:ff:ff

3: ens4: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000

    link/ether 52:54:00:00:00:1e brd ff:ff:ff:ff:ff:ff

En el ejemplo anterior, el servidor tiene tres interfaces de red: el dispositivo de bucle invertido que está conectado al propio servidor y dos interfaces Ethernet, ens3 y ens4.

Para configurar cada interfaz de red correctamente, necesita saber cuál está conectada a qué red. En muchos casos, sabrá la dirección MAC de la interfaz conectada a cada red, ya sea porque está impresa físicamente en la tarjeta o el servidor o porque es una máquina virtual y usted sabe cómo está configurada.

La dirección MAC del dispositivo aparece después de link/ether para cada interfaz. Así que ya sabes que la tarjeta de red con la dirección MAC 52:54:00:00:00:0a es la interfaz de red ens3.

Visualización de las direcciones IP

Use el comando ip para ver la información del dispositivo y de la dirección. Una sola interfaz de red puede tener varias direcciones IPv4 o IPv6.

[user@host ~]$ ip addr show ens3

2: ens3: <BROADCAST,MULTICAST, UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

Visualización de estadísticas de rendimiento

El comando ip también puede usarse para mostrar las estadísticas sobre el rendimiento de la red. Se pueden usar contadores para cada interfaz de red para identificar la presencia de problemas de red. Los contadores registran estadísticas para cosas como la cantidad de paquetes recibidos (RX) y transmitidos (TX), errores de paquetes y paquetes que se omitieron.

[user@host ~]$ ip -s link show ens3

<span style="font-size: 12.8px;">2: ens3: <BROADCAST,MULTICAST,UP,LOWER\_UP> mtu 1500 qdisc pfifo\_fast state UP qlen 1000

</span><span style="font-size: 12.8px;">link/ether 52:54:00:00:00:0a brd ff:ff:ff:ff:ff:ff

</span>RX: bytes packets errors dropped overrun mcast

269850 2931 0 0 0 0

TX: bytes packets errors dropped carrier collsns

300556 3250 0 0 0 0

Comprobación de la conectividad entre hosts

El comando ping se usa para comprobar la conectividad. El comando continúa ejecutándose hasta que se presione Ctrl+c, a menos que se indiquen otras opciones para limitar la cantidad de paquetes enviados.

[user@host ~]$ ping -c3 192.0.2.254

<span style="font-size: 12.8px;">PING 192.0.2.1 (192.0.2.254) 56(84) bytes of data.

</span>64 bytes from 192.0.2.254: icmp\_seq=1 ttl=64 time=4.33 ms

64 bytes from 192.0.2.254: icmp\_seq=2 ttl=64 time=3.48 ms

64 bytes from 192.0.2.254: icmp\_seq=3 ttl=64 time=6.83 ms

--- 192.0.2.254 ping statistics ---

<span style="font-size: 12.8px;">3 packets transmitted, 3 received, 0% packet loss, time 2003ms

</span><span style="font-size: 12.8px;">rtt min/avg/max/mdev = 3.485/4.885/6.837/1.424 ms</span>

El comando ping6 es la versión de IPv6 de ping en Linux. Se comunica a través de IPv6 y toma direcciones IPv6, pero de lo contrario funciona como ping.

[user@host ~]$ ping6 2001:db8:0:1::1

<span style="font-size: 12.8px;">PING 2001:db8:0:1::1(2001:db8:0:1::1) 56 data bytes

</span>64 bytes from 2001:db8:0:1::1: icmp\_seq=1 ttl=64 time=18.4 ms

<span style="font-size: 12.8px;">64 bytes from 2001:db8:0:1::1: icmp\_seq=2 ttl=64 time=0.178 ms

</span><span style="font-size: 12.8px;">64 bytes from 2001:db8:0:1::1: icmp\_seq=3 ttl=64 time=0.180 ms

</span>^C<strong>

</strong><span style="font-size: 12.8px;">--- 2001:db8:0:1::1 ping statistic ---

</span><span style="font-size: 12.8px;">3 packets transmitted, 3 received, 0% packet loss, time 2001ms</span><span style="font-size: 12.8px;">rtt min/avg/max/mdev = 0.178/6.272/18.458/8.616 ms

</span><span style="font-size: 12.8px;">[user@host ~]$</span>

Cuando hace ping a las direcciones de vínculo local y al grupo multidifusión de todos los nodos de vínculo local (ff02::1), la interfaz de red que se debe usar debe especificarse explícitamente con un identificador de zona de alcance (como ff02::1%ens3). Si esto se deja afuera, se muestra el error connect: Invalid argument.

Solución de problemas de ruta

El enrutamiento de la red es complejo y, a veces, el tráfico no se comporta como se espera. Aquí hay algunas herramientas de diagnóstico útiles.

Visualización de la tabla de enrutamiento

Use el ip comando con el route Opción para mostrar información de enrutamiento.

[user@host ~]$ ip route

default via 192.0.2.254 dev ens3 proto static metric 1024

192.0.2.0/24 dev ens3 proto kernel scope link src 192.0.2.2

10.0.0.0/8 dev ens4 proto kernel scope link src 10.0.0.11

Esto muestra la tabla de enrutamiento de IPv4. Todos los paquetes que estén destinados para la red 10.0.0.0/8 se envían directamente al destino mediante la ens4 del dispositivo. Todos los paquetes que estén destinados para la red 192.0.2.0/24 se envían directamente al destino mediante la ens3 del dispositivo. Todos los demás paquetes se envían al enrutador predeterminado que está ubicado en 192.0.2.254, y también mediante la ens3 del dispositivo.

Seguimiento de rutas tomadas por el tráfico

Para rastrear la ruta que toma el tráfico de la red para llegar a un host remoto a través de múltiples enrutadores, use traceroute o tracepath. Esto puede identificar si hay un problema con uno de sus enrutadores o con uno intermedio. Ambos comandos usan paquetes de UDP para realizar el seguimiento de una ruta de forma predeterminada; sin embargo, muchas redes bloquean el tráfico de UDP e ICMP. El comando traceroute tiene opciones para realizar el seguimiento de la ruta con paquetes UDP (predeterminado), ICMP (-I) o TCP (-T). Sin embargo, en general, el comando traceroute no está instalado de forma predeterminada.

[user@host ~]$ tracepath access.redhat.com

...output omitted...

<span style="font-size: 12.8px;">4: 71-32-28-145.rcmt.qwest.net 48.853ms asymm 5

</span><span style="font-size: 12.8px;">5: dcp-brdr-04.inet.qwest.net 100.732ms asymm 7

</span>6: 206.111.0.153.ptr.us.xo.net 96.245ms asymm 7

<span style="font-size: 12.8px;">7: 207.88.14.162.ptr.us.xo.net 85.270ms asymm 8

</span>8: ae1d0.cir1.atlanta6-ga.us.xo.net 64.160ms asymm 7

9: 216.156.108.98.ptr.us.xo.net 108.652m

10: bu-ether13.atlngamq46w-bcr00.tbone.rr.com 107.286ms asymm 1

...output omitted...

Cada línea de la salida de tracepath representa un router o hop por donde pasa el paquete entre el origen y el destino final. Proporciona información adicional como disponible que incluye la sincronización en ambos sentidos (RTT) y cualquier cambio en el tamaño de la unidad de transmisión máxima (MTU). La indicación asymm significa que el tráfico llegó a ese enrutador y regresó de ese enrutador usando diferentes rutas (asimétricas). Los enrutadores que se muestran son los que se usan para el tráfico saliente, no el tráfico de retorno.

Solución de problemas en puertos y servicios

Los servicios TCP usan sockets como extremos para la comunicación y se componen de una dirección IP, protocolo y número de puerto. En general, los servicios están atentos a los puertos estándares mientras que los clientes usan un puerto disponible en forma aleatoria. Los nombres más conocidos de puertos estándares están enumerados en el archivo /etc/services.

El comando ss se usa para mostrar las estadísticas del socket. El comando ss tiene por objeto reemplazarla herramienta anterior netstat que es parte del paquete net-tools, que algunos administradores de sistemas pueden conocer más, pero que no siempre está instalada.

[user@host ~]$ ss -ta

<span style="font-size: 12.8px;">State Recv-Q Send-Q Local Address:Port Peer Address:Port

</span>LISTEN 0 128 \*:sunrpc \*:\*

LISTEN 0 128 \*:ssh \*:\*

LISTEN 0 100 127.0.0.1:smtp \*:\*

LISTEN 0 128 \*:36889 \*:\*

Descripción de conceptos de NetworkManager

NetworkManager es un daemon que monitorea y administra los parámetros de red.

Además del daemon, hay un applet del área de notificaciones de GNOME que proporciona información sobre el estado de la red. Las herramientas gráficas y de la línea de comandos se comunican con NetworkManager y guardan archivos de configuración en el directorio /etc/sysconfig/network-scripts.

* Un dispositivo es una interfaz de red.
* Una conexión es un grupo de parámetros que pueden configurarse para un dispositivo.
* Solo puede haber una conexión activa para un dispositivo por vez. Pueden existir múltiples conexiones para que las usen distintos dispositivos o para permitir que una configuración se modifique para el mismo dispositivo. Si necesita cambiar temporalmente la configuración de red, en lugar de cambiar la configuración de una conexión, puede cambiar la conexión que está activa para un dispositivo. Por ejemplo, un dispositivo para una interfaz de red inalámbrica en una laptop puede usar diferentes conexiones para la red inalámbrica en un sitio de trabajo y para la red inalámbrica en el hogar.
* Cada conexión tiene un nombre o ID que la identifica.
* La utilidad nmcli se usa para crear y editar archivos de conexión desde la línea de comandos.

Visualización de información de redes

El comando nmcli dev status muestra el estado de todos los dispositivos de red:

[user@host ~]$ nmcli dev status

DEVICE TYPE STATE CONNECTION

eno1 ethernet connected eno1

ens3 ethernet connected static-ens3

eno2 ethernet disconnected --

lo loopback unmanaged --

El comando nmcli con show muestra una lista de todas las conexiones. Para enumerar solo las conexiones activas, agregue la opción --active.

[user@host ~]$ nmcli con show

NAME UUID TYPE DEVICE

eno2 ff9f7d69-db83-4fed-9f32-939f8b5f81cd 802-3-ethernet --

static-ens3 72ca57a2-f780-40da-b146-99f71c431e2b 802-3-ethernet ens3

eno1 87b53c56-1f5d-4a29-a869-8a7bdaf56dfa 802-3-ethernet eno1

[user@host ~]$ nmcli con show --active

NAME UUID TYPE DEVICE

static-ens3 72ca57a2-f780-40da-b146-99f71c431e2b 802-3-ethernet ens3

eno1 87b53c56-1f5d-4a29-a869-8a7bdaf56dfa 802-3-ethernet eno1

Cambio de nombre del host del sistema

El comando hostname muestra o modifica provisoriamente el nombre del host totalmente calificado del sistema.

[root@host ~]# hostname

host@example.com

Puede especificarse un nombre del host estático en el archivo /etc/hostname. Se usa el comando hostnamectl para modificar este archivo y puede usarse para ver el estado del nombre del host totalmente calificado del sistema. Si este archivo no existe, el nombre del host se establece mediante una consulta de DNS invertida una vez que la interfaz tiene una dirección IP asignada.

[root@host ~]# hostnamectl set-hostname host@example.com

<a href="mailto:host@example.com"></a>[root@host ~]# hostnamectl status

Static hostname: host.example.com

Icon name: computer-vm

Chassis: vm

Machine ID: 73ab164e278e48be9bf80e80714a8cd5

Boot ID: 6b1cbc4177164ef58c0e9ed4adb2904f

Virtualization: kvm

Operating System: Red Hat Enterprise Linux 8.0 beta (Ootpa)

CPE OS Name: cpe:/o:redhat:enterprise\_linux:8.0:beta

Kernel: Linux 4.18.0-60.el8.x86\_64

Architecture: x86-64

<span style="font-size: 12.8px;">[root@host ~]# cat /etc/hostname

</span><span style="font-size: 12.8px;">host@example.com

<a href="mailto:host@example.com"></a></span>[root@host ~]# cat /etc/resolv.conf

# Generated by NetworkManager

domain example.com

search example.com

nameserver 172.25.254.254

NetworkManager actualiza el archivo /etc/resolv.conf con los parámetros de configuración de DNS en los archivos de configuración de conexión. Use nmcli para modificar las conexiones.

Prueba de resolución de nombres DNS

El comando de host HOSTNAME puede usarse para probar la conectividad del servidor DNS.

[root@host ~]# host classroom.example.com

classroom.example.com has address 172.25.254.254

[root@host ~]# host 172.25.254.254

254.254.25.172.in-addr.arpa domain name pointer classroom.example.com.

DHCP reescribe automáticamente el archivo /etc/resolv.conf a medida que se inician las interfaces, a menos que usted especifique PEERDNS=no en los archivos de configuración de interfaz correspondientes. Configure esto con el comando nmcli.

[root@host ~]# nmcli con mod "static-ens3" ipv4.ignore-auto-dns yes

2.9. Almacenamiento y transferencia de archivos

El comando tar

RedHat (2022) afirma que el almacenamiento y la compresión de archivos es útil cuando se realizan copias de seguridad y se transfieren datos a través de una red. Uno de los comandos más antiguos y usados para crear y trabajar con colecciones de archivos de copias de seguridad es el comando tar.

Con el comando tar, los usuarios pueden reunir grandes conjuntos de archivos en un solo archivo (colección de archivos).

Una colección de archivos tar es una secuencia estructurada de datos de archivos mezclados con metadatos sobre cada archivo y un índice para poder extraer archivos individuales.

La colección de archivos puede comprimirse con gzip, bzip2 o xz.

Con el comando tar, también se puede enumerar el contenido de las colecciones de archivos o extraer sus archivos a su sistema actual.

Almacenamiento de archivos y directorios

La primera opción para usar cuando se crea una colección de archivos nueva es la opción c, seguida de la opción f. Luego, un solo espacio, el nombre de archivo de la colección de archivos que se creará y, por último, la lista de los archivos y directorios que deben agregarse al archivo. La colección de archivos se crea en el directorio actual, a menos que se especifique de otro modo.

Antes de crear una colección de archivos tar, verifique que no haya otra colección de archivos en el directorio con el mismo nombre de la colección de archivos nueva que se creará. El comando tar sobrescribe una colección de archivos existente sin advertencia.

Con el siguiente comando, se crea una colección de archivos con el nombre archive.tar con el contenido de file1, file2 y file3 en el directorio de inicio del usuario.

[user@host ~]$ tar -cf archive.tar file1 file2 file3

[user@host ~]$ ls archive.tar

<span style="font-size: 12.8px;">archive.tar</span>

Para que tar pueda archivar los archivos seleccionados, es obligatorio que el usuario que ejecute el comando tar pueda leer los archivos. Por ejemplo, la creación de una colección de archivos nueva de la carpeta /etc y todo su contenido requiere privilegios de root porque solo este tipo de usuario root tiene permitido leer todos los archivos que están presentes en el directorio /etc.

Un usuario sin privilegios puede crear una colección de archivos del directorio /etc, pero la colección de archivos omite los archivos que no incluyen el permiso de lectura para el usuario y omite los directorios que no incluyen permiso de lectura y ejecución para el usuario.

Para crear la colección de archivos tar con el nombre /root/etc.tar, con el directorio /etc como contenido con el usuario root:

[root@host ~]# tar -cf /root/etc.tar /etc

tar: Removing leading `/' from member names

[root@host ~]#

Transferencia de archivos con copia de segura

OpenSSH es útil para ejecutar los comandos de shell en sistemas remotos en forma segura. El comando de copia segura, scp, parte del conjunto OpenSSH, copia archivos desde un sistema remoto al sistema local o desde el sistema local a un sistema remoto. El comando usa el servidor SSH para la autenticación y cifra datos cuando se transfieren.

Puede especificar una ubicación remota para el origen o destino de los archivos que está copiando. El formato de la ubicación remota debe ser en la forma [user@]host:/path. Los user@ parte del argumento es opcional. Si falta, se utilizará su nombre de usuario local actual. Cuando ejecute el comando, su cliente scpse autenticará en el servidor SSH remoto al igual que ssh, utilizando la autenticación basada en claves o le pide su contraseña.

En el siguiente ejemplo, se muestra cómo copiar los archivos locales /etc/yum.conf y /etc/hosts en host, en el directorio de inicio de remoteuser en el sistema remoto remotehost:

[user@host ~]$ scp /etc/yum.conf /etc/hosts remoteuser@remotehost:/home/remoteuser

<a href="mailto:remoteuser@remotehost:/home/remoteuser"></a>remoteuser@remotehost's password: password

yum.conf 100% 813 0.8KB/s 00:00

hosts 100% 227 0.2KB/s 00:00

También puede copiar un archivo en la otra dirección, desde un sistema remoto al sistema de archivos local. En este ejemplo, el archivo /etc/hostname en remotehost se copia al directorio local /home/user. El comando scp autentica a remotehost como usuario remoteuser.

[user@host ~]$ scp remoteuser@remotehost:/etc/hostname /home/user

remoteuser@remotehost's password: password

hostname 100% 22 0.0KB/s 00:00

Para copiar un árbol de directorios completo de manera recursiva, use la opción -r. En el siguiente ejemplo, el directorio remoto /var/log en remotehost se copia de manera recursiva en el directorio local /tmp/ en host. Debe conectarse al sistema remoto como root para asegurarse de que puede leer todos los archivos en el directorio remoto /var/log.

[user@host ~]$ scp -r root@remoteuser:/var/log /tmp

root@remotehost's password: password

...output omitted...

Transferencia de archivos con el programa de transferencia segura de archivos

Para cargar o descargar archivos de forma interactiva desde un servidor SSH, use el programa de transferencia de archivos, sftp. Una sesión con el comando sftp usa el mecanismo de autenticación segura y la transferencia de datos cifrados desde y hacia el servidor SSH.

Al igual que con el comando scp, el comando sftp usa [user@]host para identificar el sistema de destino y el nombre de usuario. Si no especifica un usuario, el comando intentará iniciar sesión utilizando su nombre de usuario local como nombre de usuario remoto. A continuación, se le presentará un prompt sftp>.

[user@host ~]$ sftp remoteuser@remotehost

<a href="mailto:remoteuser@remotehost"></a>remoteuser@remotehost's password:<strong> </strong>password

Connected to remotehost.

sftp>

La sesión sftp interactiva acepta diversos comandos que funcionan de la misma manera en el sistema de archivos remoto que en el sistema de archivos local, como ls, cd, mkdir, rmdir y pwd. El comando put carga un archivo en el sistema remoto. El comando get descarga un archivo desde el sistema remoto. El comando exit finaliza la sesión sftp.

Sincronizar archivos y directorios con rsync

El uso del comando rsync es otra forma de copiar archivos en forma segura de un sistema a otro. La herramienta usa un algoritmo que minimiza la cantidad de datos copiados sincronizando solo las partes que se modificaron de los archivos.

Se diferencia de scp en que, si dos archivos o directorios son similares entre dos sistemas, rsync copia las diferencias entre los sistemas de archivos, mientras que scp copiaría todo.

Una ventaja de rsync es que puede copiar archivos entre un sistema local y un sistema remoto en forma segura y eficiente. Cuando una sincronización inicial de un directorio demora prácticamente el mismo tiempo que el copiado, las sincronizaciones posteriores solo requerirán que se copien las diferencias a través de la red, lo que acelera sustancialmente las actualizaciones.

Una opción importante de rsync es la opción -n para realizar un simulacro. Un simulacro es una imitación de lo que sucede cuando el comando se ejecuta. El simulacro muestra los cambios que rsync ejecutaría cuando el comando se ejecuta normalmente. Realice un simulacro antes de la operación de rsync real para garantizar que no se sobrescriban ni eliminen archivos importantes.

Las dos opciones más comunes cuando se sincronizan archivos y directorios con rsync son -v y -a.

La opción -v o --verbose proporciona un resultado más detallado. Esto es útil para solucionar problemas y para ver el progreso en vivo.

Puede usar rsync para sincronizar el contenido de un archivo o directorio local con un archivo o directorio en una máquina remota, utilizando cualquiera de las dos máquinas como origen. También puede sincronizar el contenido de dos archivos o directorios locales.

Por ejemplo, para sincronizar contenido del directorio /var/log con el directorio /tmp:

[user@host ~]$ su -

Password:<strong> </strong>password

[root@host ~]# rsync -av /var/log /tmp

receiving incremental file list

log/

log/README

log/boot.log

...output omitted...

log/tuned/tuned.log

sent 11,592,423 bytes received 779 bytes 23,186,404.00 bytes/sec

total size is 11,586,755 speedup is 1.00

[user@host ~]$ ls /tmp

log ssh-RLjDdarkKiW1

<span style="font-size: 12.8px;">[user@host ~]$</span>

Una barra al final del directorio de origen sincroniza el contenido de un directorio sin crear el subdirectorio en el directorio de destino. En este ejemplo, el directorio log no se crea en el directorio /tmp, solo el contenido de /var/log/ se sincroniza en /tmp.

[root@host ~]# rsync -av /var/log/ /tmp

sending incremental file list

<span style="font-size: 12.8px;">./

</span>README

boot.log

...output omitted...

tuned/tuned.log

sent 11,592,389 bytes received 778 bytes 23,186,334.00 bytes/sec

total size is 11,586,755 speedup is 1.00

<strong>[root@host ~]# ls /tmp

</strong>anaconda dnf.rpm.log-20190318 private

audit dnf.rpm.log-20190324 qemu-ga

boot.log dnf.rpm.log-20190331 README

...output omitted...

Importante

Cuando se ingrese el directorio de origen en el comando rsync, es importante si está presente la barra final en el nombre del directorio. Esto determina si el directorio o solo el contenido del directorio se sincroniza en el destino.

La terminación con tab de Bash agrega automáticamente una barra al final de los nombres de directorios.

Al igual que con los comandos scp y sftp, rsync especifica ubicaciones remotas usando el formato [user@]host:/path. La ubicación remota puede ser el sistema de origen o destino, pero una de las dos máquinas debe ser local.

Para preservar la propiedad del archivo, debe ser root en el sistema de destino. Si el destino es remoto,autentíquese como root. Si el destino es local, debe ejecutar rsync como root.

2.10. Acceso a los sistemas de archivos de Linux. Actualizaciones

Conceptos de la administración del almacenamiento

Conceptos de la administración del almacenamiento

RedHat (2022) afirma que se accede a los archivos en un servidor Linux a través de la jerarquía de sistemas de archivos, un solo árbol de directorios invertido. Esta jerarquía del sistema de archivos se ensambla desde sistemas de archivos provistos por los dispositivos de almacenamiento disponibles para su sistema. Cada sistema de archivos es un dispositivo de almacenamiento que ha sido formateado para almacenar archivos.

En cierto sentido, la jerarquía de sistemas de archivos Linux presenta una colección de sistemas de archivos en dispositivos de almacenamiento separados como si fuera un conjunto de archivos en un dispositivo de almacenamiento gigante en el que puede navegar. La mayoría de las veces no necesita saber en qué dispositivo de almacenamiento se encuentra un archivo en particular, solo necesita saber en qué directorio se encuentra el archivo.

Sin embargo, a veces puede ser importante. Es posible que deba determinar qué tan lleno está un dispositivo de almacenamiento y qué directorios en la jerarquía de sistemas de archivos son afectados. Es posible que haya errores en los registros de un dispositivo de almacenamiento y, por lo tanto, necesita saber qué sistemas de archivos están en riesgo. Es posible que solo desee crear un vínculo rígido entre dos archivos y necesita saber si están en el mismo sistema de archivos para determinar si es posible.

Sistemas de archivos y puntos de montaje

Para que los contenidos de un sistema de archivos estén disponibles en la jerarquía de sistemas de archivos, se debe montar en un directorio vacío. Este directorio se llama punto de montaje. Una vez montado, si usa ls para enumerar ese directorio, verá el contenido del sistema de archivos montado y puede acceder y, luego, usar esos archivos normalmente. Muchos sistemas de archivos se montan automáticamente como parte del proceso de arranque.

Si solo ha trabajado con letras de unidad de Microsoft Windows, este es un concepto fundamentalmente diferente. Es algo similar a la característica de carpetas montadas de NTFS.

Sistemas de archivos, almacenamiento y dispositivos de bloque

El acceso de bajo nivel a los dispositivos de almacenamiento en Linux se proporciona mediante un tipo especial de archivo llamado dispositivo de bloque. Estos dispositivos de bloque deben formatearse con un sistema de archivos para poder montarlos.

Los archivos de dispositivos de bloque se almacenan en el directorio /dev, junto con otros archivos de dispositivo. Los archivos del dispositivo son creados automáticamente por el sistema operativo. En muchas distribuciones Linux el primer disco duro SATA/PATA, SAS, SCSI o USB detectado se denomina /dev/sda, el segundo /dev/sdb, y así sucesivamente. Estos nombres representan el disco duro en su totalidad.

Otros tipos de almacenamiento tendrán otras formas de asignar nombres.

A blue and white table with white text

Description automatically generated

Tabla 6. Nomenclatura de dispositivos de bloque. Fuente: elaboración propia.

Nota

Muchas máquinas virtuales utilizan el almacenamiento paravirtualizado virtio-scsi más nuevo que tendrá la nomenclatura de estilo /dev/sd\*.

Particiones del disco

Normalmente, no hace que todo el dispositivo de almacenamiento se convierta en un sistema de archivos. Los dispositivos de almacenamiento normalmente se dividen en fragmentos más pequeños llamados particiones.

Las particiones le permiten compartimentar un disco; las diferentes particiones pueden formatearse con diferentes sistemas de archivos o usarse con fines distintos. Por ejemplo, una partición puede contener directorios de inicio de un usuario mientras que otra puede contener registros y datos del sistema. Si un usuario llena la partición del directorio de inicio con datos, la partición del sistema puede seguir teniendo espacio disponible.

Las particiones son dispositivos de bloque por derecho propio. En el almacenamiento adjunto a SATA, la primera partición en el primer disco es /dev/sda1. La tercera partición en el segundo disco es /dev/sdb3 y así sucesivamente. Los dispositivos de almacenamiento paravirtualizados tienen un sistema de nomenclatura similar.

Los nombres del dispositivo SSD adjunto a NVMe nombra sus particiones de manera diferente. En ese caso, la primera partición en el primer disco es /dev/nvme0p1. La tercera partición en el segundo disco es /dev/nvme1p3 y así sucesivamente. Las tarjetas SD o MMC tienen un sistema de nomenclatura similar.

Un listado extenso del archivo de dispositivo /dev/sda1 en host revela que su tipo de archivo especial es b,que significa «dispositivo de bloque».

[user@host ~]$ ls -l /dev/sda1

brw-rw----. 1 root disk 8, 1 Feb 22 08:00 /dev/sda1

Volúmenes lógicos

Otra manera de organizar discos y particiones es mediante la administración de volúmenes lógicos (LVM).Con la administración de volúmenes lógicos, uno o más dispositivos de bloque pueden agregarse a un conjunto de almacenamiento denominado grupo de volúmenes. El espacio en disco en el grupo de volúmenes se divide en uno o más volúmenes lógicos, que son el equivalente funcional de una partición que reside en un disco físico.

El sistema LVM asigna nombres a los grupos de volúmenes y volúmenes lógicos en el momento de la creación. LVM crea un directorio en /dev que coincide con el nombre del grupo y, luego, crea un enlace simbólico dentro de ese nuevo directorio con el mismo nombre que el volumen lógico. Ese archivo de volumen lógico está disponible para ser montado.

Por ejemplo, si un grupo de volúmenes se llama myvg y el volumen lógico dentro de este se llama mylv, entonces el nombre completo de la ruta al archivo de dispositivo de volumen lógico es /dev/myvg/mylv.

La forma del nombre del dispositivo de volumen lógico mencionado anteriormente se implementa, en realidad, como un enlace simbólico al archivo de dispositivo real utilizado para acceder a este que puede variar entre los arranques. Hay otra forma de nombre de dispositivo de volumen lógico vinculado desde archivos en /dev/mapper que se utilizan a menudo y, también, son enlaces simbólicos al archivo del dispositivo real.

Examen de sistemas de archivos

Para obtener una descripción general de los dispositivos de sistemas de archivos locales y remotos y la cantidad de espacio libre disponible, ejecute el comando df. Cuando el comando df se ejecuta sin argumentos, arroja un informe con el espacio en disco total, el espacio en disco usado, el espacio en disco libre y el porcentaje del espacio total en disco usado en todos los sistemas de archivos regulares montados. Informa sobre sistemas de archivos locales y remotos.

En el siguiente ejemplo, se muestran los sistemas de archivos y los puntos de montaje en host.

[user@host ~]$ df

Filesystem 1K-blocks Used Available Use% Mounted on

devtmpfs 912584 0 912584 0% /dev

tmpfs 936516 0 936516 0% /dev/shm

tmpfs 936516 16812 919704 2% /run

tmpfs 936516 0 936516 0% /sys/fs/cgroup

/dev/vda3 8377344 1411332 6966012 17% /

/dev/vda1 1038336 169896 868440 17% /boot

tmpfs 187300 0 187300 0% /run/user/1000

Las particiones en el sistema host muestran dos sistemas de archivos físicos, que se montan en / y /boot. Esto es frecuente en el caso de máquinas virtuales. Los dispositivos tmpfs y devtmpfs son sistemas de archivos en la memoria del sistema. Todos los archivos escritos en tmpfs o en devtmpfs desaparecen después de que se reinicia el sistema.

A fin de mejorar la legibilidad de los tamaños de las salidas, hay dos opciones diferentes legibles por el ojo humano: -h o -H. La diferencia entre estas dos opciones es que -h informa en KiB (210), MiB (220), o GiB (230), mientras que la opción -H informa en unidades SI: KB (103), MB (106), GB (109). Los fabricantes de discos duros normalmente usan las unidades SI cuando anuncian sus productos.

Muestre un informe sobre los sistemas de archivos en el sistema host con todas las unidades convertidas a formato legible por el ojo humano:

[user@host ~]$ df -h

Filesystem Size Used Avail Use% Mounted on

devtmpfs 892M 0 892M 0% /dev

tmpfs 915M 0 915M 0% /dev/shm

tmpfs 915M 17M 899M 2% /run

tmpfs 915M 0 915M 0% /sys/fs/cgroup

/dev/vda3 8.0G 1.4G 6.7G 17% /

/dev/vda1 1014M 166M 849M 17% /boot

tmpfs 183M 0 183M 0% /run/user/1000

Para obtener información más detallada sobre el espacio usado por un árbol de directorios en particular, use el comando du. El comando du ofrece las opciones -h y -H para convertir la salida a formato legible por el ojo humano. El comando du muestra el tamaño de todos los archivos en el árbol de directorios actual de modo recursivo.

Muestre un informe sobre el uso del disco para el directorio /usr/share en host:

[root@host ~]# du /usr/share

...output omitted...

176 /usr/share/smartmontools

184 /usr/share/nano

8 /usr/share/cmake/bash-completion

8 /usr/share/cmake

356676 /usr/share

Muestre un informe sobre el uso del disco en formato legible por el ojo humano, para el directorio /usr/share en host:

[root@host ~]# du -h /var/log

<span style="font-size: 12.8px;">...output omitted...

</span>176K /usr/share/smartmontools

184K /usr/share/nano

8.0K /usr/share/cmake/bash-completion

8.0K /usr/share/cmake

369M /usr/share

Montaje manual de sistemas de archivos

Un sistema de archivos que reside en un dispositivo de almacenamiento extraíble debe montarse para acceder a este. El comando mount permite que el usuario root monte manualmente un sistema de archivos. El primer argumento del comando mount especifica el sistema de archivos que se debe montar. El segundo argumento especifica el directorio que se usará como punto de montaje en la jerarquía de sistemas de archivos.

Hay dos formas comunes de especificar el sistema de archivos en una partición de disco para el comando mount:

* Con el nombre del archivo del dispositivo en /dev que contiene el sistema de archivos.
* Con el UUID escrito en el sistema de archivos, un identificador universalmente único.

El montaje de un dispositivo es relativamente simple. Debe identificar el dispositivo que desea montar, asegurarse de que exista el punto de montaje y montar el dispositivo en el punto de montaje.

Identificación del dispositivo de bloque

Un dispositivo de almacenamiento conectable en funcionamiento, ya sea como unidad de disco duro (HDD) o dispositivo de estado sólido (SSD) o un portadiscos de almacenamiento USB, pueden conectarse a un puerto diferente cada vez que se conectan a un sistema.

Use el comando lsblk para enumerar los detalles de un dispositivo de bloque especificado o todos los dispositivos disponibles.

[root@host ~]# lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

vda 253:0 0 12G 0 disk

├─vda1 253:1 0 1G 0 part /boot

├─vda2 253:2 0 1G 0 part [SWAP]

└─vda3 253:3 0 11G 0 part /

vdb 253:16 0 64G 0 disk

└─vdb1 253:17 0 64G 0 part

Si sabe que acaba de agregar un dispositivo de almacenamiento de 64 GB con una partición, puede deducir a partir del resultado anterior que /dev/vdb1 es la partición que desea montar.

Montaje por nombre de dispositivo de bloque

El siguiente ejemplo monta el sistema de archivos en la partición /dev/vdb1 en el directorio /mnt/data.

[root@host ~]# mount /dev/vdb1 /mnt/data

Para montar un sistema de archivos, ya debe existir el directorio de destino. El directorio /mnt existe de forma predeterminada y está destinado a ser utilizado como punto de montaje temporal.

Puedes usar el directorio /mnt, o mejor aún crear un subdirectorio de /mnt para usar como punto de montaje temporal, a menos que tenga una buena razón para montarlo en una ubicación específica en la jerarquía de sistemas de archivos.

Importante

Si el directorio que funciona como punto de montaje no está vacío, no se puede acceder a los archivos copiados en ese directorio antes de que el sistema de archivos se montara hasta que el sistema de archivos se desmonte nuevamente.

Este enfoque funciona bien a corto plazo. Sin embargo, el orden en que el sistema operativo detecta discos puede cambiar si se agregan o eliminan dispositivos del sistema. Esto cambiará el nombre del dispositivo asociado con ese dispositivo de almacenamiento. Un mejor enfoque sería montar por alguna característica incorporada en el sistema de archivos.

Montaje por UUID de sistema de archivos

Un identificador estable que está asociado con un sistema de archivos es su UUID, un número hexadecimal muy largo que actúa como identificador único universal. Este UUID es parte del sistema de archivos y permanece igual siempre que el sistema de archivos no se recree.

El comando lsblk -fp detalla la ruta completa del dispositivo, junto con los UUID y los puntos de montajes, así como el tipo de sistema de archivos en la partición. Si el sistema de archivos no está montado, el punto de montaje estará en blanco.

[root@host ~]# lsblk -fp

NAME FSTYPE LABEL UUID MOUNTPOINT

/dev/vda

├─/dev/vda1 xfs 23ea8803-a396-494a-8e95-1538a53b821c /boot

├─/dev/vda2 swap cdf61ded-534c-4bd6-b458-cab18b1a72ea [SWAP]

└─/dev/vda3 xfs 44330f15-2f9d-4745-ae2e-20844f22762d /

/dev/vdb

└─/dev/vdb1 xfs 46f543fd-78c9-4526-a857-244811be2d88

Monte el sistema de archivos por el UUID del sistema de archivos.

[root@host ~]# mount UUID="46f543fd-78c9-4526-a857-244811be2d88" /mnt/data

Montaje automático de dispositivos de almacenamiento extraíbles

Si ha iniciado sesión y está utilizando el entorno de escritorio gráfico, montará automáticamente cualquier medio de almacenamiento extraíble cuando se inserte.

El dispositivo de almacenamiento extraíble se monta en /run/media/USERNAME/LABEL donde NOMBRE DE USUARIO es el nombre del usuario que ha iniciado sesión en el entorno gráfico y ETIQUETA es un identificador, a menudo, el nombre que recibe el sistema de archivos al crearse, si estaba disponible.

Antes de retirar el dispositivo, debe desmontarlo manualmente.

Desmontaje de sistemas de archivos

Los procedimientos de apagado y reinicio desmontan todos los sistemas de archivos automáticamente.Como parte de este proceso, todos los datos del sistema de archivos almacenados en la memoria caché se transfieren al dispositivo de almacenamiento, lo que garantiza que el sistema de archivos no sufra daños en los datos.

Aviso

Los datos del sistema de archivos son a menudo almacenados en la memoria caché.

Por lo tanto, para evitar dañar los datos en el disco, es esencial que desmonte las unidades extraíbles antes de desenchufarlas. El procedimiento de desmontaje sincroniza los datos antes de liberar la unidad, lo que garantiza la integridad de los datos.

Para desmontar un sistema de archivos, el comando umount espera el punto de montaje como argumento.

[root@host ~]# umount /mnt/data

No se puede desmontar si el sistema de archivos montado está en uso. Para que el comando umount se ejecute correctamente, todos los procesos deben dejar de acceder a los datos en el punto de montaje.

En el siguiente ejemplo, el umount falla porque el sistema de archivos está en uso (la shell está usando /mnt/data como su directorio de trabajo actual) y genera un mensaje de error.

[root@host ~]# cd /mnt/data

[root@host data]# umount /mnt/data

umount: /mnt/data: target is busy.

El comando lsof enumera todos los archivos abiertos y el proceso que accede a ellos en el directorio proporcionado. Resulta útil identificar los procesos que actualmente impiden un correcto desmontaje del sistema de archivos.

[root@host data]# lsof /mnt/data

COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME

bash 1593 root cwd DIR 253,17 6 128 /mnt/data

lsof 2532 root cwd DIR 253,17 19 128 /mnt/data

lsof 2533 root cwd DIR 253,17 19 128 /mnt/data

Una vez que se identifican los procesos, puede tomarse una medida como esperar a que finalice el proceso o enviar una señal SIGTERM o SIGKILL al proceso. En este caso, basta con cambiar el directorio en funcionamiento actual por un directorio fuera del punto de montaje.

[root@host data]# cd

[root@host ~]# umount /mnt/data

Una razón común para que los sistemas de archivos no puedan desmontarse es que una shell Bash está usando el punto de montaje o un subdirectorio como directorio de trabajo actual. Use el comando cd para cambiar el sistema de archivos para resolver este problema.

Búsqueda de archivos

Un administrador de sistemas necesita herramientas para buscar archivos que coincidan con ciertos criterios en el sistema de archivos. En esta sección, se analizan dos comandos que pueden buscar archivos en la jerarquía de sistemas de archivos.

* El comando locate busca una base de datos generada previamente para nombres de archivo o rutas de archivos y arroja los resultados instantáneamente.
* El comando find busca archivos en tiempo real mediante un rastreo de la jerarquía de sistemas de archivos.

Localización de archivos por nombre

El comando locate busca archivos en función del nombre o la ruta al archivo. Es rápido porque busca esta información de la base de datos mlocate. Sin embargo, esta base de datos no se actualiza en tiempo real y debe actualizarse con frecuencia para que los resultados sean precisos. Esto también significa que locate no encontrará los archivos que se han creado desde la última actualización de la base de datos.

La base de datos locate se actualiza automáticamente todos los días. Sin embargo, en cualquier momento el usuario root puede emitir el comando updatedb para forzar una actualización inmediata.

[root@host ~]# updatedb

El comando locate restringe los resultados para los usuarios sin privilegios. Para ver el nombre del archivo resultante, el usuario debe tener permiso de búsqueda en el directorio en el que reside el archivo.

Busque archivos con passwd en el nombre o en la ruta en los árboles de directorios legibles por el usuario en host.

[user@host ~]$ locate passwd

/etc/passwd

/etc/passwd-

/etc/pam.d/passwd

/etc/security/opasswd

/usr/bin/gpasswd

/usr/bin/grub2-mkpasswd-pbkdf2

/usr/bin/lppasswd

/usr/bin/passwd

...output omitted...

Los resultados se presentan incluso cuando la ruta o el nombre de archivo es solo una coincidencia parcial con la consulta de búsqueda.

[root@host ~]# locate image

/etc/selinux/targeted/contexts/virtual\_image\_context

/usr/bin/grub2-mkimage

/usr/lib/sysimage

/usr/lib/dracut/dracut.conf.d/02-generic-image.conf

/usr/lib/firewalld/services/ovirt-imageio.xml

/usr/lib/grub/i386-pc/lnxboot.image

...output omitted...

La opción -i realiza una búsqueda que distingue entre mayúsculas y minúsculas. Con esta opción, todas las combinaciones posibles de letras en mayúsculas y minúsculas coinciden con la búsqueda.

[user@host ~]$ locate -i messages

...output omitted...

/usr/share/vim/vim80/lang/zh\_TW/LC\_MESSAGES

/usr/share/vim/vim80/lang/zh\_TW/LC\_MESSAGES/vim.mo

/usr/share/vim/vim80/lang/zh\_TW.UTF-8/LC\_MESSAGES

/usr/share/vim/vim80/lang/zh\_TW.UTF-8/LC\_MESSAGES/vim.mo

/usr/share/vim/vim80/syntax/messages.vim

/usr/share/vim/vim80/syntax/msmessages.vim

/var/log/messages

La opción -n limita el número de resultados de búsqueda arrojados por el comando locate. En el siguiente ejemplo, se limitan los resultados de búsqueda arrojados por locate a las primeras cinco coincidencias.

[user@host ~]$ locate -n 5 snow.png

/usr/share/icons/HighContrast/16x16/status/weather-snow.png

/usr/share/icons/HighContrast/22x22/status/weather-snow.png

/usr/share/icons/HighContrast/24x24/status/weather-snow.png

/usr/share/icons/HighContrast/256x256/status/weather-snow.png

/usr/share/icons/HighContrast/32x32/status/weather-snow.png

Búsqueda de archivos en tiempo real

El comando find localiza los archivos mediante la realización de una búsqueda en tiempo real en la jerarquía de sistemas de archivos. Es más lento que locate, pero más preciso. También puede buscar archivos según criterios que no sean el nombre del archivo, como los permisos del archivo, el tipo de archivo, su tamaño o la hora de su modificación.

El comando find examina los archivos en el sistema de archivos usando la cuenta de usuario que ejecutó la búsqueda. El usuario que invoca el comando find debe tener permiso de lectura y ejecución en un directorio para examinar su contenido.

El primer argumento para el comando find es el directorio en que se realizará la búsqueda. Si el argumento del directorio se omite, find comienza la búsqueda en el directorio actual y busca coincidencias en los subdirectorios.

Para buscar archivos por nombre de archivo, use la opción -name FILENAME. Con esta opción, find arroja la ruta a los archivos que coinciden exactamente con el NOMBRE DE ARCHIVO. Por ejemplo, para buscar archivos con el nombre sshd\_config comenzando con el directorio /, ejecute el siguiente comando:

[root@host ~]# find / -name sshd\_config

/etc/ssh/sshd\_config

Nota

Con el comando find, las opciones de palabra completa usan un solo guion y las opciones siguen el argumento del nombre de la ruta, a diferencia de la mayoría de otros comandos de Linux.

Los comodines están disponibles para buscar el nombre de un archivo y arrojan todos los resultados que son coincidencias parciales. Al usar comodines, es importante poner entre comillas el nombre del archivo para evitar que el terminal interprete el comodín.

En el siguiente ejemplo, busque archivos comenzando en el directorio / que finalizan en .txt:

[root@host ~]# find / -name '\*.txt'

/etc/pki/nssdb/pkcs11.txt

/etc/brltty/brl-lt-all.txt

/etc/brltty/brl-mb-all.txt

/etc/brltty/brl-md-all.txt

/etc/brltty/brl-mn-all.txt

...output omitted...

Para buscar archivos en el directorio /etc/ que contienen la palabra pass en cualquier parte de sus nombres en host, ejecute el siguiente comando:

[root@host ~]# find /etc -name '\*pass\*'

/etc/security/opasswd

/etc/pam.d/passwd

/etc/pam.d/password-auth

/etc/passwd-

/etc/passwd

/etc/authselect/password-auth

Para realizar una búsqueda que no distinga entre mayúsculas y minúsculas de un nombre de archivo determinado, use la opción -iname seguida del nombre del archivo que desea buscar. Para buscar archivos con texto que no distinga entre mayúsculas y minúsculas, messages, en sus nombres en el directorio / en host, ejecute el siguiente comando:

[root@host ~]# find / -iname '\*messages\*'

...output omitted...

/usr/share/vim/vim80/lang/zh\_CN.UTF-8/LC\_MESSAGES

/usr/share/vim/vim80/lang/zh\_CN.cp936/LC\_MESSAGES

/usr/share/vim/vim80/lang/zh\_TW/LC\_MESSAGES

/usr/share/vim/vim80/lang/zh\_TW.UTF-8/LC\_MESSAGES

/usr/share/vim/vim80/syntax/messages.vim

/usr/share/vim/vim80/syntax/msmessages.vim

Administración de paquetes de software con yum

Existen infinidad de gestores de paquetes de software. Vamos a hablar de uno de los principales como es Yum.

Yum está diseñado para ser un mejor sistema de gestión de instalación y actualizaciones de software basado en RPM. El comando yum le permite instalar, actualizar, eliminar y obtener información sobre los paquetes de software y sus dependencias. Puede obtener un historial de transacciones realizadas y trabajar con múltiples repositorios de software.

Algunas de las opciones principales de yum son las siguientes:

* yum help muestra la información de uso.
* yum list muestra los paquetes instalados y aquellos disponibles.

[user@host ~]$ yum list 'http\*'

Available Packages

http-parser.i686 2.8.0-2.el8 rhel8-appstream

http-parser.x86\_64 2.8.0-2.el8 rhel8-appstream

httpcomponents-client.noarch 4.5.5-4.module+el8+2452+b359bfcd rhel8-appstream

httpcomponents-core.noarch 4.4.10-3.module+el8+2452+b359bfcd rhel8-appstream

httpd.x86\_64 2.4.37-7.module+el8+2443+605475b7 rhel8-appstream

httpd-devel.x86\_64 2.4.37-7.module+el8+2443+605475b7 rhel8-appstream

httpd-filesystem.noarch 2.4.37-7.module+el8+2443+605475b7 rhel8-appstream

httpd-manual.noarch 2.4.37-7.module+el8+2443+605475b7 rhel8-appstream

httpd-tools.x86\_64 2.4.37-7.module+el8+2443+605475b7 rhel8-appstream

yum info PACKAGENAME arroja información detallada sobre un paquete que incluye el espacio en disco necesario para la instalación.

Para obtener información sobre el servidor HTTP Apache:

[user@host ~]$ yum info httpd

Available Packages

Name : httpd

Version : 2.4.37

Release : 7.module+el8+2443+605475b7

Arch : x86\_64

Size : 1.4 M

Source : httpd-2.4.37-7.module+el8+2443+605475b7.src.rpm

Repo : rhel8-appstream

Summary : Apache HTTP Server

URL : https://httpd.apache.org/

<a href="https://httpd.apache.org/"></a>License : ASL 2.0

Description : The Apache HTTP Server is a powerful, efficient, and extensible

: web server.

* yum install PACKAGENAME obtiene e instala un paquete de software junto con cualquier tipo de dependencia.
* yum update PACKAGENAME obtiene e instala una versión más reciente del paquete especificado, incluidas las dependencias. Generalmente, el proceso intenta preservar los archivos de configuración, pero en algunos casos se les cambiará el nombre si el empaquetador considera que el anterior no funcionará después de la actualización. Si no se especifica el PACKAGENAME, instala todas las actualizaciones relevantes.

[user@host ~]$ sudo yum update

En el siguiente vídeo *Principales comandos Linux* vamos a realizar un repaso de más de treinta comandos que un administrador de sistemas Linux debe conocer.